Annales Mathematicae et Informaticae
***36** (2009) pp. 15–28*
http://ami.ektf.hu

Approximation approach to performance evaluation of Proxy Cache Server systems

Tamás Bérczes

Department of Informatics Systems and Networks, University of Debrecen

Submitted 8 January 2009; Accepted 15 April 2009

Abstract

In this paper we treat a modification of the performance model of Proxy Cache Servers to a more powerful case when the inter-arrival times and the service times are generally distributed. First we describe the original Proxy Cache Server model where the arrival process is a Poisson process and the service times are supposed to be exponentially distributed random variables. Then we calculate the basic performance parameters of the modified performance model using the well known Queueing Network Analysis (QNA) approximation method. The accuracy of the new model is validated by means of a simulation study over an extended range of test cases.

Keywords: Queueing Network, Proxy Cache Server, Performance Models, GI/G/1 queue

1. Introduction

The Internet quickly became an essential and integral part of today's life. However, the booming use of the Web has caused congested networks and overloaded servers. So, the answer from the remote Web server to the client often takes a long time. Adding more network bandwidth is a very expensive 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. 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 [2]; the originating Web sites [3]; and the boundary between the local area network and the

Internet [4]. 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.

In this paper, we present an extended version of the performance model of a PCS (see [5, 7]) using a more powerful case when inter-arrival times and the service times are generally distributed.

The organization of the paper is as follows. In Section 2, renewal-based parametric decomposition models are reviewed (see [1, 10]). In Section 3 we introduce a modified version of the original performance model of Proxy Cache Server, where we include the repetition loop at the Proxy Server. A detailed description of the generalized model is given in Section 4. Section 5 is devoted to the validation of the numerical results of the approximation. The paper ends with Comments.

2. The GI/G/1 approximation

The GI/G/1 approximation described here is an example of a method using Parametric Decomposition (see [10]) where the individual queueing nodes are analyzed in isolation based on their respective input and output processes. In this model, the arrival process is a general (GI) arrival process characterised by a mean arrival rate and a squared coefficient of variation (SQV) of the inter-arrival time and the service time may have any general distribution. To use the approximation, we need only to know the mean and the squared coefficient of variance of the inter-arrival times and the service times. In order to apply this method, we assume that the arrival process to a network node is renewal, so the arrival intervals are independent, identically distributed random variables. Immediate feedback, where a fraction of the output of a particular queue enters the queue once again, needs special treatment. Before the detailed analysis of the queueing network is done, the method first removes immediate feedback in a queue by suitably modifying its service time.

This model contains procedures required for modeling of the basic network operations of merging, departure and splitting, arising due to the common sharing of the resources and routing decisions in the network. Futhermore, the approximation provide performance measures (i.e. mean queue lengths, mean waiting times, etc.) for both per-queue and per-network.

The parameters required for the approximation: Arrival process: (λ_A - the mean arrival rate), (c_A^2 - the SQV of the inter-arrival time) and service time (τ_S

- the mean service time), and (c_S^2 - the SQV of the service time) at a considered node.

The approximation method that transforms the two parameters of the internal flows for each of the three basic network operations and the removal of the immediate feedback, as given in [1], is described in the following:

1) *Merging GI traffic flows:* The superposed process of n individual GI flows, each characterized by λ_j and c_j^2 (j = $1, \ldots, n$), as it enters the considered node is approximated by a GI traffic flow with parameters λ_A and c_S^2, representing the mean arrival rate and SQV of the inter-arrival time of the superposed flow, respectively. The mean arrival rate and the SQV of the inter-arrival time of the superposed flow is given by:

$$\lambda_A = \sum_{j=1}^{n} \lambda_j,$$

$$c_A^2 = \varpi \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_A} c_j^2 + 1 - \varpi,$$

with

$$\varpi = \frac{1}{1 + 4(1-\rho)^2(\nu - 1)},$$

$$\nu = \frac{1}{\sum_{j=1}^{n} \left(\frac{\lambda_j}{\lambda_A}\right)^2},$$

and ρ is the utilisation at the node, defined by $\rho = \lambda_A \tau_S$.

