Service Differentiation for Delay-Sensitive Applications: An Optimisation-Based Approach

Peter Key [a] Laurent Massoulié [a] Jonathan K. Shapiro [b,1]

[a] *Microsoft Research Limited, 7 JJ Thomson Avenue, Cambridge, UK*

[b] *Department of Computer Science, University of Massachusetts at Amherst, MA, USA*

Abstract

This paper deals with the performance of delay-sensitive applications running over a network that offers multiple classes of service, where the adaption of application rates in response to network feedback is the primary mechanism available for controlling quality of service. We first evaluate the gain in utilisation allowed by the introduction of several classes of service. To this end we compare the pairs of achievable rates, or *schedulable regions*, for two types of applications with two distinct delay requirements that make use of a single resource, with either no differentiation, simple priority–based differentiation, or earliest-deadline-first scheduling–based differentiation. The main observations are that the gain achieved by differentiation is essentially affected by traffic burstiness, and that the two differentiation schemes yield very similar performance.

We then consider what feedback information should be sent to traffic sources from different classes, casting the problem in the framework of optimisation–based congestion control. We establish a connection between the *sample-path shadow price* rationale for feedback synthesis and the *rare perturbation analysis* technique for gradient estimation in discrete event systems theory. Based on this connection, we propose several marking schemes, for simple priority–based differentiation with a measure of cost based on loss or delay, and also for earliest-deadline-first–based differentiation with loss–based cost. The interaction of these marking algorithms with simple congestion control algorithms is studied via simulations.

1 Introduction

With a single best-effort service class it is possible to offer a simple form of service differentiation [1], whereby users who pay a higher price achieve a proportionally higher transmission rate. However, the applicability of such techniques is limited

[1] Partially supported by the National Science Foundation under grant number NSF???

by the assumption that applications' quality–of–service (QoS) requirements can be expressed in terms of a single performance measure, namely throughput. This limitation is a problem since QoS requirements are naturally multi-dimensional: applications vary in their sensitivities to several key performance measures including throughput, loss and delay. [2]

In this paper, we concentrate on delay-sensitive applications, and examine whether separate service classes are needed to cope with their differing requirements. This class of applications includes real time services, such as streaming media or interactive multimedia, as well as data services requiring low latency. Our focus on such applications is due not only to their growing importance, but also to the lack of any end-to-end mechanism to provide the necessary QoS. Whereas loss-sensitive applications can use error correcting codes to achieve low loss probability at the expense of reduced throughput, delay-sensitive applications have no such recourse to source coding if delay is too high. It is reasonable, therefore, to consider placing mechanisms within the network to reduce delay.

Much existing research in this area [2–4] focuses on providing deterministic delay guarantees and thus assumes the existance of traffic shaping and admission control. Although our work concerns the provision of statistical guarantees to a set of rate-adaptive sources, we derive similar results about the effect of service differentiation on network utilization. However, in the absence of admission control and traffic shaping, we must address the additional problem of finding appropriate feedback signals to enable sources to perform rate control necessary to satisfy their collective delay requirements.

The question of whether delay-sensitive applications require service differentiation was addressed by Bajaj, et al [5], who found the answer depends on the adaptive behaviour of the applications themselves and also on the burstiness of traffic. We study rate-adaptive control rather than the delay-adaptive behaviour considered in [5], but reach similar conclusions about the effect of burstiness. Alvarez and Hajek [6] also consider the necessity of multiple classes in the case of a mixed population of rate-sensitive and loss-sensitive users within the pricing framework of Gibbens and Kelly [1], which we also adopt in our work. They conclude that two classes are indeed required to satisfy the requirements of both user types, but that the benefit of service differentiation is only realized when the price per packet for each class of service is set correctly. The authors show the existence of the correct prices empirically but provide no way for the network to find them. Although we replace the loss-sensitive users with delay-sensitive ones, our research complements [6] by providing a mechanism for adaptively setting prices.

We first discuss (Section 2) the gain in utilisation offered by the introduction of two

[2] Thus, if we express user preference in terms of utility functions, utility should be a function of (at least) throughput and delay.

classes of service at a single resource. We show that two differentiation schemes—priority scheduling and earliest-deadline-first—yield very similar performance and that both outperform a single-class FIFO scheduler. Moreover, we confirm the observation in [5] that the amount of performance gain depends on traffic burstiness. We then discuss what feedback information should be sent to traffic sources from different classes. In the context of the optimization–based congestion control framework we use, this problem reduces to one of correctly setting prices. We first show how this framework can be adapted to multi-class networks using a notion of cost based on either average packet delay, or loss (Section 3). Exploiting a connection between the *sample-path shadow price* rationale for feedback synthesis and the *rare perturbation analysis* technique for gradient estimation in discrete event systems theory, we develop marking schemes for simple priority–based differentiation with a measure of cost based on loss or delay, and also for earliest-deadline-first–based differentiation with loss–based cost (Section 4). The interaction of these marking algorithms with simple congestion control algorithms is studied via simulations (Section 5).

2 Schedulable regions

In this section we discuss the performance gain achievable by introducing differentiation. We compare the efficiency of having multiple classes or not by considering the following simple scenario. Packets of two types reach one resource. Type 1 packets must be served by some time d_1, and type 2 packets must be served by some time d_2, $d_2 > d_1$. We want to compare the pairs of achievable rates (ρ_1, ρ_2) that can be sustained with different differentiation mechanisms.

2.1 Deterministic description of burstiness

In the case of deterministic guarantees, it is well-known (e.g. [2–4]) that service differentiation can improve network utilization, with the degree of improvement dependent on the relative delay requirements and the burstiness of the taffic. We now review these results for a simple leaky-bucket traffic profile.

Suppose that arrivals of type i are (σ_i, ρ_i)-constrained [7]. The capacity c needed to handle these arrivals without loss and within the deadlines is, in the FIFO case,

$$c_{FIFO} = \max(\rho_1 + \rho_2, (\sigma_1 + \sigma_2)/d_1),$$

while with an earliest deadline first scheduler, the required capacity is

$$c_{EDF} = \max(\rho_1 + \rho_2, \sigma_1/d_1, (\sigma_1 + \sigma_2 + (d_2 - d_1)\rho_1)/d_2).$$

These results follow by applying a result of [7] (see also Theorem 2.3.2, p.99 of [8]). When d_2 is large and σ_1/d_1 is larger than $\rho_1+\rho_2$, we find that the ratio of these quantities is $1+\sigma_2/\sigma_1$, which becomes significant if the burstiness parameter σ_2 is not negligible compared to σ_1. It is also shown in [7] (see also [8], p.100) that any scheduler requires as much capacity as the EDF scheduler to handle such arrivals without loss and within the deadlines. The comparison of c_{FIFO} and c_{EDF} is thus indicative of the maximal gain that can be achieved by introducing multiple classes in the network.

Using the results of Example 2.3.13, p. 61 in [9], one also sees that for priority scheduling, the minimal capacity needed is:

$$c_{Priority} = \max(\rho_1+\rho_2, \sigma_1/d_1, \rho_1+(\sigma_1+\sigma_2)/d_2).$$

We thus see that when d_2 is large, this coincides with c_{EDF}, while when d_2 is close to d_1 it might happen that this is larger than c_{FIFO}.

These evaluations based on deterministic bounds illustrate the fact that the value of introducing differentiation is directly related to the burstiness of the traffic aggregates in each class. They also suggest that earliest-deadline first scheduling, which maximises the schedulable region, should be considered as a candidate for the introduction of differentiation.

