A Case for Relative Differentiated Services and the Proportional Differentiation Model

Constantinos Dovrolis   Parameswaran Ramanathan

Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706-1691

ABSTRACT

Internet applications and users have very diverse quality-of-service expectations, making the *same-service-to-all* model of the current Internet inadequate and limiting. There is a widespread consensus today that the Internet architecture has to be extended with *service differentiation* mechanisms, so that certain users and applications can get a better service than others at a higher cost.

One approach, referred to as *absolute differentiated services*, is based on sophisticated admission control and resource reservation mechanisms in order to provide guarantees or statistical assurances for absolute performance measures, such as a minimum service rate or a maximum end-to-end delay. Another approach, which is simpler in terms of implementation, deployment, and network manageability, is to offer *relative differentiated services* between a small number of classes of service. These classes are ordered based on their packet forwarding quality, in terms of per-hop metrics for the queueing delays and packet losses, giving the assurance that higher classes are better than lower classes. The applications and users, in this context, can dynamically select the class that best meets their quality and pricing constraints, without a priori guarantees for the actual performance level of each class.

The relative differentiation approach can be further refined and quantified using the *Proportional Differentiation Model*. This model aims to provide the network operator with the *'tuning knobs'* for adjusting the *quality spacing* between classes, *independent of the class loads*. When this spacing is feasible in short timescales, it can lead to *predictable and controllable* class differentiation, which are two important features for any relative differentiation model. The proportional differentiation model can be approximated in practice with simple forwarding mechanisms (packet scheduling and buffer management), that we briefly describe here.

1 Introduction

The Internet is being used by business and user communities with widely varied service expectations from the network infrastructure. For example, there are many companies which rely on the Internet for the day-to-day management of their global enterprise. These companies are willing to pay a substantially higher cost for the best possible service level from the Internet. Similarly, there are many users who are willing to pay a higher Internet access fee in order to make use of demanding applications, such as IP-telephony and video-conferencing. At the same time, there are millions of users who want to pay as little as possible for more elementary services, like exchanging e-mails and/or surfing the Web. In addition to this variety of user expectations, there has also been a rapid evolution in the set of Internet applications. A few years ago the key Internet applications were only e-mail, ftp, or newsgroups. In contrast, the present day Internet applications have widely diverse service needs because they transfer a wide range of information types, including voice, music, video, graphics, JAVA scripts, and hypertext links. As a result of these changes in the user expectations and Internet applications, there is a growing demand for replacing the current *same-service-to-all* paradigm with a model in which users, applications, or individual packets are *differentiated* based on their service needs.

Architectures for providing service differentiation in the Internet have been the focus of extensive research in the last few years. These research efforts have identified two fundamentally different approaches for service differentiation: the *Integrated Services* approach and the *Differentiated Services* approach.

The Integrated Services Approach. The *Integrated Services* (*IntServ*) approach [1] focuses on individual packet flows, i.e., streams of IP packets between end-hosts and applications which have the same source and destination addresses, the same TCP/UDP port numbers, and the same protocol field. In this approach, each flow can request specific levels of service from the network. The levels of service are typically quantified as a minimum service rate, or a maximum tolerable end-to-end delay or loss-rate. The network grants or rejects the flow requests, based on the availability of the resources and the guarantees provided to other flows.

The three major components of the IntServ architecture are the *Admission Control* unit, which checks if the network can grant the service request, the *Packet Forwarding Mechanisms*, which perform the per-packet operations of flow classification, shaping, scheduling, and buffer management in the routers, and the *Resource Reservation Protocol (RSVP)*, which sets up some *flow-state* (e.g., bandwidth reservations, filters, accounting etc) in the routers that a flow goes through. The IntServ approach is based on a solid background of research in Quality-of-Service mechanisms and protocols for packet networks [2, 3, 4]. However, the acceptance of IntServ from network providers and router vendors has been quite limited, at least so far, mainly due to *scalability* and *manageability* problems [5]. The scalability problems arise because IntServ requires routers to maintain control and forwarding state for all flows passing through them. Maintaining and processing per-flow state for gigabit or terabit links, with millions of simultaneously active flows, is significantly difficult from an implementation point of view. Even if next generation routers can accommodate millions of flows, perhaps using approximate mechanisms such as CSFQ [6], the IntServ architecture makes the management and accounting of IP networks significantly more complicated. Additionally, it requires new application-network interfaces, and it can only provide service guarantees when all networks in the flow's path are IntServ-capable.

The Differentiated Services Approach. The *Differentiated Services* (DiffServ) approach is more recent than the IntServ approach. The main goal of DiffServ is a more scalable and manageable architecture for service differentiation in IP networks [7, 8]. The initial premise was that this goal can be achieved by not focusing on individual packet flows, but on *traffic aggregates*, where a traffic aggregate is a large set of flows with similar service requirements. Table 1 summarizes the main differences between the IntServ and DiffServ architectures.

Since the DiffServ approach is still evolving, many of its aspects are not yet clear. Research on DiffServ, however, is proceeding along two different directions. The first direction, which we refer to as *Absolute Service Differentiation*, can be thought of as trying to meet the same goals with IntServ, i.e., absolute performance levels, but without per-flow state in the backbone routers, and with only some semi-static resource reservations, instead of a dynamic resource reservation protocol. The second direction, which we refer to as *Relative Service Differentiation*, involves service models that provide assurances for the relative quality ordering between classes, rather than for the actual service level in each class. In this paper, we focus on Relative Differentiated Services, making the case that it is an easy architecture to deploy and manage and that it is capable of providing users with a better service at a possibly higher cost. We then propose the Proportional Differentiation Model, as a way to control the quality spacing between classes locally at each hop, independent of the