2) *Departure flow from a queue:* The departure flow from a queue is approximated as a GI traffic flow, characterized by λ_D and c_D^2, representing the mean departure rate and SQV of the inter-departure time of the departure flow, respectively. Under equilibrium conditions, the mean flow entering a queue is always equal to the mean flow existing the queue: $\lambda_D = \lambda_A$. The SQV of inter-departure time of the departure flow is given by:

$$c_D^2 = \rho^2 c_S^2 + \left(1 - \rho^2\right) c_A^2$$

3) *Splitting a GI flow Probabilistically:* If a GI flow with parameters λ and c^2 is split into n flows, each selected independently with probability p_i, the parameters for the i-th flow will be given by:

$$\lambda_i = p_i \lambda,$$

$$c_i^2 = p_i c^2 + (1 - p_i).$$

4) *Removing immediate feedback:* If the output traffic from a queue is fed back to this queue itself (Q_i), so that the net arrival process is the sum of the external

arrivals Λ and the fed back portion $p_{ii}\lambda_i$. The approach followed to eliminate this immediate feedback at the queue is to suitably adjust the service time at the queue and the SQV of service time. Assume that the original service parameters at the considered node are: $\tau_{S,U}$ - the mean service time, and $c^2_{S,U}$ - the SQV of the service time. Removing the immediate feedback from that node we will get the modified service parameters:

$$\begin{aligned} \tau_{S,M} &= \frac{\tau_{S,U}}{1-p_{ii}}, \\ c^2_{S,M} &= p_{ii} + (1-p_{ii})c^2_{S,U}, \\ W_{q,M} &= \frac{W_{q,M}}{1-p_{ii}}. \end{aligned}$$

This reconfigured queue without immediate feedback is used subsequently for solving the queueing network.

5) *Mean waiting time:* If the considered node is a GI/G/1 queue, the following Kramer and Langenbach-Belz approximation is used (see [12]):

$$W_q = \frac{\tau_S \cdot \rho(c_A^2 + c_D^2)\beta}{2(1-\rho)}$$

with

$$\beta_{Web} = \begin{cases} \exp\left(\frac{2(1-\rho)(1-c_A^2)^2}{3\rho(c_A^2+c_D^2)}\right), & \text{for } c_A^2 < 1, \\ 1 & \text{for } c_A^2 \geqslant 1. \end{cases}$$

3. The model of Proxy Cache Server

In this section we modified the original (M/M/1) performance model of Proxy Cache Server (see [5]). In this version of the performance model the Proxy Cache Server behaves like a Web server. So, if the size of the file that will pass through the server, is greater then the server's output buffer it will start a looping process until the delivery of all file's is completed (see [11, 6]).

Using Proxy Cache Server, if any information or file is requested to be downloaded, first it is checked whether the document exists on the Proxy Cache Server or not. (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 back to the PCS then a copy of it is delivered to the user.

