A series expansion approach for finite-capacity processor sharing queues

Dieter Fiems
Ghent University
Department TELIN
Ghent, Belgium
dieter.fiems@ugent.be

Koen De Turck
Ghent University
Department TELIN
Ghent, Belgium
koen.deturck@ugent.be

ABSTRACT

This paper investigates a finite capacity queueing system in a Markovian setting with power control and various flavours of processor sharing policies like generalised and discriminatory processor sharing. Allowing for multiple customer classes, the Markovian description of the system at hand obviously suffers from the state space explosion problem. By some numerical examples, we show that numerical Taylor series expansion techniques provide the required numerical accuracy at a limited computational cost. More precisely, the computational complexity of calculating the first N terms in the series expansion of the steady-state probability vector is $O(KMN)$, with M the size of the state space of the Markov chain and K the number of classes.

1. INTRODUCTION

Processor sharing queues and their multi-class extensions like discriminatory and generalised processor sharing queues naturally arise as a convenient abstraction for many networking and computer systems. Processor sharing queues gained particular popularity in modelling flow-level resource sharing in the Internet, see for example [11–14]. Literature on processor-sharing queueing systems is extensive and we refer to the surveys on processor sharing [18] and discriminatory processor sharing [2] for more pointers to the theory and applications of processor sharing queues and limit the present discussion to some recent applications in telecommunication networks.

Alouf et al. [3] use a processor sharing queue to study power saving at the base station and mobile terminals of a cellular network, assuming continuous connectivity. Moreover, Jin and Min [8] investigate an integrated scheduling scheme that combines priority queuing and generalized processor sharing as an efficient mechanism for quality of service differentiation in wireless networks. A generalised processor sharing queue is used to study the trade-off between performance and power consumption in servers hosting virtual machines in [4]. Also in the context of server farms, Tan et al. [15] consider a tandem network of a processor queue and a multi-server queue to assess performance of MapReduce, the popular programming framework for parallelising large data processing on a cluster. Finally, Ferragut and Paganini study a class of M/G processor sharing queues that describe populations and residual workloads in peer to peer file exchange networks [7].

ValueTools'13, December 10 – 12 2013, Turin, Italy

Exact analysis of processor sharing systems with infinite capacity buffers critically depends on structural properties of the Markov chains describing the queueing dynamics which are lost when the service speed (see e.g. [17]) or the arrival rates adapt to the queueing state. In the Markovian setting — that is, assuming Poisson arrivals and exponentially distributed service times — the balance equations of the Markov chain are easily found, even when the server speed adapts to the backlog. However, the size of the state space renders a direct numerical solution infeasible, already for limited queue capacities and a few classes. Therefore this contribution investigates numerical power series expansion techniques. Much like power series expansions extend the range of models that can be solved by analytic methods, these techniques equally well extend the range of Markov models for which a numerical analysis is computationally feasible.

Series expansion methods go by different names, including perturbation techniques, the power series method and light-traffic analysis. While the naming is not absolute, perturbation methods are mainly motivated by sensitivity analysis of the results with respect to some system parameter. The case where the perturbation does not preserve the class-structure of the non-perturbed chain — the so-called singular perturbations — has received much attention in literature [1, 10]. The power series method transforms a Markov chain of interest in a set of Markov chains parametrised by a variable γ. For $\gamma = 0$, the chain is not only easily solved, but one can also obtain the series expansion in γ. For $\gamma = 1$ one gets the original Markov chain such that the series expansion can be used to approximate the solution of the original Markov chain, provided the convergence region of the series expansion includes $\gamma = 1$ [16]. Finally, light-traffic analysis often corresponds to a series expansion in the arrival rate at a queue. For an overview on the technique of series expansions in stochastic systems, we further refer the reader to the surveys in [5, 9].

The contribution focusses on a multi-class processor sharing queue with power saving and relies on numerical series expansion techniques to assess performance of these systems. In particular, we consider light-traffic expansions, as well as

VALUETOOLS 2013, December 10-12, Torino, Italy

DOI 10.4108/icst.valuetools.2013.254394

expansions for the service rate $\mu \to 0$, which corresponds to extreme overload. For finite capacity queueing systems, the limit $\mu \to 0$ (or equivalently for the arrival rate $\lambda \to \infty$) adds additional insight in the queueing dynamics [6]. Moreover, the different expansions are combined in a single expression. Secondly, we consider both expansions for the steady-state vector and for the mean waiting time vector (conditioned on the arrival state) under the per-class random order of service (ROS) discipline, the computational complexity of these expansions being comparable. Notice that a ROS discipline is natural in the context of processor sharing as it corresponds (in the Markovian setting) to the case where all customers within a class get an equal share of the processor capacity assigned to the class.

The remainder of this paper is organised as follows. The next section introduces the multi-class queueing model at hand and sets the main notational conventions. The series expansion technique is then introduced in Section 3 and applied to the calculation of the steady state probability vector and the mean waiting times. This section also discusses the computational complexity of the approach at hand. Finally, the approach is numerically investigated in Section 4 and conclusions are drawn in Section 5.