	Integrated Services	**Differentiated Services**
Granularity of service differentiation	Individual flow	Aggregate of flows
State in routers (e.g., scheduling, buffer management)	Per-flow	Per-aggregate
Traffic classification basis	Several header fields	The DS field (6 bits) of the IP Header
Type of service differentiation	Deterministic or statistical guarantees	Absolute or relative assurances
Admission control	Required	Required for absolute differentiation only
Signaling protocol	Required (RSVP)	Not required for relative schemes. Absolute schemes need semi-static reservations or 'broker' agents.
Coordination for service differentiation	End-to-End	Local (per-hop)
Scope of service differentiation	A unicast or multicast path	Anywhere in a network or in specific paths
Scalability	Limited by number of flows	Limited by number of classes of service
Network accounting	Based on flow characteristics and QoS requirement	Based on class usage
Network management	Similar to circuit-switched networks	Similar to existing IP networks
Interdomain deployment	Multilateral agreements	Bilateral agreements

Table 1: A comparison of the IntServ and DiffServ architectures.

class loads. According to this model, certain forwarding performance metrics are ratioed proportionally to the *class differentiation parameters* that the network operator chooses, leading to controllable service differentiation. Additionally, if the specified class differentiation holds even in short timescales, users can be assured that higher classes will be better, independent of the traffic load distribution and variations (burstiness). Finally, we briefly describe packet forwarding mechanisms for approximating the proportional differentiation model, in terms of queueing delay and packet loss-rate performance metrics.

Differentiated Services vs. the Fat-Dumb-Pipe model. Besides IntServ and DiffServ, another approach for meeting the user and application service expectations is to overprovision the network so that there is no congestion, and thus no queueing delays or packet losses. This can be achieved with the deployment of very high capacity links, relative to the rate of the arriving traffic. In fact, this approach is currently being adopted by several backbone providers, since it is the simplest solution to the problem of traffic congestion. The 'fat-dumb-pipe' model, however, can be extremely inefficient in terms of network economics and resource management. The reason is that all traffic receives the same, normally very high, quality-of-service, even though not all applications need that quality level.

To illustrate this point, Figure 1 shows a generic *quality-requirement versus traffic-mix* curve for a network link. About 20% of the traffic is E-mail, FTP, or other applications that have very low-quality requirements from the network, 40% is WWW-browsing with somewhat higher requirements, 20% is audio/video conferencing, while the final 20% of the traffic is very demanding traffic, generated from applications like distributed interactive games. In the 'fat-dumb-pipe' model, corresponding to Figure 1-a, the network provider provisions the link bandwidth and buffers so that the entire traffic mix gets the quality-of-service that the most demanding applications require. The shaded area in the graph, between the quality requirements of the traffic distribution and the quality provided by the link, is related to wasted network resources, since it represents 'redundant' quality-of-service, beyond what the applications need in order to operate successfully. A DiffServ network, on the other hand, would split the traffic mix in a number of classes (say four), as shown in Figure 1-b. In this way, each traffic type would get slightly more than the quality that it needs, but not much more, i.e., the wasted resources in this case would be significantly less. Consequently, the required link capacity in the DiffServ approach would be less than in the 'fat-dumb-pipe' approach, since the network resources would be used more efficiently in the former case.

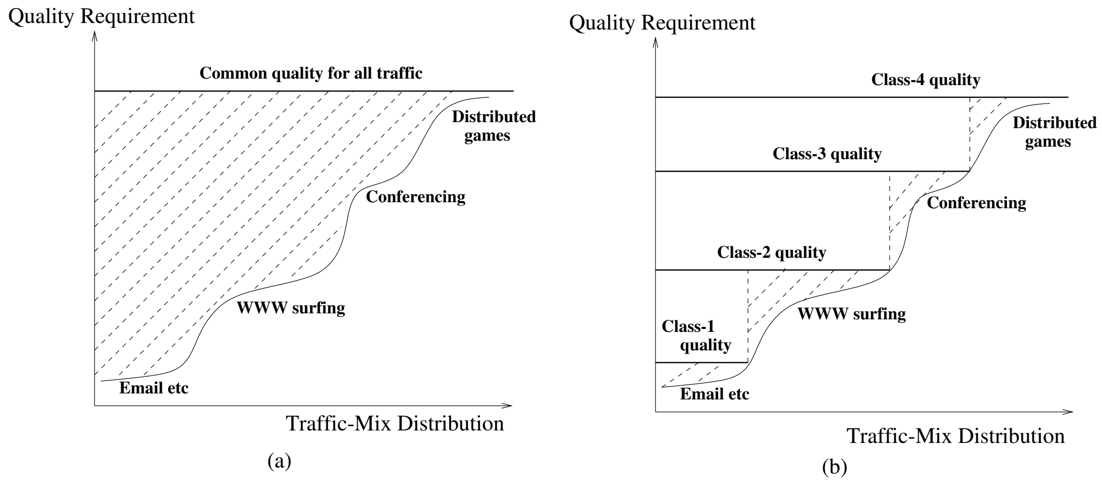


Figure 1: Two *quality-requirement versus traffic-mix* curves for the case of a link that follows the 'fat-dumb-pipe' model (a), as opposed to the DiffServ model (b). The shaded areas are related to wasted network resources.

The critical question is how can the network provider adjust the quality spacing between classes to achieve the class allocation of Figure 1-b. First, note that it is desirable for the number of classes to be roughly equal to the number of distinct quality-requirement 'groupings' of the traffic mix (in this example, four groupings). Second, the network operator must be able to control the quality spacing between classes based on certain *class differentiation parameters* provided by the router. Such tuning-knobs are necessary for adjusting the link operating point as in Figure 1-b; we will return to this issue when we discuss the proportional differentiation model. A third point, specifically for relative differentiation schemes, is that, in the absence of admission control, some quality-requirement versus traffic-mix curves may be *infeasible*, given a certain amount of link forwarding resources. Some basic results for this feasibility issue are presented in [9], for the case of average delay differentiation.

