A Survey on Discriminatory Processor Sharing*

E. Altman[†], K. Avrachenkov[†], U. Ayesta[‡]

[†]INRIA Sophia Antipolis, France
{k.avrachenkov,altman}@sophia.inria.fr

[‡]CWI, The Netherlands
ayesta@cwi.nl

Abstract

The Discriminatory Processor Sharing (DPS) model is a multi-class generalization of the egalitarian Processor Sharing model. In the DPS model all jobs present in the system are served simultaneously at rates controlled by a vector of weights $\{g_k > 0; k = 1, \ldots, K\}$. If there are N_k jobs of class k present in the system, $k = 1, \ldots, K$, each class-k job is served at rate $g_k / \sum_{j=1}^{K} g_j N_j$. The present article provides an overview of the analytical results for the DPS model. In particular, we focus on response times and numbers of jobs in the system.
Keywords. Discriminatory Processor Sharing, Asymptotic Analysis, M/G/1, Conservation Law.

1 Introduction

Conventional telephone networks provide a guaranteed service rate by allocating dedicated channels to accepted calls. In contrast, in data networks and multi-task operating systems the capacity is time-shared between all users and as a consequence the service rate allocated to each user depends on the total number of users. This technological development created a need for new mathematical models that could capture the fundamental properties of time-sharing systems. Kleinrock [32] introduced the Processor Sharing (PS) model where the capacity is shared equally among all the users present in the system. Since then, the PS model has attracted significant attention from the research community. For available results on the PS model we refer the interested reader to [32, 27, 51] and [9] in the present special issue.

*This work is part of a French-Dutch *Van Gogh* research project funded by NWO (The Netherlands Organization for Scientific Research) and EGIDE under grant VGP 61-520. This work was done while U. Ayesta was an ERCIM Postdoc fellow at CWI.

One limitation of the PS model lies in its inability to describe heterogeneous time-sharing systems, in which users from different classes obtain unequal shares of the capacity. For example, in the Internet the service rate a connection obtains depends on the characteristics of its path [34] and in modern CPUs the service rate of each task depends on its priority level [5, 47]. To model such situations multi-class time-sharing systems were proposed. Two main models have emerged: Generalized Processor Sharing (GPS) and Discriminatory Processor Sharing (DPS). The GPS model guarantees a minimum service rate to each class. When there are no jobs in one of the classes, its share of the capacity is distributed among the active classes. We refer to [40, 50] and references therein for more details on GPS. Unlike in GPS, in the DPS model the capacity that each class obtains is not guaranteed, and in fact the capacity allocated to each class depends on the number of jobs currently present in all the classes. The DPS model was proposed and studied by Kleinrock [31] under the name Priority Processor Sharing. In the DPS model all jobs present in the system are served simultaneously at rates controlled by a vector of weights $\{g_k > 0; k = 1, \dots, K\}$, where K denotes the number of classes. If there are N_k jobs of class k present in the system, $k = 1, \dots, K$, each class-k job is served at rate

$$r_k(N_1, N_2, ..., N_K) = \frac{g_k}{\sum_{j=1}^{K} g_j N_j}. \tag{1}$$

When all the weights are equal, the DPS model is equivalent to the standard PS system. By changing the DPS weights, one can effectively control the instantaneous service rates of different job classes. Thus, the possibility of providing different service rates to users of various classes makes DPS an appropriate model to study the performance of heterogeneous time-sharing systems.

After the work of Kleinrock, the paper by Fayolle, Mitrani and Iasnogorodski [16] made the most important advance in the analysis of the DPS model. In [16] the authors obtained the expected conditional response times as the solution of a system of integro-differential equations. In addition, the authors provided a thorough analysis for the case of exponentially distributed service requirements. Since the appearance of [16] until recently, publications on DPS have been very sparse. Despite the simplicity of the model description and the fact that the properties of the egalitarian PS queue are quite thoroughly understood, the analysis of DPS has proven to be extremely difficult. For example, results on an important basic metric like the distribution of the response time in the system have only been derived under certain limiting regimes (time-scale decomposition, overload etc.) Similarly, results for general service requirements are also scarce.

The range of applications of DPS is broad. Perhaps the most natural application of DPS is to model Weighted-Round-Robin (WRR) scheduling. DPS mimics the performance of WRR when the service quotas are smaller than the service requirements and when the switching delay associated with the change of job is negligible. Similarly, DPS may be an appropriate model for a modified version of WRR known as Deficit-Round-Robin [46]. The last few years have witnessed a significant increase of interest in the DPS model, partly motivated by the need to understand the performance of future communication networks that are expected to provide different quality of service to different groups of users. The first paper discussing the applications of DPS in the context of flow-level performance of bandwidth sharing mechanisms was [34], where by numerical means the authors discussed the advantages and disadvantages of discriminatory bandwidth sharing in the Internet. For more applications of DPS in communication networks see [12, 20, 1, 14, 25].

The survey is organized as follows. Section 2 introduces the notation used throughout the survey. We have followed a thematic approach for the exposition of technical results. Section 3 reviews the results on the expected conditional response time. Section 4 contains the results on the moments of the unconditional response time and numbers of jobs in the system. Section 5 presents the results on DPS in heavy-traffic and overload regimes. Finally, Section 6 reviews the results that aim at characterizing the achievable performance of DPS as a function of the weights. Throughout the paper, we state results without proofs, and when available we provide intuitive explanations.

2 Notation

Let $k \in \mathcal{K}$ be a job class index, where $\mathcal{K} = \{1, \ldots, K\}$ is the set of indices. Unless otherwise stated, we assume that the arrival process of class k is a Poisson process with rate λ_k. Thus the aggregated job arrival process is also Poisson with rate $\lambda = \sum_{j=1}^{K} \lambda_j$. Let X_k, $k = 1, \ldots, K$, denote the random variable corresponding to the service requirements of class-k jobs. We denote by $F_k(\cdot)$ the distribution of X_k, and we use $\overline{F}_k(x) = 1 - F_k(x)$ to denote the complementary service requirement distribution. We use $\overline{x_k^i} = E[X_k^i]$ to denote its i-th moment. For notational ease we write $\overline{x_k} = E[X_k^1]$. Unless otherwise specified, the service requirement distribution will be general. In the particular case of exponentially distributed service requirements, we will use the notation $\mu_k = 1/\overline{x_k}$.

