Event-Based Response Time Estimation

Dellkrantz, Manfred; Kihl, Maria; Robertsson, Anders; Åström, Karl Johan

Published in:
[Host publication title missing]

2012

Event-Based Response Time Estimation

Manfred Dellkrantz
Dept of Automatic Control
Lund University, Sweden
manfred@control.lth.se

Maria Kihl
Dept of Electrical and
Information Technology
Lund University, Sweden
maria.kihl@eit.lth.se

Anders Robertsson
Dept of Automatic Control
Lund University, Sweden
andersro@control.lth.se

Karl Johan Åström
Dept of Automatic Control
Lund University, Sweden
kja@control.lth.se

ABSTRACT

Response time is a measure of quality of service in computer systems. Estimation techniques, suitable for support systems for mobile phone systems, are explored. These systems are complex queueing systems with large databases. The traffic generated by users and system administrators changes rapidly, some loads can be measured other cannot. Attempts to capture all details give models that are not suitable for on-line control. Estimators based on continuous flow models with event based measurements are designed using extended Kalman filtering. The estimators are compared with simple-data based estimators.

Keywords

Response time estimation, modeling, event-based, extended Kalman filtering, mobile telephony, multi-activation systems.

1. INTRODUCTION

Resource management of computer systems, which has gained increased attention during recent years, was explored already in the late 60's [3, 7]. It is an essential mechanism to handle load disturbances such as traffic surges and changes in user behavior. Poorly managed resources can severely degrade the performance of a system with potentially large economical consequences.

This paper is motivated by mobile service activation systems, i.e., the systems which the network operators utilize for all processing regarding new subscribers and services in the network. Each new subscriber or service requires processing and data storage in several network nodes. The systems are in general multi-tier systems, implemented as distributed server clusters, where web and application servers process the incoming requests and database servers are used for data storage. The resource management of these systems, based of measurements and feedback of the actual utilization, is crucial for optimization of operation costs and the guarantee of service level agreements during load surges, for example during market campaigns or various events.

Any server system with software that processes requests can basically be modeled as a network of queues which store requests in waiting of service in the processors. Therefore, queuing models can be used to describe the dynamic behavior of server systems [4, 9, 19, 21]. Further, tools from control theory has emerged for both analysis and design of control of these systems [13].

Previous work on resource management for server systems has mainly been focused on the web and application servers. Large software systems have high energy consumption, and therefore, dynamic resource optimization of these systems may considerably lower the operating costs for the network operators [2, 6, 15, 10]. These types of servers has mainly CPU-intensive workload, which can rather easily be modeled as single server queuing systems [4].

Resource management solutions for server systems are usually based on dynamic control schemes, which monitor the systems, and provide actions when needed. Several types of resource management mechanisms have been proposed and evaluated. In larger server systems, load balancing is performed to distribute resources uniformly over computers, CPUs, memory, etc. to avoid that some units are overloaded while others are idle [11, 8]. During overload periods, when more resources are requested than are available, admission control mechanisms reduce the amount of work by blocking some of the requests [5, 18, 16]. Prediction based control have been shown to improve the performance compared to control systems only including feedback [14, 17, 12].

2. MOBILE SERVICE SUPPORT SYSTEM

A Mobile Service Support system (MSS) handles the setup of new subscribers and services into a mobile network (illustration in Figure 1). It presents to the operator and its business support systems a unified middleware where complex functions, such as setting up a new subscriber or modifying services for an existing subscriber, can be easily invoked.