2 Relative Differentiated Services: Why and How

The central premise in relative differentiated services is that the network traffic is grouped into N service classes, which are ordered based on their packet forwarding quality:

> Class i is better (or at least no worse) than Class $(i-1)$ for $1 < i \leq N$, in terms of local (per-hop) performance measures 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 IETF has recently standardized eight such classes, called *Class Selector* Per-Hop-Behaviors (PHBs), using the Precedence bits of the IPv4 packet header [7]. Depending on the deployment scenario, the classification of packets to different classes can be done either by the application, or by the end-host, or by the router at the boundary between two networks. For example, an IP-telephony application may dynamically adapt the classification of its packets, based on the measured delays and losses in the ongoing call. Or, in the case of an organization, the classification of packets may be based on policy rules, such as that the Management department uses the highest class, while the Engineering department uses a lower class.

A relative differentiated services model must be strongly coupled with a pricing or policy-based scheme to make higher classes more costly (or more usage-restricted) than lower classes. Otherwise, everyone would use the highest class and the relative differentiation would be ineffective. The pricing scheme can be either flat or usage-based. In the flat-rate pricing model, a user/application subscribes to a certain class by paying the appropriate fee, and in return, all generated packets belong to that class. In the usage-based pricing model, the user/application will be charged based on the number of packets generated in each class. Such a model is more suitable for cases where the user/application prefers to dynamically adapt the packet classification.

Relative vs. Absolute Service Differentiation. Before we discuss specific schemes for relative service differentiation, let us explore the differences between absolute and relative differentiated services. In the absolute model, an admitted user is assured of his/her requested performance level. The disadvantage, of course, is that a user will be rejected if the required resources are not available and the network cannot provide the requested assurances. For example, suppose that an IP telephony application requests a bandwidth of 32 Kbps and an end-to-end delay of at most 200 ms. If the user's request is accepted, then the quality of the ensuing call is assured. However, if the network is unable to provide the requested bandwidth and/or end-to-end delay, the user will receive a busy signal.

In the relative differentiation model, on the other hand, the only assurance from the network is that a higher class will receive better service than a lower class. The amount of service received by a class and the resulting quality-of-service perceived by an application depend on the current network load in each class. The users/applications, in this context, are supposed to either adapt their needs based on the observed performance level in their class, or to switch to a better class, if their cost constraints allow this transition. For example, an adaptive IP telephony application, may use a range of encoding techniques and playback-adaptation mechanisms to offer reasonable (albeit sometimes degraded) quality, when the available bandwidth is in the range of 6 Kbps to 32 Kbps and the end-to-end delays are up to 300 ms. If the delays in the current class are observed to be larger than 300 ms, the application can dynamically switch to higher classes, until it finds the lowest class in which it can operate adequately. If there is no such class, or the user has specified some maximum cost constraints, the user will experience degraded quality, since that network path was not configured and/or priced to meet this quality-requirement and cost combination.

We do not argue that absolute service differentiation is not required by any applications. However, the growing popularity of many adaptive applications (e.g., RealPlayer and IP telephony products) show that users can often tolerate variations in quality. Of course, some adaptive applications receive occasionally unacceptable levels of service today. We claim that this happens either because of badly provisioned links, or because of the 'same-service-to-all' model; once this model is replaced with service differentiation schemes, however, the users and applications will have an additional adaptation knob, namely the class of service, to control the quality that they receive. The cost of Internet access will also depend then on the class of service that users choose and the applications that they run.

Three Relative Differentiation Models. There are several different ways to provide relative service differentiation. We briefly discuss three existing schemes here, while in the next section we propose the proportional differentiation model which addresses some of the problems of the following schemes.

Strict Prioritization: It is a common misconception that a relative differentiation model has to service classes in a strict-priority manner, with higher classes being serviced before lower classes. Even though such a service scheme would maintain the desired quality ordering between classes (i.e., higher classes are always better), it has some important drawbacks. First, if the higher classes are persistently backlogged, it can result in long starvation periods for the lower classes. Second, a strict prioritization scheme is *not controllable*, i.e., it does not provide any 'tuning knobs' for adjusting the quality spacing between classes. Instead, the operating point depends only on the load distribution between classes. As we discuss later, the ability to adjust the quality spacing between classes independently of the class loads is a crucial property for a relative differentiation scheme.

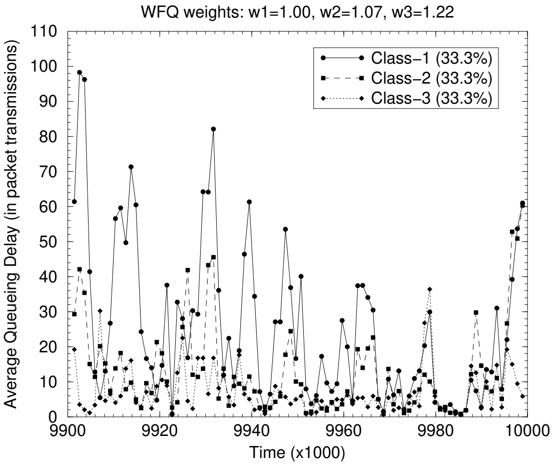
Price Differentiation: A simple case of relative service differentiation is the Paris Metro Pricing (PMP) scheme [10]. PMP is based on pricing, instead of special forwarding mechanisms, to provide relative class differentiation. It is based on the assumption that higher prices will lead to lower loads in the higher classes, and thus, better service quality. Pricing mechanisms, however, can only be effective over relatively long timescales, especially when the class tariffs cannot be modified frequently. 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*, or *unpredictable*, class differentiation.

