Proportional Differentiated Services: Delay Differentiation and Packet Scheduling

Constantinos Dovrolis
University of Wisconsin
dovrolis@ece.wisc.edu

Dimitrios Stiliadis
Bell Laboratories
stiliadi@dnrc.bell-labs.com

Parameswaran Ramanathan*
University of Wisconsin
parmesh@ece.wisc.edu

Abstract

Internet applications and users have very diverse service expectations, making the current *same-service-to-all* model inadequate and limiting. In the *relative differentiated services* approach, the network traffic is grouped in a small number of *service classes which are ordered based on their packet forwarding quality*, in terms of per-hop metrics for the queueing delays and packet losses. The users and applications, in this context, can *adaptively* choose the class that best meets their quality and pricing constraints, based on the assurance that *higher classes will be better, or at least no worse, than lower classes*. In this work, we propose the *proportional differentiation model* as a way to refine and quantify this basic premise of relative differentiated services. The proportional differentiation model aims to provide the network operator with the *'tuning knobs'* for adjusting the quality spacing between classes, *independent of the class loads*; this cannot be achieved with other relative differentiation models, such as strict prioritization or capacity differentiation. We apply the proportional model on queueing-delay differentiation only, leaving the problem of coupled delay and loss differentiation for future work. We discuss the *dynamics* of the proportional delay differentiation model and state the conditions under which it is *feasible*. Then, we identify and evaluate (using simulations) two *packet schedulers* that approximate the proportional differentiation model in heavy-load conditions, even in short timescales. Finally, we demonstrate that such per-hop and class-based mechanisms can provide consistent end-to-end differentiation to *individual flows* from different classes, independently of the network path and flow characteristics.

1 Introduction

The Internet is currently used by business and user communities with diverse service requirements from the network infrastructure. In addition, many Internet applications, even when they are adaptive, can perform well only in certain service-level conditions. Consequently, there is a growing demand for replacing the current *same-service-to-all* paradigm with a model in which packets, applications, and users are treated differently based on their service needs. The traditional approach for addressing this problem is to replace the existing *best-effort service* model with a *reservations-based* architecture in which applications and users request a certain performance level that can be guaranteed using resource reservation and admission control mechanisms. This architecture, commonly referred to as *Integrated Services*, faces some important difficulties, such as the deployment and scalability of the resource reservation protocol (RSVP) [1], the requirement for an interdomain policy and pricing infrastructure, and the mapping between application and network service parameters. Although there are proposals for alleviating some of the difficulties in the Integrated Services architecture [2, 3], the challenge of scalable service differentiation in the Internet remains open. A promising new approach is the *Differentiated Services (DS)* work within the Internet Engineering Task Force (IETF) [4]. The goal of the DS effort is to define configurable types of packet forwarding (called Per-Hop Behaviors or PHBs), that can provide local (per-hop) service differentiation for large aggregates of network traffic, as opposed to end-to-end performance guarantees for individual flows.

One approach within the DS architecture aims to provide the Integrated Services kind of performance measures, but without using per-flow state in the network core. We refer to this approach as *absolute differentiated services*, since the user receives an absolute service profile (e.g. a certain bandwidth) from the network. For example, assuming that no dynamic routing occurs, the Premium Service [5] can offer the user a performance level that is similar to that of a leased-line, as long as the user's traffic is limited to a nominal bandwidth. In the Assured Service [6], packets are classified into two levels of drop-preference ('In' and 'Out') at the network edges, depending on whether the user follows the allocated bandwidth profile or not. When congestion occurs, 'Out' packets are discarded with a higher probability than 'In' packets. An open question regarding the absolute differentiated services is whether they can provide with a high likelihood the end-to-end performance that users expect. The trade-offs in achieving high service assurance, coarse spatial granularity (i.e., users would like a certain end-to-end bandwidth in many or even all network paths), and high network utilization, have been discussed in [7]. It turns out that some form of *route pinning* is necessary for implementing such services, which may constitute by itself an obstacle for wide-scale deployment in the Internet.

* The work of C.Dovrolis and P.Ramanathan reported in this paper was supported in part by the National Science Foundation under Grant No. MIP-9526761.

A fundamentally different approach in the DS framework is the *relative differentiated services*. In this approach, the network traffic is grouped into N classes of service which are ordered, such that *Class i is better (or at least no worse) than Class $(i-1)$ for $1 < i \leq N$, in terms of local (per-hop) metrics for the queueing delays and packet losses.* Note that the elucidation 'or no worse' is required, since in low-load conditions all classes will experience the same quality level. The *Class Selector PHBs*, recently standardized by the IETF [4], follow this model of relative service differentiation. In this context, applications and users do not get an absolute service level assurance, such as an end-to-end delay bound or bandwidth, since there is no admission control and resource reservations. Instead, the network assures them that higher classes will be relatively better than lower classes, and so it is up to the applications and users to select the class that best meets their requirements, cost, and policy constraints. In this aspect, the relative service differentiation model follows the architectural principle of *end-system adaptation*, since it provides the choice of the service class as an additional dimension in the end-system adaptation space. It can be argued that the relative differentiated services approach is similar with the architecture of the postal service system, in which there are several priority classes that operate without admission control, end-to-end resource reservations, or service guarantees. We do not attempt here a comparison between relative and absolute differentiated services, since this is much larger an issue than the scope of this paper. It is likely that both approaches will coexist, since they target different applications and uses.

There are several ways in which a network can provide relative differentiated services. For example, the differentiation can be strictly based on appropriate pricing (i.e., higher classes are more expensive), or on careful capacity provisioning (i.e., higher classes have more forwarding resources relative to their expected loads). Such mechanisms, however, cannot always provide consistent differentiation between classes, because the relative differentiation between classes varies with the class loads. Especially with the Internet traffic, which is known to be bursty over a wide range of timescales, there can be long intervals in which a higher class is overloaded, to the extent that it offers worse service than a lower class. Other mechanisms, such as strict prioritization between classes, provide consistent class differentiation that does not depend on the load variations, but they do not allow the network operator to adjust the quality spacing between classes. Such 'tuning knobs' are necessary in a practical setting, since the network operators must be able to adjust the quality spacing between classes depending on pricing and policy objectives. Our basic premise, starting from these limitations of other relative differentiation models, is that in order for a relative differentiated services architecture to be effective for both users and network operators, it has to be:

- *Predictable*, in the sense that the differentiation should be consistent (i.e., higher classes are better, or at least no worse) independent of the variations of the class loads, and

- *Controllable*, meaning that the network operators should be able to adjust the quality spacing between classes based on their selected criteria.

As a target for predictable and controllable relative differentiation, the first contribution of this paper is to propose the *proportional differentiation model*. According to this model, the basic performance measures for packet forwarding locally at each hop are ratioed proportionally to certain *class differentiation parameters* that the network operator chooses. Even though there is no wide consensus on the most appropriate performance measures for packet forwarding, it is generally agreed that a better network service means lower queueing delays and lower likelihood of packet losses. In this paper, we apply the proportional model to the case of queueing delay differentiation only, leaving the more general problem of coupled delay and loss differentiation for future work. Consequently, the results presented here are not directly applicable to applications and transport protocols that depend on both queueing delays and packet losses (e.g. TCP). They are applicable to delay-sensitive applications, such as IP-telephony and video-conferencing, or to transaction-based applications. TCP-based applications, however, can also benefit from delay differentiation, especially in cases where the congested gateways have a large number of buffers and the queueing delays are a significant component of the round-trip delay; the important effect of queueing delays in TCP has been discussed in [8].

In the context of queueing delay differentiation, we first discuss the dynamics of the proportional model, and then state the conditions under which this model is feasible. We then focus on the issue of appropriate packet schedulers for proportional delay differentiation. We identify two schedulers that can approximate the proportional delay model in heavy-load conditions: the Backlog Proportional Rate (BPR) scheduler, and the Waiting-Time Priority (WTP) scheduler (originally studied by L.Kleinrock [9]). An important observation is that the two schedulers tend to the proportional differentiation model during heavy-load conditions, *even in short timescales*. This is a significant feature of the schedulers, since long-term averages do not always convey useful information when the traffic is bursty over long timescales. In the absence of appropriate analytical tools for studying the behavior of these schedulers with non-Poisson traffic models and in short timescales, we use simulations with bursty traffic in most of our evaluation study.