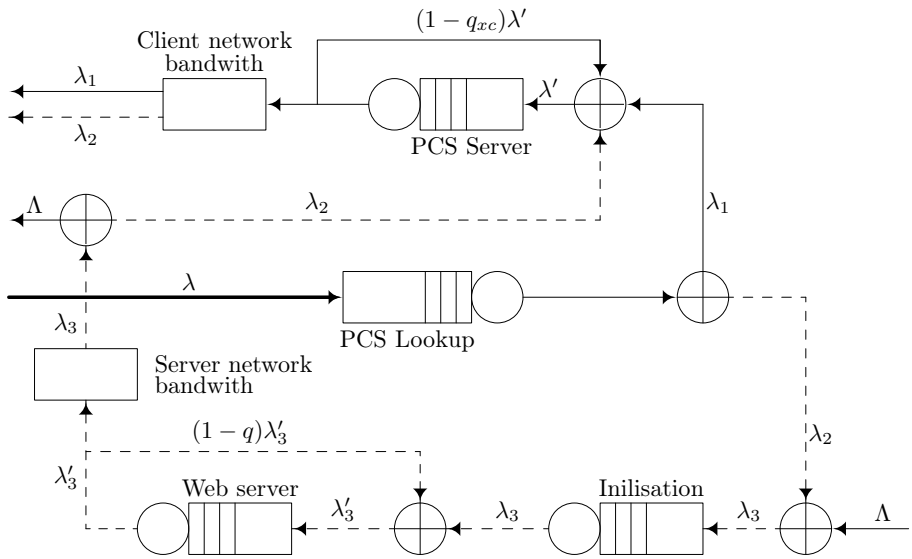

Figure 1: Network model

Figure 1 illustrates the path of a request in the original model (with feedback) starting from the user and finishing with the return of the answer to the user. The notations of the most important basic parameters used in this model are collected in Table 3.

In this section we assume that the requests of the PCS users arrive according to a Poisson process with rate λ, and the external requests arrive to the Web server according to a Poisson process with rate Λ, respectively.

The service rate of the Web server is given by:

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

where B_s is the capacity of the output buffer, Y_s is the static server time, and R_s is the dynamic server rate.

The service rate of the PCS is given by the equation:

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

where B_{xc} is the capacity of the output buffer, Y_{xc} is the static server time of the PCS, and R_s is the dynamic server rate of the PCS. The solid line in Figure 1 ($\lambda_1 = p\lambda$) represents the traffic when the requested file is available on the PCS and can be delivered directly to the user. The $\lambda_2 = (1-p)\lambda$ 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. $\lambda_3 = \lambda_2 + \Lambda$ is the flow of the overall requests arriving to the remote Web server. First the λ_3

traffic undergoes the process of initial handshaking to establish a one-time TCP connection (see [11, 7]). We denote by I_s this initial setup.

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. Let

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

be the probability that the desired file can be delivered at the first attempt. So λ_3' is the flow of the requests arriving at the Web service considering the looping process. According to the conditions of equilibrium and the flow balance theory of queueing networks

$$\lambda_3 = q\lambda_3'$$

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

$$\lambda = q_{xc}\lambda'$$

where λ' is the flow of the requests arriving to the PCS, considering the looping process.

Then we get the overall response time (see [5]):

$$\begin{aligned} T_{xc} = {} & \frac{1}{\frac{1}{I_{xc}} - (\lambda)} + p\left\{\frac{\frac{F}{B_{xc}}}{\frac{1}{\left(Y_{xc}+\frac{B_{xc}}{R_{xc}}\right)} - \frac{\lambda}{q_{xc}}} + \frac{F}{N_c}\right\} \\ & + (1-p)\left\{\frac{1}{\frac{1}{I_s} - \lambda_3} + \frac{\frac{F}{B_s}}{\frac{1}{\left(Y_s+\frac{B_s}{R_s}\right)} - \frac{\lambda_3}{q}}\right. \\ & \left. + \frac{F}{N_s} + \frac{\frac{F}{B_{xc}}}{\frac{1}{\left(Y_{xc}+\frac{B_{xc}}{R_{xc}}\right)} - \frac{\lambda}{q_{xc}}} + \frac{F}{N_c}\right\}. \end{aligned} \tag{3.1}$$

4. The GI/G/1 model of Proxy Cache Server

In this section instead of M/M/1 queues we will use GI/G/1 queues using the approximation describe in Section 2.