Capacity Differentiation: Another approach for providing relative differentiated services is to allocate a larger amount of forwarding resources (bandwidth or buffer space) to higher classes, relative to the expected load in each class. To illustrate this point, suppose that a class i has an average arrival rate λ_i. A Weighted-Fair-Queueing (WFQ) scheduler [11] with a class-i weight w_i can be configured to provide higher classes with a larger relative share of the link bandwidth, i.e.,

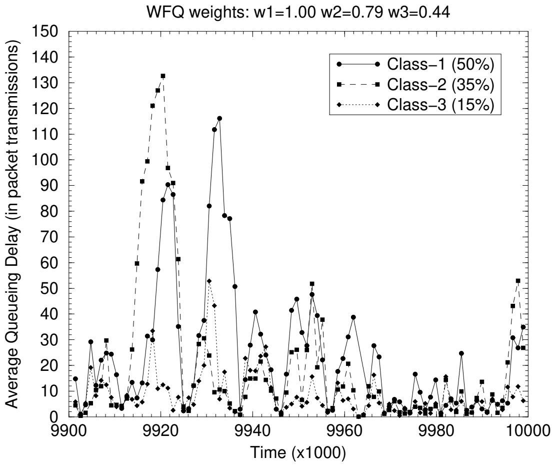
$$\frac{w_i}{\lambda_i} > \frac{w_j}{\lambda_j} \quad \text{if} \quad i > j$$

leading in this way to lower average delays for the higher classes. This approach is discussed in [12], for example, as a possible implementation of the *Olympic service model*, which consists of the 'Gold', 'Silver', and 'Bronze' classes.

The capacity differentiation approach, however, has an important drawback, which becomes clear mainly in shorter timescales. Specifically, higher classes can often provide worse quality-of-service than lower classes, invalidating the main premise of relative service differentiation. The reason for this behavior is that the service quality in each class depends on the short-term relation between the allocated service to a class and the arriving load in that class. The short-term class loads, however, may deviate from the long-term class loads over significantly large time intervals. This is especially true for the Internet traffic, which has been shown to be extremely bursty over a wide range of timescales (see [13] and references therein). Since many Internet applications are of short duration (e.g., a Web session), it is important that the class relative ordering also remains consistent in short timescales. From another point of view, it is important to give users the assurance that, independent of when they monitor the differentiation between two classes, they will always find that higher classes are better; this can be achieved if the differentiation is predictable in short timescales.



(a) Equal class load distribution



(b) Unequal class load distribution

Figure 2: Queueing delay variations with a Weighted-Fair-Queueing scheduler for two cases of load distribution. The class weights are selected so that the average delay of a class is approximately double the average delay of the next higher class.

The inadequacy of a WFQ scheduler to provide consistent delay differentiation in short timescales is illustrated in the simulation results of Figure 2. These graphs show the per-class queueing delays, averaged in successive intervals of 100 packet transmission times, with a WFQ scheduler. The per-class WFQ weights are selected based on trial-and-error, since there is no corresponding analytic methodology, so that the average delay of a class is approximately double the average delay of the next higher class. Note that the class weights are dependent on the class load distribution, making it hard to control the delay spacing between classes when the class load distribution is varying. Even more, the relative ordering between classes is often violated, i.e., higher classes often have larger delays than lower classes. Additional details for these simulations are described in Section 4, and in [14].

Two Features for Relative Service Differentiation. The drawbacks of the previous differentiation schemes point out to two important features that a relative service differentiation model should have:

- *Controllability*, meaning that the network operators should be able to adjust the quality spacing between classes based on their pricing or policy criteria.

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

In the next section we present the proportional differentiation model, which is designed to be controllable and predictable.

3 The Proportional Differentiation Model

The proportional differentiation model states that certain class performance metrics should be proportional to the 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. Consequently, we next focus on two performance metrics, one for the short-term queueing delays in each class, and one for the short-term loss-rates. A generic description of the proportional differentiation model follows. Suppose that $\bar{q}_i(t, t+\tau)$ is a performance measure for class i in the time interval $(t, t+\tau)$, where $\tau > 0$ is the *monitoring timescale*. Since we are interested in differentiation over short timescales, the value of τ should be relatively small. The proportional differentiation model imposes constraints of the following form for all pairs of classes and for all time intervals $(t, t+\tau)$ in which both $\bar{q}_i(t, t+\tau)$ and $\bar{q}_j(t, t+\tau)$ are defined:

$$\frac{\bar{q}_i(t, t+\tau)}{\bar{q}_j(t, t+\tau)} = \frac{c_i}{c_j}$$

where $c_1 < c_2 < \ldots < c_N$ are the generic *Quality Differentiation Parameters (QDPs)*. The basic idea is that, even though the actual quality level of each class will vary with the class loads, the quality ratio between classes will remain fixed and controllable by the network operator, independent of the class loads. In addition, because the class differentiation is to hold in short timescales, the relative ordering between classes is consistent and predictable from the user's perspective.

The proportional differentiation model can be applied in the context of queueing delays by setting $\bar{q}_i(t, t+\tau) = 1/\bar{d}_i(t, t+\tau)$, where $\bar{d}_i(t, t+\tau)$ is the average queueing delay of the class i packets that departed in the time interval $(t, t+\tau)$. If there are no such packets,

$\bar{d}_i(t, t+\tau)$ is not defined. The *Proportional Delay Differentiation* model states that for all pairs of classes and 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:

$$\frac{\bar{d}_i(t, t+\tau)}{\bar{d}_j(t, t+\tau)} = \frac{\delta_i}{\delta_j} \tag{1}$$

where the parameters $\{\delta_i\}$ are the *Delay Differentiation Parameters (DDPs)*, being ordered as $\delta_1 > \delta_2 > \ldots > \delta_N$. In the case of loss-rate differentiation, we set $\bar{q}_i(t, t+\tau) = 1/\bar{l}_i(t, t+\tau)$, where $\bar{l}_i(t, t+\tau)$ is the fraction of class-i packets that were backlogged at time t or that arrived during the interval $(t, t+\tau)$, and that were dropped in this same time interval. In this case, the *Proportional Loss-rate Differentiation* takes the form:

$$\frac{\bar{l}_i(t, t+\tau)}{\bar{l}_j(t, t+\tau)} = \frac{\sigma_i}{\sigma_j} \tag{2}$$

where the parameters $\{\sigma_i\}$ are the *Loss-Rate Differentiation Parameters (LDPs)*, being ordered as $\sigma_1 > \sigma_2 > \ldots > \sigma_N$.