The structure of the paper is as follows. In Section 2, we describe the proportional differentiation model and compare it with other relative differentiation approaches. In Section 3, we discuss the dynamics and feasibility conditions of the proportional delay differentiation model. In Section 4, we describe two packet schedulers that are designed for predictable and controllable delay differentiation, even in short timescales. The main evaluation and comparison of the two schedulers follows in the simulation study of Section 5. In Section 6, we take the user's perspective and investigate if such local and class-based forwarding mechanisms are able to provide consistent end-to-end differentiation to individual flows from different classes. Finally, Section 7 closes the paper by identifying some open questions in this area of Internet research.

2 The Proportional Differentiation Model

The proportional differentiation model 'spaces' certain class performance metrics proportionally to the differentiation parameters that the network operator chooses. If, say, q_i is such a performance measure for class i, the proportional differentiation model imposes constraints of the following form for all pairs of classes:

$$\frac{q_i}{q_j} = \frac{c_i}{c_j} \qquad (i, j = 1 \ldots N)$$

where $c_1 < c_2 < \ldots < c_N$ are the generic *quality differentiation parameters*. So, even though the actual quality level of each class varies with the class loads, the quality ratio between classes remains fixed and controlled by the network operator, independent of the class loads.

In this paper, we focus on queueing delay differentiation only, and so we apply the proportional differentiation model to a queueing delay metric. The simplest such metric is the long-term average queueing delay of a class. Specifically, if $\bar{d}_i$ is the average queueing delay of the class-i packets, the proportional delay differentiation model states that

$$\frac{\bar{d}_i}{\bar{d}_j} = \frac{\delta_i}{\delta_j} \quad (1)$$

for all pairs of classes i and j. The parameters $\{\delta_i\}$ are referred to as *Delay Differentiation Parameters (DDPs)*, and because higher classes are better, they are ordered as $\delta_1 > \delta_2 > \ldots > \delta_N > 0$. Since the proportional differentiation model does not depend on the class loads, it applies with the same semantics in all load conditions in which it is feasible. For example, the network operator can specify that the average delay in class-1 is double the average delay in class-2, independently of whether the delays are in the order of a few packet transmission times, or hundreds of packet transmission times. The conditions under which this model is feasible are shown in the next section.

It is desirable that the proportional differentiation model holds not only in the case of average delays over long timescales, but also for delay metrics over short timescales. Long-term averages do not always convey useful information, especially when the traffic is very bursty, or when the user/application flows are short. To illustrate this, suppose that a user generates a short packet flow (say a Web session) in a certain class j, with the expectation that the flow will encounter lower delays than if it had been sent in class i ($i < j$). Even if the long-term average delay of class j is indeed lower than the average delay of class i, it may happen that the flow was created during a time period in which class i encounters lower delays than class j because of a large burst in the latter; if this happens frequently, the relative differentiation between classes will not be consistent and predictable. A formulation of the proportional differentiation model for a short-term queueing delay metric follows: let $\bar{d}_i(t, t+\tau)$ be the average queueing delay of the class-i packets departing in the time interval $(t, t+\tau)$, where $\tau > 0$ is the *monitoring timescale*. If there are no departing packets, $\bar{d}_i(t, t+\tau)$ is undefined in this time interval. The proportional delay differentiation model for a *monitoring timescale* τ holds between a pair of classes i and j, if

$$\frac{\bar{d}_i(t, t+\tau)}{\bar{d}_j(t, t+\tau)} = \frac{\delta_i}{\delta_j} \quad (2)$$

for all time intervals $(t, t+\tau)$ in which both $\bar{d}_i(t, t+\tau)$ and $\bar{d}_j(t, t+\tau)$ are defined. Unfortunately, we were unable to derive the feasibility conditions for this short timescale formulation of the model, given a certain value of τ. Consequently, our goal in this paper is to identify and evaluate scheduling mechanisms that can *approximate* the proportional differentiation model in short timescales, even though we do not attempt to define more formally how short the monitoring timescale τ should be, and which are the implications for the feasibility of the model in that case. The feasibility conditions for the case of long-term average delays, on the other hand, are given in Section 4. We return to the 'short timescales' aspect of the proportional differentiation model in Section 5, where we evaluate using simulations the BPR and WTP schedulers in the context of the proportional differentiation model of Equation 2.

2.1 Other relative differentiation models

Strict Prioritization: In this approach, the highest backlogged class is serviced first (delay aspect), and when a packet needs to be dropped, it is from the lowest backlogged class (loss aspect). However, such a discipline does not lead to controllable differentiation, because it does not provide any means for adjusting the quality spacing between classes. Also, the lower classes can experience starvation effects if no restriction is placed on the load of higher classes.

Price Differentiation: A simple case of relative service differentiation is the Paris Metro Pricing (PMP) scheme [10]. PMP uses pricing, instead of special forwarding mechanisms, to provide relative class differentiation. It is based on the assumption that higher prices will lead to lower loads in the higher classes, and thus, better service quality. Pricing mechanisms, however, cannot be effective over relatively short timescales, especially when the class tariffs cannot be changed very often. When higher classes get overloaded (because, for example, many 'rich' users become active at the same time), they will offer worse packet forwarding than lower classes. This would be a case of inconsistent (i.e., unpredictable) class differentiation.

Capacity Differentiation: In this model, the network operator allocates the forwarding resources between classes so that higher classes have more bandwidth and packet buffers, relative to their long-term expected load, than lower classes, and so they get better service. In the case of delay differentiation, for example, a Weighted Fair Queueing (WFQ) type of scheduler can be used to distribute the link bandwidth between classes [11, 12, 13], so that the ratio of service-to-arrival rates for higher classes is larger. Although the WFQ variants provide the network operator with a set of class differentiation parameters (i.e. the proportional bandwidth share of each class), the actual queueing delays at a bandwidth allocation server depend on the bandwidth share, and the load and traffic burstiness of each class [14]. Consequently, although the bandwidth differentiation is controllable, the delay differentiation is not. One can argue that the link shares can be set based on the *expected* load in each class, adjusting the long-term queueing delays to a desired operating point. Such an approach, however, would not be capable of providing consistent differentiation in relatively short timescales, because the forwarding resources allocated to each class do not follow the actual class load variations. This issue has been illustrated in [15]. Our position is that, instead of relying on such provisioning methods, it is the *forwarding mechanisms* (i.e., packet scheduling and buffer management) that should be capable of providing consistent relative differentiation between classes, independent of the class loads and even in short timescales. So, when a high class becomes temporarily overloaded, the forwarding mechanisms should dynamically assign to it a larger share of forwarding resources so that it still remains better than the lower classes.

Additive Differentiation: The basic idea in this model is to 'space' the class service levels based on the differentiation parameters that the network operator chooses, as in the proportional model, but using additive, instead of proportional, constraints. For the case of long-term average queueing delays, the additive model states that when the load is sufficiently heavy, the difference between the class average delays is a specified constant, i.e,

$$\bar{d}_i - \bar{d}_j = D_{i,j} > 0 \quad (j > i) \tag{3}$$

where $D_{i,j} > 0$ is the delay differentiation parameter for the pair of classes (i, j). Consider a priority scheduler in which the priority of a packet in queue i at time t is $p_i(t) = w_i(t) + s_i$, where $w_i(t)$ is the waiting-time of the packet at time t, and $0 < s_1 < s_2 < \ldots < s_N$ are the *scheduler differentiation parameters*. Simulation results show that this scheduler tends to additive delay differentiation in heavy-load conditions, with $D_{i,j} = s_j - s_i$ [15]. This scheduler is also discussed in [16], together with an expression for its behavior in heavy-load conditions assuming Poisson arrivals. We mention the additive differentiation model here, as an interesting case of another relative differentiation model that deserves further investigation.

