Admission control for Web server systems - design and experimental evaluation

Robertsson, Anders; Wittenmark, Björn; Kihl, Maria; Andersson, Mikael

Published in:
43rd IEEE Conference on Decision and Control : Nassau, Bahamas, 14 - 17 December 2004

DOI:
10.1109/CDC.2004.1428685

2004

Link to publication

Citation for published version (APA):
https://doi.org/10.1109/CDC.2004.1428685

Total number of authors:
4

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Admission control for web server systems - Design and experimental evaluation

A. Robertsson, B. Wittenmark, M. Kihl, and M. Andersson

Abstract—In communication systems, all service control nodes, such as for example web sites or Mobile Switching Centers, can be modeled as a server system with one or more servers processing the incoming requests. To avoid overload at the service node, some type of admission control mechanism is usually implemented to guarantee good performance also during high traffic loads.

In this paper, we investigate, from a control point of view, the nonlinear discrete-time modeling of a server system, and the analysis and design of load control mechanisms based upon this model.

Verification of the server model behavior with respect to queue theoretic models are made and the load control mechanisms are implemented on an Apache web server and experimentally evaluated.

I. INTRODUCTION

A web site consists of one or more web servers that process the incoming HTTP requests sent by the users. When the arrival rate of new requests increases above the maximum service rate, the queues build up and the response times increase. The same kind of overload problems can occur in any modern communication network, for example in the PSTN, in a GSM- or in a UMTS-network.

A web user experiencing long response times when downloading a page, will most probably abandon the site, leading to profit loss if the site is commercial.

Modern communication networks consist of two types of nodes: switching nodes and service control nodes. The switching nodes enable the transmission of data across the network, whereas the service control nodes contain the service logic and control. The service control node consists of a server system with one or more servers processing incoming calls at a certain rate, see Fig. 1. Each server has a waiting queue where calls are queued while waiting for service. Therefore, a service control node may be modeled as a queuing system including a number of servers with finite or infinite queues. The systems can become overloaded during temporary traffic peaks when more calls arrive than the system is designed for. Since overload in general occurs rather seldom, it is not economical to over-provision the systems for these traffic peaks, instead admission control mechanisms are implemented in the nodes.

Since queuing systems have a stochastic behavior it is difficult to find equations that are simple enough to use in the analysis. A continuous-time nonlinear flow model was first developed by [1] and was further investigated in [2], [3], [4], [5] and [6]. In the references it is shown that the steady-state behavior of the single server queue is well described by the nonlinear flow model.

Different models of queuing systems for control design and analysis have been presented during recent years. These models have either been linearized models of servers or network queues, [7], [8], nonlinear ones based on the flow models with inherent queue properties such as saturations etc. [9], [6], or models derived from system identification methods, [10], [11]. The monograph [12] gives a good overview of modeling concepts and practical design methods for computational systems. In [13] an admission control algorithm was developed using optimal control theory for a discrete-time queuing system with geometrically distributed inter-arrival and service times.

The control objective has traditionally been server utilization or queue lengths in most admission control schemes. Other solutions are to use the processing delay in the system, see [14].

In particular for overload situation, it is shown that linear models of the server system are insufficient to explain the behavior, since the non-linearities in the gate and mainly in the queue affect both the stability and performance properties.

In [15], [16], we developed and validated a control theoretic model of a $G/G/1$-system that can be used for the design of load control mechanisms. In this paper, we use this model for nonlinear analysis and design of controller parameters for a PI-controller.

In Section II, we briefly recapitulate the discrete-time server model and in Section III, we examine the stability
properties for the closed loop system when the admission controller is a PI-controller. Further, the paper contains a discussion about the limitations with both linear control theoretic models of queuing systems and linear design methods. In Section IV the implementation of control algorithms and overload experiments on an Apache web server are described and the experimental results are compared with discrete-event simulations. Finally, there is a discussion about the limitations with both linear control and using the service rate as control signal was recently reported in [17].