Let τ_k be a random variable that denotes the response time (time in the system) for an arbitrary class-k job. Let $T_k^i = E[\tau_k^i]$ denote the i-th moment of the response time and let $T_k = E[\tau_k^1]$. Let $\tau_k(x)$ be the response time in steady state of a class-k job that requires x units of service. We denote by $T_k(x) = E[\tau_k(x)]$ the expected conditional response time of a class-k job whose service requirement is x and let $T_k'(x)$ be its derivative (existence is shown in Section 3). Similarly, let n_k be a random variable that denotes the number of class-k jobs in the system. Let $L_k = E[n_k]$ denote the mean number of class-k jobs and let $L_{ij} = E[n_i n_j]$, where $1 \leq i, j \leq K$. Note that by Little's law we have $L_k = \lambda_k T_k$. In sample-path arguments, we will use $N_k(t)$, $k = 1, \ldots, K$, to denote the number of class-k jobs in the system at time t.

When required, the superscript π will be added to emphasize the dependency on the particular scheduling policy π.

Since the DPS queue is work-conserving (see Section 6.1), the steady state exists in the non-saturated regime, i.e., when $\rho = \sum_{j=1}^{K} \rho_j < 1$, where $\rho_k = \lambda_k \overline{x_k}$

for $k = 1, \ldots, K$. With the exception of Section 5, it will always be assumed that the DPS queue operates in the stable regime, i.e., $\rho < 1$.

3 Expected Conditional Response Time

The expected conditional response times $T_k(x)$, $k = 1, \ldots, K$, can be found as a solution of a system of integro-differential equations. Such a system was first derived by O'Donovan [39], but unfortunately it contained an error. Then, after the error was communicated to him [37], the corrected form of the equations was presented in [16]. The derivation of these equations was inspired by similar equations that Kleinrock *et al.* [33] derived (see also [32]) for a processor sharing queue with batch Poisson arrivals. The development of the integro-differential equations relies on the so-called "tagged job" approach, in which one keeps track of the evolution of the system from the arrival until the departure of a tagged job. Let us tag a class-k job with service requirement greater than x. We note that $T_k(x)$ can also be interpreted as the average time needed for a class-k job in order to get x units of service. Then, in view of equation (1), for sufficiently small Δ the expected conditional response time satisfies

$$T_k(x + \Delta) = T_k(x) + \Delta + \sum_{j=1}^{K} \frac{g_j}{g_k} \Delta L_j(x) + o(\Delta),$$

where $L_j(x)$ is the expected number of class-j jobs in the system when the tagged job has attained service x. Note that while the tagged job obtains Δ units of service, a class-j job obtains $\frac{g_j}{g_k}\Delta$ units of service. Taking the limit $\Delta \to 0$, it is readily seen that the derivative of the expected conditional response time exists and is given by $T_k'(x) = 1 + \sum_{j=1}^{K} \frac{g_j}{g_k} L_j(x)$. Further developing the expressions for $L_j(x)$, $j = 1, \ldots, K$, it was shown in [16] that the expected conditional response times of the various classes satisfy the following system of integro-differential equations

$$\begin{aligned} T_k'(x) \quad &= \quad 1 + \sum_{j=1}^{K} \int_0^{\infty} \lambda_j \frac{g_j}{g_k} T_j'(y)[1 - F_j(y + \frac{g_j}{g_k} x)]dy \\ &\quad + \int_0^{x} T_k'(y) \sum_{j=1}^{K} \lambda_j \frac{g_j}{g_k} [1 - F_j(\frac{g_j}{g_k}(x - y))]dy, \end{aligned} \tag{2}$$

for $k = 1, \ldots, K$. The natural boundary conditions are $T_k(0) = 0$, $k = 1, \ldots, K$.

The following theorem was proved in [16] under the assumption of finite second moments of the service requirement distributions. This condition was relaxed in [2], leaving $\rho < 1$ as a sufficient condition.

Theorem 1 *The system of equations (2) has a unique solution which is given by*

$$T_k(x) = g_k \int_0^{x/g_k} a(t)dt + \int_0^{x/g_k} b(t)dt,$$

where $a(x)$ is the unique solution of the defective renewal equation

$$a(x) = 1 + \int_0^x a(y)\Psi(x-y)dy \tag{3}$$

with

$$\Psi(x) = \sum_{j=1}^{K} \lambda_j g_j \overline{F}_j(g_j x),$$

and $b(x)$ satisfies

$$b(x) = c(x) + \int_0^\infty b(y)\Psi(x+y)dy + \int_0^x b(y)\Psi(x-y)dy \tag{4}$$

with

$$c(x) = \sum_{j=1}^{K} \lambda_j g_j^2 \int_0^\infty a(y)\overline{F}_j(g_j(x+y))dy.$$

As a consequence of equation (1), if there are two classes $j, k \in \mathcal{K}$ such that $g_j = g_k$, then $\tau_k(x) \stackrel{d}{=} \tau_j(x)$ for all $x \geq 0$, where $\stackrel{d}{=}$ denotes equality in distribution. For the case $g_j \neq g_k$, the authors of [2] showed using sample-path arguments that the conditional response times are stochastically ordered according to the DPS weights.

Theorem 2 *If $g_k \geq g_l$, then $\tau_k(x) \leq_{st} \tau_l(x)$, that is, $P(\tau_k(x) > y) \leq P(\tau_l(x) > y)$ for all $y \geq 0$. Thus, for all $x \geq 0$ and for all $n \geq 1$, we have that $T_k^n(x) = E[\tau_k^n(x)] \leq E[\tau_l^n(x)] = T_l^n(x)$.*

In particular, Theorem 2 implies that for any k and j such that $k \neq j$, the two curves, $T_k(x)$ and $T_j(x)$, do not cross.

In [10] the authors analyze the asymptotic behavior of the response time distribution. Their main result relates for each class k the asymptotic tail behavior of the response time to the tail of the service requirement distribution. We recall that a random variable X is regularly varying of index ν if $P\{X > x\} = l(x)x^{-\nu}$, where $l(\cdot)$ is a slowly varying function, i.e., $\lim_{x\to\infty} l(\eta x)/l(x) = 1$, $\eta > 1$.

Theorem 3 *Let the service requirement distributions of class-k jobs and the distribution of an arbitrary job be regularly varying of index $\nu > 2$. Then*

$$\lim_{x\to\infty} \frac{P\{\tau_k > x\}}{\overline{F}_k((1-\rho)x)} = 1.$$

The result of Theorem 3 can be interpreted as follows. Let us consider a job with a very large service requirement. This job will remain in the system for a long time. If the system is stable, during this time “regular” jobs will arrive, be served and depart. On average, the server will devote a fraction of service equal to ρ to the “regular” jobs and hence, the large job will receive the remaining service, i.e., a fraction $1-\rho$. Consequently, the amount of service received by the large job during x is approximately $(1-\rho)x$. The weights of the DPS discipline do not play a role in the above explanation, and they do not appear in the statement of Theorem 3. We refer to [10] and [9, Section 5] in the present special issue for further details on the derivation of Theorem 3.