2. MODELLING ASSUMPTIONS

We consider a multi-class Markovian queueing system as depicted in Figure 1. The system consists of K finite capacity queues; let C_k denote the capacity of the kth queue, $k \in \mathcal{K} = \{1, \ldots, K\}$. As we consider a Markovian queueing system, the state of the queueing system is completely described by the number of customers in each of the queues. For further use, let $\mathcal{C}_k = \{0, 1, \ldots, C_k\}$ be the set of possible queue content of queue k and let $\mathcal{C} = \mathcal{C}_1 \times \ldots \times \mathcal{C}_K$ be the state space of the Markov chain at hand. Also, let $\mathbf{c} = [C_1, \ldots, C_K]$ be the full state and let $\mathbf{0} = [0, \ldots, 0]$ be the empty state. Arrival and departure rates at and from the different queues may depend on the queueing state. That is, let $\mathbf{x} = [x_1, \ldots, x_K]$ be the state vector (x_k is the queue content of queue k), then the arrival and departure rates equal,

$$\lambda_k(\mathbf{x}) = \lambda \alpha_k(\mathbf{x}), \quad \mu_k(\mathbf{x}) = \mu \beta_k(\mathbf{x}).$$

Here λ and μ are scale parameters of the arrival process and service process, respectively. $\alpha_k(\mathbf{x})$ denotes the (unscaled) arrival rate at queue k when the system state is $\mathbf{x}$ and $\beta_k(\mathbf{x})$ denotes the (unscaled) service rate allocated to queue k in state $\mathbf{x}$. For notational convenience, we assume $\alpha_k(\mathbf{x}) = 0$ for $x_k = C_k$ and $\beta_k(\mathbf{x}) = 0$ for $x_k = 0$. In addition, these rates adhere to the following constraints:

(A.1) In the absence of service ($\mu = 0$), the queueing system fills up completely. That is, $\sum_{k=1}^{K} \alpha_k(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{c}$.

(A.2) In the absence of arrivals ($\lambda = 0$), the queueing system empties completely. That is, $\sum_{k=1}^{K} \beta_k(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{0}$.

In the model description above, there are but few assumptions with respect to arrival and service rates. The method outlined in the next section holds for any such system but we here focus on processor sharing disciplines with power saving. Power saving is accomplished by reducing the service rate when there are but few customers in the queue.

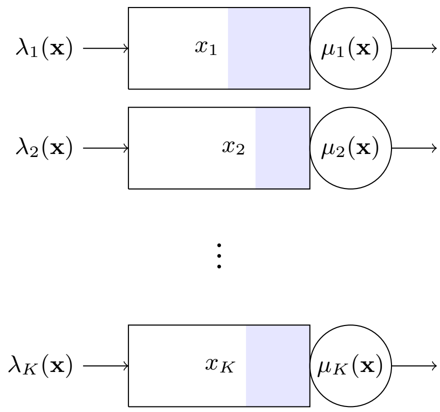


Figure 1: Coupled queueing system

We assume that the service rate $\gamma(\mathbf{x})$ only depends on $\mathbf{x}$ through the weighted queue content $q(\mathbf{x})$,

$$q(\mathbf{x}) = \sum_{i=1}^{K} w_i x_i .$$

Here $\{w_i, i = 1, \ldots, K\}$ is a set of fixed weights. For discriminatory and generalised processor sharing, the rates are:

- Discriminatory processor sharing (DPS):

$$\beta_k(\mathbf{x}) = \gamma(\mathbf{x}) \frac{g_k x_k}{\sum_{\ell=1}^{K} g_\ell x_\ell} .$$

- Generalized processor sharing (GPS):

$$\beta_k(\mathbf{x}) = \gamma(\mathbf{x}) \frac{h_k \mathbb{1}_{\{x_k > 0\}}}{\sum_{\ell=1}^{K} h_\ell \mathbb{1}_{\{x_\ell > 0\}}} .$$

Here g_k and h_k are fixed weights. Finally, the arrival rates in the different queues do not depend on the queueing state,

$$\alpha_k(\mathbf{x}) = p_k \mathbb{1}_{\{x_k < C_k\}} .$$

Remark 1. As can be seen from the modelling assumptions and from the constraints (A.1) and (A.2) above, many more queueing systems fit the assumptions. Apart from processor sharing, one can for example consider strict priority queueing,

$$\beta_k(\mathbf{x}) = \mathbb{1}\{\sum_{\ell=1}^{k-1} x_\ell = 0\} ,$$

and independent service, $\beta_k(\mathbf{x}) = g_k \mathbb{1}\{x_k > 0\}$. The arrival rates can also adapt to the queueing state. A prime example of an adaptive arrival rate is the join the shortest queue routing policy,

$$\alpha_k(\mathbf{x}) = \frac{\mathbb{1}\{x_k = m(\mathbf{x})\}}{n(\mathbf{x})} ,$$

with $m(\mathbf{x})$ the queue content of the shortest non-full queue, and $n(\mathbf{x})$ the number of such queues. Analogously, customers can join proportionally to the available buffer space policy,

$$\alpha_k(\mathbf{x}) = (C_k - x_k) / \sum_{\ell=1}^{K} (C_\ell - x_\ell) .$$

Other policies include "join the shortest local queue" (select the shortest among some neighbouring queues), join the queue with most space, etc.

3. SERIES EXPANSION TECHNIQUE

Having introduced the modelling assumptions, we now focus on Taylor series expansions of the stationary distribution and of the (conditional) waiting times. We first establish the balance equations and the mean sojourn time equations.

3.1 System equations

Recall that $\beta_k(\mathbf{x}) = 0$ for $x_k = 0$ and that $\alpha_k(\mathbf{x}) = 0$ for $x_k = C_k$: there cannot be a departure from queue k if its empty and there cannot be an arrival in queue k if its fully occupied. Let $\pi(\mathbf{x})$ be the steady state distribution of the Markov chain at hand. In view of the modelling assumptions, we easily find,

$$\pi(\mathbf{x})\sum_{k=1}^{K}(\lambda_k(\mathbf{x})+\mu_k(\mathbf{x})) = \sum_{k=1}^{K}\pi(\mathbf{x}+\mathbf{e}_k)\mu_k(\mathbf{x}+\mathbf{e}_k) + \sum_{k=1}^{K}\pi(\mathbf{x}-\mathbf{e}_k)\lambda_k(\mathbf{x}-\mathbf{e}_k)\,. \quad (1)$$

Here we assume $\pi(\mathbf{x}) = \alpha_k(\mathbf{x}) = \beta_k(\mathbf{x}) = 0$ for $\mathbf{x} \notin \mathcal{C}$ to simplify the balance equation. Further, $\mathbf{e}_k$ is a row vector of zeros apart from its kth element which equals 1.