The proportional differentiation model is controllable from the network operator's perspective, using the Quality Differentiation Parameters. It is also predictable, since, if the value of τ is sufficiently small, higher classes are consistently better than lower classes even in short timescales. On the negative side, though, the proportional differentiation model is not always *feasible* using work-conserving forwarding mechanisms. Depending on the load in each class, the specified QDPs, and the monitoring timescale τ, there are cases in which the proportional differentiation specified by the QDPs cannot be achieved with a work-conserving mechanism. This can occur, for example, when the highest class is much more heavily loaded than the lower classes, but the network operator specifies a very high QDP for that class. Or, more specifically, even if the highest class is given strict priority over all other classes, there is still a bound on how low delay it can get, due to its own inherent load. The feasibility test for the case of long-term average delays is presented in [9], based on fundamental results from [15]. The feasibility issues for short-term performance metrics, like the ones used in (1) and (2), however, are an open question which requires further investigation. In the next section, we describe a packet scheduler and a buffer manager that can *approximate* the proportional differentiation model quite closely in heavy-load conditions, when the monitoring timescale is in the order of tens or hundreds of packet transmission times.

	Premium Service	**Assured Service**	**Proportional Differentiation**
Type of Service	Strictly absolute	Depends on provisioning algorithm	Strictly relative
Bandwidth-Broker Agent	Required	Not required	Not required
Route-Pinning	Important	Important	Not important
Operation at Network Ingress	Policing	Marking (IN/OUT or drop-preference)	Class-usage accounting
End-Point Adaptation	Not required	Important	Important or required

Table 2: A comparison of three DiffServ models.

Comparison to two other DiffServ models. Two other service models that are being actively considered in the context of DiffServ are the Premium Service [16], and the Assured Service [17]. We briefly discuss them next, while Table 2 compares them with the proportional differentiation model.

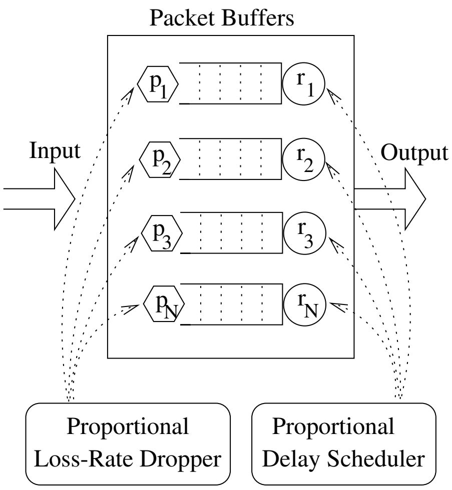
Premium (or Virtual Lease Line) service: In this model, a premium-service user is given the guarantee for a nominal bandwidth with minimal queueing delays and losses along a certain network path, independent of the behavior of the rest of the traffic in that path. That is, the 'assurance-level' of this service is similar to that of the Guaranteed service in the IntServ approach [1]. The Premium service requires some form of semi-static bandwidth reservations, that a 'bandwidth broker' protocol or agent has to setup across domains. It also requires some form of *route-pinning* for holding the Premium traffic in the links where the bandwidth reservations have been setup, despite of any dynamic re-routing that often happens in IP networks. Finally, since the resource reservations are made in a less dynamic manner, they often need to be quite conservative in order to allow for concurrent activation of large volumes of Premium traffic in the same links.

Assured Service: The Assured Service also provides users with bandwidth assurances along certain network paths or in an entire network, but without strict guarantees that this bandwidth will always be available. In other words, the Assured service is based more on *provisioning and statistical assurances*, rather than on bandwidth reservations for each user. However, some recent works have shown that it is difficult, if not impossible, to design provisioning algorithms that achieve simultaneously good service quality and high resource utilization for such services, with large spatial granularities [18]. Besides, some form of route-pinning is also necessary to support this service model. As a result, the Assured service can also be viewed as a relative service differentiation scheme, in which users that paid for a higher bandwidth profile get a better service than those who paid for a lower profile, even though they do not get exactly the profile that the network promised them. A similar approach is the User-Share-Differentiation scheme [19], in which the per-hop allocation of bandwidth is performed in proportion to the profile that each user (or groups of users) paid for.

4 Forwarding Mechanisms for Proportional Differentiation

An important question is whether there are packet forwarding mechanisms that can achieve or approximate the Proportional Differentiation model. If approximations are necessary, under which conditions these mechanisms perform adequately close, from a practical point of view, to the proportional model? In this section we briefly summarize some first results in this direction; it has to be emphasized, though, that this is an ongoing research effort and that we do not claim that the presented mechanisms are 'optimal' in any sense, or that we have conclusive answers for these questions. The model of a forwarding engine that implements the Proportional Differentiation model is shown in Figure 3. The buffers for a particular output interface (assuming an output-queued router) are organized into a set of N logical queues, one for each class. The N queues share the link bandwidth and the physical buffer space using a packet scheduler and a buffer manager, respectively. In this model, a 'Proportional Delay Scheduler' distributes *dynamically* the link bandwidth to the N classes, attempting to maintain the proportional delay constraints of Equation 1. This is a fundamentally different scheduling approach, compared to the WFQ [11], CBQ [20], H-PFQ [21], or H-FSQ [22] schedulers, in which each class is guaranteed a certain minimum bandwidth. These latter schedulers are designed in the link-sharing context, where different organizations or users are guaranteed a certain fraction of a link's capacity, sharing dynamically any available excess bandwidth. Since these schedulers do not directly adjust the link shares, however, in order to make the short-term queueing delays in the higher classes lower than the delays in the lower classes, they are not ideal for relative differentiated services.