One request to the MSA from an upstream system normally

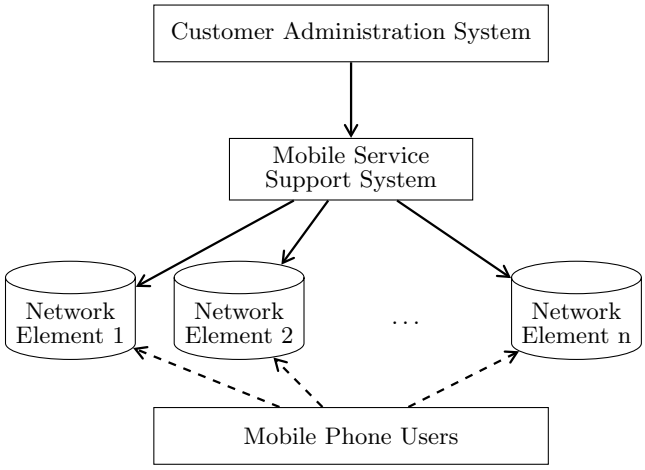

Figure 1: Schematic diagram of a support system for mobile service providers.

results in a number of requests downstream out on the mobile network to several different network elements (NEs). A network element is usually a database storing subscriber and service data, for example, the Home Location Register (HLR). A user id which needs to be fetched from one database needs to be supplied in a query to another database to get the system consistent.

In parallel to the changes and setups that the MSA performs, the network is also used by the end users. Services being set up by the MSA are queried by base stations and other systems requiring that information. In respect to the MSA, this traffic can be considered as an unknown background traffic, in contrast to the known traffic flowing through the MSA. These two loads may interfere with each other, creating a race for resources and may put a too high load on an NE.

One NE that becomes overloaded and unresponsive may result in the entire transaction requiring rollback to avoid inconsistencies in the network. Such a rollback may require manual work which is of course costly for the operator. To protect against such situations, traffic monitoring and control is crucial.

3. MODELING

The system in Figure 1 is complicated with many different queues, caches and databases. Attempting to capture all details give models that are too complex for on-line control. Therefore we will develop simpler models that capture the gross input-output behavior. The models will be evaluated based on the quality of the estimates of the response times.

The input-output behavior of the system can be captured by the response times for each individual request. Since such a model is by nature event based we will make a further simplification by attempting to capture the gross behavior by a continuous flow model. We will recover the event-based behavior in the design of the estimators.

A simple flow model of a queue is given by [1]

$$\frac{dx}{dt} = \lambda - \mu_{max} f(x) \tag{1}$$

where λ is the arrival rate, μ_{max} is the service rate and f a monotone function with the range $[0, 1]$, [1]. The response time is

$$T = t_0(1 + x) = t_0\big(1 + f^{-}(\rho)\big), \tag{2}$$

where $t_0 = {\mu_{max}}^{-1}$ is the average time to serve one customer when the queue is empty and ρ is the normalized service rate or the utility $\rho = \lambda/\mu_{max}$. For the simple M/M/1 queue we have $f = x/(x + 1)$ [20].

If the function f in (1) is monotone the general behavior is that the response time increases with increasing arrival rate. The response time goes to infinity as λ approaches μ_{max} if the function f has the range $[0, 1]$. Since the parameter μ_{max} is uncertain it may be desirable to have models where response rates increase significantly but that they do not go to infinity for finite λ, which can be accomplished by other choices of the function f.

When (1) is used to model an NE in Figure 1 the variable x accounts for the aggregated effect of the storage. Therefore x and T should be interpreted as apparent queue length and response time, they represent the aggregated behavior of many different queues in the real system. It is not possible to measure the apparent queue length directly but the response time can be measured. Requests that enter in a known way can also be used as an inputs.

Linearizing the model around the equilibrium x_e gives a first order system with the time constant

$$\tau = \frac{1}{\mu_{max} f'(x_e)} \geq \frac{1}{\mu_{max}}. \tag{3}$$

The inequality follows from f being monotone and $f(0) = 1$. Notice that the time constant increases significantly with increasing queue length.

4. ESTIMATION

Different ways to estimate the response time from available measurements will now be discussed. There are significant variations in the arrival rate. The response time increases dramatically when the admission rate approaches the capacity of the system. The queue length x in the model (1) cannot be measured directly because it represents an aggregate effect of many queues as discussed in Section 3. It follows from (2) that a measurement of the response time T directly gives the queue length.