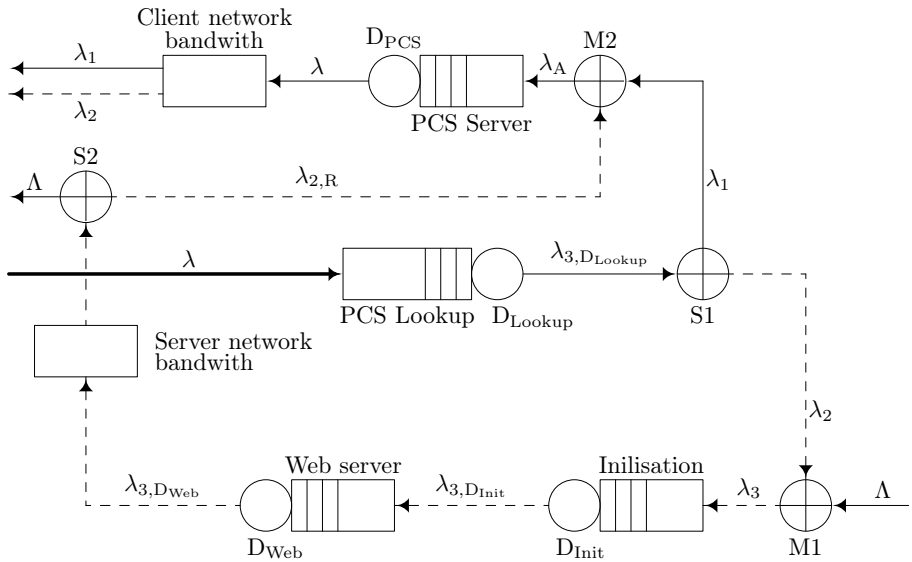

Figure 2: Modified Network model

The requests of the PCS users is assumed to be generalized inter-arrival (GI) process (see [10]) with λ mean arrival rate and with c_λ^2 SQV of the inter-arrival time, and the external arrivals at the remote Web server are generalized inter-arrival process too with parameters Λ and c_Λ^2.

The parameters of the queue PCS Lookup and the queue of the TCP initialization (see Figure 2) are μ_{Lookup}, c_{Lookup}^2 and μ_{Init}, c_{Init}^2 and the parameters of the Web and PCS servers are μ_{Web}, c_{Web}^2 and μ_{PCS}, c_{PCS}^2 where:

$$\mu_{Lookup} = \frac{1}{I_{xc}},$$

and

$$\mu_{Init} = \frac{1}{I_S},$$

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

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

where I_{xc} is the lookup time of the PCS (in second) and I_s is the TCP setup time. The B_s and B_{xc} parameters are the capacity of the output buffer of the Web server and the PCS, Y_s and Y_{xc} are the static server times, and R_s and R_{xc} are the dynamic server rates for the Web and Proxy servers (see [7]).

In the first step we removed the immediate feedbacks from the Web server and from the PCS, respectively. Figure 2 shows the modified model. After the removal

we have to modify the parameters of the corresponding servers:

$$\mu_{Web,M} = \mu_{Web} q,$$
$$c^2_{Web,M} = (1-q) + q c^2_{Web},$$
$$\mu_{PCS,M} = \mu_{PCS} q_{xc},$$
$$c^2_{PCS,M} = (1-q_{xc}) + q_{xc} c^2_{PCS}.$$

In the model we have 2 superposition point (S1,S2), 2 merging point (M1,M2) and 4 separate queue where we have to recalculate the basic parameters. In S1 position the flow of the requests split in two flows with probability p and $1-p$.

The recalculated parameters of the departure flow after checked of the availability of the required file are:

$$\lambda_D = \lambda,$$
$$c^2_{D_{Lookup}} = \rho^2 c^2_{Lookup} + \left(1-\rho^2\right) c^2_{\lambda},$$

where

$$\rho = \frac{\lambda}{\mu_{I_{xc}}}.$$

The solid line (λ_1) represents those requests, which are 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.

