Class provisioning using proportional delay differentiation

Constantinos Dovrolis[a] and Parameswaran Ramanathan[b]

[a]Computer and Information Sciences, University of Delaware
[b]Electrical and Computer Engineering, University of Wisconsin-Madison

ABSTRACT

We consider the problem of *link provisioning* in a differentiated services network that offers N *classes of service*. At the provisioning phase, the network manager configures the link to support the requirements of M distinct traffic types. Each traffic type is specified by an expected arrival rate and an average delay requirement. The objective of the provisioning phase is to jointly determine: the *minimum link capacity* needed to support the M given traffic types, the *nominal class of service* for each traffic type, and the *appropriate resource allocation* between classes.

We propose such a *class provisioning methodology*. Our methodology is based on Proportional Delay Differentiation (PDD) scheduling. The major advantage of PDD scheduling is that it avoids the computation of an explicit bandwidth share for each class. Instead, PDD scheduling determines the order of packet transmissions in order to meet the $N-1$ ratios of the N target class delays. Having fixed the delay ratios with PDD, we then set the N class delays to their target values adjusting a single knob, which is the link capacity. The methodology is illustrated with examples.

1. INTRODUCTION

The IETF has recently standardized eight differentiation classes, called *Class Selector Compliant Per-Hop-Behaviors (CSC PHBs)*, or simply *Class Selectors*.[1] These classes of service are *ordered*, in the sense that higher classes receive better performance (lower queueing delays and lower loss rate). Such a *relative differentiation architecture* is easier to deploy and manage, because it does not require admission control, bandwidth brokers, resource reservations, signaling, or route pinning.[2]

Since there is no admission control, however, the offered load at a link cannot be controlled. Consequently, it is possible that an application with absolute QoS requirements will not find an acceptable class, even if it makes use of the highest service class. This depends on the amount of forwarding resources at the link (transmission capacity and number of buffers), on the allocation of forwarding resources between classes, and on the volume and performance requirements of the rest of the traffic. Intuitively, if the link is *well-provisioned*, there should be an acceptable class for each traffic type. But what does it exactly mean for the link to be well-provisioned? And how can a network manager perform such provisioning?

This is the kind of questions that we attempt to answer in this paper. Specifically, we investigate the following instance of the provisioning problem: *how can a network manager provision a link to meet an average delay requirement for each traffic type, requiring the minimum link capacity?* In order to provision a link, the network manager needs a *workload profile*. The workload profile is a specification of the anticipated traffic types in the link, in terms of their arrival rate and average delay requirement. Given this profile, a provisioning methodology has the following outcomes:

1. The nominal service class for each traffic type.
2. The minimum link capacity for the given workload profile.
3. The required capacity allocation between classes.

Contact authors at dovrolis@cis.udel.edu, parmesh@ece.wisc.edu.

We propose such a *class provisioning methodology.* The methodology uses *Proportional Delay Differentiation (PDD) scheduling.*[3] The major advantage of PDD scheduling is that it avoids the computation of an explicit bandwidth share for each class. Instead, PDD scheduling determines the order of packet transmissions in order to meet the $N-1$ ratios of the N target class delays. Having fixed the delay ratios with PDD, we then set the N class delays to their target values adjusting a single knob, which is the link capacity.

Section 2 explains the class provisioning problem in more detail. Section 3 describes the link and packet scheduling model. Section 4 derives the nominal service class for each traffic type. Section 5 computes the minimum link capacity and the required Delay Differentiation Parameters for the PDD scheduler. Section 6 comments on the 'average backlog function', which is required for the calculation of the minimum link capacity. Section 7 summarizes the paper and suggests possible extensions.

2. THE PROBLEM OF CLASS PROVISIONING

In the provisioning phase, the objective of the network manager is to configure a network link at a *desired operating point.* The "knobs" that the manager can control are the link forwarding resources, as well as the allocation of these resources between classes. An important issue in the provisioning phase is to use the minimum capacity, especially if the cost of the link increases with its capacity. Even when this is not strictly the case (say in optical networks), the network manager would still be interested to *at least know* the minimum link capacity required.