Apart from the steady state distribution of the joint queue content, we also study mean sojourn times of customers, given the state of the system when they arrive. Let W be the remaining sojourn time of a random customer, let Q be the queue in which this customer resides and let $\mathbf{X}$ be the state of the queueing system. We then define,

$$w(\mathbf{x};i) = \mathrm{E}[W\mathbf{1}_{\{\mathbf{X}=\mathbf{x}\}}|Q=i]\,.$$

We consider the expectations of the sojourn times under ROS scheduling (within the class). By conditioning on the first event (arrival or departure in any of the queues) during the sojourn time, we find the following recursion for the class i sojourn times,

$$w(\mathbf{x};i) = \frac{1-\mu_i(\mathbf{x})/x_i w(\mathbf{x}-\mathbf{e}_i;i)}{\sum_{k=1}^{K}(\mu_k(\mathbf{x})+\lambda_k(\mathbf{x}))} + \frac{\sum_{k=1}^{K}(\mu_k(\mathbf{x})w(\mathbf{x}-\mathbf{e}_k;i)+\lambda_k(\mathbf{x})w(\mathbf{x}+\mathbf{e}_k;i))}{\sum_{k=1}^{K}(\mu_k(\mathbf{x})+\lambda_k(\mathbf{x}))}\,, \quad (2)$$

for all $\mathbf{x} \in \{\mathbf{x} \in \mathcal{C} : x_i > 0\}$ and with $w(\mathbf{x};i) \doteq 0$ for $x_i = 0$ to simplify notation. The expansions of the mean conditional sojourn times are motivated by game-theoretic analysis of observable queueing systems. Indeed, if one is merely interested in the mean sojourn time, the expansion of the mean queue content above and Little's result immediately result in the expansion of the mean sojourn time. In addition, we note that the method exemplified above also allows for expansions of higher-order conditional moments which cannot be obtained by Little's theorem.

3.2 Series expansions for $\pi(\mathbf{x})$

Assuming but a few classes and limited buffer space, the balance equations and waiting time equations introduced above can be easily solved numerically. However, even for only a few classes and moderate buffer size, direct computation of these stationary measures is no longer computationally feasible due to the size of the state space. To overcome the state space explosion problem, series expansion techniques are used which allow to approximate these steady state measures. We consider expansions in λ around $\lambda = 0$ (light traffic) and in μ around $\mu = 0$ (overload).

Before proceeding, we note that such expansions are justified. Indeed, a finite Markov chain that reduces to a Markov chain with a single absorbing state in the limit, permits an analytic series expansion. In the present case, this is an immediate consequence of the fact that the steady-state probabilities are rational functions of μ (λ), $\mu = 0$ ($\lambda = 0$) not being a pole. Hence, the constraint (A.1) on the Markov chain introduced above is sufficient to guarantee the existence of an analytic expansion in μ of the steady state solution in a neighbourhood of $\mu = 0$ as there is a single absorbing state for $\mu = 0$. Analogously, constraint (A.2) guarantees the existence of an analytic expansion in λ around $\lambda = 0$.

3.2.1 Light-traffic expansion

Let $\widehat{\pi}_n(\mathbf{x})$ be the nth coefficient in the series expansion of $\pi(\mathbf{x})$ around $\lambda = 0$. That is,

$$\pi(\mathbf{x}) = \sum_{n=0}^{\infty}\widehat{\pi}_n(\mathbf{x})\lambda^n\,.$$

Plugging the series expansion into the balance equation (1), and comparing terms in λ^n yields,

$$\widehat{\pi}_0(\mathbf{x})\sum_{k=1}^{K}\mu_k(\mathbf{x}) = \sum_{\substack{k=1\\x_k<C_k}}^{K}\widehat{\pi}_0(\mathbf{x}+\mathbf{e}_k)\mu_k(\mathbf{x}+\mathbf{e}_k)\,, \quad (3)$$

and,

$$\widehat{\pi}_n(\mathbf{x})\sum_{k=1}^{K}\mu_k(\mathbf{x}) + \widehat{\pi}_{n-1}(\mathbf{x})\sum_{k=1}^{K}\alpha_k(\mathbf{x}) = \sum_{\substack{k=1\\x_k<C_k}}^{K}\widehat{\pi}_n(\mathbf{x}+\mathbf{e}_k)\mu_k(\mathbf{x}+\mathbf{e}_k) + \sum_{\substack{k=1\\x_k>0}}^{K}\widehat{\pi}_{n-1}(\mathbf{x}-\mathbf{e}_k)\alpha_k(\mathbf{x}-\mathbf{e}_k)\,, \quad (4)$$

for all $\mathbf{x} \in \mathcal{C} \setminus \{\mathbf{c}\}$. Moreover, the normalisation condition $\sum_{\mathbf{x}\in\mathcal{C}}\pi(\mathbf{x}) = 1$ yields,

$$\sum_{\mathbf{x}\in\mathcal{C}}\widehat{\pi}_0(\mathbf{x}) = 1\,, \quad \sum_{\mathbf{x}\in\mathcal{C}}\widehat{\pi}_n(\mathbf{x}) = 0\,, \quad (5)$$

for $n > 0$. These equations allow for recursive calculation of the terms of the series expansion. The first term of the series expansion is,

$$\widehat{\pi}_0(\mathbf{x}) = \mathbf{1}_{\{\mathbf{x}=\mathbf{0}\}}\,.$$

That is, in the absence of arrivals, the queueing system is empty. For the higher order terms ($n > 0$) we have,

$$\widehat{\pi}_n(\mathbf{x}) = \frac{\sum_{k=1,x_k<C_k}^{K}\widehat{\pi}_n(\mathbf{x}+\mathbf{e}_k)\mu_k(\mathbf{x}+\mathbf{e}_k)}{\sum_{k=1}^{K}\mu_k(\mathbf{x})} - \frac{\widehat{\pi}_{n-1}(\mathbf{x})\sum_{k=1}^{K}\alpha_k(\mathbf{x})}{\sum_{k=1}^{K}\mu_k(\mathbf{x})}$$

$$+\frac{\sum_{k=1,x_k>0}^{K}\widehat{\pi}_{n-1}(\mathbf{x}-\mathbf{e}_k)\alpha_k(\mathbf{x}-\mathbf{e}_k)}{\sum_{k=1}^{K}\mu_k(\mathbf{x})}\,, \quad (6)$$

for $\mathbf{x}\neq\mathbf{0}$ and

$$\widehat{\pi}_n(\mathbf{0}) = -\sum_{\mathbf{x}\in\mathcal{C}\setminus\{\mathbf{0}\}}\widehat{\pi}_n(\mathbf{x})\,.$$