The parameters of the two flows are:

$$\lambda_1 = p\lambda_D,$$
$$c^2_1 = p c^2_{D_{Lookup}} + (1-p),$$
$$\lambda_2 = (1-p)\lambda_D,$$
$$c^2_2 = (1-p) c^2_{D_{Lookup}} + p.$$

In M1 position the λ_2 flow and the external requests are merging, and we get the λ_3 flow with the parameters defined below:

$$\lambda_3 = \lambda_2 + \Lambda,$$
$$c^2_3 = w\left(\frac{\lambda_2}{\lambda_3} c^2_2 + \frac{\Lambda}{\lambda_3} c^2_{\Lambda}\right) + (1-w)$$

where

$$w = \frac{1}{4(1-\rho)^2(\nu-1)},$$
$$\nu = \frac{1}{\left(\frac{\lambda_2}{\lambda_3}\right)^2 + \left(\frac{\Lambda}{\lambda_3}\right)^2},$$

and

$$\rho = \frac{\lambda_3}{\mu_{Init}}$$

The parameters of the departure flow ($\lambda_{3,D_{Init}}$) after the TCP initialisation are:

$$\lambda_{3,D_{Init}} = \lambda_3,$$

$$c^2_{3,D_{Init}} = \rho^2 c^2_{Init} + \left(1 - \rho^2\right) c^2_3,$$

where

$$\rho = \frac{\lambda_{3,D_{Init}}}{\mu_{Init}}.$$

The parameters of the departure flow ($\lambda_{3,D_{Web}}$) from the Web server are:

$$\lambda_{3,D_{Web}} = \lambda_3,$$

$$c^2_{3,D_{Web}} = \rho^2 c^2_{Web,M} + \left(1 - \rho^2\right) c^2_{3,D_{Init}},$$

where

$$\rho = \frac{\lambda_3}{\mu_{Web,M}}.$$

Then in S2 position the $\lambda_{3,D_{Web}}$ flow splits into two parts. One part is the traffic of the external requests, with probability $\frac{\Lambda}{\lambda_2+\Lambda}$, and the second part is the flow ($\lambda_{2,R}$) of the returning requests to the PCS. The parameters of the $\lambda_{2,R}$ traffic are:

$$\lambda_{2,R} = \lambda_2,$$

$$c^2_{2,R} = \frac{\lambda_2}{\lambda_2 + \Lambda} c^2_{3,D_{Web}} + \left(1 - \frac{\lambda_2}{\lambda_2 + \Lambda}\right)$$

In M2 position the λ_1 traffic and the $\lambda_{2,R}$ traffic are merging into λ_A traffic which is described by parameters:

$$\lambda_A = \lambda_1 + \lambda_{2,_R} = \lambda_1 + \lambda_2 = \lambda,$$

$$c^2_A = w\left(\frac{\lambda_1}{\lambda} c^2_1 + \frac{\lambda_2}{\lambda} c^2_{2,R}\right) + (1 - w)$$

where

$$w = \frac{1}{4(1-\rho)^2(\nu - 1)},$$

$$\nu = \frac{1}{\left(\frac{\lambda_1}{\lambda}\right)^2 + \left(\frac{\lambda_2}{\lambda}\right)^2},$$

and

$$\rho = \frac{\lambda}{\mu_{PCS,M}}$$

The overall response time can be calculated as follows (see [5, 11]):

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

where

$$T_{Lookup} = W_{Lookup} + \frac{1}{\mu_{Lookup}} = \\ = \frac{\frac{1}{\mu_{Lookup}}\rho_{Lookup}\left(c_\lambda^2 + c_{Lookup}^2\right)\beta}{2\left(1-\rho_{lookup}\right)} + \frac{1}{\mu_{Lookup}},$$

$$\beta = \begin{cases} \exp\left(-\frac{2(1-\rho_{Lookup})\left(1-c_\lambda^2\right)^2}{3\rho_{Lookup}\left(c_\lambda^2+c_{Lookup}^2\right)}\right) & \text{for } c_\lambda^2 < 1 \\ 1 & \text{for } c_\lambda^2 \geqslant 1 \end{cases}$$