The desired operating point is determined by the link's workload, i.e., by the traffic types that the link carries. By 'traffic type', we mean an aggregation of flows that have the same performance requirements. In order to perform provisioning, the network manager needs a profile for the link workload. This *workload profile* consists of the offered load and the QoS requirement for each traffic type. Such a profile is often available, based on operational data and statistics, in stable networks that are well monitored.

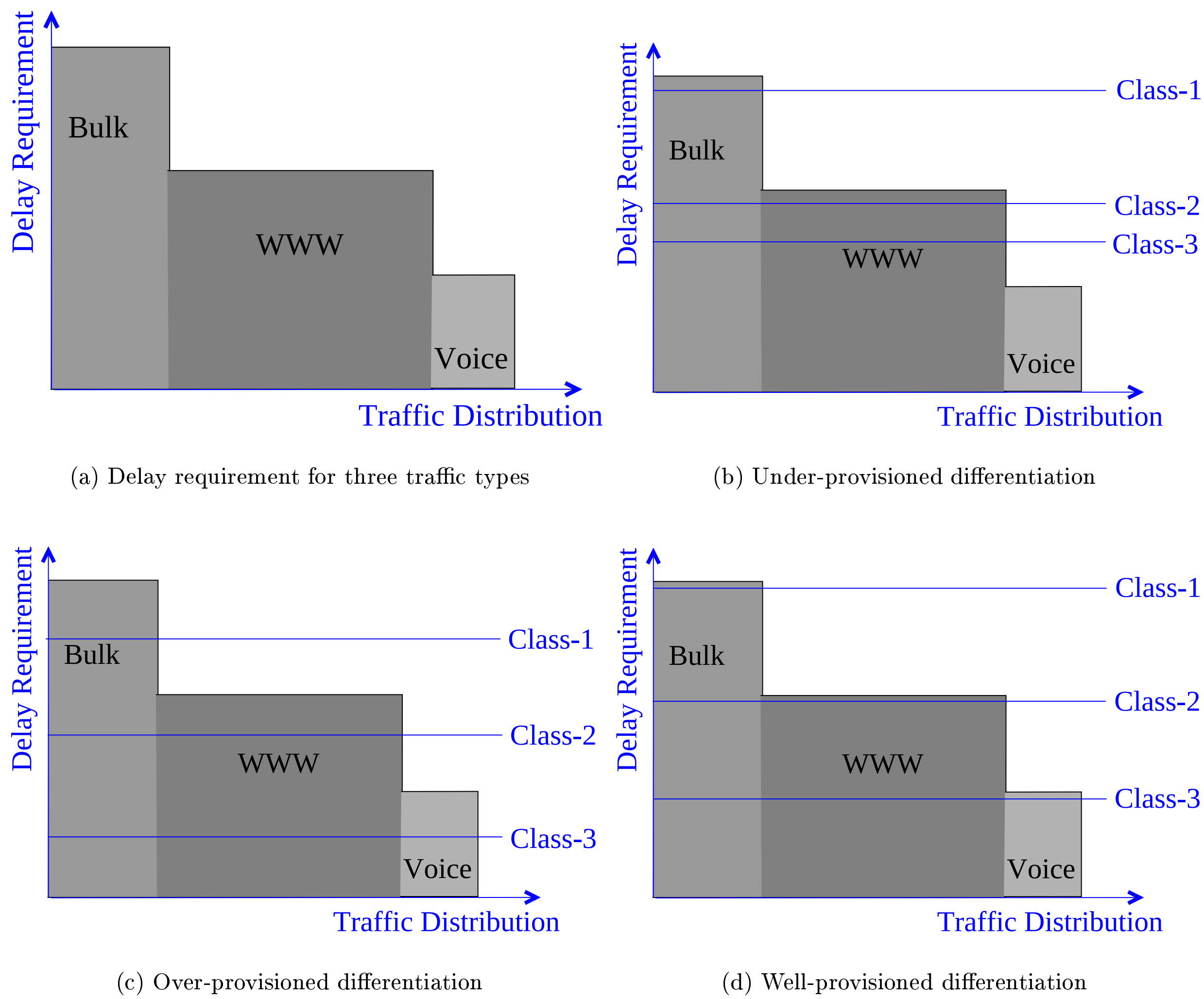
The exact form of link provisioning that we develop in this paper *can meet an average queueing delay requirement for each traffic type.* For example, a network provider can provision an average delay of 50msec for the E-mail, Network-News (NNTP), and other 'Bulk' traffic, 20msec for the WWW traffic, and 10 msec for the IP telephony and video conferencing traffic.