At the buffer management level, a 'Proportional Loss-Rate Dropper' decides which



.

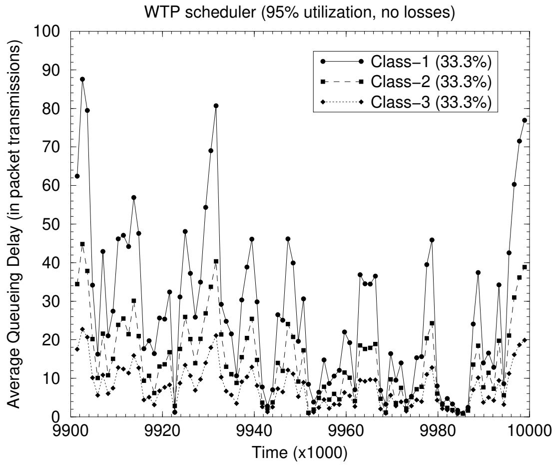
Figure 3: The main components of a forwarding engine in the context of the Proportional Differentiation Model.

class's packets to drop, whenever there is need to do so, so that the proportional loss-rate constraints of Equation 2 are closely approximated. Note that the decision to drop a packet is independent of the selection of the class from which the packet will be dropped. For example, packet drops can occur whenever there is a shortage of buffers, using a Drop-Tail buffer management scheme. Alternatively, an active buffer management scheme, such as RED [23], can be used to decide that a packet needs to be dropped. In that case, the RED module would monitor the *aggregate* load in the N classes, and then based on a 'loss-rate vs. aggregate load' law it would provide feedback to the traffic sources to reduce their rate by dropping some packets. The selection of the class from which the packet will be dropped from would be performed by the Dropper module, though.

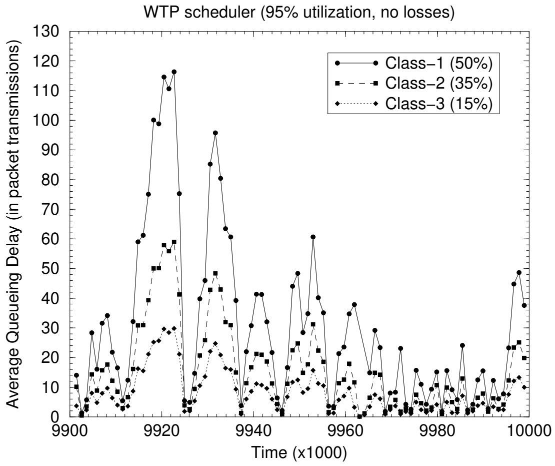
A Scheduler for Proportional Delay Differentiation. A packet scheduler that can approximate the Proportional Delay Differentiation model in short timescales is the *Waiting-Time Priority* (WTP) scheduler. In this scheduler the priority of a packet increases proportionally with its waiting-time. Specifically, the priority of a packet in queue i at time t is

$$p_i(t) = \frac{w_i(t)}{\delta_i}$$

where $w_i(t)$ is the waiting-time of the packet at time t. The DDPs $\{\delta_i\}$ determine the rate with which the priority of the packets of a certain class increases with time. The WTP algorithm was first studied by L. Kleinrock in 1964, with the name Time-Dependent-Priorities



(a) Compare with Figure 2-a



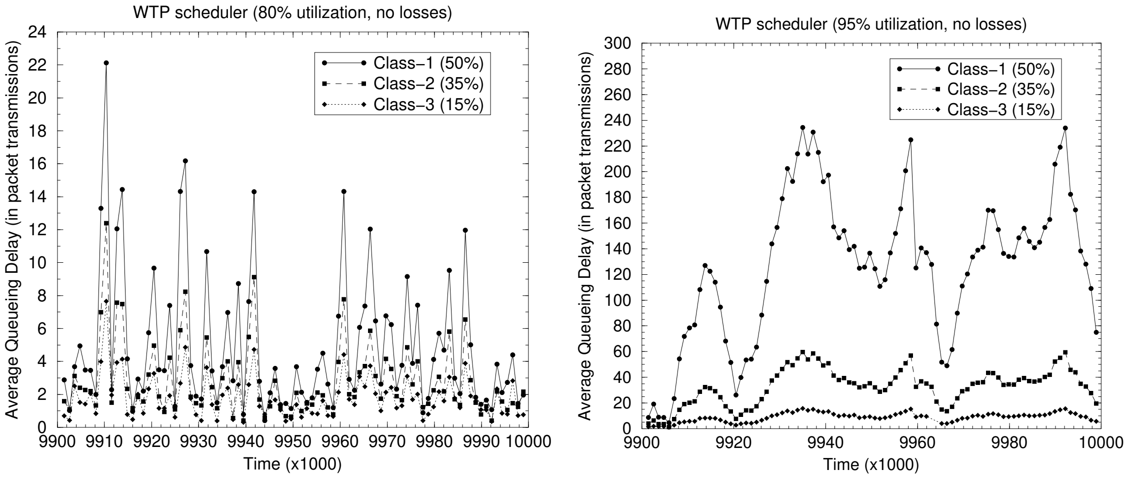
(b) Compare with Figure 2-b

Figure 4: Queueing delay variations with the WTP scheduler when $\delta_1 = 1, \delta_2 = 1/2, \delta_3 = 1/4$. The average delays are measured every 100 packet transmission times.