We note that Theorem 3 does not require the assumption of Poisson arrivals. Hence, in the particular case of $K=1$, Theorem 3 generalizes the main result of [53] for the egalitarian PS model, where the assumption of Poisson arrivals was required.

Another point of view on the same phenomenon was provided in [16], where it was shown that as the service requirement tends to infinity the slowdown in the DPS system approaches the slowdown of the PS system, that is,

$$\lim_{x\to\infty} \frac{T_k(x)}{x} = \frac{1}{1-\rho}.$$

We note that similarly to Theorem 3, the weights do not appear in the result. This result was strengthened in [2], where it was proved that the expected conditional response time of class k has an asymptote with slope $1/(1-\rho)$. In addition a simple closed-form expression for the asymptotic bias was provided.

Theorem 4 *Let $\overline{x_k^2}$ be finite for all $k=1,\ldots,K$. Then the expected conditional response time of class k has an asymptote with slope $1/(1-\rho)$ and the following bias*

$$\lim_{x\to\infty}\left(T_k(x) - \frac{x}{1-\rho}\right) = \frac{\sum_{j=1}^{K}\lambda_j(1-\frac{g_k}{g_j})\overline{x_j^2}}{2(1-\rho)^2}. \tag{5}$$

We note that the value of the bias depends on the value of the weights. In fact the value of the bias depends on the second moments of the classes that have different weights. This result was proved by combining the integro-differential equations (2) and the conservation law introduced in Theorem 11.

3.1 Exponential Service Requirements

For the case of exponentially distributed service requirements, the authors of [16] further developed Theorem 1 and obtained a closed-form expression for the expected conditional response times. Let m be the number of different elements in the vector $v=(g_k\mu_k)_{k=1,\ldots,K}$.

Theorem 5 *Let the service requirement distributions be exponential. Then the expected conditional response time of a class-k job with required service time x is equal to*

$$T_k(x) = \frac{x}{1-\rho} + \sum_{j=1}^{m} \frac{g_k c_j \alpha_j + d_j}{\alpha_j^2}\left(1 - e^{-\alpha_j x/g_k}\right), \tag{6}$$

where $-\alpha_j$, $j = 1, 2, \ldots, m$, *are the* m *distinct negative roots of*

$$\sum_{j=1}^{K} \frac{\lambda_j g_j}{\mu_j g_j + s} = 1, \tag{7}$$

and c_j *and* d_j, $j = 1, \ldots, m$, *are given respectively by*

$$c_j = \frac{\prod_{k=1}^{m} (g_k \mu_k - \alpha_j)}{-\alpha_j \prod_{k \neq j}^{m} (\alpha_k - \alpha_j)},$$

and

$$d_j = \frac{\left(\sum_{k=1}^{K} \lambda_k g_k^2 / (\mu_k^2 g_k^2 - \alpha_j^2)\right) \prod_{k=1}^{m} (\mu_k^2 g_k^2 - \alpha_j^2)}{\prod_{k \neq j}^{m} (\alpha_k^2 - \alpha_j^2)}.$$

Theorem 5 can be readily extended to the case of a hyperexponential service requirement distribution. Assume that there are two classes i, j such that $g_i = g_j$. Then $T_i(x) = T_j(x)$ for all $x \geq 0$ and hence the jobs of the two classes can be seen as belonging to a single class with hyperexponential distribution $\lambda_i/(\lambda_i + \lambda_j) F_i(x) + \lambda_j/(\lambda_i + \lambda_j) F_j(x)$. This argument can be generalized to an arbitrary number of phases.

4 Moments of Response Times and Numbers of Jobs

In a stable PS system the mean number of jobs in the system is finite [27] regardless of the characteristics of the service requirement distribution. In the context of single-class systems, this result illustrates the benefits of time-sharing disciplines compared to more traditional disciplines such as First Come First Served, where the expected number of jobs is infinite if the second moment of the service requirement is infinite. In [3] the authors proved that the finiteness of the mean number of jobs is preserved by DPS.

Theorem 6 *An upper bound for the mean number of class-k jobs present in the system is as follows:*

$$L_k \leq \frac{\rho_k}{1-\rho} \left(1 + \frac{1}{g_k (1-\rho)} \sum_{j=1}^{K} g_j \rho_j \right).$$

In particular, the mean number of class-k jobs in the system is finite.

We would like to emphasize that the above upper bound is insensitive to the service requirement distributions. Theorem 6 shows the benefits of time-sharing scheduling disciplines compared to strict priority rules in the context of multi-class systems. Under strict priority disciplines, the mean number of jobs in the system is infinite if the distribution of the service requirement of a class has an infinite second moment. The robustness of DPS comes from the fact that the share of the server that a given class obtains depends on its weight as well as on the numbers of jobs present in the system. As a consequence, the share obtained by a class will increase proportionally as the numbers of jobs of this class grows and in this way the DPS discipline prevents classes with small weights from experiencing starvation.

In [8] the authors derived stochastic upper and lower bounds for the number of jobs by considering a DPS system as a PS network, where each node in the network represents a class of jobs. The nodes of such a network are coupled through their service capacity. The service speed of each node depends on the numbers of jobs present at all nodes. The DPS network turns out to be monotonic, which means that removing a customer from any node increases the service rate of all customers. The obtained bounds require a more restrictive condition than $\rho < 1$ and they rely on the calculation of the so-called balance function.

4.1 Exponential Service Requirements

In this subsection we turn the attention to exponentially distributed service requirements. Several of the results reported below have been extended to phase-type and general distributions (see Section 4.2).

In [24] the authors calculate the expected unconditional response time of a class-k job conditioned on the number of jobs found upon arrival $T_k(\mathbf{N})$, $k = 1, \ldots, K$, where $\mathbf{N} = (N_1, \ldots, N_K)$.

Theorem 7 *Let the service requirement distributions be exponential. Then $T_k(\mathbf{N})$ is an affine function in $\mathbf{N}$, i.e., there exist constants B_{kj}, $j = 0, \ldots, K$, such that*

$$T_k(\mathbf{N}) = B_{k0} + \sum_{j=1}^{K} N_j B_{kj},$$

where for all $k = 1, \ldots, K$,

$$B_{kj} = \begin{cases} \frac{1}{\mu_k(1-\sigma_k)} & \text{if } j=0, \\ \frac{g_j}{(g_k\mu_k+g_j\mu_j)(1-\sigma_k)} & \text{otherwise}, \end{cases} \tag{8}$$

where the constants σ_k, $k = 1, \ldots, K$, are given by

$$\sigma_i = \sum_{k=1}^{K} \frac{\lambda_k g_k}{g_i\mu_i + g_k\mu_k}. \tag{9}$$

Theorem 7 provides crucial insights into the dynamics of DPS by decomposing the response time (conditioned on the number of jobs present in the system) into $\sum_{j=1}^{K} N_j$ independent summands. In fact, each of the jobs in the system contributes to the tagged job's response time with an additive component, which is independent of other jobs' contributions. For more discussion on decomposition results see Section 4.2 below.