2.2 Probabilistic description of burstiness

We next adopt a probabilistic model for the description of burstiness, which is better suited to reasoning about the case of offering statistical delay guarantees—the primary interest of this work. We assume Poisson arrivals with rates λ_1, λ_2 in each class, unit capacity $c = 1$, and fixed packet sizes σ. In this simple model of traffic the burstiness is captured by the parameter σ. We assume an infinite buffer, so that packets don't get dropped by the link. We define the schedulable region as the set of parameters (ρ_1, ρ_2), where $\rho_i := \lambda_i\sigma$, so that the probability of a class i packet experiencing a delay larger than d_i is less than some fixed value θ, say $\theta = 1\%$. As before, our aim is to compare schedulable regions (for a two-class resource with two distinct delay bounds) when there is single class and FIFO scheduling, with two classes using priority scheduling or EDF. We use heavy traffic approximations to obtain tractable formulas.

For FIFO, the total arrival rate $\lambda = \lambda_1 + \lambda_2$ must be such that the probability of experiencing a delay larger than the smallest of the two, i.e. d_1, is smaller than θ. Let W denote the stationary workload in the queue. Then we need to ensure that $\mathbf{P}(W+\sigma > d_1) < \theta$. In heavy traffic, the distribution of W is close to that of

$\lambda\sigma^2/(2(1-\rho))X$, where X is an exponential random variable with unit mean [3] (see e.g. [11]), thus $\mathbf{P}(X > x) < \theta \Rightarrow x > -\log\theta$. Hence, we must ensure

$$\rho_1 + \rho_2 \leq \frac{2(d_1 - \sigma)}{-\sigma\log(\theta) + 2(d_1 - \sigma)}. \tag{1}$$

We now consider preemptive priority scheduling. Class 1 packets do not see class 2 packets. Applying the same heavy traffic approximation for the stationary workload due to class 1 packets only[2] shows that we must ensure

$$\rho_1 \leq \frac{2(d_1 - \sigma)}{-\sigma\log(\theta) + 2(d_1 - \sigma)}. \tag{2}$$

For class 2 packets, we proceed as follows. Let W be the workload seen by a class 2 packet arriving at time 0. Then its sojourn time T is given by

$$T = \inf\{t > 0 : t \geq W + \sigma + \sigma N_1(t)\},$$

where $N_1(t)$ is the number of class 1 packets that arrive in the interval $(0,t)$. Let

$$X_t = e^{iuN_1(t) + \lambda_1 t(1-e^{iu})}.$$

This is a martingale. Also, T is a stopping time, so by Doob's sampling theorem applied to the bounded stopping time $T \wedge n$, letting n go to infinity (which is justified by a dominated convergence argument under the stability condition $\rho_1 + \rho_2 < 1$), we have

$$\mathbf{E}[X_T | W] = 1.$$

Since $\sigma N_1(T) = T - \sigma - W$, we get

$$\mathbf{E}\left[e^{T(\lambda_1(1-e^{iu}) + iu/\sigma)} | W\right] = e^{iu(W+\sigma)/\sigma}.$$

Convergence in distribution of $(1-\rho)W$ implies that

$$(1-\rho)T \xrightarrow{\mathcal{D}} \frac{\lambda\sigma^2}{2(1-\rho_1)}\mathrm{Exp}(1).$$

Using this approximation, we therefore need to ensure that

$$\frac{2(1-\rho)(1-\rho_1)(d_2 - \sigma)}{\lambda\sigma^2} \geq -\log\theta,$$

[3] The exact distribution of the unfinished work in an M/D/1 queue is given in explicit form in [10], p.112. Using heavy traffic approximation rather than the exact distribution affects only marginally the schedulable regions we derive, at least for small θ.

or equivalently

$$\rho_2 \leq \frac{2(1-\rho_1)(d_2-\sigma)}{-\sigma\log(\theta)+2(1-\rho_1)(d_2-\sigma)} - \rho_1. \tag{3}$$

The analysis of Earliest Deadline First scheduling in heavy traffic has been presented in [12]. We follow the approach of [12] and consider preemptive resume EDF scheduling, where all customers are accepted into the system, and can thus have negative deadlines. Applying the results of Section 4, formulas (4.4) to (4.7) in [12], we obtain that in stationarity, given the total workload in the system W, the workload due to customers with current lead-times not larger than x is approximately

$$W(x) \approx W - H(x \vee F),$$

where

$$F = H^{-1}(W)$$

and the function H is derived from the distribution of deadlines. In the present situation, where deadlines equal d_i with probability λ_i/λ, $i = 1,2$, we obtain that

$$H(x) = \begin{cases} 0 & \text{if } x > d_2, \\ (d_2 - x)(1-\lambda_1/\lambda) & \text{if } d_1 \leq x \leq d_2, \\ (1-\lambda_1/\lambda)(d_2-d_1) + (d_1 - x) & \text{if } x < d_1. \end{cases}$$

It follows from this definition that

$$H^{-1}(x) = \begin{cases} d_2 - (1-\lambda_1/\lambda)^{-1}x & \text{if } 0 \leq x \leq (d_2-d_1)(1-\lambda_1/\lambda), \\ d_1 - (x - (d_2-d_1)(1-\lambda_1/\lambda)) & \text{if } x \geq (d_2-d_1)(1-\lambda_1/\lambda). \end{cases}$$

Thus from the global workload one can infer the workload profile. An incoming type 1 packet will not meet its deadline if and only if $W(d_1) + \sigma > d_1$. Therefore we need to evaluate the probability that $W(d_1) > d_1 - \sigma$. From the definition of H one can establish:

$$W(d_1) + \sigma > d_1 \Leftrightarrow d_1 - F > d_1 - \sigma \Leftrightarrow W > H(\sigma). \tag{4}$$

In the case of a type 2 packet arrival, its deadline can be violated because $W(d_2) > d_2 + \sigma$, but it could also be violated at a later time, due to class 1 packet arrivals in the interval of length $d_2 - d_1$ following the arrival. In order to assess the probability of such an event, we resort to the so-called snapshot principle [11] that applies in a heavy traffic regime, and according to which the state of the system does not change significantly during the sojourn of a customer. This coupled with the generic shape of the workload profile (zero on $(-\infty, F]$, concave non-decreasing on $[F, \infty)$, with slope 1 on $[F, d_1]$) implies that a type 2 packet will not meet its deadline if and only if

$$W(d_2) > d_2 - \sigma \text{ or } W(d_1) > d_1,$$

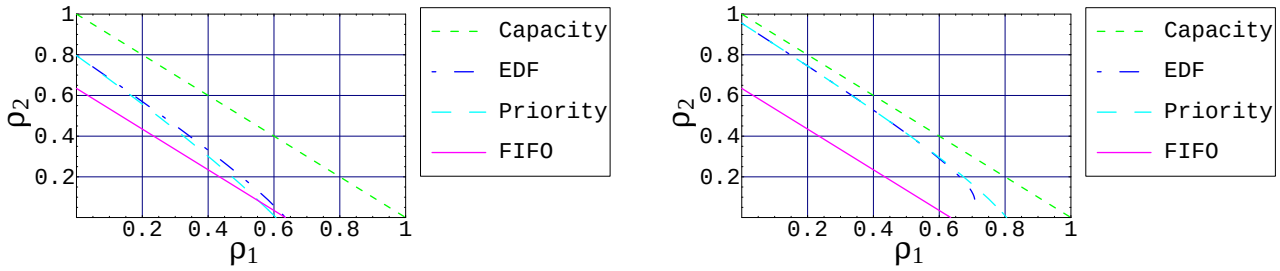

Fig. 1. Schedulable regions for $d_2 = 10$ (left) and $d_2 = 50$ (right), $\theta = 0.01$, $\sigma = 1$, $d_1 = 5$.

which is equivalent to

$$W > H\left(\max\left(0, \sigma - (d_2 - d_1)\lambda_1/\lambda\right)\right). \tag{5}$$

Because the function H is non-increasing, event (5) is less likely than event (4). Finally, using the heavy traffic approximation for the distribution of W, together with the expression of $H(\sigma)$ (recall that necessarily, $\sigma < d_1$) we obtain the condition

$$\frac{2(1-\rho)(d_1 - \sigma + (d_2 - d_1)(1 - \rho_1/\rho))}{\sigma\rho} \geq -\log\theta. \tag{6}$$

Figure 1 illustrates the different schedulable regions. As can be seen, priority can be outperformed by FIFO. However, a more typical situation is that FIFO is significantly outperformed by Priority, while Priority is only very slightly outperformed by EDF. For some sets of parameters, Priority can outperform EDF, which is unexpected in view of the optimality properties of EDF.