4.1 Exponential Smoothing

A simple way to estimate both response time and arrival rate is to use a moving average estimate. Since this estimator does not require a mathematical model it is used as a reference case. The estimator is given by

$$\hat{x}^{+} = \hat{x} + k(x_m - \hat{x}) \tag{4}$$

where x_m is the measured quantity, $\hat{x}$, and $\hat{x}^{+}$ is the estimates before and after an event, and k is the filter gain. The filter can be used to estimate both response time and arrival

rate. The filter coefficient can be chosen to minimize some measure of the error. The filter has the advantage that it does not require any model.

4.2 Kalman Filtering - Known Arrival Rate

In this case it is assumed that the arrival rate is measured and that the arrival time of each request and the time it takes to serve it are measured. There are significant variations in the response time. For a Poisson process the mean value and the variance are the same. A smoothed estimate is required to obtain information that is useful for control.

If the arrival rate λ is known, the apparent queue length can be predicted by the model (1) when there are no events. Hence

$$\frac{d\hat{x}}{dt} = \lambda - \mu_{max} f(\hat{x}) + k\left(T - t_0(1+\hat{x})\right), \quad \hat{T} = t_0(1+\hat{x}), \tag{5}$$

where the initial condition is taken as the estimate obtained at the most recent event. When an event occurs the estimate is updated as

$$\hat{x}^+ = \hat{x} + k(T - \hat{T}) \tag{6}$$

where T is the measured response time, $\hat{x}$ and $\hat{x}^+$ are estimates before and after an event, and k is a filter gain. The filter gain k can be computed if the statistics of $\hat{x}$ and T are known. Since it is unrealistic to assume that this information is available we will instead determine the filter gain from simulation and experiments.

4.3 Kalman Filtering - Unknown Arrival Rate

It is not always the case that all traffic can be controlled, so here we investigate if both arrival rate λ and queue length x can be estimated from measurements of response time. For simplicity we will assume that the arrival rate is constant but unknown or a random walk. Both assumptions lead to the same filter. Linearization of (1) and (2) around the equilibrium x_e, λ_e gives a dynamical system with

$$A = \begin{bmatrix} -\mu_{max} f'(x_e) & 1 \\ 0 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} t_0 & 0 \end{bmatrix}. \tag{7}$$

Estimation is possible because the system is observable. If the measurements were continuous the extended Kalman filter is

$$\begin{aligned} \frac{d\hat{x}}{dt} &= \hat{\lambda} - \mu_{max} f(\hat{x}) + k_1\left(T - t_0(1+\hat{x})\right) \\ \frac{d\hat{\lambda}}{dt} &= k_2\left(T - t_0(1+\hat{x})\right). \end{aligned}$$

When the measurements are event-based the model (1) is used to update the estimate when there are no events. The estimates are given by

$$\frac{d\hat{x}}{dt} = \hat{\lambda} - \mu_{max} f(\hat{x}), \qquad \frac{d\hat{\lambda}}{dt} = 0, \tag{8}$$

where the initial conditions are the estimates $x = \hat{x}^+$ and $\lambda = \hat{\lambda}^+$ obtained after a request has been serviced.

When a measurement of response time T is available the estimates are updated by

$$\begin{aligned} \hat{x}^+ &= \hat{x} + k_1(T - \hat{T}) \\ \hat{\lambda}^+ &= \hat{\lambda} + k_2(T - \hat{T}) \end{aligned} \tag{9}$$

where $\hat{T} = t_0(1+\hat{x})$ from the time at which the request entered the system.