Unconditioning on the number of jobs found upon arrival and using the fact that Poisson Arrivals See Time Averages (PASTA) as well as Little's law, it is readily seen from Theorem 7 that the expected unconditional response times T_k, $k = 1, \ldots, K$, can be found as the solution of a system of linear equations.

Theorem 8 *Let the service requirement distributions be exponential. Then the unconditional expected response times satisfy the following system of linear equations:*

$$T_k = B_{k0} + \sum_{j=1}^{K} T_j \lambda_j B_{kj}, \tag{10}$$

where for all $k = 1, \ldots, K$ and $j = 0, 1, \ldots, K$, B_{kj} are given in Theorem 7.

We note that the system of equations (10) can be derived from the integral equation (2) (see [16] for more details), by multiplying both sides of the integral equation by $e^{\mu_k t}$, integrating over $t \in (0, \infty)$ and using

$$T_k = \int_0^\infty T_k(x) \mu_k e^{-\mu_k x} dt = \int_0^\infty T_k'(x) e^{-\mu_k x} dx.$$

A closed-form solution of the system of equations (10) is only available for the case of $K = 2$, and reads

$$T_1 = \frac{1}{\mu_1(1-\rho)} \left(1 + \frac{\mu_1 \rho_2 (g_2 - g_1)}{\mu_1 g_1 (1-\rho_1) + \mu_2 g_2 (1-\rho_2)} \right), \tag{11}$$

and

$$T_2 = \frac{1}{\mu_2(1-\rho)} \left(1 + \frac{\mu_2 \rho_1 (g_1 - g_2)}{\mu_1 g_1 (1-\rho_1) + \mu_2 g_2 (1-\rho_2)} \right). \tag{12}$$

By Little's law, Theorem 8 can be easily modified to obtain an equivalent system of equations for the mean number of jobs in the system.

In the case of exponential service requirements, the vector consisting of the numbers of jobs in each of the classes is a Markov process. Based on the Chapman-Kolmogorov equations for this process, the authors of [41, 28] provide methods to obtain higher moments of the unconditional number of jobs in the system. Basically, the proposed algorithms require solving multiple systems of linear equations. For the second moment, it turns out that the expectations $L_{k_1 k_2}$, for all $1 \le k_1, k_2 \le K$ satisfy

$$L_{k_1 k_2} - \sum_{i=1}^{K} g_i \frac{\lambda_{k_1} L_{k_2 i} + \lambda_{k_2} L_{i k_1} + \lambda_i L_{k_1 k_2}}{g_{k_1} \mu_{k_1} + g_{k_2} \mu_{k_2} + g_i \mu_i} = (g_{k_1} + g_{k_2}) \frac{\lambda_{k_1} L_{k_2} + \lambda_{k_1} L_{k_1}}{g_{k_1} \mu_{k_1} + g_{k_2} \mu_{k_2}},$$

where the $L_k = \lambda_k T_k$ and T_k, $k = 1, \ldots, K$, correspond to the solution of the system of equations (10).

It is worthwhile noting that in addition to the unconditional number of jobs in the system, the authors of [28] also provide a system of differential equations in order to determine moments of the conditional response time.

4.2 Phase-Type and General Service Requirements

Researchers have succeeded in extending some of the results on DPS with exponential service requirement distributions to the larger class of phase-type distributions. We refer the interested reader to [48, 49, 24] for extensions of the results of Section 4.1 to phase-type service requirement distributions.

In [41] the authors consider general service requirement distributions. By conditioning not only on the number of jobs, but also on their remaining service requirements, the authors of [41] show that the response times allow a decomposition into independent summands. Unfortunately, in the absence of information on the initial conditions this decomposition does not yield steady-state results. For this reason, this approach has proven to be more fruitful in the analysis of the egalitarian PS queue [52, 19].

In [13] the authors develop an approximate but more tractable decomposition result for DPS which turns out to be quite accurate for moderate differences in service weights.

4.3 Asymptotic Analysis: Time-Scale Decomposition

In [8] and [49] the authors consider a DPS queue with general service requirement distribution and perform a time-scale separation. As a consequence, from the perspective of a given class, the arrival and service completions of the other classes occur either on an extremely fast or slow time scale. The analysis of this limiting regime yields closed-form expressions for the marginal distributions of the queue lengths. The obtained formulae provide insights into the performance of the system as a function of the weights. The time-scale decomposition has also been used for an approximate analysis of response times in a DPS system with admission control [11].

For the sake of clarity, we present here results for the mean number of jobs and only for the case $K = 2$. The exposition below is based on [49]. The dynamics of class 1 are modified by introducing a scaling parameter $r > 0$. The job arrival process is Poisson with mean rate $\lambda_1^r = \lambda_1 r$ and the service requirement is $F_1^r(x) = F_1(rx)$. It is easy to see that after these modifications the class-1 load remains invariant, that is $\rho_1^r = \rho_1$, for all $r > 0$. When $r \to \infty$ a complete time-scale separation occurs. From the perspective of class-2 jobs, the dynamics of class 1 will average out on its relevant time-scale and thus

$$L_2^\infty = \frac{\rho_2}{1-\rho}. \tag{13}$$

On the other hand, from the perspective of class 1 the dynamics of class 2 vanish and hence if there are N_2 class-2 jobs in the system, class 1 approximately behaves as a standard PS system with $(g_2/g_1)N_2$ "permanent jobs". Thus, we obtain

$$L_1^\infty = \frac{\rho_1}{1-\rho_1}\left(\frac{g_2}{g_1}L_2^\infty + 1\right) = \frac{\rho_1}{1-\rho_1}\left(\frac{g_2}{g_1}\frac{\rho_2}{1-\rho} + 1\right). \tag{14}$$

From equations (13) and (14), the authors of [49] conclude that by giving larger weight to class 1 ($g_1 > g_2$), the number of class-1 jobs can be reduced, whereas the number of class-2 jobs will not change significantly. In contrast, if class 2 were given higher preference ($g_2 > g_1$), the number of class-2 jobs will not be reduced, while the number of class-1 jobs would drastically increase. In [49] the authors report numerical results that illustrate that the performance of a relatively slow class would be quite insensitive with respect to the choice of the weights and that the performance becomes more sensitive as the dynamics of the class accelerate. More guidelines on setting weights are presented in Section 6.