The comparison of the schedulable regions suggests that service differentiation can provide significant advantages in the presence of burstiness. In most cases, it appears that the improvement offered by EDF is larger than that offered by Priority, but the two are almost indistinguishable, which would argue in favour of implementing the simpler Priority scheme. [4]

3 The Congestion Pricing Framework for Delay-Sensitive Users

We now describe a framework which identifies user preferences with multidimensional utility functions. We make use of the resource pricing framework described by Kelly et al. [1,13] which requires the network to send feedback signals (or charges) to users, perhaps conveyed by a packet marking strategy, that reflect the

[4] Is is possible, however, that EDF would provide a significant improvement over Priority under a more realistic model of burstiness. We therefore continue to consider both mechanisms throughout this work.

marginal cost of congestion in the network. We consider both delay-based and loss-based congestion costs, and relate them to user utilities expressed in terms of either average delays or delay bounds.

3.1 Delay–based congestion cost

Define the delay–based congestion cost at a link to be $C(x) := xD$, where x and D are the packet sending rate and the average packet delay at that link, respectively. Consider a population of users indexed by r, and assume for now that the utility to user r of sending at rate x_r is given by $V_r(x_r)$. In other words, assume (temporarily) that user utility does not depend explicitly on delay but that the network imposes an additional delay–based cost. For the moment, we assume a single class of traffic, and also assume a network reduced to a single bottleneck. It is well known that if the network sets the packet price p to the marginal cost $C'(x) = xD'(x) + D(x)$, users trying to maximise their net benefit $V_r(x_r) - x_r p$ will increase the total welfare $\sum_r V_r(x_r) - C(x)$, and if furthermore the V_r are strictly concave and C is convex, a global optimum will be reached.

Noting that $x = \sum x_r$, the total welfare can also be written as a sum of user utility functions $\sum_r U_r(x_r, D)$, where $U_r(x_r, D) := V_r(x_r) - x_r D$. The delay–based cost is thus appropriate in a scenario where user utilities depend explicitly on average packet delay in this specific manner and the network imposes no additional cost. In this case, letting users optimise their net benefit $V_r(x_r) - x_r p = U_r(x_r, 0) - x_r p$, where disutility due to delay is accounted for in the price p as above, will achieve maximal welfare.

For the case of utility functions $U_r(x_r, D)$ depending on average delay in a general manner, we assume that each user neglects the impact of its own rate on average delay (i.e. $\partial D/\partial x_r = 0$)[5] Under this assumption, user r maximizes its benefit $U_r(x_r, D) - x_r p$ by adapting its rate x_r so as to set $\partial_{x_r} U_r$ equal to the congestion price p. The *correct* congestion price, which yields rates optimising total welfare $\sum_r U_r(x_r, D)$, then becomes

$$p = -D'(x) \sum_s \partial_D U_s(x_s, D).$$

This price is the sensitivity of delay at that link with respect to aggregate rate x multiplied by the sensitivity of utility with respect to delay aggregated over all users using the bottleneck link. Therefore the network needs to know the aggregate sensitivity of utility to delays if it is to infer the correct price p and signal it back to the users. This sensitivity could be signaled by letting each packet sent by user r carry a tag with value $\partial_D U_r / x_r$. The division by x_r has the effect that the aggregate

[5] This assumption is reasonable in the case of many small users.

rate of tags received by a link is equal to the desired sensitivity of utility with respect to delays.

If the network provides two classes of service, letting x_i and D_i be the sending rate and average delay of class i packets respectively, the delay-based congestion cost now reads $C(x_1,x_2) = x_1D_1 + x_2D_2$. The corresponding congestion price per type i packet is then given by $p_i = \partial_{x_i}C(x_1,x_2)$. As in the single class case, this is appropriate for users seeking to maximise total welfare when the utility to user r of sending jointly at rates x_{r1} and x_{r2} in classes 1 and 2 takes the form $U_r(x_{r1},x_{r2},D_1,D_2) = V_r(x_{r1},x_{r2}) - x_{r1}D_1 - x_{r2}D_2$.

3.2 Loss–based congestion cost

Alternatively, one can consider (as in [14,1]) the rate of packet loss events, $x\ell(x)$, as the cost of congestion, where $\ell(x)$ is the packet loss probability. This might be more appropriate in the situation where user utilities do not depend on average packet delays, but rather on maximal packet delay. For example, for a voice over IP application, only the rate of packets which experience a delay not greater than D_{max} contribute to the user utility. The network might implement scheduling policies that enforce specific delay bounds on the transmission of packets in each class, for instance by implementing EDF scheduling and dropping packets that would cause a deadline violation.

As in the case of delay-based cost, for utility functions $U_r(x_r,\ell)$ of the form $V_r(x_r) - x_r\ell$, the loss-based cost, given by $p = \partial_x(x\ell(x))$, is adequate, and can be interpreted as transferring the penalty term $x_r\ell$ from the users to the network. Another approach consists in taking utilities that depend only on the goodput, i.e. $U_r(x_r,\ell) = V_r(x_r(1-\ell))$. In that case, the correct congestion price is given by

$$p = \ell' \sum_s x_s V_s'(x_s(1-\ell)).$$

As for utilities depending on average delays, we see here that in general the network would need information on the sensitivity of utilities in order to infer the correct congestion price.

4 Rare Perturbation Analysis (RPA) Basis for Sample Path Shadow Prices

Based on the previous section, we want to signal to the end-users the sensitivity of loss–based or delay–based cost measures with respect to sending rates. The problem of computing this signal based on the observed system evolution is similar to that of sensitivity analysis in discrete event systems, where the goal is to estimate

the gradient of a performance measure with respect to a parameter vector using a single simulated sample path. An alternate view of the appropriate feedback is the *sample path shadow price* (s.p.s.p.) of a packet, which is defined (see [14] or [15]) as the difference between the actual cost and that which would occur if that packet had not been submitted. In this section, we relate sample path shadow prices to RPA estimates of the derivative of a cost function with respect to the aggregate rates of one or more classes of service, and thus to the sensitivity of social welfare cost, identified as a suitable congestion price in the last section.

Consider an M/G/1 queue that supports multiple classes of service, indexed by $i \in \{1,\dots,I\}$. Arrivals for each class are assumed to be Poisson, with the vector $\mathbf{x} = (x_i)$ describing their respective rates. A cost function $C(\mathbf{x})$ is given, which could be the rate of packet loss events, or the average packet delay incurred per time unit. In both cases, this cost measure takes the form $C(\mathbf{x}) = (\sum_i x_i)J(\mathbf{x})$, where $J(\mathbf{x})$ is the per-packet cost, while $C(\mathbf{x})$ is the cost per time unit. Assume packets are labeled by the order of their arrivals, irrespective of their classes. We further assume that packets $1,\dots,N$ constitute a busy period for the queue under consideration. We let $t(n) \in \{1,\dots,I\}$ denote the class of packet n. For the two cost structures under consideration, the average per-packet cost $J(\mathbf{x})$ can be expressed according to the well-known cycle formula as

$$J(\mathbf{x}) = \frac{\mathbf{E}\sum_{n=1}^{N} Z_n}{\mathbf{E}(N)}, \tag{7}$$

where Z_n is defined to be the cost to packet n, which for loss-based cost equals 1 if packet n is lost and zero otherwise, while for delay-based cost Z_n is simply the sojourn time of packet n in the queue.