4.4 Kalman Filtering - Two Arrival Streams

A characteristic feature of the system in Figure 1 is that there are two different input streams to the network elements. The stream coming from the service provider side is known but the traffic generated by the users enters the system in many different ways and cannot be measured. To capture this situation we will assume that there are two input streams. One stream is measured and the other is unknown, manifested only through variations in response time. The corresponding flow model is

$$\begin{aligned} \frac{dx}{dt} &= \lambda_c + \lambda_u - \mu_{max} f(x) \\ \hat{T} &= t_0(1+x), \end{aligned} \tag{10}$$

where λ_c is a controllable/known arrival rate and and λ_u is an uncontrollable/unknown arrival rate. Assuming that λ_u is a constant it follows that both x and λ_u are observable from measurements of T and λ_c. The event-based extended Kalman filter is obtained as a simple extension of the filter in Section 4.3.

5. SIMULATION

5.1 Introduction

To test the the estimators we will apply them to a known situation with an M/M/1 queue where many quantities can be evaluated analytically.

The simulated queue server system is a one-server, infinite queue system with exponentially distributed process times with mean ${\mu_{max}}^{-1}$. Jobs arrive at the queue, are finished in FIFO order and are acknowledged upon completion. The jobs were generated as a Poisson processes, with exponentially distributed inter-arrival times.

The arrival rate for the controllable stream coming from the service provider side in Figure 1, $rate_c(t)$, is a combination of a constant and a sine function variation. The arrival rate for the uncontrollable stream coming from the user side, $rate_u(t)$ was set as constant.

$$\begin{aligned} rate_c(t) &= C_c + a\sin(kt) \\ rate_u(t) &= C_u \end{aligned} \tag{11}$$

The parameters were chosen so that the system can handle the workload over long time but with periodic overloads, hence

$$\mu_{max} - a < C_c + C_u < \mu_{max}.$$

The numerical values used in the simulations are

$$\mu_{max} = 100, \qquad C_c = 42.5, \qquad C_u = 42.5, \qquad a = 20. \tag{12}$$

The same realizations were used in all simulations. For experiments with only one fully controllable stream, requests from both streams were directed to the controllable side.

The differential equations describing the behavior of the estimates between events were approximated using first order forward Euler discretization.

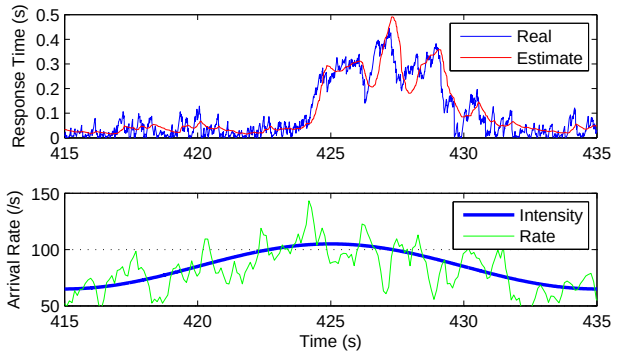

Figure 2: Estimation of response time by exponential smoothing.

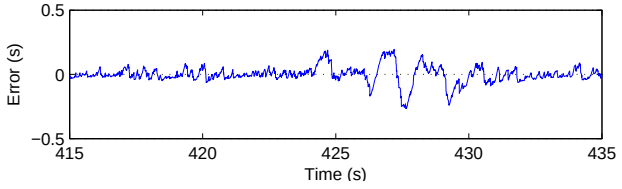

Figure 3: Error of the estimates in Figure 2. The mean square error is $\sigma = 0.018$. Notice the time variability of the error.

5.2 Exponential Smoothing

A simple way to estimate both response time and arrival rate is to use a moving average estimate. Since this estimator does not require a mathematical model it is used as a reference case. The estimator is given by (4). At the departure of a job the estimated queue length is updated as

$$\hat{x}^+ = \hat{x} + k_1 \left(\frac{T}{t_0} - \frac{\hat{T}}{t_0} \right) \tag{13}$$

where T is the measured response time of the request and $\hat{T}$ is the response time estimation from when the request arrived at the system.