3 The Dynamics and Feasibility of Proportional Delay Differentiation

The objective of this section is to provide further intuition on the dynamics of the proportional delay differentiation model, and to show the conditions under which this model is feasible, for the case of long-term average delays. We consider a *lossless* and *work-conserving* packet scheduler that services N queues, one for each class.[1] The lossless assumption requires that the scheduler operates in the stable region, i.e., the offered load is less than the service capacity; otherwise the queues can be unrealistically long. This operation scenario can be achieved in practice with sources that react to the Explicit Congestion Notification (ECN) bit, without requiring loss-induced congestion control [17]. A more realistic case would be to model a lossy multiplexer with sources that adjust their rates based on packet losses, as the current TCP sources do, but we do not take this model further, since we do not consider loss-rate differentiation in this paper. The assumption of work-conserving forwarding mechanisms is also important, because with a non-work-conserving scheduler it is possible to set the delay spacing between classes to arbitrary levels. We believe that only work-conserving forwarding mechanisms will be used in practice, because of the competition for the best possible service between providers; this is mainly a non-technical issue however.

Let λ_i be the average arrival rate in class i, and $\lambda = \sum_{i=1}^{N} \lambda_i$ be the aggregate arrival rate in the system. Assume that there exists a scheduling discipline which can enforce the proportional delay differentiation model for the case of long-term average queueing delays,

$$\frac{\bar{d}_i}{\bar{d}_j} = \frac{\delta_i}{\delta_j} \quad (i, j = 1 \ldots N) \tag{4}$$

i.e., we assume that the proportional differentiation is *feasible*. If the packet length distribution is the same in all classes, the *conservation law* mandates that:

$$\sum_{i=1}^{N} \lambda_i \, \bar{d}_i = \lambda \, \bar{d}(\lambda) \tag{5}$$

where $\bar{d}(\lambda)$ is the average queueing delay that would result if the aggregate traffic was serviced by a work-conserving FCFS server of the same capacity as the scheduler that enforces the proportional delay model. From Little's law, the conservation law states that the average backlog in a work-conserving system is independent of the scheduling discipline. For a more general form of the conservation law, which does not require the same packet length distribution in all classes, see [16]. Note that $\bar{d}(\lambda)$ depends strongly on the traffic characteristics (e.g., burstiness), and it requires detailed measurements of the actual traffic dynamics in a certain link in order to be estimated.

Combining Equations (4) and (5), the average queueing delay of class i is:

$$\bar{d}_i = \frac{\delta_i \, \bar{d}(\lambda)}{\delta_1 \frac{\lambda_1}{\lambda} + \delta_2 \frac{\lambda_2}{\lambda} + \ldots + \delta_N \frac{\lambda_N}{\lambda}} \quad (i = 1 \ldots N) \tag{6}$$

From Equation (6) we can verify the following properties for the 'dynamics' of the proportional delay differentiation model:

1. The average delay of a class i increases[2] with the arrival rate of *each* class j.
2. Increasing the load of a higher class causes a larger increase in the average delay of a class than increasing the load of a lower class.
3. If the delay differentiation parameter of a class increases, the average delay of all other classes decreases, while the average delay of that class increases.
4. Suppose that a fraction of the class i load switches to class j, while the aggregate load remains the same. The average delay of each class increases if $i < j$, and decreases if $i > j$.

We assumed earlier that the proportional delay differentiation model is *feasible*, i.e., that there exists a work-conserving scheduler that can enforce the constraints of (4). It is easy to see that this is not always true. For example, the average delay of a class cannot be lower than the average delay of that class in a FCFS server, with the traffic from the other classes removed. *Given the class arrival rates $\{\lambda_i\}$ and the average delay $\bar{d}(\lambda)$ of the aggregate traffic, we say that a set of DDPs $\{\delta_i\}$ is feasible if there is a work-conserving scheduler that can set the average delay of each class as given in Equation (6)*; in that case, the constraints of (4) are satisfied as well. Necessary and sufficient conditions for the feasibility of a set of average class delays, given the class loads, were derived in [18] (see also [19]). For general traffic assumptions,[3] a set of N average delays $\{\bar{d}_i\}$ is feasible if and only if the following $2^N - 2$ inequalities hold,

$$\sum_{i \in \phi} \lambda_i \, \bar{d}_i \geq (\sum_{i \in \phi} \lambda_i) \, \bar{d}(\sum_{i \in \phi} \lambda_i) \quad \text{for all} \quad \phi \in \Phi \tag{7}$$

where Φ is the set of $2^N - 2$ nonempty proper subsets of $\{1, 2, \ldots N\}$. The term $\bar{d}(\sum_{i \in \phi} \lambda_i)$ is the average queueing delay that the aggregate traffic of the classes in $\phi \in \Phi$ would experience in a work-conserving FCFS server. These conditions express the fact that the average backlog of a subset

[1] We use the terms 'class' and 'queue' interchangeably.

[2] In the following properties we use the terms 'increase' and 'decrease' informally, since the actual terms should be 'non-decrease' and 'non-increase', respectively.

[3] Most of [18] assumes Poisson arrivals. The particular result that we include here holds however, as [18] also states, for a general arrival model.

of the N classes cannot be lower than the backlog of these classes in a FCFS server, independently of the scheduling discipline.

It has to be noted that these feasibility conditions can be applied in practice only if it is possible to estimate the average delays $\bar{d}(\sum_{i \in \phi} \lambda_i)$ with measurements of the traffic dynamics in a specific link. Such an experimental procedure is by itself a challenging open issue, that we do not pursue further here. An important point to keep from this discussion, however, is that even if there is an 'ideal proportional delay scheduler', the DDPs that a network operator specifies might not be feasible under certain conditions on the system load, the class load distribution, and the traffic characteristics. In the rest of the paper, we only consider cases of feasible DDPs. Since it is possible to examine the conditions stated in (7) in simulation experiments (by simply simulating the FCFS server), we have verified that the experiments in Section 5 (and specifically in Figures 1 and 2) refer to cases of feasible delay differentiation. Consequently, the deviations from the proportional delay model shown there are due to the inefficiencies of the packet schedulers, rather than due to the proportional model or the selected DDPs.

4 Two Packet Schedulers for Relative Delay Differentiation

In this section, we identify two packet schedulers that are designed for predictable and controllable relative delay differentiation. They both approximate the proportional delay differentiation model in heavy-load conditions, as the simulation study of the next section shows, even in short timescales.

4.1 Backlog-Proportional Rate (BPR) scheduler

The basic idea in this scheduler is to use the bandwidth distribution model of a GPS server [12], but with the following modification: *dynamically readjust the class service rates so that they are always ratioed proportionally to the corresponding ratios of class loads.* The relation between class loads is reflected on the relation of the class backlogs, since if a certain class has received a small amount of service relative to the amount of arrivals in a recent time interval, then that class will also have a relatively larger backlog. Specifically, let $r_i(t)$ be the service rate that is assigned to queue i at time t. If the queue i is empty at time t, $r_i(t) = 0$. For two backlogged queues i and j, the service rate allocation in BPR satisfies the proportionality constraint:

$$\frac{r_i(t)}{r_j(t)} = \frac{s_i}{s_j} \frac{q_i(t)}{q_j(t)} \tag{8}$$

where $q_i(t)$ is the backlog of queue i at time t. The actual service rate of each class during a busy period can be calculated from the work-conservation constraint:

$$\sum_{i=1}^{N} r_i(t) = R \tag{9}$$

where R is the link capacity. The parameters $\{s_i\}$ are the *Scheduler Differentiation Parameters (SDPs)* that the network operator selects, and as will be shown in the next section, they are directly related to the Delay Differentiation Parameters (DDPs) $\{\delta_i\}$ in heavy-load conditions. Following our convention on the ordering of classes, $s_1 < s_2 < \ldots < s_N$.

The main finding of the simulation study in the next section is that the the BPR scheduler approximates the proportional delay differentiation model of Equation (1) in heavy-load conditions, with *the DDP ratios tending to the inverse of the corresponding SDP ratios*, i.e.,

$$\frac{\bar{d}_i}{\bar{d}_j} \rightarrow \frac{\delta_i}{\delta_j} = \frac{s_j}{s_i} \tag{10}$$

Further work is required however in order to analytically examine the validity of this property for general traffic models.