A sample path for the system with rate x_i reduced to $x_i(1-h)$ is generated from the nominal sample path with arrival rate x_i by removing independently with a probability h each packet of type i. For all $\xi = \xi_1,\dots,\xi_n \in \{0,1\}^N$, denote by $Z_n(\xi)$ the cost to packet n when packets m such that $\xi_m = 0$ are removed from the arrival process, while those packets m with $\xi_m = 1$ are kept. We adopt the convention that $Z_n(\xi) = 0$ whenever $\xi_n = 0$. Let $f(h)$ be defined as

$$f(h) := \mathbf{E}_h \sum_{n=1}^{N} Z_n(\xi),$$

where the variables ξ_n are independent conditionally on the nominal sample path randomness, with $\mathbf{P}(\xi_n = 0) = h1_{t(n)=i}$. Similarly, define

$$g(h) := \mathbf{E}_h \sum_{n=1}^{N} \xi_n.$$

It then holds that $J(\mathbf{x} - x_i h\,\mathbf{e}_i) = f(h)/g(h)$, where $\mathbf{e}_i$ is the i-th unit vector, so that we can evaluate the derivative of J with respect to x_i by evaluating the derivatives

of f and g.

Letting $Z_n^{(-m)}$ denote the value of Z_n when the only packet that is removed is m, and N_i the number of type i packets in the busy period under consideration, we may write

$$f(h) = \mathbf{E}\left[(1-h)^{N_i}\sum_{n=1}^{N} Z_n + h(1-h)^{N_i-1}\sum_{m=1}^{N} 1_{t(m)=i}\sum_{n=1}^{N} Z_n^{(-m)} + o(h^2)\right].$$

Performing a binomial expansion of $f(0)$, we have

$$\frac{f(0)-f(h)}{h} = \mathbf{E}\left[(1-h)^{N_i-1}\sum_{m=1}^{N} 1_{t(m)=i}\sum_{n=1}^{N}(Z_n - Z_n^{(-m)})\right] + R(h),$$

where the term $R(h)$ accounts for the differences $Z_n - Z_n(\xi)$ for all ξ with at least two entries equal to zero. One expects this term to vanish as h goes to zero, since the probability of having more than two ξ_n equal to zero is of order $O(h^2)$. Assuming interchanges between limits and expectations are valid [6], we obtain

$$x_i\,\partial_{x_i} f = \lim_{h\to 0}\frac{f(0)-f(h)}{h} = \mathbf{E}\left[\sum_{m=1}^{N} 1_{t(m)=i}\sum_{n=1}^{N}(Z_n - Z_n^{(-m)})\right].$$

Similarly,

$$x_i\,\partial_{x_i} g = \lim_{h\to 0}\frac{g(0)-g(h)}{h} = \mathbf{E}\left[\sum_{m=1}^{N} 1_{t(m)=i}\right] = E[N_i].$$

This together with the quotient rule for differentiation, $(f/g)' = f'/g - (f/g)\,(g'/g)$, yields the following expression:

$$x_i\partial_{x_i} J(x) = \frac{\mathbf{E}\left[\sum_{m=1}^{N}\sum_{t(m)=i}\sum_{n=1}^{N}(Z_n - Z_n^{(-m)})\right]}{\mathbf{E}[N]} - \frac{\mathbf{E}[\sum_{n=1}^{N} Z_n]}{E[N]}\times\frac{\mathbf{E}[N_i]}{E[N]},$$

Using $E[N_i]/E[N] = x_i/\sum_j x_j$ we obtain an estimate for the derivative of cost per unit time,

$$\partial_{x_i} C(\mathbf{x}) = \frac{\mathbf{E}\sum_{m=1}^{N} 1_{t(m)=i}\sum_{n=1}^{N}\left(Z_n - Z_n^{(-m)}\right)}{\mathbf{E}[N_i]}. \tag{8}$$

Recall that the sample path shadow price consists in charging each packet m for the marginal cost it caused to other packets m. Since packet m has an impact only on packets belonging to the same busy period, the sample path shadow price for

[6] A rigorous proof of this fact typically involves a dominated convergence argument, and dominating variables have to be found for each specific situation; see [16] for more details.

packet m is exactly

$$\sum_{n=1}^{N} \left(Z_n - Z_n^{(-m)} \right).$$

Applying the cycle formula, it is then seen that the corresponding average packet price for packets of a given type, say type i, does coincide with that derived from the RPA analysis as in (8).

On-line computation of the sample-path shadow price of a packet is typically not feasible, because removal of the packet has an impact on future, not yet observed arrivals. In the next subsections we propose practically implementable alternatives to the exact sample path shadow price, for loss or delay–based cost, and for priority or EDF-based differentiation in the multi-class case.

4.1 RPA Estimators for a Single Class of Service

4.1.1 Loss-Based Shadow Price

For loss–based cost, the corresponding cost Z_n to packet n is 1 if packet n is lost and zero otherwise. As argued in [14], the sample path shadow price is 1 if in the nominal trajectory, one of the packets $m, m+1, \ldots, N$ is lost, and zero otherwise. Hence all packets up to and including the last lost packet in the busy period are marked (i.e. have a unit s.p.s.p.).

Kelly and Gibbens [14] suggest using an approximate marking scheme, according to which all packets in the busy period from the *first* lost packet, included until the end of the busy period, get marked. The motivation for this scheme is that it marks the correct number of packets per busy period on average, a fact that derives from the stochastic reversibility of sample paths of the M/D/1/C queue. We refer to the charge computed by this time-reversal approach as the *reversed s.p.s.p*.

One drawback of the reversed s.p.s.p. is that it charges the 'wrong packets'—that is, packets are charged according to the cost imposed on them rather than the cost they impose on others. This suggests the alternative where a packet is charged by its expected s.p.s.p. given the information available. Assuming for simplicity exponentially distributed service times with mean μ^{-1}, an incoming packet finding $n-1$ packets in the queue, would then be charged by the amount $(\rho^{-n}-1)/(\rho^{-C-1}-1)$, where $\rho = x/\mu$, and x is the packet arrival rate.

4.1.2 Delay-Based Shadow Price

For delay-based shadow price, the cost Z_n to packet n is simply its sojourn time in the queue. For FIFO queueing, one obviously has $(Z_n - Z_n^{(-m)}) = 0$ when $m > n$

and $(Z_n - Z_n^{(-m)}) = Z_n$ when $m = n$. For the case when $m < n$, we require some additional notation to characterize the effect of a packet's service time on the waiting time of future packets. Let T_n and σ_n be the arrival time and service requirement of packet n and let S_n be the time at which packet n begins service. Denote by $\delta_n^{(m)}$ the difference $Z_n - Z_n^{(m)}$ for $n > m$. Setting by convention $\delta_n^{(n)} = \sigma_n$, using Lindley's recursion one can show that the following recurrence holds for $n \geq m$:

$$\delta_{n+1}^{(m)} = \min\{[S_n + \sigma_n - T_{n+1}]^+, \delta_n^{(m)}\}.$$

An adaptation of the time reversal approach taken in [14] would let packet n arriving in busy period due to packets $1, \ldots, N$ be charged by the amount $Z_n + \sum_{n+1}^{N} \delta_n^{(m)}$. This scheme is such that the cumulated charge of packets $1, \ldots, N$ coincides with the cumulated sum of their exact s.p.s.p.. It is however not practical, as it requires the queue to keep track of the values $\delta_n^{(m)}$ for all packets m of the current busy period.

