Modelling and Performance Evaluation of Router Transparent Web cache Mode

Emad Hassan Al-Hemiary

Network Engineering Department, College of Information Engineering, Nahrain University
Baghdad, Iraq
emad@ie-nahrain.org

Abstract— **This paper models and evaluates the performance of router transparent web cache mode in local area networks. A nearby transparent Web cache server intercepts client's requests; forwarded by a layer 3 router or a layer 4 switch, and looks for a copy in its local cache store; if the object requested exits in its local cache, then a cache hit results and the object is forwarded to the router immediately which in turns delivers it to the client. Otherwise, the Web cache establishes a one-time TCP connection with the origin HTTP server and downloads that object. The analytical modelling is carried out using state transition diagrams and open queuing network modelling where the average response time is evaluated for different parameters like probability of cache hit, client HTTP requests (arrival) rate and external arrivals at the origin HTTP server. We also include the delay effects of transmission media between the local network and the Internet where the origin servers reside. Results obtained validate the model and show the significance of deploying Web cache server in a network.**

Keywords—**Web cache model, performance of transparent Web cache, Web cache state transition diagram.**

I. INTRODUCTION

Web caching has been employed to improve the efficiency and reliability of data delivery over the Internet. A nearby Web cache can serve a (cached) page quickly even if the originating HTTP server is swamped or the network path to it is congested [1]. While this argument provides the self-interested user with the motivation to exploit caches, it is worth noting that using widespread use of caches also engenders a general good: if requests are intercepted by nearby caches, then fewer go to the source server, reducing load on the server and network traffic to the benefit of all users.

Besides the advantage of reducing load on webservers, caching frequently visited internet objects reduces ISPs bandwidth effectively. This fact may change when these objects are dynamic (non-cacheable contents) [2]. This is due to the cache-control pragma webservers issue in their headers which in turn cause cache hit rate (percentage number of requested objects fetched from the cache store to the total number of clients requests) to be low when most of clients requests are for dynamic contents. To overcome this limitation, an HTTP violation should be enabled in the Web cache to modify this cache-control pragma and have that object cached.

The other advantage of using web cache servers is related to client speed directly. If we assume that the client has a limited bandwidth controlled by his ISP, then in general the overall speed is enhanced especially for loaded webservers. This is done by allowing temporarily high speed for clients requesting objects from the cache storage.

The most obvious approach, of providing a group of users with a single, shared caching server in standalone mode, has several drawbacks. If the caching machine fails, all users are cut off from the Web. Even while running, a single cache is limited in the number of users it can serve, and may become a bottleneck during periods of intense use. This is directly related to hardware and software used for web caching. The hardware part is the storage media and dynamic memory allocation. The software controls the way web cache reads, writes, swap-out and replace objects. Web cache deployment includes: standalone, router transparent and switch transparent modes [3]. In standalone mode, the Web cache server acts as a gateway between clients and web. In transparent mode, client requests are redirected using a router or a layer 4 switch to Web cache as shown in Fig. 1.

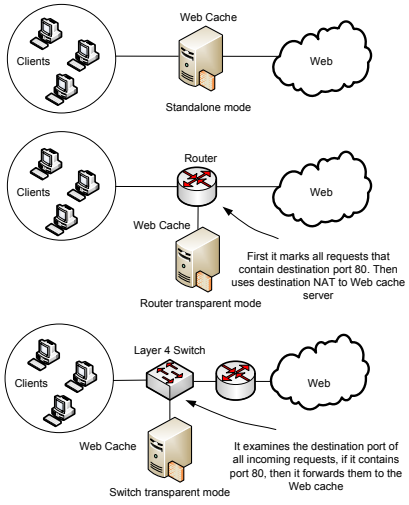

Fig. 1 Web cache deployment methods

II. RELATED WORK

Web content delivery using Web caches has been discussed in the literature for quite long time. There have been recent

researches on using open network queuing models for web cache performance evaluation. The following is a list of related works in this area:

Bose and Cheng [4] examine the impact of installing a proxy cache server (PCS) on overall response time to Web requests. They analyzed how various factors affect the performance of that server. Their analysis is based on modelling the proxy cache server using open queuing network approach and the research specifically identifies a "crossover probability", the minimum cache hit rate probability at which installing a proxy cache server becomes beneficial. Berczes and Sztrik [5] modified the performance model in [4] to a more realistic case when external visits are also allowed to the remote Web servers and the Web servers have limited buffer. They analyzed how many parameters affect the performance of a Proxy Cache Server. Numerical results for the overall response time with and without a PCS are obtained. These results show that the benefit of a PCS depends on various factors. Several numerical examples illustrate the effect of visit rates, visit rates for the external users, and the cache hit rate probability on the mean response times. Berczes [6] evaluates the performance of a PCS when the incoming arrivals and service time are generally distributed. The work considers the network model of [4] as the first step then an expression for the overall system time is evaluated using the proposed distributions. For other literature references, we point to other related references [7]-[10].

In this paper, we evaluate the performance of a transparent Web cache deployed in a local area network where we first model the network using state transition diagrams then we give analytical formulation to the average response time with and without Web cache server.

III. Transparent Web Cache Operation

In transparent Web cache mode, a router or a layer 4 switch is configured to intercept client's request for object of the type "HTTP" (request with destination port 80) and redirect it to a nearby Web cache (called an interceptor cache sometimes) [11]. The latter resolves that request and look for any available "not expired" copy of the requested object in its local cache. If the object exists then a cache hit is resulted and the object is delivered to that client with an overall response time dependent on the Web cache speed and link speed. On the other and, a cache miss is resulted when the requested object is not found in the local cache store which forces the Web cache to contact the origin server and fetches that object, delivered to the client while creating a copy in its local cache for future requests of the same object. The overall response time in the case of cache miss depends on many factors like Web cache speed, link speed between the Web cache and the origin HTTP server, its Network bandwidth and the probability of blocking at the origin HTTP server itself.

We will adopt state transition diagrams to model the network response time with transparent Web cache server. Fig. 2 shows the state transition diagrams of cache hit and cache miss for transparent router mode operation with incoming arrivals from clients denoted by λ. In Fig. 2, the state transition probabilities are given as follows:

p_{80}: the probability that the router detects an HTTP request (port 80) of an arrival and direct it to the Web cache. The rest $(1 - p_{80})$ are directed to the corresponding non-HTTP servers.

p_{hit}: the probability of cache hit, in which the Web cache finds a copy of the requested object in its local cache and delivered it to the router immediately.

$1 - p_{hit}$: the probability of cache miss, in which the Web cache does not find a copy of the requested object in its local cache and forced to download it from the origin HTTP server.

p_{wr}: the probability that the Web cache response back to router with requested object (cache-Miss).

p_{nr}: the probability that the non-HTTP server responds back to the router for non-HTTP requests

p_{hw}: the probability that the HTTP server responds back to Web cache request for cache-Miss requests

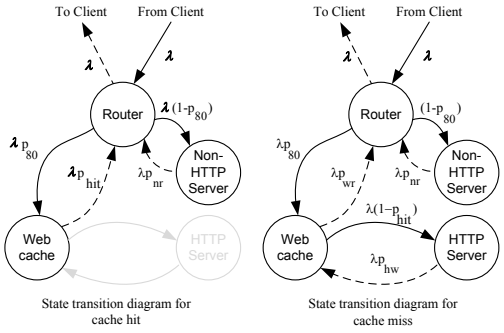

Fig. 2 State transition diagrams of router transparent Web cache mode

The state transition diagram shown in Fig. 2 can be converted into the equivalent open queuing network model shown in Fig. 3. The following section gives detailed analytical formulation to the overall response time based on the queue model given in Fig. 3.