We will initially consider an approach based on a linear queue model without any saturation nonlinearities and compare with the admissible control parameters derived from nonlinear analysis. The analysis is based on the Tsypkin/Jury-Lee stability criterion (discrete-time versions of the Popov criterion) [18], [19], [20]. In the analysis only the dominating 'queue-limitation' $\varphi$ will be considered. See Section V for comments on the saturation.

A. Linear approximation model

Under the assumption that the queue is modeled as a pure delay element in feedback, (i.e., assuming $\varphi(z) = z$) and using a standard PI-controller $G_{cl}(z) = K(1 + \frac{1}{T_i} \cdot \frac{h}{z - 1})$ the closed-loop dynamics will be

$$ G_{cl} = \frac{G_c(1 + G_q)G_m}{1 + G_c(1 + G_q)G_m} $$

$$ = \frac{z \cdot K / \sigma (z - 1 + h/T_i)}{z \cdot (z^2 + (K/\sigma - 2)z + (1 - K/\sigma + Kh/(\sigma Ti))} \tag{1} $$

where $G_q$ and $G_m$ represent the queue and monitor dynamics, respectively. To match the characteristic polynomial of (1) with a desired characteristic polynomial

$$ z \cdot (z^2 + a_1 z + a_2) \tag{2} $$

the control parameters should be chosen as

$$ K = (2 + a_1) \sigma, T_i = h(2 + a_1)/(1 + a_1 + a_2) $$

For the linear model it is thus possible to make an arbitrary pole-placement, except for the pole $z = 0$, which corresponds to a time delay.

Choosing $\{K, T_i\}$ such that

$$ a_1 = -2 + K/\sigma, \quad a_2 = 1 - K/\sigma (h/T_i - 1) \tag{3} $$

belong to the stability triangle, [21],

$$ \{ a_2 < 1, \quad a_2 > -1 + a_1, \quad a_2 > -1 - a_1 \} $$

thus predicts stability for the linear closed-loop system.

In next subsection we will consider the influence of the queue nonlinearity $\varphi(\cdot)$. 

Fig. 2. Discrete-time model with controller saturation and saturation $\varphi$ for positive queue lengths.
B. Model with queue limitation

Consider the admission control scheme in Fig. 3 where we have introduced the states \( \{x_1, x_2, x_3\} \) corresponding to the queue length, the (delayed) utilization \( \rho \) and the integrator state in the PI-controller, respectively.

The state space model will be

\[
\begin{align*}
  x_1(kh+h) &= \phi (u+x_1(kh)-\sigma) \\
  x_2(kh+h) &= \frac{1}{\sigma}(u+x_1(kh)) \\
  x_3(kh+h) &= Kh/T_i(\rho_{ref}-x_2(kh))+x_3(kh)
\end{align*}
\]

(5)

where \( u = K(\rho_{ref}-x_2) + x_3 \) and \( \phi(\cdot) \) is the saturation function in Fig. 3.

This system can be decomposed into a linear subsystem \( G_z(z) \) in negative feedback connection with a (new) nonlinearity \( \phi(y) = \phi(y-\sigma(1-\rho_{ref})) \). Note that for \( \rho_{ref} \in [0,1] \) and \( \sigma > 1 \) the nonlinear function \( \phi(\cdot) \) will belong to the same cone as \( \phi(\cdot) \), namely \( [\alpha, \beta] = [0, 1] \). The maximal incremental variation will also be the same (\( \epsilon = 1 \)).

The transfer function \( G_z = G_{uc-yc}(z) \) from cut \( B \) to cut \( A \) in Fig. 3 will be

\[
G_z = C_z(zI - A_z)^{-1}B_z = \frac{-z(z-1)}{z(z^2+(-1+K/\sigma)z+K(h-T_i)/\sigma T_i)}
\]

(6)

C. Stability analysis for discrete-time nonlinear system

The stability analysis for the nonlinear system in Fig. 3 will rely on the property that a sector condition for the nonlinearity is satisfied.