As an approximation, it may be useful to introduce the simplifying assumption that a packet imposes a delay equal to its service time on all later-arriving packets in the busy period, or equivalently to replace $\delta_n^{(m)}$ by the service time σ_m, for $m < n$ and m, n belonging to the same busy period. This amounts to marking packet n by its waiting time Z_n, plus the length of the current busy period when it enters service. The amount by which this overestimates the correct packet price on average can be computed exactly in the case of an M/D/1 queue, using, for instance, results of [17]). We find that the resulting average price is never more than twice the correct price, and approaches the correct price as load increases to 1.

An alternative approach is to charge each packet according to the conditional expectation of its s.p.s.p., given the state of the queue found upon arrival. Unlike the reversed s.p.s.p., this approach does not introduce extra charges, nor does it charge the wrong packets. In the case of the M/D/1 queue with constant service times σ, the resulting expected s.p.s.p. $p(w)$ of a virtual packet entering at time zero the queue and finding a workload of w is seen to be

$$p(w) = w + \sigma + \sigma \mathbf{E}(N(0, T(w)]) + K,$$

where $T(w)$ is the time at which the busy period started at 0 with workload w ends, $N(0, T(w)]$ is the corresponding number of packet arrivals and the constant K does not depend on w. Indeed, $w + \sigma$ is the sojourn time of the packet arriving at time 0, all packets arriving in the interval $(0, T(w)]$ are delayed by an amount σ, while the workload at time $T(w)$ is exactly σ hence the impact on later arriving packets does not depend on w. Applying the integration formula (8.3.3), p. 49 in [18] yields the expression $w\rho/(1-\rho)$ for the middle-term in the right-hand side of this expression. The constant K is then determined by using the expression $\partial_x(x\mathbf{E}(W + \sigma))$ for the

average s.p.s.p., yielding

$$p(w) = \frac{w}{1-\rho} + \sigma + \frac{x\sigma^2}{2(1-\rho)} \cdot \tag{9}$$

We note that the constant term in the above is exactly the expectation of the stationary workload. This suggests the simpler marking scheme, according to which a packet finding an amount of w is charged $w[1+1/(1-\rho)]$. This collects the correct amount on average, and under-charges slightly packets finding a near-empty system. Note that the expected length of a busy period is given by $\sigma/(1-\rho)$, so that the term $1/(1-\rho)$ in this scheme can thus be estimated as $\hat{B}/\sigma$, where $\hat{B}$ is a sample mean of the observed busy periods.

4.2 Sample Path Shadow Prices for Two Classes of Service

4.2.1 Delay-Based Shadow Price with Priority Scheduling

Recall that for a delay-based cost structure, the quantity Z_n represents the sojourn time of packet n. We adopt a multiclass version of the simplifying assumption from the single class case–that a packet imposes either no delay or a delay equivalent to its complete service time on later departing packets in the same busy period. Additionally, we assume a preemptive priority discipline.

The busy period of a priority queue is composed of smaller high priority busy periods separated by intervals of low priority service when there are no class 1 packets in the system. In the case of two high priority packets m and n, m will delay n if m arrives before n and both packets fall within the same high priority busy period. That is, if any class 2 packet is served between m and n, then m imposes no additional delay on n. Under the assumption of preemption, a low priority packet is delayed by any earlier departing high priority packet in the same busy period, as well as by earlier arriving low priority packets. The marginal cost contributions for other class combinations are easily verified. As in the single class case, the shadow price $\partial_{x_i} C(\mathbf{x})$ can be separated into two terms, where the first term is simply the average delay for class i.

$$\partial_{x_i} C(\mathbf{x}) = \frac{\mathbf{E}[\sum_{n=1}^{N} Z_n \mathbf{1}_{t(n)=i}]}{\mathbf{E}[N_i]} + \frac{\mathbf{E}[\sum_{m=1}^{N} \mathbf{1}_{t(m)=i} \sum_{n=1, n\neq m}^{N} (Z_n + Z_n^{(-m)})]}{\mathbf{E}[N_i]}. \tag{10}$$

The second term in (10), or rather its time-reversed counterpart, can be estimated with the help of the following charging mechanism.

The charge for a high priority packet has two components. The first is the cost imposed on subsequent high priority packets within the same high priority busy period. For this charge, we can do as in the single class case, and charge the number

of high priority packets in the same high priority busy period served prior to itself. The second component is the cost imposed on low priority packets and reversal cannot be applied as we don't want low priority packets to pay for the harm done to them by high priority packets. The high priority packet delays all those low priority packets currently in the queue plus those that will arrive before the end of the busy period. As an approximation, we charge the length of low priority queue at the time the high priority packet enters service, multiplied by a factor $1 \leq \alpha \leq 2$. [7] To charge the low priority packets, we can treat the class 2 arrival process as reversible and charge the number of low priority packets served since the start of the current busy period.

As in the single class case, alternative schemes which do not introduce over-charging can be designed, based on formulas available for mean delays in M/G/1 priority queues (see e.g. [18], p.188-192). We shall not pursue this here.

4.2.2 Loss-Based Shadow Price with Priority Scheduling

In this section, we define the sample path shadow price for a priority queue with finite buffer serving two classes of traffic in strict priority order. Gibbens and Kelly suggest a model for such a queue [1] in which the packet dropping policy is governed by two parameters, B_1 and B_2 so as to enforce the following constraints:

$$q_1 \leq B_1 \tag{11}$$
$$q_1 + q_2 \leq B_2, \tag{12}$$

where q_1 and q_2 are the buffer occupancies of high priority and low priority packets, respectively. It is assumed that high priority arrivals do not push out low priority packets if the high priority constraint (11) is slack while (12) is binding.

Gibbens and Kelly propose treating such a queue as a pair of virtual single class resources. The first virtual resource is a single class queue with buffer capacity B_2 used by all packets, while the second is a single class queue with buffer capacity B_1 used only by high priority packets. Sample path shadow prices are computed independently for each virtual resource according to the single-class loss-based marking scheme originally proposed by the same authors in previous work [14] and discussed above. The resulting marking scheme requires that, for each lost packet, we distinguish which of constraints (11) and (12) has been violated. Following the first loss due to constraint (11), the queue marks all high priority packets until the high priority queue becomes idle. Following the first loss due to constraint (12), the queue marks all packets until the end of the busy period. Observe that a high priority packet has two chances to be marked, but carries only a single marking bit.

[7] In simulations, we observe that α has very little effect on the performance measures of interest. For the experimental results presented later, $\alpha = 1$.