5 Heavy-Traffic and Overload Regimes

In this section we consider a DPS system when it operates in the heavy-traffic regime ($\rho \uparrow 1$) as well as in overload ($\rho > 1$). In addition, we briefly discuss related results in a closed DPS system, where the number of jobs tends to infinity.

5.1 Heavy-Traffic Regime

The authors of [48] have considered the DPS model in a heavy-traffic regime with phase-type service requirements when $\rho = \sum_{j=1}^{K} \rho_j \to 1$ while ρ_j/ρ_i remains constant for all $i, j \in \mathcal{K}$. Their result is a generalization of the result for exponentially distributed service requirements obtained in [42]. The following result has been obtained in [48], but we note that the analysis presented in [42] is much more detailed.

Theorem 9 *If the service requirement distributions are of phase-type, then for* $[\rho_1, ..., \rho_K] \to [\bar\rho_1, ..., \bar\rho_K]$, *with* $\sum_{j=1}^{K} \bar\rho_j = 1$,

$$(1-\rho)[n_1, n_2, ..., n_K] \xrightarrow{d} \mathcal{E} \cdot [\frac{\bar\rho_1}{g_1}, \frac{\bar\rho_2}{g_2}, ..., \frac{\bar\rho_K}{g_K}],$$

where $\xrightarrow{d}$ *denotes convergence in distribution and* $\mathcal{E}$ *is an exponentially distributed random variable with mean*

$$\frac{\sum_k p_k \overline{x_k^2}}{\sum_k p_k \overline{x_k} \sum_k \frac{\bar\rho_k \overline{x_k^2}}{g_k \overline{x_k}}},$$

where $p_k = \lambda_k/\lambda$, $k = 1, \dots, K$.

We note that the mean of the exponential distribution in Theorem 9 is equal to 1 in the case of standard PS, that is, when $g_j = 1$ for $j = 1, \dots, K$.

Theorem 9 reflects a so-called state-space collapse. In heavy traffic, the random vector $(1-\rho)[n_1, n_2, ..., n_K]$ converges in distribution to a constant vector multiplied with an exponentially distributed random scalar. The fact that the scaled queue lengths are proportional to a common exponentially distributed random variable can be explained by the fact that the distribution of the workload in an $M/G/1$ queue (scaled by $1-\rho$) is also exponentially distributed in heavy-traffic [29].

5.2 Overload Regime

In [1] the authors study a DPS queue in overload, that is, when $\rho > 1$. Their analysis is based on techniques developed in [26], where egalitarian PS was analyzed. In the main result the authors proved that under rather general conditions, the number of jobs of any class k in the system grows asymptotically linearly with a rate β_k. Interestingly, β_k depends on the distribution of the job sizes in a complex manner, and not just on their first moments. In contrast, the dependence of β_k on the arrival process is only through the arrival rates of the various classes.

Theorem 10 *Let $\rho > 1$. In addition assume that the arrival rate of class-k jobs is λ_k (not necessarily Poisson). Then*

$$\lim_{t\to\infty} \frac{N_k(t)}{t} = \beta_k \quad a.s. \tag{15}$$

where $y_k = \beta_k$, $k = 1, \ldots, K$, are the unique positive solutions of

$$y_k = \lambda_k \left(1 - E\left[e^{-X_k \frac{1}{g_k} \sum_{j=1}^{K} y_j g_j}\right]\right) \quad k = 1, \ldots, K. \tag{16}$$

The response time τ_k^n of the n-th class-k job satisfies

$$\lim_{n\to\infty} \frac{\tau_k^n}{n} - \frac{e^{\sigma_k^n \frac{\gamma}{g_k}} - 1}{\lambda_k} = 0 \tag{17}$$

in distribution, where σ_k^n denotes the service requirement of the n-th class-k job and $\gamma = \sum_{j=1}^{K} g_j \beta_j$.

Note that it follows from equation (17) that even in overload conditions, eventually all jobs complete and leave the system in a finite time.

Assume now that the service requirements are exponentially distributed. By definition, the class-k job departure rate is $\lambda_k - \beta_k$. Then, in view of equations (1) and (15) we have that for all $k = 1, \ldots, K$, $\lambda_k - \beta_k = \frac{g_k \beta_k}{\sum_{j=1}^{K} g_j \beta_j} \mu_k$. Hence, the values of β_k, $k = 1, \ldots, K$, are given by

$$\beta_k = \frac{\lambda_k \gamma}{\mu_k g_k + \gamma}, \tag{18}$$

where γ is given in Theorem 10. On the other hand, since the DPS queue is a work-conserving system it holds that $\rho - 1 = \sum_{j=1}^{K} \beta_j \frac{1}{\mu_j}$. Then, after substituting the expression of equation (18) and straightforward manipulations we obtain that $\nu = \gamma$ is the unique strictly positive solution of

$$1 = \sum_{i=1}^{K} \frac{\lambda_i g_i}{\mu_i g_i + \nu}. \tag{19}$$

We note that equations (18) and (19) can be derived directly from equation (16) under the assumption of exponential service requirement distributions [1].

Interestingly, equation (19) is the same as equation (7), and whose roots enabled the computation of the expected conditional response times in steady state.

5.3 Closed DPS Models

In [35] the authors consider a DPS queue with K classes of permanent jobs. Each job alternates between an *off* state corresponding to an exponentially distributed thinking time, and an *on* state, in which a session with exponentially distributed service requirement is sent to the DPS queue. There are N_k jobs of type k and $M = \sum_{j=1}^{K} N_j$ permanent jobs altogether. The system is studied as M becomes large. It turns out that the job growth rate in time in the standard overloaded DPS is the same as the growth rate in M for the closed queue. For instance, equations (14) and (15) in [35] are exactly the same as equations (18) and (19) and in fact they were derived with similar arguments as the ones used in Section 5.2.

In [38] the author considered the same model as in [35] but in moderately heavy-traffic regime so that $\rho = 1 - a/\sqrt{M}$, where $a = O(1)$ as $M \to \infty$. Two-term asymptotic approximations to the mean numbers of jobs, and the mean response times of all classes are obtained. In addition, the author of [38] obtained the leading term in the asymptotic approximation to the joint distribution of the numbers of jobs in the DPS node, which is a zero-mean multivariate Gaussian distribution.

6 Setting the Weights

We now turn the attention to the problem of weight setting. Sections 6.1 and 6.2 review guidelines for setting the weights from a system point of view, that is, when a central entity chooses the weights in order to optimize some system-wide metric (for example the mean number of jobs in the system). In Section 6.3 we review the problem of setting weights from the jobs' (customers') point of view. With that objective in mind, DPS is studied as a non-cooperative game where jobs may pay some amount in return for a higher priority weight.