Recursive calculation is indeed possible. If the $n-1$st expansion is known, we can calculate all probabilities by iterating the state space in inverse lexicographical order; see the right-hand side of (6).

REMARK 2. We have chosen to express the balance equations, rather then giving an expression for the generator matrix of the Markov chain. We opted for this approach as this more closely aligns with the software implementation of the algorithm. This may however somewhat obfuscate why a series expansion approach is feasible and a direct solution approach is not. To make this clear, note that the balance equations (1) can be expressed as,

$$\boldsymbol{\pi}(\mathcal{A}+\lambda\mathcal{B}) = 0\,,$$

where $\boldsymbol{\pi}$ is the steady state vector and $\mathcal{A}$ and $\mathcal{B}$ are known matrices that do not depend on λ. Moreover, $\mathcal{A}$ is an upper triangular matrix as all transitions to higher values in the state space (lexicographically ordered) depend on λ. Let $\boldsymbol{\pi}_n$ be the nth term in the series expansion, we then have,

$$\boldsymbol{\pi}_n\mathcal{A} = -\boldsymbol{\pi}_{n-1}\mathcal{B}\,,$$

or,

$$\boldsymbol{\pi}_n = -\boldsymbol{\pi}_{n-1}\mathcal{B}\mathcal{A}^{-1}\,.$$

Inversion of $\mathcal{A}$ is trivial as this is a triangular matrix.

3.2.2 Overload

Let $\widetilde{\pi}_n(\mathbf{x})$ be the nth coefficient in the series expansion of $\pi(\mathbf{x})$ around $\mu=0$. That is,

$$\pi(\mathbf{x}) = \sum_{n=0}^{\infty}\widetilde{\pi}_n(\mathbf{x})\mu^n\,.$$

Plugging the series expansion above into the balance equation (1), and comparing terms in μ^n yields,

$$\widetilde{\pi}_0(\mathbf{x})\sum_{\substack{k=1\\x_k<C_k}}^{K}\lambda_k(\mathbf{x}) = \sum_{\substack{k=1\\x_k>0}}^{K}\widetilde{\pi}_0(\mathbf{x}-\mathbf{e}_k)\lambda_k(\mathbf{x}-\mathbf{e}_k)\,, \quad (7)$$

and,

$$\widetilde{\pi}_n(\mathbf{x})\sum_{\substack{k=1\\x_k<C_k}}^{K}\lambda_k(\mathbf{x}) + \widetilde{\pi}_{n-1}(\mathbf{x})\sum_{\substack{k=1\\x_k>0}}^{K}\beta_k(\mathbf{x}) = \sum_{\substack{k=1\\x_k<C_k}}^{K}\widetilde{\pi}_{n-1}(\mathbf{x}+\mathbf{e}_k)\beta_k(\mathbf{x}+\mathbf{e}_k) + \sum_{\substack{k=1\\x_k>0}}^{K}\widetilde{\pi}_n(\mathbf{x}-\mathbf{e}_k)\lambda_k(\mathbf{x}-\mathbf{e}_k)\,, \quad (8)$$

for $n>0$. Moreover, as for the light-traffic case, the normalisation condition of the steady-state vector yields,

$$\sum_{\mathbf{x}\in\mathcal{C}}\widetilde{\pi}_0(\mathbf{x}) = 1\,, \quad \sum_{\mathbf{x}\in\mathcal{C}}\widetilde{\pi}_n(\mathbf{x}) = 0\,. \quad (9)$$

The first term of the series expansion is,

$$\widetilde{\pi}_0(\mathbf{x}) = \mathbf{1}_{\{\mathbf{x}=\mathbf{c}\}}\,,$$

as all queues are full in the absence of service. Moreover, the following terms can be calculated by iterating over the state space in lexicographical order,

$$\widetilde{\pi}_n(\mathbf{x}) = \frac{\sum_{k=1,x_k<C_k}^{K}\widetilde{\pi}_{n-1}(\mathbf{x}+\mathbf{e}_k)\beta_k(\mathbf{x}+\mathbf{e}_k)}{\sum_{k=1,x_k<C_k}^{K}\lambda_k(\mathbf{x})} + \frac{\sum_{k=1,x_k>0}^{K}\widetilde{\pi}_n(\mathbf{x}-\mathbf{e}_k)\lambda_k(\mathbf{x}-\mathbf{e}_k)}{\sum_{k=1,x_k<C_k}^{K}\lambda_k(\mathbf{x})} - \frac{\widetilde{\pi}_{n-1}(\mathbf{x})\sum_{k=1,x_k>0}^{K}\beta_k(\mathbf{x})}{\sum_{k=1,x_k<C_k}^{K}\lambda_k(\mathbf{x})}\,, \quad (10)$$

for $\mathbf{x}\neq\mathbf{c}$ and,

$$\widetilde{\pi}_n(\mathbf{c}) = -\sum_{\mathbf{x}\in\mathcal{C}\setminus\{\mathbf{c}\}}\widetilde{\pi}_n(\mathbf{x})\,.$$

3.3 Expansions for $\mathrm{w}_i(\mathbf{x})$

We now consider the mean waiting times. For overload, particular care needs to be taken as customers will wait forever when there is no service.