The choice of filter gain is a compromise, large values give a fast response with large fluctuations, small values give smoother estimates with slower response. After some experimentation the gain was chosen as $k = 0.03$, which corresponds to a time constant of about 30 events. Figure 2 shows that the simple exponential smoothing estimator gives reasonable results. It gives an efficient smoothing when the response times are small. There is however a lag in the response when the response times are changing significantly for example around times 424 and 427. The arrival rate is around 100 and the time delay is approximately 0.3 s, which matches the time constant of the estimator. The magnitude of the estimation error is very different at different periods the mean square error is $\sigma = 8.4 \cdot 10^{-4}$ in the interval $415 < t < 420$ and $\sigma = 1.5 \cdot 10^{-2}$ in the interval $425 < t < 430$. The mean square error for the entire 600 second experiment is $\sigma = 1.8 \cdot 10^{-2}$. The different behaviors for different queue lengths indicate that it may be useful to schedule the estimator gains.

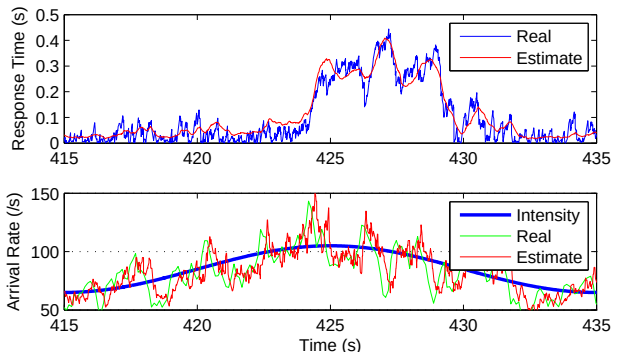

Figure 4: Estimates with known arrival rate. Arrival rates are smoothed and the response time is estimated using an event-based Kalman filter.

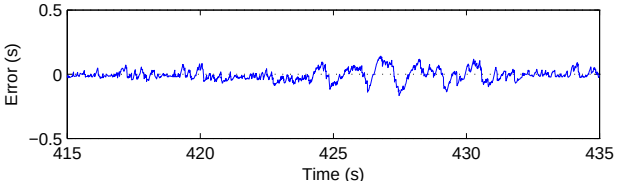

Figure 5: Error of the estimates in Figure 4. The mean square error is $\sigma = 0.0069$. Notice the time variability of the error.

5.3 Known Arrival Rate

Since all traffic passes through the filter, exponential smoothing can be used to estimate the mean inter-arrival time which is the inverse of the arrival rate. This rate is used with the flow model (1) to estimate the response time using an extended Kalman filter. The estimates used on arrival are

$$\begin{aligned} \hat{i}^+ &= \hat{i} + k_3(h_a - \hat{i}) \\ \hat{\lambda}^+ &= (\hat{i}^+)^{-1} \\ \hat{x}^+ &= \hat{x} + h_a \left(\hat{\lambda}^+ - \mu f(\hat{x}) \right) \end{aligned} \tag{14}$$

where $\hat{i}$ is the estimate of the mean inter-arrival time, $\hat{\lambda}$ is the estimate of the arrival rate, $\hat{x}$ is the estimate of the effective queue length, and h_a is the time from the last arrival.

On departure the queue length is updated as

$$\hat{x}^+ = \hat{x} + k_1 \left(\frac{T}{t_0} - \frac{\hat{T}}{t_0} \right) \tag{15}$$

Figure 4 shows the arrival rate and the response time and their estimate. The error of the response time estimate is shown in Figure 5. A comparison with Figure 2 shows that a significant improvement is obtained at the times when the response times changes rapidly. Compare the behaviors around times 424 and 427. The improvement is particularly important to avoid overload during rapid increases in traffic. The magnitude of the estimation error is different at different periods the mean square error is $\sigma = 6.3 \cdot 10^{-4}$ in the interval $415 < t < 420$ and $\sigma = 4.5 \cdot 10^{-3}$ in the interval $425 < t < 430$. The total mean square error is