It can be shown that the Gibbens-Kelly marking scheme, modulo time reversal, indeed computes the correct shadow price for loss. The essential step is deriving expressions for $Z_n - Z_n^{(-m)}$. Consider the case where $t(m) = 2$. Regardless of the class of n, the removal of m prevents a loss of n only if n is lost due to a violation of the total capacity constraint (12) and if there is no intervening loss due to its violation. Formally,

$$Z_n - Z_n^{(-m)} = \mathbf{1}_{q_{1n}+q_{2n}=B_2} \prod_{k=m}^{n-1} \mathbf{1}_{q_{1k}+q_{2k}<B_2} \quad t(m) = 2. \tag{13}$$

The same relation applies when $t(m) = 1$ and $t(n) = 2$. When both m and n are high priority packets ($t(m) = t(n) = 1$), a loss of n can be prevented by the removal of m if n violates either of constraints (11) and (12).

$$Z_n - Z_n^{(-m)} = \mathbf{1}_{q_{1n}=B_1} \prod_{k=m}^{n-1} \mathbf{1}_{q_{1k}<B_1} + \mathbf{1}_{q_{1n}+q_{2n}=B_2} \prod_{k=m}^{n-1} \mathbf{1}_{q_{1k}+q_{2k}<B_2} - R(m,n) \tag{14}$$

$$R(m,n) = \mathbf{1}_{q_{1n}=B_1} \mathbf{1}_{q_{1n}+q_{2n}=B_2} \prod_{k=m}^{n-1} \mathbf{1}_{q_{1k}<B_1} \mathbf{1}_{q_{1k}+q_{2k}<B_2} \tag{15}$$

When considering marking mechanisms using a single bit, we may ignore the intersection term (15) since the bit can only be marked once even if both constraints are binding.

The cost imposed by a low priority packet on other packets is equivalent to that of a packet passing through the first virtual resource, while the cost imposed by a high priority packet is equivalent to that of packet passing through both logical resources.

4.2.3 Loss-Based Shadow Price with Earliest Deadline First Scheduling

In this section, we define a sample path shadow price for the single server queue serving packets of unit size with two service classes in an earliest deadline first (EDF) discipline. Under EDF, a two class system is implemented by granting high priority arrivals a short deadline d_1 and low priority arrivals a deadline $d_2 \gg d_1$. As with priority scheduling, we will approximate the operation of a real non-preemptive server by assuming preemption to simplify the analysis.

An arriving packet is lost if the EDF scheduler determines that accepting the packet would lead to a violation of either its own deadline, or that of another packet already in the queue. As with other loss-based schemes, $Z_n = 1$ if packet n is lost, otherwise $Z_n = 0$. Thus, $Z_n - Z_n^{(-m)} = 1$ for those packets whose removal from the busy period would allow a lost packet n to be accepted.

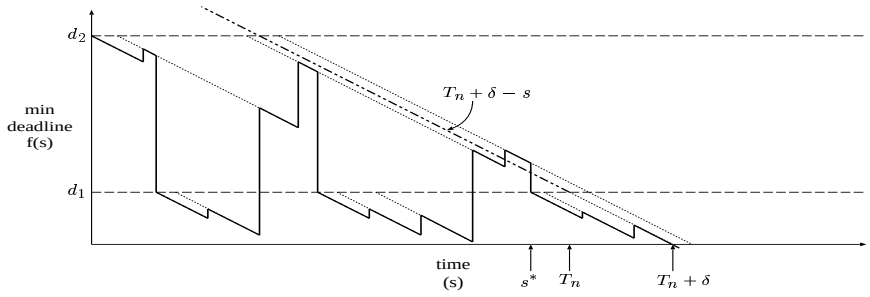

Fig. 2. A sample path of the process $f(s)$, showing the beginning of the local busy period at s^* for a loss at time T_n. Dotted lines show the evolution of packet lead times. The bold solid line shows $f(s)$, the minimum deadline currently in the system at time s.

Suppose packet n is rejected by the scheduler. Let δ be the time at which the first deadline violation would occur if the packet were accepted. The *lead time* of a packet k at time t, denoted $d_k(t)$ is the time remaining before its deadline expires. Observe that since d_1 is the minimum deadline that can be assigned to the arriving packet, $\delta \geq d_1$. In other words, packet k with lead time $d_k(T_n) < d_1$ would not experience a deadline violation if the arriving packet were accepted. It is also clear that any packet k in the queue at time T_n with $d_k(T_n) > \delta$ is not responsible for the loss of the arriving packet since the presence or absence of such a packet would not affect the time at at which the deadline violation occurs.

If the rejected packet n is low priority ($d_n(T_n) = d_2$), then all previous packets in the busy period are responsible. To compute the reversed sample path shadow price, we may simply begin marking packets once a low priority packet has been rejected and continue until the queue is idle.

The more complicated (and more likely) case is that a high priority packet is rejected. Such a loss is due not only to high priority packets but any low priority packet m with $d_m(T_n) < d_1$ as well. The effect of such packets on the loss of n can be understood in terms of the process $f(s)$, defined as the value of the earliest deadline present in the system at time s. The arrival of n occurs in what we will call a *local* busy period. High priority packets arriving before the start of this local busy period have no effect on the loss of n. The local busy period associated with n begins with the arrival of a prior high priority packet at time s^*, such that

$$s^* = \sup\{s < T : f(s^+) < T + \delta - s < f(s^-)\}, \tag{16}$$

where $f(s^-)$ and $f(s^+)$ respectively represent the earliest deadline just prior to and following an arrival at s. If no such time s exists, then the local busy period begins at the start of the global busy period.

Figure 2 shows a sample path for $f(s)$ that includes a packet loss at time T_n. We see that the local busy period begins with the arrival of a high priority packet that causes $f(s)$ to fall below the line defined by $T_n + \delta - s$. This arrival and all subsequent high priority arrivals (up to n) contribute to the loss since the removal of any of them

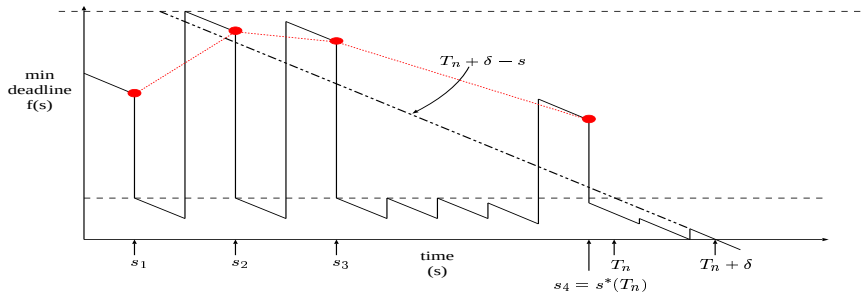

Fig. 3. Effect of update rule (17). The dots show successive updates of $(s^*, f(s^*))$. When a loss occurs at time T_n, s_4 is the current estimate for s^* and also happens to be the correct beginning of the local busy period.

would prevent the deadline violation at $T_n + \delta$. High priority arrivals that precede the local busy period are not responsible for the loss since the loss would still occur if any one of them were removed.

If the local busy period is shorter than the global busy period, then there will be no low priority packets served between s^* and T_n. However, if the local and global busy periods coincide, then there may be a sufficiently old low priority packet m in the queue with $d_m(T_n) < \delta$. In particular, any low priority packets arriving *prior* to $t = T_n + \delta - d_2$ within the same global busy period should be charged. We can thus define two intervals during which high and low priority arrivals should be charged for the loss at time T_n. High priority arrivals in the interval $[s^*, T_n]$ and low priority arrivals in the interval $[s^*, t]$ should be marked. Note that if $t < s^*$, no low priority packets should be charged.

It is clear that the charging mechanisms described for lost high priority packets are non-causal. We implement causal approximations of these mechanisms by marking packets for the appropriate duration following a loss. Such a marking scheme can be implemented using a timer and a pair of busy flags that are turned on when a loss occurs and turned off after the appropriate interval has expired. The main challenge in implementing this non-causal mechanism is determining s^*. Candidate values for s^* correspond to the beginning of the busy period and the arrival times of any high priority packet for which the value of $f(s)$ is reduced (i.e. a high priority arrival that occurs when the system contains only low priority packets with lead times greater than d_1). One way to determine s^*, therefore, is to maintain a list of candidates as the busy period evolves and to evaluate (16) when a loss occurs and T_n and δ become known.