A property of the BPR scheduler is that when the relative backlog of a queue is quite small, the relative service rate given to that queue is also small. As a result, the last few packets in a queue either before the queue gets empty or before new arrivals occur can experience a much larger delay than their predecessors. This causes *sawtooth-type* of variations in the queueing delays of consecutive packets (see Figure 4). This pathological short-term behavior is related to the *simultaneous queue clearing* property of the BPR scheduler:

Proposition 1 *In a BPR scheduler, all queues that are backlogged during a busy period become empty at the same time.*

This property is proved in Appendix-1. Since the BPR scheduler is based on the fluid server model, it can only be approximated in practice. A possible way to 'packetize' the BPR scheduler is given in Appendix-3. That is also the algorithm implemented in our simulation study.

4.2 Waiting-Time Priority (WTP) scheduler

This is a priority scheduler in which the priority of a packet increases proportionally with its waiting-time. Specifically, the priority of a packet in queue i at time t is

$$p_i(t) = w_i(t)\, s_i \tag{11}$$

where $w_i(t)$ is the waiting-time of the packet at time t. The Scheduler Differentiation Parameters $\{s_i\}$ determine the rate with which the priority of the packets of a certain class increases with time. As in BPR, $s_1 < s_2 < \ldots < s_N$. The WTP algorithm was first studied by L. Kleinrock in 1964, with the name Time-Dependent-Priorities [9, 20]. Using Kleinrock's words: *The utility of this new priority structure is that it provides a number of degrees of freedom with which to manipulate the relative waiting times for each priority group.* The WTP scheduler distributes the service rate between backlogged classes dynamically based on the load of each class, but it does so in a different way than the BPR scheduler. Specifically, in the WTP scheduler the load of a queue in the recent past is reflected on the waiting-time of the packet at the head of that queue, since if a queue has received a small amount of service relative to the corresponding amount of arrivals in a recent past interval, it will have packets with large waiting times close to its head. As in BPR, the Scheduler Differentiation Parameters function as weights in the service rate distribution, and they are directly related to the Delay Differentiation Parameters.

The main finding of the simulation study in the next section is that the WTP scheduler approximates the proportional delay differentiation model of Equation (1) in heavy-load conditions, with *the DDP ratios tending to the inverse of the corresponding SDP ratios*, as in Equation (10). The simulation study also shows that the WTP scheduler achieves this goal more accurately than the BPR scheduler.

Additionally, the WTP scheduler approximates the short-timescale proportional differentiation model of Equation (2) in time intervals of high load, even when then the monitoring timescale τ is a few tens of packet transmission times. As in the case of BPR, though, further work is required in order to analytically examine the validity of these properties for general traffic models.

However, the WTP scheduler can also exhibit a problematic behavior in short timescales. This behavior is probably due to its priority nature (as opposed to the 'simultaneous link-sharing' nature of BPR). Specifically, under certain conditions on the SDPs, an arriving burst in a high-class queue can exclude lower classes from service until this burst is completely serviced; this short-term starvation effect can happen for arbitrarily long high-class bursts.

Proposition 2 *Let R_I be the peak input rate in the scheduler, and R be the output link (or service) rate. If $R < R_I$ and*

$$1 - \frac{R}{R_I} > \frac{s_i}{s_j}, \qquad (s_i < s_j) \tag{12}$$

a sequence of Λ consecutive class j packets that starts arriving at time t_0 will be serviced before any class i packets that arrived at t_0 or later, and this is true for arbitrary large values of Λ.

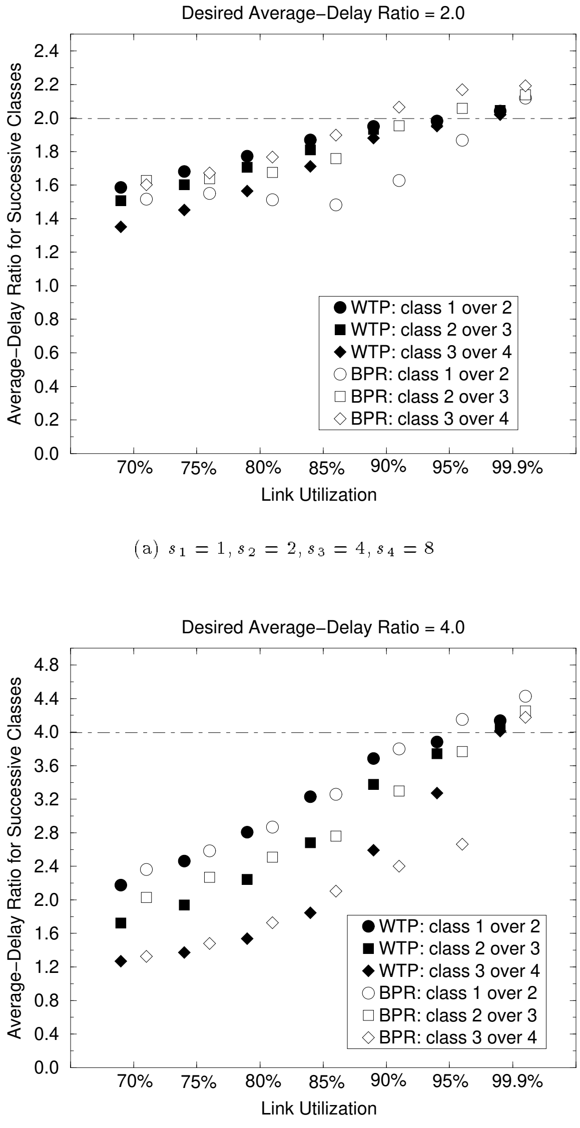
This property is proved in Appendix-2. The complexity of the WTP scheduler is $O(N)$, where N is the number of classes, since a priority has to be calculated for every backlogged class after a packet departure. For a small number of classes, the implementation of this scheduler should be feasible even at very high-speed links. Also, packets have to be timestamped upon arrival, which may be a significant overhead in certain implementations.

5 Simulation Study A: Evaluation of BPR and WTP

The objective of this simulation study is to evaluate the BPR and WTP packet schedulers in the context of the proportional delay differentiation model. We first investigate the effect of the aggregate load and of the class load distribution on the long-term average delay differentiation. Then, we examine if BPR and WTP can satisfactorily approximate the proportional delay model in short timescales. Finally, we show instances of the 'microscopic' views of the queueing delays in different classes, and highlight some problems in the short-term behavior of the two schedulers.

The simulated model is as follows. A BPR/WTP scheduler services N=4 packet sources, with one source for each service class. Unless stated otherwise, the Service Differentiation Parameters for both schedulers are $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 8$. The interarrivals between packets of the same class follow a Pareto distribution with a shape parameter $\alpha = 1.9$; because of the infinite-variance property of this distribution we do not determine confidence intervals for the average-delay measurements. The class load distribution is specified in each graph, and in most cases it is set to: Class-1:40%, Class-2:30%, Class-3:20%, Class-4:10%. The packet length distribution is the same for all classes (40% of the packets are 40 bytes, 50% are 550 bytes, and 10% are 1500 bytes). Normalizing to an arbitrary link speed, the average packet transmission time (referred to as *p-unit*) is 11.2 time units. The following graphs show only queueing delays, which are also measured in these arbitrary time units. The utilization factor ρ is set to the ratio of the average packet transmission time and the average interarrival of the aggregate packet stream. In most of the simulations we examine the behavior of the schedulers between moderate-load ($\rho \approx 0.70$) and heavy-load ($\rho \approx 0.95$) conditions. As mentioned earlier, such stable and high-utilization operation can be achieved in practice without packet losses only if there is an adequately large number of packet buffers and the sources adjust their rate successfully using the ECN bit set by congested routers. If the utilization is less than around 70%, the queueing delays are fairly small, and so, no service differentiation is probably needed.