6.1 Conservation Law and Achievable Performance

The so-called work conservation property is fundamental to single-server (multi-class) systems and provides useful insights into the behavior of a system. We recall that a discipline is called work-conserving if the server works at full speed whenever there is work in the system. As a direct consequence of equation (1), it is easy to verify that a DPS queue is indeed work-conserving. Exploiting the work conservation property, the so-called Conservation Laws have been derived [30, 32, 4]. In [2], the authors proved the following conservation law for DPS queues.

Theorem 11 *Let the second moments of the service requirement distributions be finite, i.e.,* $\overline{x_k^2} < \infty$*, for all* $k = 1, \ldots, K$*. Then,*

$$\sum_{j=1}^{K} \lambda_j \int_0^{\infty} T_j(x)\overline{F}_j(x)dx = \frac{\sum_{j=1}^{K} \lambda_j \overline{x_j^2}}{2(1-\rho)}. \tag{20}$$

The term $\lambda_j \int_0^\infty T_j(x)\overline{F}_j(x)dx$, $j = 1, \ldots, K$, can be interpreted as the contribution of class-j jobs to the total mean unfinished work [32, Sections 3.4 and 4] and [4, Section 3.2]. Summing the contribution over all classes we obtain the total mean unfinished work in the queue, which is the same for all work-conserving disciplines. Since the compound arrival process is Poisson, the total mean unfinished work in the system is given by the well known Pollaczek-Khintchine formula which corresponds to the expression on the right-hand side of equation (20).

In the particular case of exponentially distributed service requirements, equation (20) becomes

$$\sum_{j=1}^{K} \rho_j T_j = \frac{1}{(1-\rho)^2} \sum_{j=1}^{K} \frac{\lambda_j}{\mu_j^2}. \tag{21}$$

We note that in this case equation (21) not only holds for DPS queues, but for general non-anticipating disciplines (both non-preemptive and preemptive resume) [15]. Based on equation (21), the authors of [36] characterized the achievable performance of a DPS queue and proved the so-called "almost complete" property of DPS.

Theorem 12 *Assume that the service requirement distributions are exponential. Let* Π *and* Π^{DPS} *denote the set of non-anticipating disciplines in a multi-class* $M/M/1$ *queue and the set of DPS disciplines, respectively. Let* $\mathcal{P}$ *and* $\mathcal{P}^{DPS}$ *denote the polyhedra that contain all the achievable mean-delay vectors, that is,* $\mathcal{P} = \{\mathbf{T}^\pi : \pi \in \Pi\}$ *and* $\mathcal{P}^{DPS} = \{\mathbf{T}^\pi : \pi \in \Pi^{DPS}\}$*, where* $\mathbf{T}^\pi = (T_1^\pi, \ldots, T_K^\pi)$*. Then* $\mathcal{P}^{DPS}$ *is the interior subset of* $\mathcal{P}$*.*

Theorem 12 states that if a performance vector is an interior point of $\mathcal{P}$, then it can be achieved by a DPS discipline with a suitable choice of weights. If the desired performance vector is on the boundary of $\mathcal{P}$, then it can be approximated by DPS as closely as desired. We note that an arbitrary boundary point corresponds to a strategy where certain subsets of classes have priority over other subsets, and within the classes of a given subset, the capacity is shared according to some appropriate weights. In particular, the performance vectors of the $K!$ vertices of $\mathcal{P}$ are achieved by the $K!$ strict preemptive resume priority disciplines that give priority to classes according to a permutation of the class indices.

We denote by $\pi(\phi)$ the strict preemptive resume priority discipline that gives priority to classes according to the permutation of the indices $\phi = (\phi_1, \ldots, \phi_K)$. It is known (see for example [15, 45, 18]) that under the assumption of exponential service requirements, the optimal scheduling discipline with respect to the objective

$$\min\{\sum_{j=1}^{K} c_j L_j^\pi : \pi \in \Pi\}, \tag{22}$$

is a strict preemptive resume priority discipline $\pi(\phi)$, where

$$c_{\phi_1}\mu_{\phi_1} \geq c_{\phi_2}\mu_{\phi_2} \geq \ldots \geq c_{\phi_K}\mu_{\phi_K}.$$

In particular, the strategy that minimizes the mean number of jobs in the system will be a strict preemptive resume priority discipline $\pi(\phi)$ such that

$$\mu_{\phi_1} \geq \mu_{\phi_2} \geq \ldots \geq \mu_{\phi_K}.$$

Hence, in the case of exponential service requirements and in view of Theorem 12, DPS will always be sub-optimal. It is important to note that this situation changes completely when we consider distributions with an infinite second moment. If a class has an infinite second moment, the mean number of jobs in the system will be infinite, and as a consequence the objective function (22) will be unbounded. Thus, in view of Theorem 6, any DPS system will outperform all strict priority systems. Unfortunately, for the case of general service requirement distributions, very little is known about how to set the weights of DPS in order to improve the overall performance.

Under the assumption of exponential service requirements, one would expect that in order for DPS to outperform PS, one should set the weights in such a way that the DPS system approaches the optimal strict priority. Specifically, one should give larger weights to the classes with smaller mean. In the particular case of two classes we obtain from equations (11) and (12)

$$L^{DPS} - L^{PS} = \frac{-\rho_1\rho_2(g_1 - g_2)(\mu_1 - \mu_2)}{(1-\rho)\left(\mu_1 g_1(1-\rho_1) + \mu_2 g_2(1-\rho_2)\right)},$$

where $L^{DPS} = \sum_{k=1}^{2} L_k^{DPS}$ and $L^{PS} = \sum_{k=1}^{2} L_k^{PS}$. Thus, we note that if $\mu_1 \geq \mu_2$ and $g_1 \geq g_2$, then $L^{DPS} \leq L^{PS}$. In [2] the authors compared the performance of DPS and PS with an arbitrary number of classes and proved that, under an additional technical condition, if the weights of DPS are chosen in decreasing order with respect to the mean service requirements of the classes, the mean number of jobs in a DPS queue is smaller than in PS. This result can be seen as a multi-class counterpart of the classical single-class results on age-based scheduling, where it is well known that giving preferential treatment to short jobs reduces the mean number of jobs in the system [44, 43, 17].

In a slightly different setting, the authors of [22] consider a DPS model with two job classes, entry fees and waiting costs, in which the server is allowed to set the weights with the objective of maximizing the profit.