To illustrate the importance of class provisioning, let us consider the following simple example. Figure 1-a shows a *delay requirement curve* for three traffic types (Bulk, WWW, and Voice) at a certain link. In this example, about 25% of the link's traffic is Bulk transfers that can tolerate large delays, 60% is WWW flows with moderate delay requirements, and 15% is Voice, having low delay requirements. The link, in this example, offers three classes of service: Class-1, Class-2, and Class-3. It is noted that the number of classes may be different than the number of traffic types, and in practice it is likely that the traffic types will be more than the offered classes.

When the link is *under-provisioned*, one or more traffic types cannot meet their delay requirements even in the highest class of service. In Figure 1-b, Voice does not get an acceptable delay even in Class-3. A link can be under-provisioned either because it does not have an adequate amount of forwarding resources (capacity), or because the differentiation between classes (i.e., the delay spacing in this case) is not appropriately configured.

When the link is *over-provisioned*, an acceptable class exists for each traffic type, but the link may operate with more than the minimum required capacity. In Figure 1-c, Bulk meets its requirement in Class-1, WWW in Class-2, and Voice in Class-3. Notice, however, that each class offers a much lower delay that what the corresponding traffic type needs.

Finally, when the link is *well-provisioned*, an acceptable class exists for each traffic type, and additionally, the link operates with the minimum required capacity. In Figure 1-d, the link is well-provisioned when Bulk uses Class-1, WWW uses Class-2, and Voice uses Class-3. Such a *nominal class allocation* can be enforced by a network provider using an ingress classifier that operates based on the packet port numbers. Notice that each class provides (almost) the delay requirement of the corresponding traffic type. Also, none of the traffic types would be able to meet their delay requirement in a lower class.



(a) Delay requirement for three traffic types

(b) Under-provisioned differentiation

(c) Over-provisioned differentiation

(d) Well-provisioned differentiation

Figure 1. An under-provisioned, over-provisioned, and well-provisioned link with three classes and three traffic types.

3. LINK MODEL AND PDD SCHEDULING

Suppose that we provision a network link $\mathcal{L}$ that offers N *service classes* and carries M *traffic types*. A traffic type j is characterized by an average queueing delay requirement χ_j, and an average input rate ζ_j. Without loss of generality, the traffic types are ordered based on their delay requirements, so that $\chi_1 > \chi_2 > \ldots > \chi_M > 0$. For simplicity, we assume that all traffic types have the same average packet size $\bar{L}$, normalized as $\bar{L} = 1$. The set $\{(\chi_j, \zeta_j), j = 1 \ldots M\}$ is the input of the class provisioning methodology.

We assume that the network manager provisions $\mathcal{L}$ for lossless operation. This is a reasonable assumption, as most backbone providers today provision their links for lossless operation. The link capacity, which is an outcome of the class provisioning methodology, is denoted by C. The offered rate in class i is λ_i, while the aggregate offered rate is $\lambda = \sum_i^N \lambda_i = \sum_j^M \zeta_j$. The utilization is denoted by $u = \lambda/C$. Note that λ depends on the traffic type rates, and is constant for a given workload profile. The capacity, and thus the utilization, are variables, however, that are to be computed by the provisioning methodology.

The delay differentiation between classes in $\mathcal{L}$ follows the Proportional Delay Differentiation (PDD) model.[3] According to the PDD model, if $\bar{d}_i$ is the average queueing delay in class i, the ratios between class delays are fixed to:

$$\frac{\bar{d}_i}{\bar{d}_j} = \frac{\delta_i}{\delta_j} \qquad 1 \leq i, j \leq N \tag{1}$$

where δ_i are the Delay Differentiation Parameters (DDPs) ($\delta_1 = 1 > \delta_2 > \ldots > \delta_N > 0$). Notice that the PDD model consists of $N-1$ ratios of N class delays.