The effect of the aggregate load. Figure 1 shows the ratio of the average-delays between successive classes in



(a) $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 8$

(b) $s_1 = 1, s_2 = 4, s_3 = 16, s_4 = 64$

Figure 1: The ratios of average-delays between successive classes with WTP and BPR. The traffic load distribution is Class-1: 40%, Class-2: 30%, Class-3: 20%, Class-4: 10%.

moderate and heavy-load conditions. The average delay for each class in these experiments is computed from the entire simulation run (after an initial 'warm-up period'), i.e., they are long-term average delays. Each point in these figures resulted from averaging over ten simulation runs with different seeds, while the simulation time in each run was 10^8 time units. The SDPs were chosen as $s_i/s_{i-1} = 2$ in Figure 1-a, and as $s_i/s_{i-1} = 4$ in Figure 1-b. Notice that as the aggregate load increases, the WTP scheduler tends to proportional average-delay differentiation with:

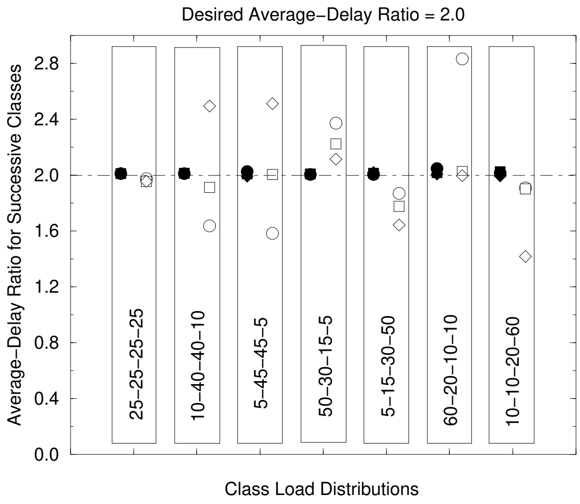
$$\frac{\delta_i}{\delta_j} \to \frac{s_j}{s_i} \tag{13}$$

i.e., the DDP ratios are just the inverse of the corresponding SDP ratios. The BPR scheduler has a similar trend, but it does not converge exactly to the specified ratio, probably because of the approximations done in the 'packetization' of the scheduler (see Appendix-3).

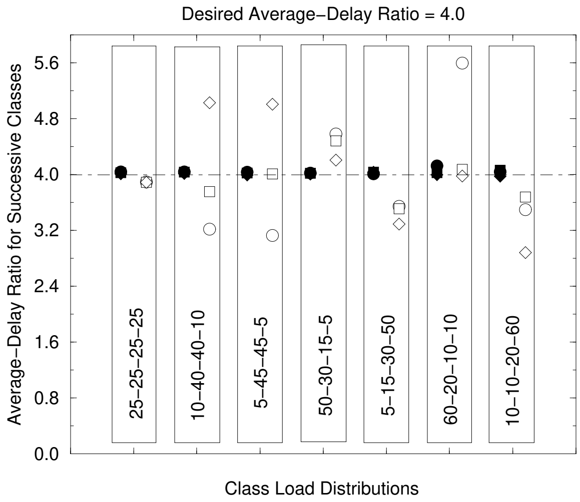
Although these figures show only the delay ratios, the actual average delays in the heavy-load region in which both schedulers approximate the proportional differentiation model are in the order of a few tens of packet transmission times for the high classes, and of a few hundreds of packet transmission times for the low classes. This can be also seen in Figures 4 and 5. Such delays are common in practice, especially for links with a high bandwidth-delay product, and thus, we can argue that the high-load operating region in which the two schedulers perform close to the proportional differentiation model is a realistic case, and not an impractical regime in which even the highest classes encounter excessive queueing delays.

However, neither scheduler manages to maintain the proportional delay differentiation in moderate loads. For example, when the utilization is 70% the differentiation ratio is about 1.5 when it should be 2, and about 1.7 when it should be 4. The deviations increase as we widen the differentiation spacing between classes, by having higher SDP ratios s_i/s_{i-1}. We repeat that these experiments refer to feasible proportional delay differentiation, and so the inaccuracies shown are because of the schedulers, and not because of the chosen SDPs (or DDPs). These inaccuracies are not surprising; these schedulers were not a priori designed for proportional delay differentiation, but for controllable relative delay differentiation. They tend to proportional delay differentiation only under sufficiently heavy-load conditions. An interesting open question is whether there is a work-conserving scheduler that can achieve the proportional delay differentiation constraints, whenever this is feasible.

The effect of the class load distribution. Figure 2 shows the ratio of the average-delays between successive classes in seven different load distribution cases. The simulation methodology is as in the previous paragraph. The SDPs are chosen as $s_i/s_{i-1} = 2$ in Figure 1-a, and as $s_i/s_{i-1} = 4$ in Figure 1-b. The link utilization is 95% in all cases. Notice that the WTP scheduler provides the specified average-delay differentiation ratio (Equation 13) independent of the load distribution in a very precise manner. The BPR scheduler, on the other hand, is in a certain degree dependent on the class load distribution. Specifically, although it can achieve the proportional average-delay model when all classes have the same load, it cannot do so in an accurate manner when some classes are more loaded than others; the highly loaded classes get higher delays than what the SDPs specify. The deviations from the proportional differentiation model diminish as the aggregate load tends to 100%. These results



(a) $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 8$

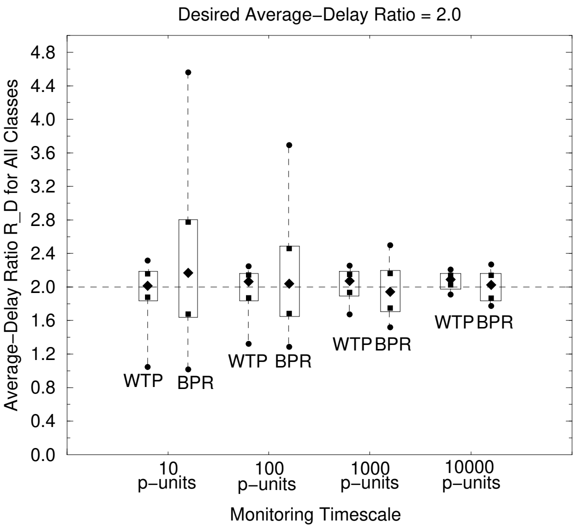


(b) $s_1 = 1, s_2 = 4, s_3 = 16, s_4 = 64$

Figure 2: The symbols in this graph are as in Figure 1. The four numbers in each bar denote the fraction of the four classes in the aggregate packet stream, starting from class 1 up to class 4. The utilization is 95% in all cases.

strengthen the observation that WTP is better than BPR in the context of proportional delay differentiation.

The behavior of BPR and WTP in short timescales. The previous two experiments are based on measurements of long-term averaging delays. A critical issue, however, is to investigate whether the WTP and BPR schedulers can also approximate the short-timescale proportional differentiation model of Equation (2). In this experiment we measure the ratios of average-delays between successive classes in consecutive time-intervals of length τ, where τ is the *monitoring timescale*. The four values of τ are 10, 100, 1000, and 10000 p-units, where a p-unit is the average packet transmission time (11.2 time units in this



(a) $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 8$

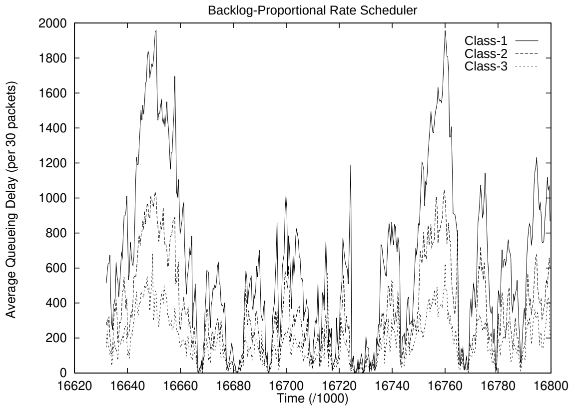
Figure 3: Five percentiles of the R_D measure (see text) for four values of the monitoring timescale τ. The diamonds represent the 50% percentiles (median), the squares at the horizontal edges of the boxes represent the 25% and 75% percentiles, while the circles at the end points of the dashed lines represent the 5% and 95% percentiles.

study). For example, a monitoring timescale τ of 1000 p-units corresponds to about 3 seconds in a T1 link, and to about 30 milliseconds in an OC-3 link. The average delay of a class in a certain time interval of length τ is measured as the average delay of the packets that departed in that time interval from the queue of that class. At the end of the simulation run we compute, for each time interval, the ratios of average-delays between successive classes; then, we average these ratios over all pairs of classes in order to get a single measure R_D for the ratio of average-delays between successive classes in the corresponding time interval. When one or more classes are not 'active' in a certain time interval (i.e., there are no packet departures from that class), we normalize the ratios of average delays of the 'active' classes in order to compute R_D.