3.3.1 Light-traffic expansion

We first consider the expansion of $w(\mathbf{x};i)$ in λ around $\lambda=0$. We assume the following expansion,

$$w(\mathbf{x};i) = \sum_{n=0}^{\infty}\widehat{w}_n(\mathbf{x};i)\lambda^n\,.$$

Plugging the former expansion in (2) and comparing terms in λ^n yields,

$$\widehat{w}_0(\mathbf{x};i)\sum_{k=1}^{K}\mu_k(\mathbf{x}) = 1 - \mu_i(\mathbf{x})/x_i\widehat{w}_0(\mathbf{x}-\mathbf{e}_i;i) + \sum_{k=1}^{K}\mu_k(\mathbf{x})\widehat{w}_0(\mathbf{x}-\mathbf{e}_k;i)\,, \quad (11)$$

and,

$$\widehat{w}_n(\mathbf{x};i)\sum_{k=1}^{K}\mu_k(\mathbf{x}) = \sum_{k=1}^{K}\left(\mu_k(\mathbf{x})\widehat{w}_n(\mathbf{x}-\mathbf{e}_k;i) + \alpha_k(\mathbf{x})\widehat{w}_{n-1}(\mathbf{x}+\mathbf{e}_k;i)\right) - \widehat{w}_{n-1}(\mathbf{x};i)\sum_{k=1}^{K}\alpha_k(\mathbf{x}) - \frac{\mu_i(\mathbf{x})}{x_i}\widehat{w}_n(\mathbf{x}-\mathbf{e}_i;i)\,. \quad (12)$$

Plugging in $\mathbf{x}=\mathbf{e}_i$ in the former equations yields,

$$\widehat{w}_0(\mathbf{e}_i;i) = \frac{1}{\mu_i(\mathbf{e}_i)}\,,$$

and,

$$\widehat{w}_n(\mathbf{e}_i;i) = \frac{\sum_{k=1}^{K} \alpha_k(\mathbf{e}_i)\left(\widehat{w}_{n-1}(\mathbf{e}_i+\mathbf{e}_k;i) - \widehat{w}_{n-1}(\mathbf{x};i)\right)}{\mu_i(\mathbf{e}_i)} .$$

Given these boundary expectations, equations (11) and (12) then allow for recursively calculating $w_n(\mathbf{x};i)$ for all $\mathbf{x}$ and n.

3.3.2 Overload expansion

We now consider the expansion of $w(\mathbf{x};i)\mu$ in the case of overload as sojourn times grow unbounded in the absence of service. By means of Little's law, it is easy to show that $\lim_{\mu\to 0} w(\mathbf{x};i)\mu$ is finite for all $\mathbf{x}$ and i. Hence, in contrast to the preceding expansions, we now consider an expansion of the form,

$$w(\mathbf{x};i) = \sum_{n=-1}^{\infty} \widetilde{w}_n(\mathbf{x};i)\mu^n .$$

It turns out that for the overload expansion, it takes a little more work to determine the boundary expectations $\widetilde{w}_n(\mathbf{c};i)$ as can be seen below.

As before, plugging the former expansion in (2) and comparing terms in μ^n yields,

$$\widetilde{w}_{-1}(\mathbf{x};i)\sum_{k=1}^{K}\lambda_k(\mathbf{x}) = \sum_{k=1}^{K}\lambda_k(\mathbf{x})\widetilde{w}_{-1}(\mathbf{x}+\mathbf{e}_k;i) , \qquad (13)$$

and,

$$\sum_{k=1}^{K}(\widetilde{w}_{n-1}(\mathbf{x};i)\beta_k(\mathbf{x}) + \widetilde{w}_n(\mathbf{x};i)\lambda_k(\mathbf{x})) = \mathbb{1}_{\{n=0\}} - \beta_i(\mathbf{x})/x_i\widetilde{w}_{n-1}(\mathbf{x}-\mathbf{e}_i;i) + \sum_{k=1}^{K}(\beta_k(\mathbf{x})\widetilde{w}_{n-1}(\mathbf{x}-\mathbf{e}_k;i) + \lambda_k(\mathbf{x})\widetilde{w}_n(\mathbf{x}+\mathbf{e}_k;i)) , \qquad (14)$$

for $n \ge 0$.

Plugging $\mathbf{x} = \mathbf{c} - \mathbf{e}_\ell$ in (13) yields,

$$\widetilde{w}_{-1}(\mathbf{c}-\mathbf{e}_\ell;i) = \widetilde{w}_{-1}(\mathbf{c};i) . \qquad (15)$$

In view of this expression, evaluating (14) in $\mathbf{x} = \mathbf{c}$ and $n = 0$, and solving for $w_{-1}(\mathbf{c};i)$ yields,

$$\widetilde{w}_{-1}(\mathbf{c};i) = \frac{C_i}{\beta_i(\mathbf{c})} . \qquad (16)$$

Equation (13) then allows to recursively calculate $w_{-1}(\mathbf{x};i)$ for all $\mathbf{x}$.