The packet scheduler in $\mathcal{L}$ is a work-conserving, non-preemptive, Proportional Delay Differentiation scheduler that can meet the PDD model, when the specified DDPs are feasible. Such schedulers have been the subject of recent research.[3–8]

The proposed class provisioning methodology consists of two parts. First, we determine the *target average delay* $\hat{v}_i$ and the corresponding *target offered rate* $\hat{h}_i$ for each class $i = 1 \ldots N$. The objective in the selection of the N pairs $\{(\hat{v}_i, \hat{h}_i), i = 1 \ldots N\}$ is that $\mathcal{L}$ *meets the average delay requirement* $\{\chi_j, j = 1, \ldots, M\}$ *of the M traffic types, with the minimum link capacity.* Second, we compute this minimum link capacity $\hat{C}$, and the required DDPs $\{\hat{\delta}_i, i = 2 \ldots N\}$.

4. CLASS OPERATING POINT (COP) SELECTION

We define a *Class Operating Point (COP)* as a vector $\mathbf{v} = \{v_1, \ldots, v_N\}$, such that $v_1 \geq v_2 \geq \ldots \geq v_N > 0$, where v_i is the desired (target) average delay in class i. A COP $\mathbf{v}$ is *acceptable* when for each traffic type j there exists at least one class i such that $v_i \leq \chi_j$. Let V be the set of acceptable COPs. If $\mathbf{v} \in V$, then for each traffic type j there exists a class $n(j) \in \{1 \ldots N\}$ such that $v_{n(j)} \leq \chi_j < v_{n(j)-1}$ ($v_0 = \infty$).

Given an acceptable COP $\mathbf{v}$, each traffic type j is assigned to class $n(j)$, since that is the *minimum class* that satisfies the delay requirement of j. We say that traffic type j is *mapped* to class $n(j)$, or that $n(j)$ is the *nominal class* for traffic type j. Note that when $M > N$ (which is probably the more practical case), there will be more than one traffic types mapped to some classes. Some classes, that are referred to as *void*, may not be nominal for any traffic type. To denote the inverse mapping, from classes to traffic types, $t(i)$ is the *maximum* traffic type that maps to class i; if class i is void, then $t(i)$=0.

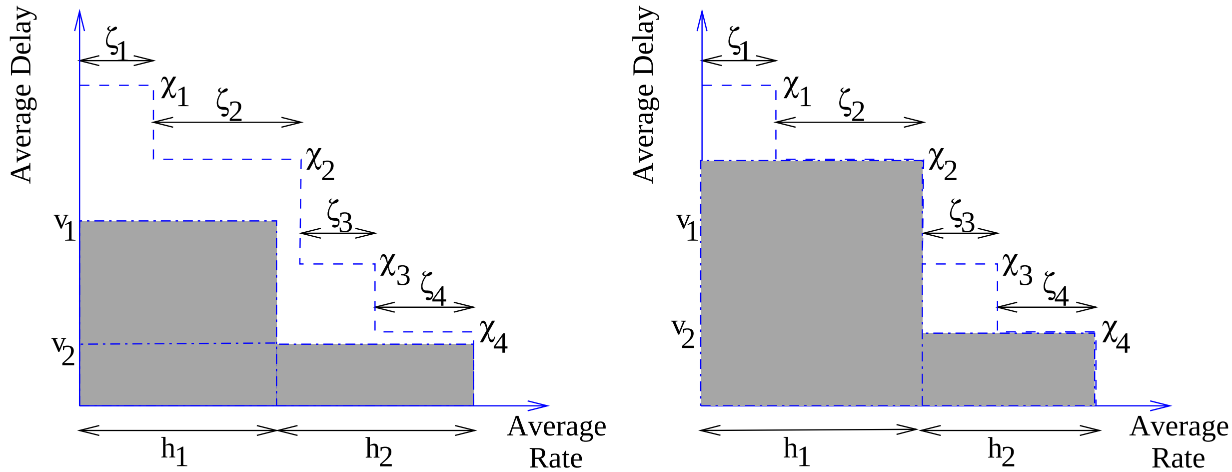
The *expected rate* h_i in class i is the aggregate rate of all traffic types that map to class i. Since an acceptable COP $\mathbf{v}$ determines the nominal class for each traffic type, the expected rates are a function of $\mathbf{v}$,

$$\mathbf{h}(\mathbf{v}) = \{h_1(\mathbf{v}), h_2(\mathbf{v}), \ldots, h_N(\mathbf{v})\} \quad \text{with} \quad h_i(\mathbf{v}) = \sum_{j:n(j)=i} \zeta_j \geq 0 \tag{2}$$