$$\rho_{Lookup} = \frac{\lambda}{\mu_{Lookup}}$$

and

$$T_{PCS} = W_{PCS} + \frac{1}{\mu_{PCS,M}} = \\ = \frac{\frac{1}{\mu_{PCS,M}}\rho_{pcs}\left(c_A^2 + c_{pcs,M}^2\right)\beta}{2\left(1-\rho_{pcs}\right)} + \frac{1}{\mu_{pcs,M}},$$

where

$$\beta = \begin{cases} \exp\left(-\frac{2(1-\rho_{pcs})\left(1-c_A^2\right)^2}{3\rho_{pcs}\left(c_A^2+c_{pcs,M}^2\right)}\right) & \text{for } c_A^2 < 1 \\ 1 & \text{for } c_A^2 \geqslant 1 \end{cases}$$

$$\rho_{pcs} = \frac{\lambda_A}{\mu_{pcs,M}}$$

and

$$T_{Init} = W_{Init} + \frac{1}{\mu_{Init}} = \\ = \frac{\frac{1}{\mu_{Init}}\rho_{Init}\left(c_3^2 + c_{Init}^2\right)\beta_{Init}}{2\left(1-\rho_{Init}\right)} + \frac{1}{\mu_{Init}},$$

where

$$\beta_{Init} = \begin{cases} \exp\left(-\frac{2(1-\rho_{Init})\left(1-c_3^2\right)^2}{3\rho_{Init}\left(c_3^2+c_{Init}^2\right)}\right), & \text{for } c_3^2 < 1 \\ 1 & \text{for } c_3^2 \geqslant 1 \end{cases}$$

$$\rho_{Init} = \frac{\lambda_3}{\mu_{Init}}$$

and

$$T_{Web} = W_{Web} + \frac{1}{\mu_{Web,M}} = \\ = \frac{\frac{1}{\mu_{Web,M}}\rho_{web}\left(c^2_{D_{Init}} + c^2_{web,M}\right)\beta_{web}}{2\left(1-\rho_{web}\right)} + \frac{1}{\mu_{web,M}},$$

$$\beta_{Web} = \begin{cases} \exp\left(-\frac{2(1-\rho_{web})\left(1-c^2_{D_{Init}}\right)^2}{3\rho_{web}\left(c^2_{D_{Init}}+c^2_{web,M}\right)}\right), & \text{for } c^2_{D_{Init}} < 1 \\ 1 & \text{for } c^2_{D_{Init}} \geqslant 1 \end{cases}$$

$$\rho_{web} = \frac{\lambda_3}{\mu_{web,M}}$$

5. Numerical results

For the numerical explorations the corresponding parameters of Cheng and Bose [7] are used. The value of the other parameters for numerical calculations are: $I_s = I_{xc} = 0.004$ 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 [9].

For validating the approximation, we wrote a simulation program in Microsoft Visual Basic 2005 under .NET framework 2.0. It was run on a PC with a T2300 Intel processor (1.66 GHz) with 2 GB RAM. First we validated the simulation program using exponential distributions. For validation we calculated the analytical results of the overall response time given by Eq(3.1) and compared to the simulation results. In Table 1, we can see that the corresponding mean of the total response times are very close to each other; they are the same at least up to the 4th decimal digit.

For the validation of the approximation method we used the following distributions (see [8]). In case $0 < c_X < 1$ we use an $E_{k-1,k}$ distribution, where $\frac{1}{k} \leqslant k < \frac{1}{k-1}$. In this case the approximating $E_{k-1,k}$ distribution is with probability p (resp. $1-p$) the sum of $k-1$ (resp. k) independent exponentials with common mean $\frac{1}{\mu}$. Choosing

$$p = \frac{1}{1+c_X^2}\left(kc_X^2 - \left(k\left(1+c_X^2\right) - k^2c_X^2\right)^{1/2}\right) \quad \text{and} \quad \mu = \frac{k-p}{E(X)}$$

the $E_{k-1,k}$ distribution matches $E(X)$ and c_X.