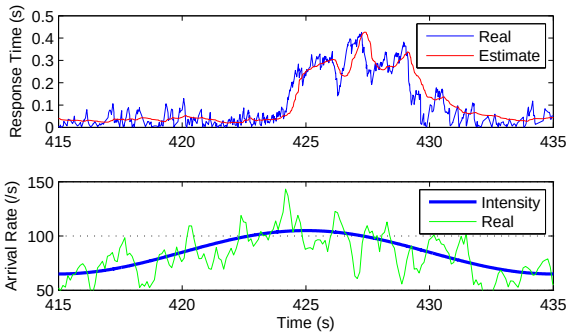

Figure 6: Exponential smoothing response times with estimate, arrival rate has no estimate.

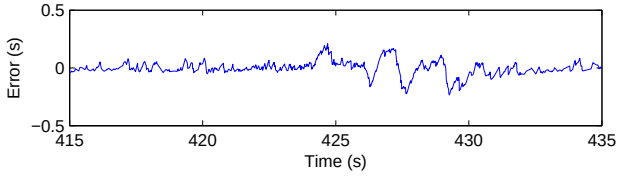

Figure 7: Exponential smoothing prediction error.

$\sigma = 6.9 \cdot 10^{-3}$ which is significantly smaller that the error obtained by the simple exponential smoothing estimate which had $\sigma = 1.8 \cdot 10^{-2}$.

5.4 Two Arrival Streams

In this experiment we separate the two streams of traffic to simulate the two sides of the NEs in Figure 1. One stream passes the observer and one stream enters the shared resource in the background, only showing itself as an added load on the system.

Running this scenario with the simple exponential smoothing estimator presented in section 5.2 results in the response times and estimations shown in Figure 6. The estimation error is shown in Figure 7. Since the filter gets only half the amount of measurements, this situation is not identical to Figure 2. Here the mean square error is $\sigma = 8.6 \cdot 10^{-4}$ for the period $415 < t < 420$ and $\sigma = 1.1 \cdot 10^{-2}$ for the period $425 < t < 430$. The mean square error for the entire experiment is $\sigma = 9.9 \cdot 10^{-3}$.

k_1	k_2	k_3
250	110	0.031

Table 1: Parameters used in this experiment

To try the Kalman filter we use the parameters shown in table 1. The observer follows the inter-arrival times of the controllable traffic using the exponential smoothing described in Section 5.3 and equation (14). The controllable arrival rate is then

$$\hat{\lambda}_c^+ = \hat{i}^{-1}. \tag{16}$$

On every departure we get a response time measurement

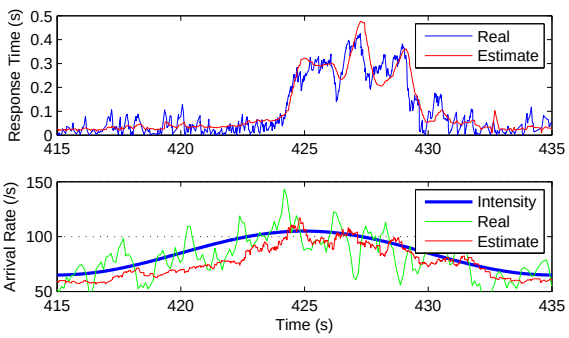

Figure 8: Kalman filter response times with estimate, arrival rate with estimate.