6.2 Utility-Based Resource Allocation

The allocation provided by a DPS system can be interpreted as the allocation that maximizes the aggregate utility of the various classes for a given user population. More specifically, consider a single server shared by K traffic classes. Assume that there are N_k class-k jobs in the system, $k = 1, \ldots, K$. In [7] the authors consider the following optimization problem

$$\max_{r_k, k\in\mathcal{K}} \left\{ \sum_{k=1}^{K} w_k N_k U\left(\frac{r_k}{w_k}\right) \right\}, \tag{23}$$

subject to the capacity constraint

$$\sum_{k=1}^{K} N_k r_k \leq 1, \tag{24}$$

where the constants w_k, $k = 1, \ldots, K$, denote the weights for class-k jobs and the utility function $U(\cdot)$ is increasing and strictly concave. Then it can be shown that the service rates r_k, $k = 1, \ldots, K$, that solve the above optimization problem correspond to a DPS system, that is, the service rates r_k are given by equation (1) with weights $g_k = w_k$.

In [6] the authors considered a specific choice of the utility function $U(r) = r^{(1-\alpha)}/(1-\alpha)$ and studied the optimal allocation in the context of networks.

6.3 A Non-Cooperative Game Approach

Game theory provides a framework to study how jobs would choose their weights in order to maximize their own profit. Several non-cooperative models based on the DPS model are presented in [21, Section 4.3]. By non-cooperative we mean that each arriving job would take his own decision about the choice of his weight so as to maximize his personal payoff, which can be expected to be composed of a utility part (the expected response time) and a cost part (the price to pay for choosing a given weight). The decision of which weight to choose can be expected to depend on the available information. The solution concept in the non-cooperative setting is the Nash equilibrium, i.e., a set of decisions for all jobs such that no one can strictly improve his payoff by deviating from the equilibrium point.

We present here one model, and we refer to [21, Section 4.3] (see also [23]) for more results in the same spirit. It is assumed that the service requirements are exponentially distributed with parameter μ and the cost per unit of time spent in the system is C. In addition, jobs are not aware of their own service requirement and have no information on the state of the system. Two weights $g_1 > g_2 \geq 0$ are available. Without loss of generality assume that $g_1 + g_2 = 1$. A strategy is characterized by a probability q of choosing the higher priority weight g_1. Each job has a choice of paying an amount of $\theta > 0$ and obtaining the priority parameter g_1, or else getting the priority parameter g_2. Jobs decide whether to buy priority after comparing the price to pay θ and the achievable reduction in the response time as a consequence of purchasing the high priority weight g_1. The following result holds [21].

Theorem 13 *Let the service requirements be exponential with parameter μ. Consider a non-cooperative model in which jobs may decide to pay an amount θ in order to get a higher priority weight g_1 instead of g_2. Then,*

- *if $\theta < \frac{C\rho(g_1-g_2)}{\mu(1-\rho)(1-\rho g_2)}$, $q = 1$ is a Nash equilibrium,*
- *if $\theta > \frac{C\rho(g_1-g_2)}{\mu(1-\rho)(1-\rho g_1)}$, $q = 0$ is a Nash equilibrium,*

- *if* $\frac{C\rho(g_1-g_2)}{\mu(1-\rho)(1-\rho g_2)} < \theta < \frac{C\rho(g_1-g_2)}{\mu(1-\rho)(1-\rho g_1)}$*, there are three equilibria:* $q = 0, q = 1$*, and* $q = \frac{1}{\rho(g_1-g_2)} - \frac{C}{\theta(\mu-\lambda)} - \frac{g_2}{g_1-g_2}$.

Theorem 13 shows that the larger the fraction of jobs who purchase high priority, the more valuable it becomes for a job to become high priority. The authors of [21] described the above strategies with the term "follow the crowd". In the first (second) case, the price is that low (high) that regardless of other jobs' choice, it is optimal to purchase (not purchase) priority. In the third case, for a given medium price, if nobody buys priority it is better not to buy priority either, but if all jobs choose priority, then it is better to do so as well. Thus, there are two pure equilibria. In addition, there is a third, mixed and unstable equilibrium, where a certain fraction of jobs purchase priority, such that any job is indifferent between the two options (and might in fact also adopt a mixed strategy).

Let us now consider the case $g_1 = 1$ and $g_2 = 0$. Then, Theorem 13 shows that if θ is smaller than the cost of the mean waiting time in an $M/M/1$ queue, that is, $\theta < \frac{C\rho}{\mu(1-\rho)}$, then all jobs will purchase priority in equilibrium. As a consequence, the possibility of buying priority worsens the situation for everybody. In equilibrium all jobs pay θ but in practice no one benefits from it since the mean delay will remain the same.

Acknowledgments

The authors wish to thank the anonymous reviewers for the careful reading and for providing valuable comments that helped improve the presentation. The authors also gratefully acknowledge the kind help and advice of Rudesindo Núñez-Queija.

References

[1] E. Altman, T. Jimenez, and D. Kofman. DPS queues with stationary ergodic service times and the performance of TCP in overload. In *Proceedings of IEEE INFOCOM*, 2004.

[2] K.E. Avrachenkov, U. Ayesta, P. Brown, and R. Núñez-Queija. Discriminatory processor sharing revisited. In *Proceedings of IEEE INFOCOM*, 2005.

[3] K.E. Avrachenkov, U. Ayesta, P. Brown, and R. Núñez-Queija. Discriminatory processor sharing revisited, 2006. Extended version submitted for publication.

[4] F. Baccelli and P. Brémaud. *Elements of Queuing Theory: Palm Martingale Calculus and Stochastic Recurrences*. Springer, 2003.

[5] M.J. Bach. *The design of the UNIX operating system.* Prentice-Hall, 1986.

[6] T. Bonald and L. Massoulié. Impact of fairness on Internet performance. In *Proceedings of ACM SIGMETRICS/Performance*, pages 82–91, 2001.

[7] T. Bonald and A. Proutière. Insensitive bandwidth sharing in data networks. *Queueing Systems*, 44:69–100, 2003.

[8] T. Bonald and A. Proutière. On stochastic bounds for monotonic processor sharing networks. *Queueing Systems*, 47:81–106, 2004.

[9] S.C. Borst, R. Núñez-Queija, and A.P. Zwart. Sojourn time asymptotics in processor sharing queues, 2006. In this special issue.

[10] S.C. Borst, D.T.M.B. van Ooteghem, and A.P. Zwart. Tail asymptotics for discriminatory processor sharing queues with heavy-tailed service requirements. *Performance Evaluation*, 61(2–3):281–298, 2005.