We proceed analogously for the higher-order terms. Evaluating (14) in $\mathbf{x} = \mathbf{c}$, we have,

$$\widetilde{w}_n(\mathbf{c};i)\sum_{k=1}^{K}\beta_k(\mathbf{c}) = -\beta_i(\mathbf{c})/C_i\widetilde{w}_n(\mathbf{c}-\mathbf{e}_i;i) + \sum_{k=1}^{K}\beta_k(\mathbf{c})\widetilde{w}_n(\mathbf{c}-\mathbf{e}_k;i) , \qquad (17)$$

whereas evaluation in $\mathbf{x} = \mathbf{c} - \mathbf{e}_\ell$ gives,

$$\widetilde{w}_n(\mathbf{c}-\mathbf{e}_\ell;i) = \kappa_{i,\ell,n} + \widetilde{w}_n(\mathbf{c};i) , \qquad (18)$$

with

$$\kappa_{i,\ell,n} = \Big(\mathbb{1}_{\{n=0\}} - \sum_{k=1}^{K}\widetilde{w}_{n-1}(\mathbf{c}-\mathbf{e}_\ell;i)\beta_k(\mathbf{c}-\mathbf{e}_\ell) - \beta_i(\mathbf{c}-\mathbf{e}_\ell)/(C_i - \mathbb{1}_{\{\ell=i\}})\widetilde{w}_{n-1}(\mathbf{c}-\mathbf{e}_\ell-\mathbf{e}_i;i) + \sum_{k=1}^{K}\beta_k(\mathbf{x})\widetilde{w}_{n-1}(\mathbf{c}-\mathbf{e}_\ell-\mathbf{e}_k;i)\Big)(\lambda_\ell(\mathbf{c}-\mathbf{e}_\ell))^{-1} . \qquad (19)$$

Note that $\kappa_{i,\ell,n}$ can be calculated once the terms up to order $n-1$ have been calculated. Plugging (18) in (17) and solving for $\widetilde{w}_n(\mathbf{c};i)$ yields,

$$\widetilde{w}_n(\mathbf{c};i) = \frac{C_i}{\beta_i(\mathbf{c})}\sum_{k=1}^{K}\beta_k(\mathbf{c})\kappa_{i,k,n} - \kappa_{i,i,n} , \qquad (20)$$

Equation (14) then allows to recursively calculate $w_n(\mathbf{x};i)$ for all $\mathbf{x}$.

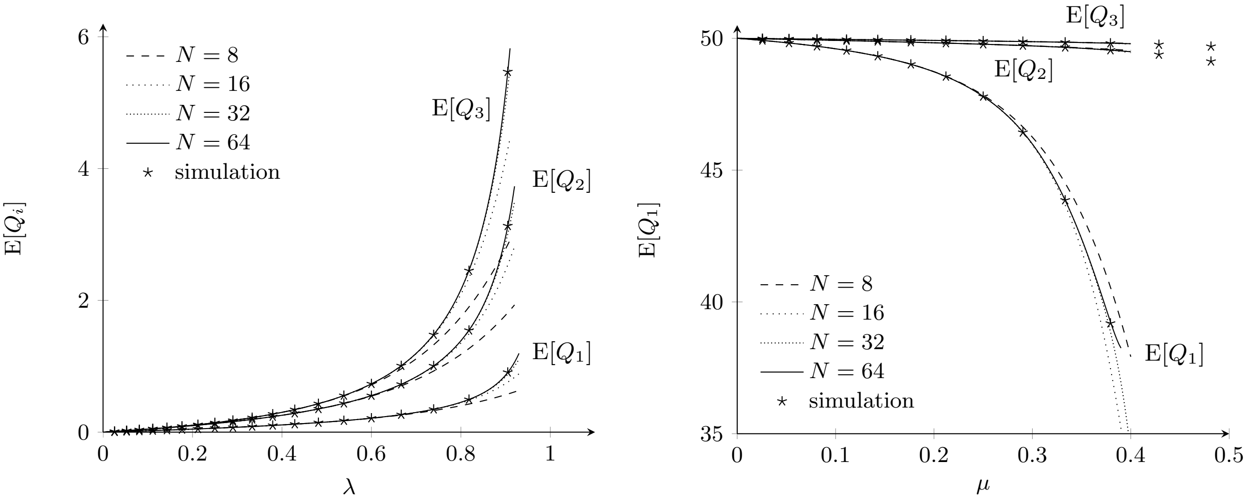
3.4 Computational complexity

From the equations above, calculation of a single term in the expansion takes about $O(K)$ additions and one division. Hence, the computational complexity of calculating π_n (or w_n) is $O(KM)$, with $M = |\mathcal{C}|$ the size of the state space. Having the same complexity for every additional term in the expansion, calculating the first N coefficients then has complexity $O(KMN)$.

As the size of the state space is very large, limited memory consumption is equally important. To limit memory consumption to the size of storing only one steady-state vector one can proceed as follows. Assuming one is mainly interested in the expansion of a number of performance measures, note that once the nth term of the expansion of the steady state vector is determined, the corresponding terms in the expansions of various performance measures can be determined as well. Hence, there is no need to keep track of previous terms of the expansion of steady-state probabilities unless they are required for further calculations of coefficients. From the equations above one sees that for the light-traffic approximation only values of the preceding order are used of states which are lexicographically larger. Hence, as we calculate in lexicographical order, we can overwrite the preceding vector during our calculations. Memory for only one vector of size M is needed.

4. NUMERICAL RESULTS

Having established the numerical procedures, we now evaluate them by some numerical examples. We first consider our approximations for DPS and GPS queues without power saving. In figure 2 we depict the mean queue content of the different classes of a DPS queue with three classes. The weights are $g_1 = 4$, $g_2 = 2$ and $g_3 = 1$ and the (unscaled) arrival load for the different classes are set to $\alpha_1 = 0.2$, $\alpha_2 = 0.4$ amd $\alpha_3 = 0.4$. The queue capacity of every class is $C_i = 50$ such that the size of the state space is 132651. Figures 2(a) and 2(b) depict the mean queue content in light-traffic and in overload, respectively. Approximations up to order $N = 64$ are displayed and compared with simulation results. The simulation uses uniformisation of the Markov chain and we have generated 10^8 events, which ensures that the simulation value is not visually discernible from the real value. The calculation time of the approximations is indeed fast: about $0.13s$ per term in the expansion on a 2.93 GHz



(a) Light traffic (b) Overload

Figure 2: Light traffic approximation by a series expansion around $\lambda = 0$ (a) and overload approximation by a series expansion around $\mu = 0$ (b).

Intel Core 2 processor.