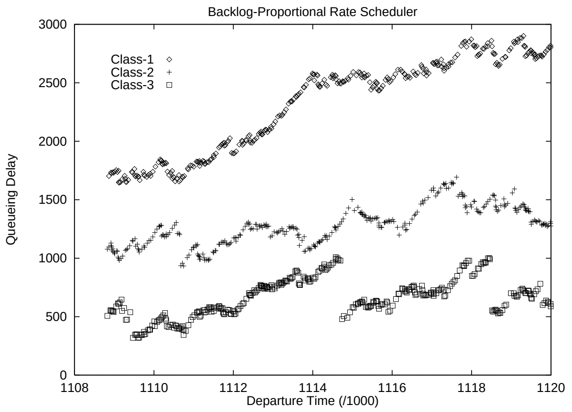
Figure 3 shows five percentiles of the R_D values obtained from all time-intervals of length τ, for the four different values of τ; the five percentiles are: 50% (median), 5%, 25%, 75%, and 95%. The SDPs are $s_i/s_{i-1} = 2$, and the aggregate load is 95%. As we increase the monitoring timescale τ to 10000 p-units, both schedulers approximate the short-term proportional differentiation model of Equation (2) in almost all time-intervals of length τ. Also, if we focus in the range between the 25 and 75 percentiles, the WTP approximates the proportional constraints even with a monitored timescale of only tens of p-units. The BPR, on the other hand, has a quite 'spread' range of average-delay ratios in timescales of hundreds of p-units or less. The improved behavior of WTP over BPR in short timescales is also illustrated in the next paragraph.

Microscopic views of the behavior of BPR and WTP. To further illustrate the short-timescale behavior of the two schedulers, we next show two pairs of *microscopic views* of the queueing delays in each class. In the microscopic views I (Figures 4-a and 5-a) each point represents the average queueing delay of a class in consecutive time intervals of 30 p-units, for a time-window of about 15,000 packet transmission times. In the microscopic views II (Figures 4-b and 5-b) each point shows the queueing delay of an individual packet at the time of its departure, for a time-window of about 1,000 packet transmission times. Figure 4 shows the microscopic views in the case of BPR, while Figure 5 shows the case of WTP. Each pair of microscopic views covers the same simulation time interval, and the same arriving packet streams in each class. For simplicity of illustration, the number of classes in these experiments is three instead of four. The SDPs in both schedulers are $s_1 = 1, s_2 = 2$, and $s_3 = 4$, while the aggregate link utilization is 95%.

The microscopic views I are typical for these load conditions, while the microscopic views II cover an overloaded time interval. Notice that, as mentioned earlier, even in such overloaded periods the actual delays for the lowest class are a few thousands of time-units, which corresponds to a few hundreds of packet transmission times, while the actual delays for the highest class are a few hundreds of time-units, which

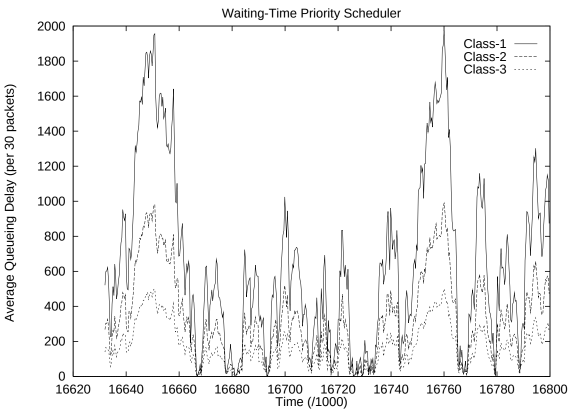


(a) Microscopic view I

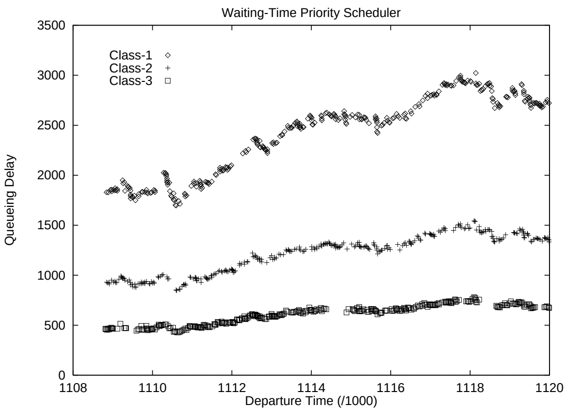


(b) Microscopic view II

Figure 4: Queueing delays with the BPR scheduler when $s_1 = 1, s_2 = 2, s_3 = 4$.



(a) Microscopic view I



(b) Microscopic view II

Figure 5: Queueing delays with the WTP scheduler when $s_1 = 1, s_2 = 2, s_3 = 4$.

corresponds to a few tens of packet transmission times. The general observation for both schedulers from the microscopic views I is that the proportional delay differentiation model is better approximated during intervals of high load, where the class queues and delays are large. In the case of BPR, it is easy to note, especially in the microscopic view II, that it deviates quite often from the proportional delay model in very short timescales. Specifically, the *sawtooth-type* of variations in the queueing delays of the microscopic view II are common: the queueing delays of consecutive packets gradually increase, until they suddenly drop at a certain time, after the arrival of new packets in that class. WTP, on the other hand, approximates more precisely the proportional delay differentiation model, even in the microscopic view II. Although there are several noticeable deviations, but the general trend is quite satisfactory. Comparing Figures 4 and 5 shows that even though both schedulers approximate the proportional average-delay model under heavy-load conditions, WTP achieves this in a more precise manner in short timescales, while BPR creates 'noisy' queueing delay variations.

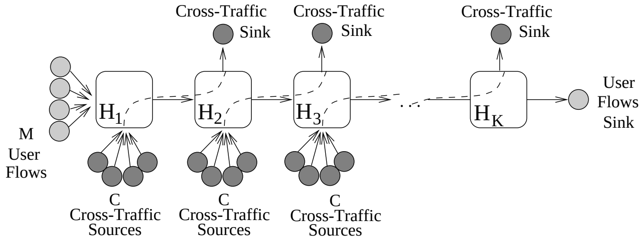


Figure 6: The multi-hop traffic configuration in this simulation study.

6 Simulation Study B: The User's Perspective

The simulation study of the previous section focused on a single link and on the relative delay differentiation between different classes. Although such a study would be of interest to a network operator, the network users would obviously be concerned for the *end-to-end performance of their packet flows*. To illustrate the issues that arise in the user's perspective, consider the following scenario. Suppose that two identical flows from classes i and j (with $i > j$) enter a network at the same time and they traverse a common path. The network attempts to provide locally in each link proportional delay differentiation in the granularity of class traffic, but as it was shown in the previous section, this is often impossible to do in all timescales and load conditions. The fair question that arises in the user's perspective is: 'since class i is higher (and probably more expensive) than class j, will I get lower delays in this path if my packet flow belongs to class i instead of class j?' More generally, the issue here is: *can a local and class-based relative differentiation lead to consistent end-to-end and flow-based relative differentiation, independent of the network path and user-flow characteristics?* If not, there is obviously a mismatch between what the network offers and what the users expect.