[11] O.J. Boxma, N. Hegde, and R. Núñez-Queija. Exact and approximate analysis of sojourn times in finite discriminatory processor sharing queues. *AEU International Journal on Electronic Communications*, 60:109–115, 2006.

[12] T. Bu and D. Towsley. Fixed point approximation for TCP behaviour in an AQM network. In *Proceedings of ACM SIGMETRICS/Performance*, pages 216–225, 2001.

[13] S.K. Cheung, J.L. van den Berg, and R.J. Boucherie. Decomposing the queue length distribution of processor sharing models into queue lengths of permanent customer queues. *Performance Evaluation*, 62(1–4):100–116, 2005.

[14] S.K. Cheung, J.L. van den Berg, R.J. Boucherie, R. Litjens, and F. Roijers. An analytical packet/flow-level modelling approach for wireless LANs with quality-of-service support. In *Proceedings of ITC-19*, 2005.

[15] E.G. Coffman and I. Mitrani. A characterization of waiting time performance realizable by single-server queues. *Operations Research*, 3(28):810–821, 1980.

[16] G. Fayolle, I. Mitrani, and R. Iasnogorodski. Sharing a processor among many job classes. *Journal of the ACM*, 27(3):519–532, 1980.

[17] H. Feng and V. Misra. Mixed scheduling disciplines for network flows. *Performance Evaluation Review*, 31(2):36–39, 2003.

[18] T.C. Green and S. Stidham. Sample-path conservation laws, with application to scheduling queues and fluid systems. *Queueing Systems*, 36:175–199, 2000.

[19] S. Grishechkin. On a relationship between processor sharing queues and crump-mode-jagers branching processes. *Adv. Appl. Prob.*, 24(3):653–698, 1992.

[20] L. Guo and I. Matta. Scheduling flows with unknown sizes: Approximate analysis. In *Proceedings of ACM SIGMETRICS (Extended Abstract)*, pages 276–277, 2002. Extended version available as a Boston University Technical Report BU-CS-2002-009.

[21] R. Hassin and M. Haviv. *To Queue or not to Queue: Equilibrium Behavior in Queueing Systems.* Kluwer Academic Publishers, Boston etc., 2003.

[22] R. Hassin and M. Haviv. Who should be given priority in a queue. *Operations Research Letters*, 34(2):191–198, 2006.

[23] M. Haviv and J. van der Wal. Equilibrium strategies for processor sharing and random queues with relative priorities. *Probability in the Engineering and Informational Sciences*, 11:403–412, 1997.

[24] M. Haviv and J. van der Wal. Waiting times discriminatory processor sharing. 2005. Submitted for publication.

[25] Y. Hayel and B. Tuffin. Pricing for heterogeneous services at a discriminatory processor sharing queue. In *Proceedings of Networking*, 2005.

[26] A. Jean-Marie and P. Robert. On the transient behavior of the processor sharing queue. *Queueing Systems*, 17:129–136, 1994.

[27] F. Kelly. *Stochastic Networks and Reversibility.* Wiley, Chichester, 1979.

[28] J. Kim and B. Kim. Sojourn time distribution in the M/M/1 queue with discriminatory processor sharing. *Performance Evaluation*, 58(4):341–365, 2004.

[29] J.F.C. Kingman. On queues in heavy traffic. *Journal of the Royal Statistical Society. Series B, Methodological*, 24:383–392, 1962.

[30] L. Kleinrock. A conservation law for a wide class of queueing disciplines. *Naval Research Logistics Quarterly*, 12(2):181–192, 1965.

[31] L. Kleinrock. Time-shared systems: A theoretical treatment. *Journal of the ACM*, 14(2):242–261, 1967.

[32] L. Kleinrock. *Queueing Systems, vol. 2.* John Wiley and Sons, 1976.

[33] L. Kleinrock, R.R. Muntz, and E. Rodemich. The processor sharing queueing model for time-shared systems with bulk arrivals. *Networks Journal*, 1(1):1–13, 1971.

[34] L. Massoulié and J. Roberts. Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, 15(1):185–201, 2000.

[35] D. Mitra and A. Weiss. A closed network with a discriminatory processor sharing server. *Performance Evaluation Review*, 17(1):200–208, 1989.

[36] I. Mitrani and J.H. Hine. Complete parameterized families of job scheduling strategies. *Acta Informatica*, (8):61–73, 1977.

[37] I. Mitrani and T.M. O'Donovan. Private communications.

[38] J.A. Morrison. Asymptotic analysis of a large closed queueing network with discriminatory processor sharing. *Queueing Systems*, 9:191–214, 1991.

[39] T.M. O'Donovan. Direct solutions of M/G/1 processor sharing models. *Operations Research*, 22:1232–1235, 1974.

[40] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, 1993.

[41] K.M. Rege and B. Sengupta. A decomposition theorem and related results for the discriminatory processor sharing queue. *Queueing Systems*, 18:333–351, 1994.

[42] K.M. Rege and B. Sengupta. Queue length distribution for the discriminatory processor sharing queue. *Operations Research*, 44(4):653–657, 1996.

[43] R. Righter and J.G. Shanthikumar. Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures. *Probability in the Engineering and Informational Sciences*, 3:323–334, 1989.

[44] L.E. Schrage. The queue M/G/1 with feedback to lower priority queues. *Management Science*, 13:466–471, 1967.

[45] J. Shanthikumar and D. Yao. Multiclass queueing systems: Polymatroidal structure and optimal scheduling control. *Operations Research*, 40(2):293–299, 1992.

[46] M. Shreedhar and G. Varghese. Efficient fair queuing using Deficit Round-Robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, 1996.

[47] A. Silberschatz, P. Galvin, and G. Gagne. *Applied Operating System Concepts*. John Wiley and Sons, 2000.

[48] G. van Kessel, R. Núñez-Queija, and S.C. Borst. Asymptotic regimes and approximations for discriminatory processor sharing. *Performance Evaluation Review*, 32(6):44–46, 2004.

[49] G. van Kessel, R. Núñez-Queija, and S.C. Borst. Differentiated bandwidth sharing with disparate flow sizes. In *Proceedings of IEEE INFOCOM*, 2005.

[50] M.J.G. van Uitert. *Generalized Processor Sharing Queues.* PhD thesis, Eindhoven University of Technology, 2003.

[51] S.F. Yashkov. Processor sharing queues: Some progress in analysis. *Queueing Systems*, 2:1–17, 1987.

[52] S.F. Yashkov. Mathematical problems in the theory of processor sharing queueing systems. *Journal of Soviet Mathematics*, 58:101–147, 1992.

[53] A.P. Zwart and O.J. Boxma. Sojourn time asymptotics in the M/G/1 processor sharing queue. *Queueing Systems*, 35:141–166, 2000.