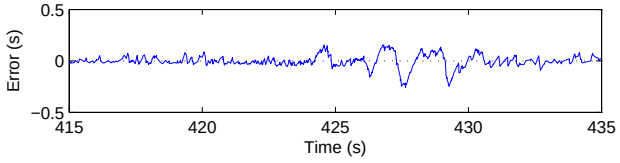

Figure 9: Kalman filter prediction error.

and update the estimation of the queue length and uncontrollable arrival rate as

$$\begin{aligned} \hat{x}^+ &= \hat{x} + h_d \left(\hat{\lambda}_c + \hat{\lambda}_u - \mu f(\hat{x}) + k_1 (T - \hat{T}) \right) \\ \hat{\lambda}_u^+ &= \hat{\lambda}_u + h_d k_2 (T - \hat{T}) \end{aligned} \tag{17}$$

where h_d is the time since the last departure. Figure 8 shows the response times and the arrival rate, both real values and estimates. The estimate error is shown in Figure 9. Once again we can see how the Kalman filter manages to follow the real system during the quick rises in response time around time 424 and 427. Here the mean square error is $\sigma = 7.4 \cdot 10^{-4}$ for the period $415 < t < 420$ and $\sigma = 1.1 \cdot 10^{-2}$ for the period $425 < t < 430$. The mean square error for the entire experiment is $\sigma = 1.9 \cdot 10^{-2}$.

6. FUNDAMENTAL LIMITATIONS

It is useful to know the factors that fundamentally limits how accurate the response time can be estimated. Since response times are stochastic, our best guess is the expected value. If there are n jobs in the system, our best guess will be that the next job will take $t_0 \cdot (1 + n)$, where $t_0 = \mathrm{E}[X], X \sim \mathrm{Exp}(\mu_{max})$. However, it will actually take $\sum_{i=1}^{n+1} X_i, X_i \sim \mathrm{Exp}(\mu_{max})$ which is a stochastic variable. Since the sum of several exponentially distributed variables follows the *Erlang distribution* the expected value of the error as a function of queue length will be

$$E_{err}(n) = \mathrm{E}\left[\left| \mathrm{E}[Y] - Y \right| \right]$$

where $Y \sim \mathrm{Erlang}(n + 1, \mu_{max})$ and n is the number of requests already in the system. This gives us the following calculations for the minimal error:

$$E_{err}(n) = \frac{{\mu_{max}}^{n+1}}{n!} \int_0^\infty \left| \frac{n+1}{\mu_{max}} - x \right| x^n e^{-\mu_{max} x} \mathrm{dx} =$$

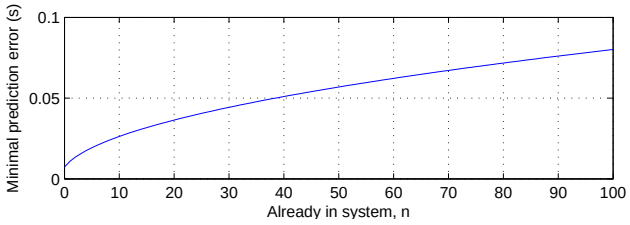

Figure 10: Minimal prediction error for a M/M/1 queue system with $\mu_{max} = 100$.

$$= \frac{2(n+1)^{n+1}}{n!\mu_{max}e^{n+1}}.$$

The smallest value is obtained for $n = 0$.

Figure 10 shows the minimal prediction error as a function of queue length with $\mu_{max} = 100$.

7. SUMMARY

Feedback control is essential for resource management in computer systems. We have investigated several ways of estimating response time which is a key measure of service quality. Simple estimators that do not require models as well as more sophisticated model based schemes have been investigated. The model-based estimators use flow models of the queuing systems and provide event-based estimates using extended Kalman filtering. The estimators have been tested by simulation for scenarios for resource management for mobile telephone operators. The simple model-free estimators give reasonable estimates but the estimates are delayed when the the queue length increases due to system overload. The delay can be reduced by using model-based estimators both in the case of a measured incoming traffic and when the incoming traffic is a mix of known and unknown background traffic.

8. ACKNOWLEDGMENT

This work has been partly funded by the Lund Center for Control of Complex Engineering Systems (LCCC) and the Swedish Research Council grant VR 2010-5864.