Since the number of candidate points is potentially unbounded, we propose an approximation in which only the most promising candidate is retained. At the beginning of the busy period, we initialize $s^* = 0, f(s^*) = 0$. When a candidate high priority arrival occurs at time s, we apply the following update rule:

$$\text{if} f(s^*) - f(s^-) > s^* - s, \text{then} (s^*, f(s^*)) \leftarrow (s, f(s^-)), \tag{17}$$

where $f(s^-)$ is the value of the earliest deadline just prior to the arrival at time s. The effect of this update rule is shown graphically in Fig. 3.

5 Simulation Results

We ran simulations to compare the performance of the three proposed mechanisms with that of a single-class FIFO scheduler. The objectives of these experiments are twofold. First, we would like to observe the predicted efficiency gain due to the introduction of a low delay service class. Second, we would like to determine whether our marking schemes provide the correct congestion feedback to users of both classes while treating the low priority class fairly.

To obtain these results, we performed an experiment proposed by Gibbens and Kelly [1]. The queue in this experiment operates in discrete time, serve one packet per time slot, and is shared by a population of N users with willingness-to-pay $w_i = w_0\, i, \quad i = 1, \ldots, N$. In each simulation run, each user is designated as either high or low priority and generates packets of the appropriate type. Users send packets in accordance with the *elastic–user* algorithm proposed for in [14], and which corresponds to user i having a utility function $w_i \log(x_i)$ for sending at rate x_i in the framework of Section 3. In the single-class case with a population elastic-users sharing a single link, user i will achieve throughput proportional to its relative willingness-to-pay $w_i / \sum_{j=1}^{N} w_j$. Associated with the high and low service classes, respectively, are delay bounds d_1 and d_2, with $d_1 < d_2$. Packets served past their delay bounds are discarded by the users and thus considered lost.

For the initial run of the simulation, all users are designated as low priority and the system simulated is run for enough time steps to observe convergence of the sending rates. A low priority user is then selected at random to promote to the high priority class and the simulation is run again. Users are repeatedly promoted and the system simulated until all users generate high priority packets.

5.1 Schedulable Regions

Our first result is an experimental validation of the analytical results of Section 2. Recall that these results describe, qualitatively, the relative amounts of available capacity that can be consumed while still respecting delay bounds under various scheduling disciplines. In the experimental counterpart to this result, we record the *goodput*—the throughput of packets that respect delay bounds $d_1 = 5$ and $d_2 = 50$—for each user. Plotting aggregated goodput for high priority traffic against that for low priority traffic for the different mixes of traffic generated over the simulation iterations defines the boundary of the schedulable region. We compare the resulting

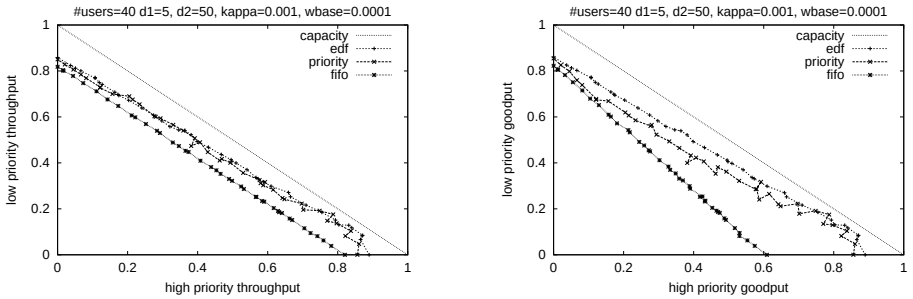

Fig. 4. Aggregate throughput (left) and goodput for high vs. low priority traffic. The topmost diagonal line represents the available capacity in the system. We observe that as the level of high priority traffic increases, the FIFO scheduler has great difficulty making use of the available capacity while still satisfying the delay bounds.

plots for EDF and loss-based priority schedulers with that of a single class FIFO scheduler. [8]

The left-hand plot of Fig. 4 shows the raw throughputs for the three disciplines and gives some idea of the level of utilization achieved by each. Note that the elastic-user parameters for this experiment were not tuned to optimize utilization. Rather we sought (empirically) a consistent set of parameters that would yield roughly comparable utilization for all three scheduling disciplines. We observe that all three scheduling disciplines achieved utilizations between 0.8 and 0.9.

When we consider only packets that arrive within the delay bounds, we obtain an equivalent plot for goodput, shown on the right in Fig. 4. When compared with the results presented in Section 2, these results show the same qualitative behavior predicted by theory. EDF and the priority scheduler both achieve substantially higher goodput than FIFO as the proportion of high priority traffic increase. Also, EDF makes somewhat better use of available capacity than strict priority scheduling. The difference between EDF and priority scheduling appears to be more substantial than the analytical results suggest. We conjecture that this difference is due to the heavy traffic approximations employed in the analysis.

5.2 Class performance: Incentive compatibility and fairness

The performance of the system is evaluated on the basis of the average waiting time for each class and the average transmission rate (normalized by willingness-to-pay) for each class. By plotting these performance metrics for each class with respect to the proportion of high priority demand, we can get some idea of whether the pricing mechanisms we propose indeed provide the correct incentives to users of

[8] Note that we do not provide a plot for the delay-based pricing mechanism in this section since this mechanism cannot be tuned to provide deterministic delay bounds for either class with the elastic-user strategy.

each class and whether members of the low priority class are treated fairly. We make the assumption, similar to Alvarez and Hajek [6] and Hurley et al. [19], that users of the low priority class are delay tolerant, but seek to maximize throughput whereas high priority users seek low delay, perhaps at the expense of lower throughput. Our criteria for incentive compatibility and fairness as the proportion of high priority demand increases are that (a) low priority users get throughput at least as good as when no high priority traffic is present and thus have no incentive to switch classes, (b) high priority users experience lower delay than low priority users, and (c) the delay for low priority users is finite.

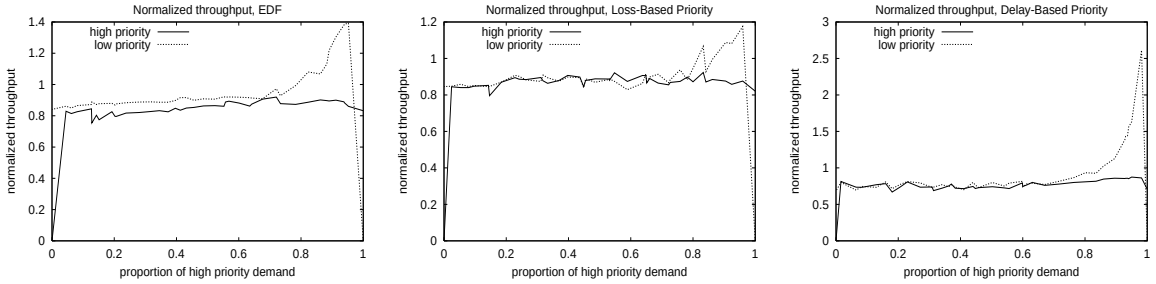

Fig. 5. Average normalized throughput for each class, plotted for all three service disciplines.

Figure 5 shows the normalized throughputs for high and low priority classes. [9] These plots show that all three disciplines satisfy the throughput fairness criterion. Under heavy high priority demand, the low priority class actually can receive substantially better throughput than in the single class case. [10] We also observe that, for moderate amounts of high priority demand (< 0.8) the high priority class achieves a throughput comparable to the low priority class, although this was not an explicit design goal.