We will use the Tsypkin criterion or the Jury-Lee criterion which are the discrete-time counterparts of the Popov criterion for continuous time systems [22].

Sufficient conditions for stability are that \( G_z \) has all its poles within the unit circle \(|z| < 1\), and that there exists a (positive) constant \( \eta \) such that

\[
\text{Re}[(1+\eta(1-z^{-1}))G_z(z)] + \frac{1}{k} \geq 0 \text{ for } z = e^{j\omega}, \omega \geq 0 \quad (7)
\]

where the nonlinearity \( \phi \) belongs to the cone \([0,k = 1]\).

The condition of \( G_z \) being Hurwitz implies that the coefficients \( \{a_1, a_2\} \) of the characteristic polynomial in 6 should belong to the stability triangle (4) with the following parameter relations

\[
K/\sigma = a_1 + 1, \quad h/T_i = (1+a_1+a_2)/(1+a_1)
\]

(8)

or

\[
a_1 = (-1+K/\sigma), \quad a_2 = K/\sigma \cdot ((h/T_i)-1)
\]

(9)

By comparing Eqs. (9) and (3), we see that for a fixed set of parameters \( \{K, T_i\} \) there will be a unit shift in the \{\(a_1, a_2\}\}-parameters for \( G_{cl} \) and \( G_z \) respectively.

In the upper plot of Fig. 4 we show the stability triangle for the characteristic polynomial of Eq.(1).

We see that control parameters for \( G_{cl} \) which will give coefficients for the characteristic polynomial (1) in the upper left triangle (A1) also will give a stable transfer function \( G_z \). The corresponding poles are plotted in the lower diagram of Fig. 4. In Fig. 5 we show a graphical representation of the Tsypkin condition (7) for this set of control parameters. The green (dashed) non-intersecting line in Fig. 5 corresponds to the existence of a positive parameter \( \eta \) satisfying the frequency condition in Eq.(7), and absolute stability for the nonlinear system is thus also guaranteed for this choice of parameters.

IV. SIMULATIONS AND EXPERIMENTS

To compare simulations with the experiments, a discrete-event simulation program was implemented in C, and the control theoretic models were implemented with the Matlab/Simulink package. The traffic generators in the discrete-time model were built as Matlab programs. They generate arrivals and departures according to the given statistical distributions.

A. Experimental Setup

The admission control mechanism was implemented in the Apache web server [23]. The server was a PC Pentium III 1700 MHz with 512 MB RAM running Windows 2000 as operating system. A new module was added in the Apache core containing our admission control algorithm written in C and was called every time a request was made to the web server. Based on the control signal the arrival request was either rejected or admitted.

To compare the results we present performance metrics such as the server utilization distribution and step responses.
for $G_2$.

The computer representing the clients was a PC Pentium II 400 MHz with 256 MB RAM running RedHat Linux 7.3. Apache 2.0.45 was installed in the server. We used the default configuration of Apache. The client computer was installed with a HTTP load generator, which was a modified version of S-Client [24]. The S-Client is able to generate high request rates even with few client computers by aborting TCP connection attempts that take too long time. The original version of S-Client uses deterministic waiting times between requests. We modified the code to use Poissonian arrivals instead. The client program was programmed to request dynamically generated HTML files from the server. The CGI script was written in Perl. It generates a random number of random numbers, adds them together and returns the summation. The average request rate was set to 100 requests per second in all experiments. In all experiments, the control interval was set to one second.

Fig. 4. (Upper:) The large triangle is the stability area a linear model would predict. However, from this parameter set, nonlinear analysis guarantees only stability for parameters $\{a_1, a_2\}$ in the upper left triangle (A1, ′s′). If $G$ has the coefficients of the characteristic polynomial in $A_1$, $G_2$ will have its parameters in area $A_2$ for the same values of $(K, T)$. (Lower:) Corresponding pole location

Fig. 5. Set of Tsypkin plots which all satisfy the frequency condition (7) for $G_2 = G_2(K, T)$, where $(K, T)$ correspond to pole locations in Fig. 4

B. Model Validation

We have validated that the open-loop system is accurate in terms of average server utilization. For a single-server queue, the server utilization is proportional to the arrival rate which aligns well with the measurements, and the slope of the server utilization curve is given by the average service time. From experiments, an estimate of the average service time in the web server is $1/\mu = 0.0255$. For experiments and simulations with bursty arrival traffic using a two-state Markov Modulated Poisson Process, see Fig. 6.