[24]. Our main finding in [9] is that the WTP scheduler approximates the proportional delay differentiation model of Equation (1) in heavy-load conditions, even with a monitoring timescale τ of a few tens of packet transmission times, when the delay differentiation that the DDPs specify is feasible.

In the following we show some simulation results for the per-class queueing delay variations with a WTP scheduler, illustrating this property of WTP, as well as the magnitude of the deviations from the proportional delay differentiation model under different conditions. The simulation scenario is a WTP scheduler that is loaded with random traffic from three classes. In all cases, the packet interarrivals follow an infinite-variance Pareto distribution ($\alpha = 1.9$) in order to create bursty traffic. 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 is 11.2 time units. The following graphs show only queueing delays, measured in (average) packet transmission units. The average link utilization and the class load distribution is given in each graph, together with the Delay or Loss-Rate Differentiation Parameters.

Figure 4 shows the queueing delay variations for the same time interval and the same input traffic streams as the WFQ simulations of Figure 2. The monitoring timescale is $\tau = 100$ packet transmission times. Note that the delay differentiation between classes is more predictable with WTP than with WFQ, and that there are no cases in which



(a) Moderate load conditions

(b) Wider delay differentiation spacing

Figure 5: The WTP behavior in moderate-loads (a'), and in wider differentiation constraints ($\delta_1 = 1, \delta_2 = 1/4, \delta_3 = 1/16$) (b'). The average delays are measured every 100 packet transmission times.

higher classes have larger delays than lower classes. Also, note that WTP approximates the proportional differentiation model fairly well. Since $\delta_1 = 1, \delta_2 = 1/2, \delta_3 = 1/4$, the queueing delay in each class is approximately double the delay of the next higher class. The deviations from these proportionality constraints are smaller during high-load intervals, in which the delay differentiation is even more important.

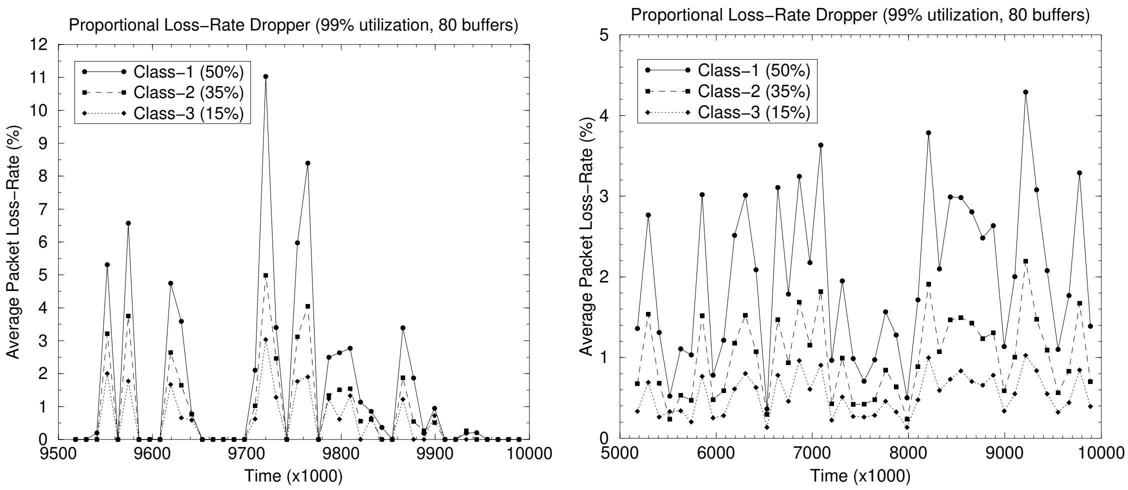
It is interesting to examine the behavior of WTP in other operating regimes, such as lower link utilization, or wider delay differentiation constraints. Figure 5-a shows the queueing delay variations when the aggregate utilization is 80%, which we consider moderate-load conditions. When the aggregate utilization is lower than 70% or so, the queueing delays even with the very bursty Pareto traffic are too low, reducing the need for a service differentiation scheme. Note that the deviations of WTP from the proportional differentiation model in these moderate-load conditions are significantly higher, especially during periods where the average delays are in the order of one or two packet transmission times. This is due to both WTP's operation, and also because the proportional differentiation model may not be feasible in these short timescales and load conditions. On the other hand, it is important that the behavior of WTP is quite close to the proportional model in high-load conditions, because it is exactly in that operating region that service differentiation mechanisms are mostly needed. Figure 5-b shows the queueing delay variations when the delay differentiation spacing is wider, i.e., the queueing delay in each class is approximately four times the delay of the next higher class. Note that the deviations from the proportional differentiation model are, again, minor during high-load conditions. However, as we increase the required delay spacing between classes, the deviations from the proportional model increase, given the same load conditions.

A Dropper for Proportional Loss-Rate Differentiation. A *Proportional Loss-Rate Dropper* is described next. The dropper maintains a *Loss-History-Buffer (LHB)*, which is a cyclical queue. The LHB stores loss-related information for the last K arrived packets: which class they belong to, and if they were dropped while they were recorded in the LHB. Using these two fields, the dropper computes a loss-rate l_i for each class i, as the fraction of class-i packets recorded in LHB that were dropped. Note that another approach would be to define the loss-rate in terms of bytes instead of packets; in that case the LHB would have to record also the length of each packet. When a packet needs to be dropped, the dropper selects the backlogged class j with the minimum normalized loss-rate, i.e.,

$$j = \text{argmin}_{\ i} \ \{\frac{l_i}{\sigma_i}\}$$

Dropping a packet from that class increases the normalized loss-rate l_j/σ_j, reducing its distance from the normalized loss-rates of other classes. The expectation is that if this dropper makes the normalized loss-rates roughly equal, the proportionality constraints of Equation 2 will be approximately met, as long as the monitoring timescale τ is in the same order as the size K of the LHB. Note that in order to achieve loss-rate differentiation in short-timescales, we would prefer lower values of K. However, as we decrease K, the deviations from the proportional loss-rate model increase, in general, because it becomes harder to equalize the normalized loss-rates.