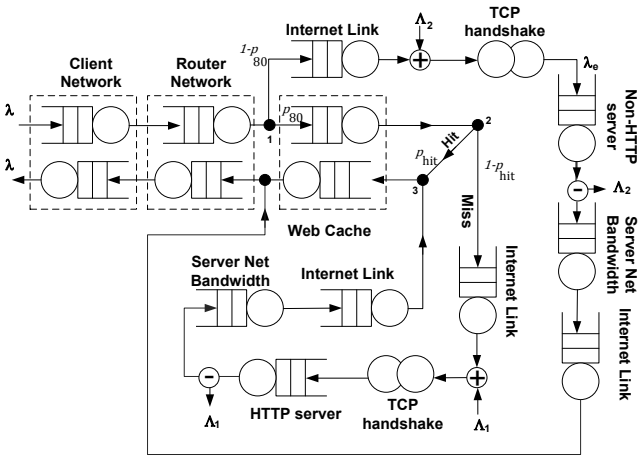

Fig. 3 open queuing network model of router transparent Web cache mode

IV. ANALYTICAL MODELLING

As shown in Fig. 3, the incoming arrivals λ from client network are handled by the gateway router which examines the destination port and route the traffic into two directions. HTTP traffic with probability p_{80} will be routed to the Web cache server and non-HTTP traffic with probability $1-p_{80}$ will be routed to their destination servers (node 1). The Web cache server upon an arrival of a request searches its local cache store for a copy of the requested object, if the object exists then p_{hit} represents the probability of cache hit. Otherwise, $(1-p_{hit})$ represents the probability of cache misses and the requested object is fetched from the origin HTTP server (node 2). Since the connection between the router or/and the Web cache to the origin servers is done through internet, then we include this link as a queue model as shown in Fig. 3. We also assume that this connection starts with a one-time TCP 3-way handshake which is represented by the two circles in Fig. 3. This handshake depends on the round trip time (RTT) of the link between the two ends (i.e., ISP link). Therefore, we have represented each one-way link with a queue to highlight the delay amount this link adds to the overall response time. The performance of the origin servers are characterized by the capacity of its output buffer B, the static server time$S$, and the dynamic server rate D [7,12]. In our model we assume that the server has a buffer of capacity K. Let P_b be the probability that a request will be denied by the HTTP server (i.e., probability of blocking). From basic queuing theory the blocking probability P_b for the M/M/1/K queuing system is given by [13]:

$$P_b = \frac{(1-\rho)\cdot\rho^K}{1-\rho^{K+1}}, \quad \rho = \frac{\lambda_e}{\mu} \tag{1}$$

with [12],

$$\mu = \frac{D_{http}B_{http}}{F(S_{http}D_{http}+B_{http})} \tag{2}$$

gives,

$$\rho = \frac{\lambda_e F(S_{http}D_{http}+B_{http})}{D_{http}B_{http}} \tag{3}$$

Where the subscript $http$ will denote the HTTP server and λ_e is the sum of arrivals from our Web cache server and external arrivals denoted by Λ_1 as shown in Fig. 4. Then:

$$\lambda_e = (1-P_b)[p_{80}(1-p_{hit})\lambda + \Lambda_1] \tag{4}$$

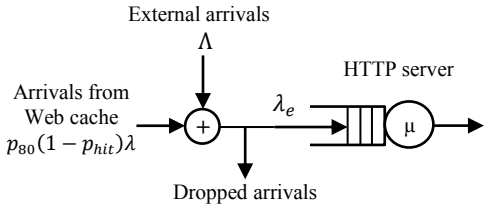

Fig. 4 The origin HTTP server queuing model showing external arrivals

In order to find the response time for the modelled transparent router mode, we must consider the sum of responses from HTTP cache hit, cache miss and non-HTTP requests. Then, the overall response time can be stated as:

$$T_{res} = T_{router} + p_{80}T_{cache} + (1-p_{80})T_s \tag{5}$$

where, T_{router} is the average router system time in seconds, T_{cache} is the average Web cache system time in seconds and T_s is the average non-HTTP system time in seconds

The average response time spent in the router queue T_{router} in Eq. (5) can be represented by an M/M/1 queue. Assuming the processing rate of this router is much more than the arrival rate, then:

$$T_{router} = \frac{1}{u_r - \lambda} \tag{6}$$

where, u_r is the router service rate. Considering a FIFO router queue discipline, incoming requests from clients will have to pass the router input queue then processed by the NAT chain and redirected to the corresponding destination. The same process will occur when a response comes from the Web cache. The second term in Eq. (5) is the average response time for the Web cache for both cache hit and cache miss as shown in Fig 3:

$$T_{cache} = T_{check} + p_{hit}T_{hit} + (1-p_{hit})T_{miss} \tag{7}$$

where, T_{check} is the average Web cache lookup time to check whether the requested object exists in its local cache store or not. This is derived from the waiting time in an M/M/1 queue where the arrivals form a Poisson process with rate $p_{80}\lambda$ and the service rate is u_{ch}. It can be formed as:

$$T_{check} = \frac{1}{u_{ch} - p_{80}\lambda} \tag{8}$$

The second term in Eq. (7) is the response time of a cache hit. This can be obtained by summing the average response time of the Web cache object fetch, delivered to the router and finally to the client network. Thus:

$$T_{hit} = T_{RTT} + \frac{1}{\frac{B_{cache}}{F\cdot\left(S_{cache}+\frac{B_{cache}}{D_{cache}}\right)} - \frac{p_{80}p_{hit}\lambda}{min(B_{cache}/F)}} + \frac{1}{u_r - p_{hit}p_{80}\lambda} + \frac{1}{\frac{N_c}{F} - p_{hit}p_{80}\lambda} \tag{9}$$

Inside the brackets of Eq. (9), the first item is the system time in the Web cache, where B_{cache} is the Web cache output buffer in bytes, S_{cache} is the Web cache static server time in seconds, D_{cache} is the Web cache dynamic server rate in bytes/seconds and F is the average object file size in bytes. The second item in Eq. (9) corresponds to the required time for the router to route the requested object to client. Finally, the last term is the time required to transfer the object over the

client network bandwidth where N_c is the client network bandwidth in bits/seconds.

The last term in Eq. (7) is the average response time when there is a cache miss. Therefore:

$$T_{miss} = \frac{1}{u_l - p_{80}(1-p_{hit})\lambda} + \left\{T_{RTT} + \frac{1}{u_{tcp} - [p_{80}(1-p_{hit})\lambda + \Lambda_1]}\right\} + \frac{1}{\frac{B_{http}}{F\cdot\left(s_{http}+\frac{B_{http}}{D_{http}}\right)} - \frac{\lambda_e}{min(B_{http}/F)}} + \frac{1}{\frac{N_s}{F} - p_{80}(1-p_{hit})(1-P_b)\lambda} + \frac{1}{\frac{B_{cache}}{F\cdot\left(s_{cache}+\frac{B_{cache}}{D_{cache}}\right)} - \frac{p_{80}(1-p_{hit})(1-P_b)\lambda}{\min(B_{cache}/F)}} + \frac{1}{u_r - p_{80}(1-p_{hit})(1-P_b)\lambda} + \frac{1}{\frac{N_c}{F} - p_{80}(1-p_{hit})(1-P_b)\lambda} \quad (10)$$

The reader can notice that the service rate of the Web cache server is the same as the HTTP server! This is due to the fact that the Web cache acts as an HTTP server relative to client when delivering objects, and as a client when connecting to the origin HTTP server. Returning back to Eq. (10), the first term represents the internet link delays where u_l is the link service rate which is a function of the ISP's link speed and object size. When the Web cache does not find a copy of the requested object in its local cache store, it establishes a one-time TCP connection with the origin HTTP server to fetch that object. This operation is modelled by the second term in Eq. (10) where u_{tcp} is the TCP handshake service lookup rate. The response time of the HTTP server for delivering the object is shown in the third term of Eq. (10) and the server network response time is shown in the fourth term, where N_s is the HTTP server network bandwidth. The rest terms in Eq. (10) are the same as those in Eq. (9) which models the response from the Web cache server when it receives the requested object from the HTTP server.

If the size of the requested object is greater than the HTTP server's output buffer it will start a looping process until the delivery of all requested objects is completed [6]. This operation is expressed by the $\min(B_{http}/F)$ value in the third term of Eq. (10) and the first term in Eq. (9).

Finally, the time T_s for non-HTTP requests can be modelled by:

$$T_s = \frac{1}{u_l - (1-p_{80})\lambda} + \left\{T_{RTT} + \frac{1}{u_{tcp} - [(1-p_{80})\lambda + \Lambda_2]}\right\} + \frac{1}{\frac{B_n}{F_n\cdot\left(s_n+\frac{B_n}{D_n}\right)} - \frac{(1-P_b)[(1-p_{80})\lambda+\Lambda_2]}{min(B_n/F_n)}} + \frac{1}{\frac{N_n}{F_n} - (1-p_{80})(1-P_b)\lambda} + \frac{1}{u_r - (1-p_{80})(1-P_b)\lambda} + \frac{1}{\frac{N_c}{F_n} - (1-p_{80})(1-P_b)\lambda} \quad (11)$$