In case $c_X > 1$ we fits a $H2(p_1; p_2; \mu_1; \mu_2)$ hyper-exponential distribution with balanced mean (see [8]):

$$\frac{p_1}{\mu_1} = \frac{p_2}{\mu_2}$$

So the parameters of this H_2 distributions are:

$$p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{c_X^2 - 1}{c_X^2 + 1}}\right), \quad p_2 = 1 - p_1,$$

and

$$\mu_1 = \frac{2p_1}{E(X)}, \quad \mu_2 = \frac{2p_2}{E(X)}.$$

For the easier understanding we used for all queues the same SQV rate. In Table 2, we show the results of the simulations and approximations with various parameters.

Parameters	Analytical result	Simulation result	Approximation	Difference
$\lambda = 20$, $\Lambda = 100$	0.425793	0.425706	0.425793	0.000087
$\lambda = 80$, $\Lambda = 100$	0.430135	0.430136	0.430135	0.000001

Table 1: **Exponential distribution**

Arrival intensity	SQV	Simulation result	Approx. result	Difference
$\lambda = 20$ $\Lambda = 100$	0.1	0.423084	0.423129	0.000045
	0.8	0.425127	0.425189	0.000062
	1.2	0.423042	0.426328	0.003286
	1.8	0.421744	0.427979	0.006235
$\lambda = 80$ $\Lambda = 100$	0.1	0.423591	0.423974	0.000383
	0.8	0.428624	0.428725	0.000101
	1.2	0.425821	0.431507	0.005686
	1.8	0.424049	0.435869	0.011820

Table 2: **Approximation**

6. Comments

In this paper we modified the performance model of Proxy Cache Server to a more powerful case when the arrival processes is a GI process and the service times may have any general distribution. To obtain the overall response time we used the QNA approximation method, which was validated by simulation. As we can see in Table 2, when the $SQV < 1$ the overall response time obtained by approximation is very close to response time obtained by simulation; they are the same at least up to the 3–4th decimal digit. In case when the $SQV > 1$ the response times are the same only to 2–3th decimal digit. We can see, when the $SQV = 1.2$ the difference between the response times are 0.005686, and when the $SQV = 1.8$ the difference between the response times is greater (0.01182). So, using greater SQV the approximation error is greater.

Table 3: **Notations**

λ:	arrival rate from the PCS
Λ:	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 second)
R_{xc}:	dynamic server time of the PCS (in byte/second)
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 (in second)
R_s:	dynamic server time of the Web server (in byte/second)
N_s:	server network bandwidth (in bit/second)

References

[1] Atov I., QNA Inverse Model for Capacity Provisioning in Delay Constrained IP Networks. *Centre for Advanced Internet Architectures. Technical Report 040611A* Swinburne University of Technology Melbourne, Australia, (2004).

[2] 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.

[3] 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 8. 13 (1996).

[4] Arlitt, M.A. and Williamson, C.L. Internet Web servers: workload characterization and performance implications. *IEEErACM Transactions on Networking,* 5 (1997), 631–645.

[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] Henk C. Tijms *Stochastic Modelling and Analysis. A computational approuch.* John Wiley & Sons, Inc. New York, (1986).

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

[10] Sanjay K. Bose *An introduction to queueing systems.* Kluwer Academic/Plenum Publishers, New York, (2002).

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

[12] Whitt, W., The Queueing Network Analyzer. *Bell System Technical Journal.*, Vol. 62, No. 9 (1983) 2799–2815.

Tamás Bérczes
Department of Informatics Systems and Networks
University of Debrecen
P.O. Box 12
H-4010 Debrecen
Hungary
e-mail: berczes.tamas@inf.unideb.hu