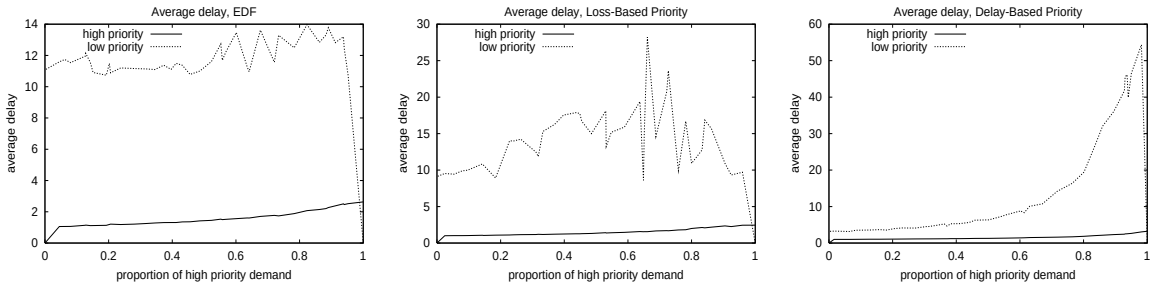

Fig. 6. Average delay for each class, plotted for all three service disciplines.

[9] Observe that while the sum of throughputs clearly must be limited by the total capacity of the link, no such restriction applies to the sum of *normalized* throughputs shown here.
[10] The single class case is shown by the endpoints of each plot–the throughput of the low (resp. high) priority class when the proportion of high (resp. low) priority traffic is zero.

Figure 6 shows the average delays for both classes. Under all three disciplines, high priority class traffic sees a substantially lower delay than low priority traffic. Furthermore, the average delay seen by the low priority traffic is well below its delay bound of 50ms, a property guaranteed only by the EDF scheduler, but that appears to be satisfied by the loss-based priority scheduler. The variability in the low priority delay in the loss-based priority scheduler is due to the presence of bursts in the low priority queue length. The number of such bursts in an experiment iteration contributes significantly to the average waiting time. While bursts are also present in the time series for the EDF and delay-based priority schedulers, they appear to be more periodic in nature and thus make a comparable contribution in each iteration of the experiment as high priority demand is increased.

An explanation of the plot for the delay-based priority scheduler requires some care. Although not shown in Fig. 4, the link operates at fairly low utilization under the delay-based pricing mechanism for the parameters we have selected, which explains the existence of very low delays for low priority traffic in the presence of low high priority demand. At the same time, this scheduler employs an infinite buffer, which can allow low priority delays to become large. Indeed, it is only the pricing mechanism that prevents low priority delays from growing without bound. The parameters used in this experiment were tuned empirically to achieve normalized throughput that is comparable to the other disciplines. In general, it is difficult to achieve high utilization under our delay-based pricing mechanism since the queues must clear frequently to keep the price per packet low. It is worth noting that an alternative mechanism based on conditional expectation (see Section 4) might allow the queue to stabilize at a modest size, but we leave it as an area of future research.

6 Conclusions

When users are sensitive to delay and there is but a single class of service and FIFO scheduling, delay requirements for such users can only be met if the network operates in lightly loaded region. The introduction of scheduling, such as priority or EDF allows much more efficient use of the network. We have quantified the benefits analytically for unresponsive traffic, where the gains are related to the burstiness, and by simulation for responsive users.

For responsive users, we have shown how to calculate the correct ‘shadow-price’ or feedback signals that reflect congestion costs, allowing decentralized optimization of social welfare which reflects multidimensional utility functions. We have derived sample-path shadow prices for loss-based and delay-based priority queues, and loss-based prices for EDF and used these as a basis for practical marking schemes. These were then used with a simple user-adaptation algorithm in simulations.

Our results show that such marking schemes are effective for controlling delay to sensitive users, with tangible gains over a single-class FIFO, and also provide a way of implementing differentiation for proposals such as ABE [19]. In the simulations, delay-based marking for priority ran at a low utilisation, suggesting some further tuning is required. EDF performed slightly better than priority with loss-based marking, but the latter is simpler to implement. This resonates with the analytic heavy-traffic results which also showed that little was gained by using EDF rather than priorities for unresponsive traffic. It is worth noting, however, that EDF offers the additional flexibility of allowing end-users to specify their delay requirements directly.

The difficult question remains of where, if anywhere, such service differentiation should, in fact, be deployed in the network. This decision ought to depend on whether actual burstiness of future network traffic makes the over-provisioning of capacity economically unattractive to network operators. The availability of efficient and controllable alternatives to over-provisioning, such as those we have shown, allows this choice to be grounded in economic concerns, rather than technological ones.

References

[1] R. J. Gibbens, F. P. Kelly, Resource pricing and the evolution of congestion control, Automatica .

[2] D. E. Wrege, E. W. Knightly, H. Zhang, J. Liebeherr, Deterministic delay bounds for vbr video in packet-switching networks: Fundamental limits and practical tradeoffs, IEEE/ACM Trans. on Networking 4 (3) (1996) 352–362.

[3] V. Sivaraman, F. M. Chiussi, M. Gerla, Traffic shaping for end–to–end delay guarantees with edf scheduling, in: Proc. of IWQoS 2000, 2000.

[4] Z. Zhang, Z. Duan, Y. T. Hou, Fundamental trade-offs in aggregate packet scheduling, in: Proc. ICNP 2001, 2001.

[5] S. Bajaj, L. Breslau, S. Shenker, Is service priority useful in networks?, in: Proceedings of ACM Sigmetrics '98, 1998, pp. 66–77.

[6] J. Alvarez, B. Hajek, On using marks for pricing in multiclass packet networks to provide multidimensional QoS, submitted (available at http://tesla.csl.uiuc.edu/ hajek/Papers/AlvarezHajek.ps) (2001).

[7] J. Liebeherr, D. Wrege, D. Ferrari, Exact admission control for networks with bounded delay services, ACM/IEEE Transactions on Networking 4 (1996) 885–901.

[8] J.-Y. L. Boudec, P. Thiran, Network Calculus, a theory of deterministic queuing systems for the Internet, Springer Verlag, 2001.

[9] C.-S. Chang, Performance guarantees in communication networks, Springer, 2000.

[10] E. project COST 224, Performance evaluation and design of multiservice networks, European Community, 1992.

[11] M. Reiman, Open queueing networks in heavy traffic, Mathematics of Operations Research 9 (3) (1984) 441–458.

[12] B. Doytchinov, J. Lehoczky, S. Shreve, Real-time queues in heavy traffic with earliest-deadline-first queue discipline, Annals of Applied Probability 11 (2) (2001) 332–378.

[13] F. Kelly, A. Maulloo, D. Tan, Rate control for communication networks: shadow prices, proportional fairness and stability, Journal of the Operational Research Society 49.

[14] R. J. Gibbens, F. P. Kelly, A note on packet marking at priority queues, to appear in IEEE Transactions on Automatic Control (available at http://www.statslab.cam.ac.uk/ frank/PAPERS/tac.html) (2000).

[15] D. Wischik, How to mark fairly, in: Workshop on Internet Service Quality Economics, MIT 1999, 1999, (Available at http://www.statslab.cam.ac.uk/ djw1005/Stats/Research/marking.html).

[16] F. J. Vazquez-Abad, Discrete Event Systems, Analysis and Control (R. Boel and G. Stremersch, ed.), Kluwer Academic Publishers, 2000, Ch. A Course on Sensitivity Analysis for Gradient Estimation of DES Performance Measures.

[17] J. Walraevens, H. Bruneel, Analysis of a single server atm queue with priority scheduling, in: Proceedings of the COST 257 9th Management Committee Meeting, 1999.

[18] F. Baccelli, P. Brémaud, Elements of Queueing Theory, Springer Verlag, 1994.

[19] J.-Y. L. B. P. Hurley, M. Kara, P. Thiran, ABE: Providing a low-delay service within best-effort, IEEE Network, Special issue on Control of Best-Effort Traffic 15 (3).