When there is no Web cache server, then the state transition diagram is as shown in Fig. 5 and the average response time can be formed as:

$$T_{res} = T_{router} + p_{80}T_{http} + (1-p_{80})T_s \quad (12)$$

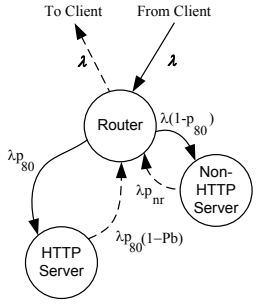

Fig. 5 State transition diagram without transparent Web cache mode

Equation (12) is the same as Eq. (5) except for the second term that corresponds to the response time from the HTTP server. It is given by:

$$T_{http} = \frac{1}{u_l - p_{80}\lambda} + \left\{T_{RTT} + \frac{1}{u_{tcp} - [p_{80}\lambda + \Lambda_1]}\right\} + \frac{1}{\frac{B_{http}}{F\cdot\left(s_{http}+\frac{B_{http}}{D_{http}}\right)} - \frac{(1-P_b)[p_{80}\lambda+\Lambda_1]}{min(B_{http}/F)}} + \frac{1}{\frac{N_s}{F} - p_{80}(1-P_b)\lambda} + \frac{1}{u_r - p_{80}(1-P_b)\lambda} + \frac{1}{\frac{N_c}{F} - p_{80}(1-P_b)\lambda} \quad (13)$$

V. Simulation Results

To evaluate the performance of the modelled router transparent mode Web cache network, we adopt the following parameters value: $B_{cache} = B_{http} = B_n = 32kB$, $F = F_n = 8kB$, $u_r = u_{ch} = u_{tcp} = 250\frac{req}{s}$, $u_l = \frac{4Mbps}{F}$, $S_{cache} = S_{http} = S_n = 16\ \mu s$, $D_{cache} = D_{http} = D_n = 1250\ MBps$, $N_s = 1544\ kbps$, $N_c = 128\ kbps$ $T_{RTT} = 140\ ms$.

We conduct several tests for the modelled network with and without transparent Web cache to evaluate the overall average response time for a client request. The following subsections discuss these tests.

A. *Effect of existence of transparent Web cache and its probability of cache hit*

The first result obtained by evaluating the average response time versus arrival rate λ for different values of probability of hit rates p_{hit} as shown in Fig. 6. The dotted line in Fig. 6 is the average response time for the network without a transparent Web cache, while the other curves are the responses when p_{hit} is varied from 0 to 50% cache hit rates. It is clear that a transparent Web cache with higher cache hit rates improve the overall network response time. For example, 10% cache hit rate has close performance to that without the Web cache. While for λ above 20% the response time dramatically changes to lower values. In practice however, cache hit rate is related to the amount of stored frequently visited objects in Web

cache disks and therefore cannot be configured to have fixed values.

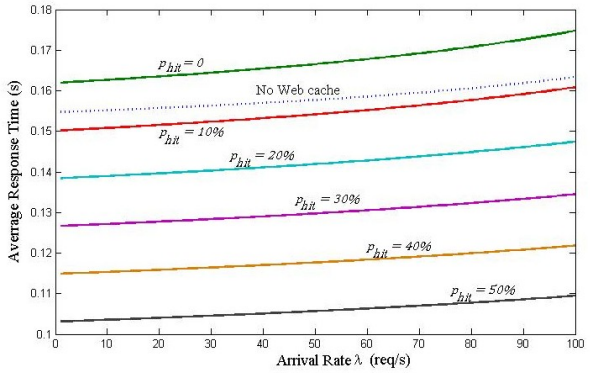

Fig. 6 Average response time versus arrival rate. 80% HTTP, $\Lambda_1 = \Lambda_2$=100, K=100.

B. Effect of External Arrivals rate at the Origin HTTP Server

One of the advantages of deploying Web cache server in a network is to improve the performance of internal clients browsing heavily loaded HTTP servers. Fig. 7 shows the results obtained when we vary the external incoming rate to the HTTP server Λ from 90 to 140 req/s and evaluate the average response time versus λ with $p_{hit} = 20\%$. The upper curves denote performance without Web cache while the lower curves are for transparent Web cache. It is clear that even when the origin HTTP server response time is increased due to increase in incoming external arrivals Λ_1, the average response time for the network with a transparent web cache remains at lower levels.