A key issue with series expansions is the establishment of the region in which the expansions are sufficiently accurate. From a theoretical perspective, establishing this region is not trivial. First, the region where the expansion converges to the stationary distribution is bounded and may not span the entire region of interest (for λ and μ). It is easy to show that the stationary probabilities are rational functions of λ (or μ) such that the pole of these rational functions with smallest modulus determines the region of convergence. In view of the order of the denominator, this pole cannot easily be determined. Secondly, even if convergence can be established for some λ or μ, bounds on the error term are not easily found. Therefore, we here rely on the following simple heuristic. Let $f_N(x)$ be the Nth order expansion in x ($x = \mu$ or $x = \lambda$), we then accept our Nth order approximation provided,

$$\frac{|f_{2N}(x) - f_N(x)|}{f_{2N}(x)} < \epsilon \,,$$

or equivalently,

$$1 - \epsilon < \frac{f_N(x)}{f_{2N}(x)} < 1 + \epsilon \,.$$

Let Ω_N denote the region where these inequalities hold (the implicit dependency on ϵ and on the performance measure under consideration will be clear from the context). In figure 2, the curves are only depicted in Ω_{64} with $\epsilon = 0.1$. As can be observed, the simulation results visually match the $N = 64$-approximation in the displayed regions. Further notice that Ω_{64} for the light-traffic expansions is about twice the size of Ω_{64} for the overload expansions. Nevertheless, the regions are still considerable: up to more than 80% load for a light-traffic approximation and for loads of 250% and more for the overload expansion.

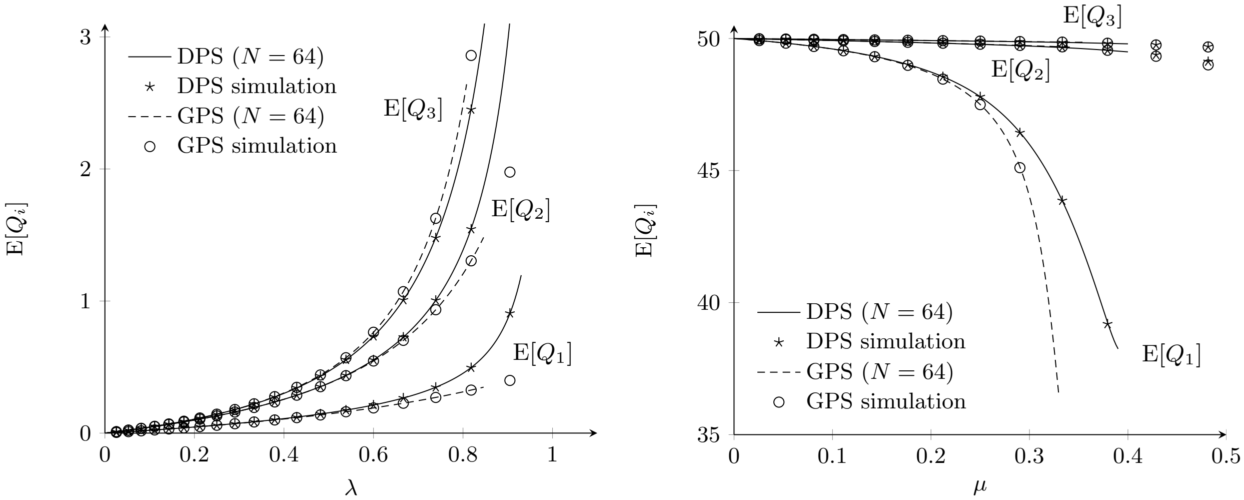
Assuming the same parameters as in figure 2, figure 3 compares DPS and GPS. The same weights for DPS and GPS are used ($g_1 = h_1 = 4$, $g_2 = h_2 = 2$, $g_3 = h_3 = 1$). We now only depict the $N = 64$ approximation in Ω_{64} and verify the approximations by simulation. Again, the visual match is very good, the often small differences between GPS and DPS being accurately depicted. Again, we can note differences in the Ω_{64} regions, the regions for GPS being slightly smaller than the corresponding region for DPS.

In figure 4, we consider the effect of power saving on DPS performance in light traffic. The assumptions on arrivals, queue sizes and DPS weights are retained but we now adapt the processor speed to the total queue size. In figure 4(a), the processor rate increases from 1/2 to 1, proportionally to the total queue size. In figure 4(b), a threshold policy is adopted: the server serves at full capacity when there are more than 10 customers in the queue, and serves at half capacity if this not the case. As for the preceding examples, our approximation is compared with simulation results and no difference between approximation and simulation result can be discerned. Compared to the preceding examples, it should be noted that the Ω_{64} regions are notably smaller.

5. CONCLUSION AND FUTURE WORK

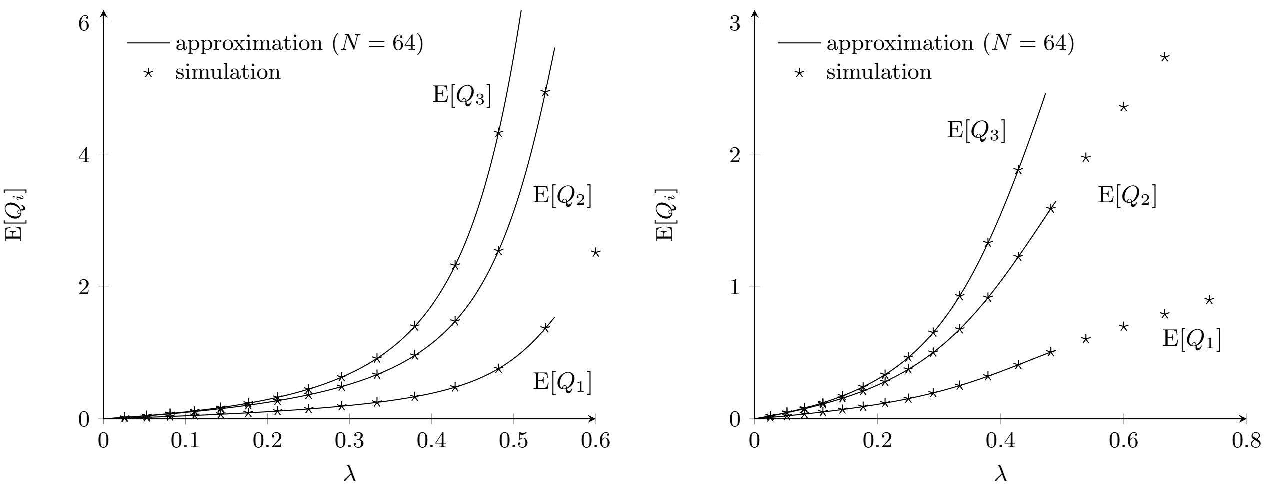
In this paper we investigated numerical series expansion techniques to assess performance of DPS and GPS queues with finite capacity. We showed that the computational complexity is $O(KMN)$ with K the number of classes, M the size of the state space and N the order of the expansions considered. Numerical experimentation confirmed that the series expansion technique is fast. Establishing the region in which expansions are accurate is a key issue with series expansions techniques. We therefore proposed a simple heuristic which compares the Nth and $2N$th order expansions. Several numerical experiments confirmed that performance assessment is accurate in the heuristically determined region.