Figure 6 shows the per-class loss-rate variations using the Proportional Loss-Rate Dropper and a Drop-Tail buffer management scheme, for two values of K. In each case, the monitoring timescale τ is set to K average packet transmission times (although the exact equality of the two is not required), while the simulation intervals are adjusted so that the two graphs have roughly the same number of points. Note that as K increases from 1000 to 10000 packets, the deviations from the proportional loss-rate differentiation model decrease significantly. On the other hand, it is not clear if $\tau = 10000$ packet transmission times is an adequately short timescale; for a packet size of 500 bytes, this interval corresponds to about 270 milliseconds in an OC-3 link and to about 27 seconds in a T1 link.



(a) $K = 1000$ packets

(b) $K = 10000$ packets

Figure 6: Loss-Rate variations with the Proportional Loss-Rate Dropper when $\sigma_1 = 1, \sigma_2 = 1/2, \sigma_3 = 1/4$.

5 Summary

The DiffServ architecture can provide the means for extending the Internet forwarding paradigm with scalable and easy-to-deploy service differentiation mechanisms. In the absolute differentiation approach, these mechanisms can offer absolute assurances, which are mainly useful for 'unelastic' applications and for deployment scenarios that require specific service measures (such as in Virtual Private Networks). In the relative differentiation approach, the DiffServ architecture can provide different applications and users with the flexibility of selecting the forwarding class that best matches their quality-cost tradeoff, assuming some adaptation at the end systems and applications. In this paper, we first made a case for the relative differentiation approach, as a simple to implement, deploy, and manage solution for service differentiation in the global Internet. We then proposed a specific type of relative services, based on the Proportional Differentiation Model. This model allows the network operator to control the quality spacing between classes independent of the class loads, and it can provide consistent class differentiation in short timescales. Finally, we described a packet scheduling and a buffer management mechanism for approximating the proportional differentiation model in the forwarding engine of a router.

Acknowledgments

This work has greatly benefited from discussions with Dimitrios Stiliadis.

References

[1] P. P. White, "RSVP and Integrated Services in the Internet: A Tutorial," *IEEE Communications Magazine*, pp. 100–106, May 1997.

[2] A. Banerjea, D. Ferrari, B. A. Mah, M. Moran, D. C. Verma, and H. Zhang, "The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 1–10, Feb. 1996.

[3] A. K. Parekh, *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. PhD thesis, Massachusetts Institute of Technology, 1992. LIDS-TH-2089.

[4] A.Enwalid, D.Heyman, T.V.Lakshman, D.Mitra, and A.Weiss, "Fundamental Bounds and Approximations for ATM Multiplexers with Applications to Video Teleconferencing," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1004–1016, Aug. 1995.

[5] 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.

[6] 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.

[7] 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.

[8] S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang, and W.Weiss, *An Architecture for Differentiated Services*, Dec. 1998. IETF RFC 2475.

[9] C.Dovrolis, D.Stiliadis, and P.Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," in *ACM SIGCOMM*, Sept. 1999.

[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] J.Heinanen, F. Baker, W. Weiss, and J. Wroclawski, *Assured Forwarding PHB Group*, June 1999. RFC 2597.

[13] A.Feldmann, A.C.Gilbert, and W.Willinger, "Data networks as cascades: Investigating the multifractal nature of the Internet WAN traffic," in *Proceedings SIGCOMM Symposium*, 1998.

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

[15] 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.

[16] V.Jacobson, K. Nichols, and K.Poduri, *An Expedited Forwarding PHB*, June 1999. RFC 2598.

[17] 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.

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

[19] Z. Wang, "A Case for Proportional Fair Sharing," in *International Workshop on QoS*, May 1998.

[20] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 365–386, Aug. 1995.

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

[22] I.Stoika, H.Zhang, and T. Ng, "A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services," in *Proceedings ACM SIGCOMM*, Sept. 1997.

[23] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

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

Authors' Biographies

Constantinos Dovrolis is a Ph. D. candidate in the Department of Electrical and Computer Engineering at the University of Wisconsin, Madison. His research interests include the architecture and implementation of Internet routers, network traffic engineering, and network pricing. Constantinos was an intern at Bell Laboratories in the summer of 1998, and at the Cooperative Association for Internet Data Analysis (CAIDA) in the summer of 1999. He has received the Computer Engineering degree from the Technical University of Crete, Greece, in 1995, and the M. S. degree in VLSI design from the University of Rochester in 1996. His email address is: dovrolis@ece.wisc.edu.

Parameswaran Ramanathan (M'89/ACM'89) is an Associate Professor in the Department of Electrical & Computer Engineering and in the Department of Computer Sciences at the University of Wisconsin, Madison. He received the B. Tech degree from the Indian Institute of Technology, Bombay, India, in 1984, and the M. S. E. and Ph. D. degrees from the University of Michigan, Ann Arbor, in 1986 and 1989, respectively.

From 1984 to 1989 he was a Research Assistant in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. He was an Assistant Professor in the Department of Electrical & Computer Engineering at the University of Wisconsin, Madison from 1989 to 1995. In 1997-98, he was on sabbatical leave from the University. He spent his sabbatical leave at AT&T Labs, Whippany, New Jersey and Bellcore, Morristown, New Jersey. His research interests include the areas of wireless and wireline networks, real-time systems, fault tolerant computing, and distributed systems. His email address is: parmesh@ece.wisc.edu.