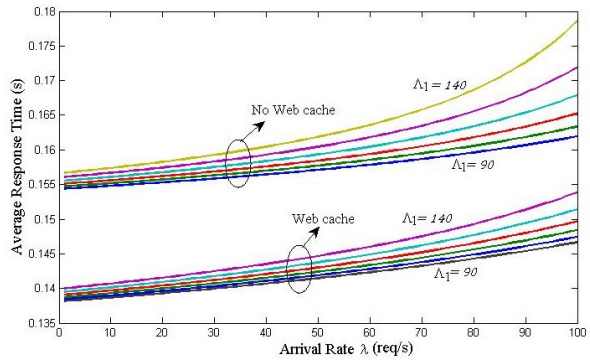

Fig. 7 Average response time versus arrival rate with variable external arrival rates at the origin HTTP server. 80% HTTP, Λ_2=100, 20% cache hit rate, K=100

VI. CONCLUSIONS

In this paper, we use analytical modelling to find the average response time of a network with transparent Web cache using state transition diagrams and open network queuing modelling. To examine this model, we carried numerical tests by varying different parameters like the client arrival rate, probability of cache hit and external arrivals to the origin HTTP server. The modelled network in this work includes effects of transmission delay between the local network and the origin servers, the processing delay time of the router and the effect of client network bandwidth. Verification of the analytical modelling is carried out using numerical analysis where different outcomes are discussed. The results obtained indicate that the network average response time can be enhanced using a transparent Web cache given that the latter one can produce considerable amount of cache hit rates more than of 20% and more. On the other hand, we allowed through analytical modelling that the origin HTTP server can handle external arrival rates from other users over the Internet. When we increase this rate, we found that the performance of the deployed transparent router mode Web cache is satisfactory which confirms one of the advantages of deploying Web cache in a network; that is, reducing the amount of traffic on the origin HTTP servers.

REFERENCES

[1] D. M. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Yoav Yerushalmi, "Web caching with consistent hashing", Int. Journal of Computer and Telecommunications Networking Vol. 31, Issue 11-16, pp. 1203-1213, May 1999.

[2] C. Amza, G. Soundararajan, E. Cecchet and R. Alpes, "Transparent caching with strong consistency in dynamic content web sites", Proc. 19th annual int. conf. on Supercomputing, pp. 264-273, 2005.

[3] G. Barish and K. Obraczke, "World Wide Web caching: trends and techniques", IEEE Comm. Magazine, Vol. 38, Issue 5, pp. 178-184, May 2000.

[4] I. Bose and H. K. Cheng, "Performance models of a firms proxy cache server", Decision Support Systems and Electronic Commerce, Vol. 29, Issue 1, pp. 47-57, July 2000.

[5] T. Berczes and J. Sztrik, "Performance Modeling of Proxy Cache Servers", Journal of Universal Computer Science, vol. 12, no. 9, pp. 1139-1153, 2006.

[6] T. Berczes, "Approximation approach to performance evaluation of Proxy Cache Server systems", Annales Mathematicae et Informaticae, Vol. 36, pp. 15-28, 2009.

[7] T. Berczes, G. Guta, G. Kusper, W. Schreiner and J. Sztrik, "Analyzing a Proxy Cache Server Performance Model with the Probabilistic Model Checker PRISM", Proc. 5th Int. Workshop on Automated Specification and Verification of Web Systems (WWV 2009), 2009.

[8] J. Z. Wang, Z. Du and P.K. Srimani, "Network cache model for wireless proxy caching" IEEE Int. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp. 311-314, 2005.

[9] T. Berczes and J. Sztrik, "A queueing network model to study Proxy Cache Servers, Proc. of the 7th Int. Conf. on Applied Informatics Eger, Hungary, Vol. 1. pp. 203-210, Jan. 2007.

[10] R. Srivastava, "Estimation of Web Proxy Server Cache Size using G/G/1 Queuing Model", Int. Journal of Applied Research on Information Technology and Computing (IJARITAC), Vol. 1, No. 1, pp.46-58, April 2010.

[11] K. Saini, *Squid Proxy Server 3.1*, Packt Publishing Ltd, 2011.

[12] L. P. Slothouber, "A Model of Web Server Performance", Proc. 5th Int. World Wide Web Conference, 1996.

[13] C. G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems*, Springer, 2008.