9. REFERENCES

[1] Carson E. Agnew. Dynamic modeling and control of congestion-prone systems. *Operations Research*, 24(3):pp. 400–419, 1976.

[2] R. Bianchini and R. Rajamony. Power and energy management for server systems. *IEEE Computer*, 37(11), 2004.

[3] B. Brawn and F. Gustavson. Program behavior in a paging environment. *Proceedings of the AFIPS Fall Joint Computer Conference*, pages 1019–1032, 1968.

[4] J. Cao, M. Andersson, C. Nyberg, and M. Kihl. Web server performance modeling using an m/g/1/k*ps queue. In *International Conference on Telecommunication*, 2003.

[5] X. Chen, H. Chen, and P. Mohapatra. Aces: an efficient admission control scheme for qos- aware web servers. *Computer Communication*, 26(14), 2003.

[6] H. Claussen, L.T.W Ho, and F. Pivit. Leveraging advances in mobile broadband technology to improve environmental sustainability. *Telecommunications Journal of Australia*, 59(1), 2009.

[7] Crocus. *Systemes d'Exploitation des Ordinateurs*. Dunod, Paris, 1975.

[8] Y. Diao, C. Wu, J. Hellerstein, A. Storm, M. Surendra, S. Lightstone, S. Parekh, C. Garcia-Arellano, M. Carroll, L. Chu, and J. Colaco. Comparative studies of load balancing with control and optimization techniques. In *American Control Conference*, 2005.

[9] J. Dilley, R. Friedrich, T. Jin, and J. Rolia. Web server performance measurement and modeling techniques. *Performance Evaluation*, 33(1), 1998.

[10] E. Elnozahy, M. Kistler, , and R. Rajamony. Energy-efficient server clusters. In *Lecture Notes in Computer Science 2325*. Springer-Verlag Berlin Heidelberg, 2003.

[11] Y. Fu, H. Wang, C. Lu, and R. Chandra. Distributed utilization control for real-time clusters with load balancing. In *IEEE International Real-Time Systems Symposium*, 2006.

[12] K. Gilly, C. Juiz, S. Alcaraz, and R. Puigjaner. Adaptive admission control algorithm in a qos-aware web system. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on*, pages 1 –3, sept. 2009.

[13] J. Hellerstein, Y. Diao, S. Parekh, and D. Tilbury. Control engineering for computing systems. *IEEE Control System Magazine*, 25(6), 2005.

[14] D. Henriksson, Y. Lu, and T. Abdelzaher. Improved prediction for web server delay control. In *16th Euromicro Conference on Real-Time Systems*, 2004.

[15] T. Horvath, T. Abdelzaher, K. Skadron, and X. Liu. Dynamic voltage scaling in multitier web servers with end-to-end delay control. *IEEE Transactions on Computers*, 56(4), 2007.

[16] M. Kihl, A. Robertsson, M. Andersson, and B. Wittenmark. Control theoretic analysis of admission control mechanisms for web server systems. *The World Wide Web Journal*, 11(1), 2008.

[17] M. Kjaer, M. Kihl, and A. Robertsson. Resource allocation and disturbance rejection in web servers using slas and virtualized servers. *IEEE Transaction on Network and Service Management*, 6(4), 2009.

[18] X. Liu, J. Heo, L. Sha, and X. Zhu. Adaptive control of multi-tiered web applications using queuing predictor. In *10th IEEE/IFIP Network Operation Mangement Symposium*, 2006.

[19] D. A. Menascé and V. A. F. Almeida. *Capacity Planning for Web Services*. Prentice Hall, 2002.

[20] D. Tipper and M.K. Sundareshan. Numerical methods for modeling computer networks under nonstationary conditions. *Selected Areas in Communications, IEEE Journal on*, 8(9):1682 –1695, dec 1990.

[21] R. D. van der Mei, R. Hariharan, and P. K. Reeser. Web server performance modeling. *Telecommunication Systems*, 16(3), 2001.