The modelling approach allows for several extensions, some of which are discussed below. Foremost, the processor sharing queue at hand can be seen as a network of queues with a common server. The method proposed here is not limited to multi-class queues but can be applied to any acyclic network of finite capacity queues with state-dependent arrival



(a) Light traffic (b) Overload

Figure 3: Comparison of DPS and GPS in light-traffic (a) and overload (b).



(a) linear (b) threshold

Figure 4: Comparison of linear (a) and threshold-based (b) power saving in light-traffic.

and service rates. Key is that the state space of the queueing system is ordered such that arrivals correspond to increasing state transitions and service to decreasing state transitions. For acyclic networks of queueing systems, such an ordering is possible. As the network is acyclic, the queues can be indexed such that departures in a queue only yield arrivals in queues with higher indices. By using this indexation for the state vector and lexicographical ordering of the state space, an arrival and a departure indeed correspond to an increase and a decrease respectively. In terms of conditional sojourn times, the methodology for the conditional mean sojourn time vector easily extends to higher order moments, although the recursion will be more involved.

Acknowledgment

The second author is a Postdoctoral Fellow with the Research Foundation, Flanders, Belgium. This research was partially funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

6. REFERENCES

[1] E. Altman, K.E. Avrachenkov, R. Nunez-Queija. Perturbation analysis for denumerable Markov chains with application to queueing models. *Advances in Applied Probability* 36(3):839–853, 2004.

[2] E. Altman, K. Avrachenkov, U. Ayesta. A survey on discriminatory processor sharing. *Queueing Systems* 53(1–2):53–63, 2006.

[3] S. Alouf, V. Mancuso, N.C. Fofack. Analysis of power saving and its impact on web traffic in cellular networks with continuous connectivity. *Pervasive and Mobile Computing* 8:646–661, 2012.

[4] J. Anselmi, I. Verloop. Energy-aware capacity scaling in virtualized environments with performance guarantees. *Performance Evaluation* 68(11):1207–1221, 2011.

[5] B. Błaszczyszyn, T. Rolski, V. Schmidt. *Advances in Queueing: Theory, Methods and Open Problems*, chapter Light-traffic approximations in queues and related stochastic models. CRC Press, Boca Raton, Florida, 1995.

[6] K. De Turck, E. De Cuypere, S. Wittevrongel, D. Fiems. Algorithmic approach to series expansions around transient Markov chains with applications to paired queuing systems. Proceedings of the 6th International Conference on Performance Evaluation Methodologies and Tools (Valuetools), Cargese, France, 2012.

[7] A. Ferragut, F. Paganini. Content Dynamics in P2P Networks from Queueing and Fluid Perspectives Proc. of 24th International Teletraffic Congress, Krakow, Poland. pp. 145–152, 2012.

[8] X.L. Jin, G.Y. Min. Modelling and Analysis of an Integrated Scheduling Scheme with Heterogeneous LRD and SRD Traffic. *IEEE Transactions on Wireless Communications* 12(6):2598–2607, 2013.

[9] I. Kovalenko. Rare events in queueing theory. A survey. *Queueing systems* 16(1):1–49, 1994.

[10] J.B. Lasserre. A Formula for Singular Perturbations of Markov Chains *Journal of Applied Probability* 31(3):829–833, 1994.

[11] L. Massoulié, J.W. Roberts. Bandwidth sharing: objectives and algorithms. In: Proc. of IEEE Infocom, New York NY, USA, pp. 1395–1403, 1999.

[12] L. Massoulié, J.W. Roberts. Bandwidth sharing and admission control for elastic traffic. Telecommunication Systems 15:185–201, 2000.

[13] M. Nabe, M. Murata, H. Miyahara. Analysis and modeling of World Wide Web traffic for capacity dimensioning of Internet access lines. *Performance Evaluation* 34:249–271, 1998.

[14] J.W. Roberts. A survey on statistical bandwidth sharing. *Computer Networks* 45:319–332, 2004.

[15] J. Tan, S.C. Meng, X.Q. Meng, L. Zhang. Delay Asymptotics for Heavy-tailed MapReduce Jobs In: Proc. of 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton IL, USA, pp. 1637–1639, 2012.

[16] W.B. van den Hout. The power-series algorithm. PhD Thesis. University of Tilburg. 1996.

[17] A. Wierman, L.L.H. Andrew,A. Tang. Power-aware speed scaling in processor sharing systems: Optimality and robustness *Performance Evaluation* 69:601–622, 2012.

[18] S.F. Yashkov, A.S. Yashkova. Processor Sharing: A Survey of the Mathematical Theory *Automation and Remote Control* 68(9):1662–1731, 2007.