C. Controller parameters

Control parameters for the PI-controller are chosen from the stability area in Fig. 4 based on a root locus argument. In the simulations and experiments below we use $(K, T) = \{20, 2.8\}$.

D. Evaluation

It is important for the admission control mechanism to meet objectives. Firstly, the control error, $e = \rho_{ref} - \rho$, should have zero mean and small variance. Secondly, the settling time for reference changes should be short.

E. Utilization distribution

The distribution function shows how well the control mechanism meets the first control objective. An ideal admission control mechanism would show a utilization distribution function that is zero until the desired load, and is one thereafter. In Fig. 7 we show the measured (estimated) distribution function from the Apache server together with simulations of controlled $M/M/1$ and the $M/D/1$ systems. The distribution function is estimated from measurements during 1000 seconds. In this case, the load was kept at 80%, and the parameter setting, $(K, T) = \{20, 2.8\}$, results in a controller that behaves very well in this sense. Also, as comparison, results from simulations of the $M/D/1$-system and the $M/M/1$-system are included in Fig. 7, when using $(K, T) = \{20, 2.8\}$. They show that the system behaves as expected. Experimental results with a bursty arrival process (MMPP-2) can be seen in Fig. 8.
Fig. 7. Server utilization distribution of measurements from the real system for two different parameter settings (solid) and (dashed) together with simulations of the M/M/1 (dotted) and the M/D/1 system (dash-dotted).

Fig. 8. Utilization for closed-loop system: Web-server measurements from Fig.7 compared with web-server measurements from a bursty traffic situation using MMPP-2 requests for ratios \((r_1, r_2) = (0.05, 0.95)\).

**F. Step response**

Fig. 9 shows the behavior of the web server during the transient period. The measurements were made on an empty system that was exposed to 100 requests per second. The parameter setting, \(\{K, T_i\} = \{20, 2.8\}\), exhibits a short settling time with a relatively steady server utilization. Comparisons to M/D/1 and M/M/1 simulations, also in Fig. 9, show that the model is accurate.

**V. DISCUSSION**

The analysis in Section III-C gives sufficient conditions and a region for control parameters which guarantee stability of the nonlinear closed loop as well as for the simplified linear model. We are of course not restricted to choose parameters from only this region as the main objective is that the nonlinear system should be stable.

Fig. 10 shows some pole locations where we can use the Tsypkin criterion to show stability for the nonlinear system which could not be predicted by linear system analysis. However, these results are only sufficient and simulations indicate that control parameters which would render the linear system unstable (poles outside the unit circle) and which do not satisfy the Tsypkin criterion actually shows good performance, see Fig. 11.

During simulations the dominant nonlinear effect has come from the queue nonlinearity \(\phi\). The saturation due to limited arrival rate can be handled with a standard implementation of an anti-reset windup scheme, see [6].

The experimental evaluation on the admission controlled Apache server aligns well with both discrete-time and discrete-event simulations.

**VI. CONCLUSIONS**

When investigating server systems, queuing theory has traditionally been used. However, within queuing theory there are few mathematical tools for design and stability analysis of, for instance, admission control mechanisms. Therefore, these mechanisms have mostly been developed with empirical methods. In this paper, we have designed a load control mechanism for a web-server system with control theoretic methods and analyzed its stability properties, taking into account the dominating queue nonlinearity.
We have shown that the PI-controlled system behaves quite well considering transients, stationary behavior, and robustness. The designs have been experimentally verified with simulations and experiments on an Apache web-server system.

REFERENCES