We attempt a first investigation of this critical issue using simulations that we performed with ns-v2 [21]. The simulated model is based on the previous scenario: A set of $N(=4)$ identical flows, one from each class, traverse a congested network path that consists of K hops. Each flow has a length of F packets (500 bytes per packet) which are periodically transmitted at 1.5Mbps to generate an average rate of R_u kbps. Note that the periodic nature of these flows is just a technicality to ensure that the corresponding packets enter the network at the same time; alternatively, we could use bursty precomputed arrivals, common for all flows. In the rest of this section, these N flows are referred to as *User* flows. The network path is also loaded with *Cross-traffic*. The Cross-traffic at each node is generated from $C(=8)$ sources that randomly generate packets (500 bytes) from different classes, following the distribution: Class-1:40%, Class-2:30%, Class-3:20%, Class-4:10%. These sources have Pareto-distributed interarrivals ($\alpha = 1.9$). Their average rate R_c is adjusted based on the desired average utilization ρ in each network link. The traffic configuration that we simulated is shown in Figure 6. The bandwidth of the network links is 25Mbps. Each link uses a WTP scheduler (since it performs better than BPR), and the SDPs are: $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 8$. In order to examine the effectiveness of the relative delay differentiation, we ignore propagation and transmission delays (which are common to all packets) and focus only on queueing delays. The parameters that we vary in this model, together with the two values

	F=10 R_u=50	F=10 R_u=200	F=100 R_u=50	F=100 R_u=200
K=4 ρ=85%	2.3	2.2	2.2	2.1
K=4 ρ=95%	2.1	2.1	2.1	2.0
K=8 ρ=85%	2.0	2.0	2.0	2.0
K=8 ρ=95%	2.0	2.0	2.0	2.0

Table 1: The metric $\bar{R}_D$: Ideally, it should be 2.00 in all cases. These results have been consistent in five simulation runs with different random seeds.

that have been simulated for each parameter, are:
a) the length F of the User flows (10 and 100 packets),
b) the rate R_u of the User flows (50 and 200 kbps),
c) the utilization ρ of the network links (85% and 95%),
d) the number of (congested) hops K in the network path (4 and 8 hops).
The first two refer to the characteristics of the User flows, while the last two refer to the network path.

The simulation methodology is as follows. After 'warming up' the network for 100 seconds, we generate in every second M identical User flows, one from each class. Each of these periodic events is referred to as a 'user experiment', since it is what a user would do in order to evaluate the end-to-end delay differentiation between classes in a certain network path. The simulation runs long enough to generate M=100 user experiments. At the end of the simulation run, we process the packet delays of each User flow, and calculate the ten end-to-end delay percentiles: 10%, 20%, ... 90%, and 99%. In the last step of the process, we compare the corresponding delay percentiles of the M flows in each user experiment. If in a certain user experiment a flow from a higher class experienced larger delays than a lower class, in terms of *any* of these percentiles, we identify a case of *inconsistent delay differentiation*. Finally, we compute the end-to-end delay ratio $\bar{R}_D$ between successive classes, averaged over the $N-1$ pairs of successive classes, over the M user experiments, and over the ten delay percentiles. Although $\bar{R}_D$ is a very 'spread' average, it provides a simple figure-of-merit for the end-to-end queueing delay relative differentiation.

The first, and perhaps most important, result of this simulation study is that there were no cases of inconsistent delay differentiation observed in any simulation run. In other words, *the local and class-based differentiation translates to consistent end-to-end and flow-based differentiation.* The results for the metric $\bar{R}_D$ are also quite satisfactory, as shown in Table 1. In most cases $\bar{R}_D$ is quite close to 2.0, which is the value that would result in the case of ideal proportional delay differentiation with $\delta_i/\delta_j = s_j/s_i$. As expected, $\bar{R}_D$ tends to 2.0 as the load increases, because WTP converges to the proportional delay differentiation model. It is interesting that as the number of path hops increases, the per-hop deviations from the proportional differentiation model tend to cancel-out, leading to a better $\bar{R}_D$. It has to be noted, finally, that one of the reasons $\bar{R}_D$ is so close to the ideal differentiation ratio, is because it is an average over three dimensions: all successive class pairs, all user experiments, and all delay percentiles. The deviations from the proportional differentiation model can be larger when we focus on individual user experiments, or class pairs.

7 Conclusions and Open Issues

The relative differentiated services architecture is a promising approach for addressing the issue of scalable service differentiation in the Internet. Attempting a preliminary investigation of the problems that arise in this context, this paper made two contributions. First, we proposed the proportional differentiation model as a target for predictable and controllable relative differentiation between classes. Second, we identified and evaluated two packet schedulers that approximate the proportional delay differentiation model in heavy-load conditions, even in short timescales. Although both schedulers are appropriate for relative delay differentiation, our studies illustrate that WTP is significantly better than BPR in the context of proportional delay differentiation.

We emphasize that our work is far from conclusive and that there are several open issues that need to be further investigated. First, is the proportional delay differentiation model, as formulated in this paper, the most appropriate means for predictable and controllable differentiation? Another choice could be the additive differentiation model, briefly described here, which promises an absolute difference between the class delays, when the load is sufficiently high. A different approach would be to provide some type of controllable differentiation between two classes, only in the intervals where both these classes are backlogged. However, it is is not clear if such differentiation would be of any value for the users, since they cannot know or control when classes are simultaneously backlogged. Second, the proportional differentiation model has to be extended in the direction of coupled delay and loss differentiation, since both performance measures are significant in most applications and transport protocols. BPR and WTP may not be good schedulers in that context, since they perform well when the queues are sufficiently long (i.e., in heavy-load conditions). In the presence of limited buffers and losses, however, the queues may not be long, even when the offered load is high. This issue is also related with the poor behavior of WTP and BPR in moderate loads or large SDP ratios; in either case, the class queues are not sufficiently long for these schedulers to distribute the class delays as in the proportional differentiation model. It is interesting to know the form of an 'optimal proportional differentiation scheduler', even if it is too complicated to be practical. Third, it is important to combine the feasibility conditions presented here with experimental procedures and measurements, in order to be able to determine efficiently the space of feasible DDPs for a certain network link. Even though it is hard to come up with a 'universal' model for the Internet traffic, it may be possible to estimate the required delay-vs-rate curve $\bar{d}(\lambda)$ for a specific link, perhaps as a function of time too. Finally, a major question from a network operator's point of view is how to choose the class differentiation parameters. This issue, which is also related with the the space of feasible DDPs, can be answered if the network operator knows a typical 'profile' of the quality requirements and tariff constraints of the user population in that network link. The exact algorithms for solving this network-design and pricing problem, however, require further research.

Acknowledgments

We are grateful to the anonymous Sigmetrics and Sigcomm reviewers for their constructive criticism, and to Roch Guerin for his invaluable suggestions while shepherding the final revision of this paper. The motivation source for this work was the interesting discussions at the IETF DiffServ mailing list. We thank all the participants in the DiffServ working group for both their agreements and objections to the ideas presented in this paper. We also thank T.V.Lakshman for bringing to our attention Kleinrock's Time-Dependent-Priorities scheduler, and for his comments in the early phase of this work. The first author (C.D) is also grateful to S.Tatikonda, M.Kasbekar, K.Kar and D.Katabi for the inspiring discussions during the Summer-98 Bell Labs internship program.

References

[1] A.Mankin, F.Baker, B.Braden, S.Bradner, M.O'Dell, A.Romanow, A.Weinrib, and L.Zhang, *RSVP Version 1: Applicability Statement, Some Guidelines on Deployment*, Sept. 1997. IETF RFC 2208.

[2] R.Guerin, S.Kamat, V.Peris, and R.Rajan, "Scalable QoS Provision Through Buffer Management," in *Proceedings ACM SIGCOMM*, Sept. 1998.

[3] I.Stoika, S.Shenker, and H.Zhang, "Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks," in *Proceedings ACM SIGCOMM*, Sept. 1998.

[4] K.Nichols, S.Blake, F. Baker, and D.L.Black, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, Dec. 1998. IETF RFC 2474.

[5] V.Jacobson, K. Nichols, and K.Poduri, *An Expedited Forwarding PHB*, 1999. draft-ietf-diffserv-phb-ef-02.txt (work in progress).

[6] D. D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 362–373, Aug. 1998.

[7] I.Stoika and H.Zhang, "LIRA: An Approach for Service Differentiation in the Internet," in *Proceedings NOSSDAV*, 1998.

[8] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New Tecnhiques for Congestion Detection and Avoidance," in *Proceedings ACM SIGCOMM*, Aug. 1994.

[9] L.Kleinrock, "A Delay Dependent Queue Discipline," *Journal of the ACM*, vol. 14, no. 2, pp. 242–261, 1967.

[10] A.M.Odlyzko, "Paris Metro Pricing: The Minimalist Differentiated Services Solution," in *Proceedings IEEE/IFIP International Workshop on Quality of Service*, June 1999.

[11] A.Demers, S.Keshav, and S.Shenker, "Analysis and Simulation of a Fair Queueing Algorithm," in *Internetworking: Research and Experience*, pp. 3–26, 1990.

[12] 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*, vol. 1, pp. 344–357, June 1993.

[13] J.C.R.Bennett and H.Zhang, "Hierarchical Packet Fair Queueing Algorithms," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 675–689, Oct. 1997.

[14] D.Stiliadis and A.Varma, "Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 611–625, Oct. 1998.

[15] C.Dovrolis and D.Stiliadis, "Relative Differentiated Services in the Internet: Issues and Mechanisms," in *ACM SIGMETRICS*, May 1999. Short paper.

[16] G.Bolch, S.Greiner, H.Meer, and K.S.Trivedi, *Queueing Networks and Markov Chains*. John Wiley and Sons, 1999.

[17] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communication Review*, vol. 24, Oct. 1994.

[18] E.G.Coffman and I.Mitrani, "A Characterization of Waiting Time Performance Realizable by Single-Server Queues," *Operations Research*, vol. 28, pp. 810–821, May 1980.

[19] I.Mitrani and J.H.Hine, "Complete Parameterized Families of Job Scheduling Strategies," *Acta Informatica*, vol. 8, pp. 61–73, 1977.

[20] L.Kleinrock, *Queueing Systems, Volume II*. John Wiley and Sons, 1976.

[21] Network Simulator (ns), version 2. http://www-mash.cs.berkeley.edu/ns/.

Appendix 1: Proof of Proposition 1

Consider a busy period in which a queue i is backlogged. Let t_1 be the time instant that queue i becomes empty. If t_1 is also the end of the busy period, then either all backlogged queues become empty at t_1, which is what we want to show, or queue i was the only backlogged queue before t_1, which is a trivial case that we can ignore. So, assume that the busy period does not end at t_1 and that there is another queue, say j, that was backlogged before t_1 and remains backlogged at t_1. We show next that this assumption leads to a contradiction.

Let t_0 be the time of the last arrival in any queue before t_1. We focus on the time interval $(t_0, t_1]$. Since there are no arrivals, the backlog of class i is decreased with the service rate $r_i(t)$ that is given to this queue:

$$\frac{dq_i(t)}{dt} = \lim_{\epsilon \downarrow 0} \frac{q_i(t) - q_i(t-\epsilon)}{\epsilon} = -\lim_{\epsilon \downarrow 0} r_i(t-\epsilon) \qquad (14)$$

Note that the backlog function $q_i(t)$ is differentiable in the interval $(t_0, t_1]$, because there are no arrivals and the server follows the fluid model. The service rate $r_i(t)$, however, is not necessarily continuous from the left at t_1. In fact, for $t = t_1$ the service rate converges from the left to

$$\lim_{\epsilon \downarrow 0} r_i(t_1 - \epsilon) = \lim_{\epsilon \downarrow 0} \frac{q_i(t_1-\epsilon)}{\epsilon} > 0 \qquad (15)$$

(because $q_i(t_1) = 0$, $q_i(t_1 - \epsilon) > 0$, and $q_i(t)$ is differentiable at $t = t_1$), while $r_i(t_1) = 0$.

From the definition of the BPR scheduler, however, we have that for any $t \in (t_0, t_1]$

$$r_i(t) = \frac{s_i}{s_j}\, r_j(t)\, \frac{q_i(t)}{q_j(t)} \qquad (16)$$

because queue j is backlogged during this interval. $q_i(t)$ tends continuously to zero

$$\lim_{\epsilon \downarrow 0} q_i(t_1 - \epsilon) = 0 \tag{17}$$

while $q_j(t_1)$ and $r_j(t_1)$ were assumed to be non-zero. So,

$$\lim_{\epsilon \downarrow 0} r_i(t_1 - \epsilon) = 0 \tag{18}$$

which contradicts (15). Consequently, queue i cannot become empty while queue j is backlogged, and since this holds for any queues i and j, all the backlogged queues become empty at the same time. Note that when all queues become empty at t_1 we cannot use Equation (16) to calculate $\lim_{\epsilon \downarrow 0} q_i(t_1 - \epsilon)$ because both $q_i(t)$ and $q_j(t)$ tend to zero.

Appendix 2: Proof of Proposition 2

Without loss of generality, let $t_0 = 0$. Starting at t_0, a sequence $\{\pi_j^k,\ k = 1, 2, \ldots \Lambda\}$ of class j packets starts arriving at the system in the peak input rate R_I, and so the arrival time of the k'th packet of the sequence is $(k-1)/R_I$. Additionally, a class i packet π_i^1 arrives at t_0. We next show using induction over the range $\{k = 1, 2, \ldots \Lambda\}$ that if the condition (12) holds, π_j^k will be transmitted before π_i^1 for any $k = 1, 2, \ldots \Lambda$. For simplicity, we assume that all packets have the same unit length.

We first show that independent of the backlog in each queue at $t = 0$, the packet π_j^1 will be transmitted before the packet π_i^1. Both these packets arrive at t_0, and so they encounter the same waiting time in the interval that they are both queued. Since $s_i < s_j$, the priority of π_j^1 will be always higher than the priority of π_i^1, and because WTP serves higher priority packets first, π_j^1 will be transmitted before π_i^1.

Assume, now, that packet π_j^k $(1 \leq k < \Lambda)$ has just been transmitted at t_k. From the inductive assumption, π_i^1 is still at the head of queue i. At least k packets have been transmitted in $[0, t_k]$, and hence, $t_k \geq k/R$. At t_k, the priority of π_i^1 is

$$p_i(t_k) = t_k\, s_i \tag{19}$$

and the priority of π_j^{k+1}, which is now at the head of queue j, is

$$p_j(t_k) = (t_k - \frac{k}{R_I})\, s_j \tag{20}$$

and so,

$$p_j(t_k) - p_i(t_k) = t_k(s_j - s_i) - \frac{k}{R_I} s_j \geq (\frac{k}{R} - \frac{k}{R_I}) s_j - \frac{k}{R} s_i$$

It is easy to see now that if the condition (12) holds, $p_j(t_k) - p_i(t_k) > 0$, and so π_j^{k+1} will be transmitted before π_i^1. This completes the inductive proof.

Appendix 3: A packetized BPR scheduler

The BPR packet scheduler is based on a *virtual service function* $v_i(t)$ for each queue i. $v_i(t)$ approximates the service that the packet at the head of queue i at time t in the BPR packet scheduler would have received by that time if it was serviced by the BPR fluid server.

Let $t^1, t^2, \ldots, t^k, \ldots$ be the departure times from the BPR packet scheduler. The service rate $r_i(t)$ that is allocated to a queue i is computed from Equations (8) and (9) after each departure. Let $B(t^k)$ be the set of backlogged queues at t^k. If $i \notin B(t^k)$, $r_i(t^k) = 0$ and $v_i(t^k) = 0$. Otherwise, if $i \in B(t^k)$, the virtual service function $v_i(t^k)$ is computed as follows:

$$v_i(t^k) = \begin{cases} 0 & \text{if } t^{k-1} \leq a_i \\ v_i(t^{k-1}) + r_i(t^{k-1})\,(t^k - t^{k-1}) & \text{otherwise} \end{cases}$$

where a_i is the arrival time of the packet at the head of queue i at t^k. After computing the virtual service function for all queues, the BPR packet scheduler chooses the packet to transmit next as follows: if L_i is the length of the packet at the head of queue i, the scheduler services the queue j with

$$j = \mathrm{argmin}_{i \in B(t^k)} [L_i - v_i(t^k)] \tag{21}$$

Ties are broken in favor of higher classes.

The complexity of the packet scheduler is $O(N)$, where N is the number of classes. For a small number of classes, the BPR scheduler should be implementable even in very high-speed links. Notice that there are two approximations in this packetized version of the BPR scheduler. First, the service rate $r_i(t^k)$ that is allocated to a backlogged queue after a packet departure is assumed to remain constant until the next departure. This is not the case in the BPR fluid server, but we make this approximation in order to simplify the packet scheduler. Additionally, the virtual service $v_i(t)$, as computed here, approximates the service that a packet would have received by the fluid server, if it had started being serviced when it reached the head of the queue *in the packet scheduler, and not in the the fluid server.* Consequently, we cannot argue that the packets leave the packet scheduler in the same order that they would leave the fluid server. It is likely that there are more accurate packetized approximations of the BPR fluid server, but we do not pursue this issue further here.