When the particular $\mathbf{v}$ that we consider is obvious, we write $\mathbf{h}$ or h_i, instead of $\mathbf{h}(\mathbf{v})$ or $h_i(\mathbf{v})$, respectively. The *total expected rate* in the link is

$$h = \sum_{i=1}^{N} h_i = \sum_{j=1}^{M} \zeta_j \tag{3}$$

that is independent of $\mathbf{v}$.



(a) An acceptable COP

(b) The optimal COP

Figure 2. An acceptable COP and the optimal COP for a link with N=2 classes and M=4 traffic types.

We say that an acceptable COP is *realized* if the average delay in each class i becomes $\bar{d}_i = v_i$, when the class rates are $\lambda_k = h_k$ for $k = 1 \ldots N$. The link capacity that is required for realizing $\mathbf{v}$ is called the *capacity requirement* of $\mathbf{v}$ and is denoted by $C(\mathbf{v})$. When $\mathbf{v}$ is realized, the aggregate backlog in $\mathcal{L}$ becomes

$$\bar{q}_{ag}(\mathbf{v}) = \sum_{i=1}^{N} \bar{d}_i \lambda_i = \sum_{i=1}^{N} v_i h_i \tag{4}$$

Note that the average backlog $\bar{q}_{ag}$ depends on the link utilization and the statistical characteristics of the traffic, and not on the scheduler or the class load distribution.[9]

If we want to compute the capacity requirement $C(\mathbf{v})$ of a COP $\mathbf{v}$, we need to know how the average backlog $\bar{q}_{ag}$ varies with the link utilization u. We refer to this relation as the *average backlog function* $\bar{q}_{ag} = \beta(u)$. $\beta(u)$ is assumed to be strictly increasing and convex when $u \in (0, 1)$, and it is unbounded as $u \to 1$. $\beta(u)$ is thus invertible, meaning that the link utilization u can be computed from the average backlog through the *inverse backlog function* $\beta^{-1}(\bar{q}_{ag})$. The problem of estimating the average backlog function is discussed in §6. Given the inverse backlog function, we can determine the capacity requirement of $\mathbf{v}$ from

$$C(\mathbf{v}) = \frac{h}{u(\mathbf{v})} = \frac{h}{\beta^{-1}(\bar{q}_{ag}(\mathbf{v}))} \tag{5}$$

where $u(\mathbf{v})$ is the link utilization that creates an average backlog $\bar{q}_{ag}(\mathbf{v})$.

An important part of the class provisioning methodology is to select the *optimal COP* $\hat{\mathbf{v}}$ among all acceptable COPs. The optimality constraint in the selection of $\hat{\mathbf{v}}$ is that *it has to be the acceptable COP with the minimum capacity requirement*,

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v} \in V} C(\mathbf{v}) \tag{6}$$

Since the average backlog function $\bar{q}_{ag} = \beta(u) = \beta(\lambda/C)$ is strictly increasing though, *the COP with the minimum capacity requirement is the COP with the maximum average backlog.* So, the optimal COP is the acceptable COP with the maximum average aggregate backlog,

$$\hat{\mathbf{v}} = \arg\max_{\mathbf{v} \in V} \bar{q}_{ag}(\mathbf{v}) \tag{7}$$

To determine $\hat{\mathbf{v}}$ in practice, we only need to consider a finite set of acceptable COPs. To see why, consider the example of Figure 2. The example refers to a link with N=2 classes and M=4 traffic types, and it shows two

```
optimal_cop (t_1, t_2, ..., t_{N-1}, i, max_q, best_cop)
{
// t_i: maximum traffic type (∈ {1...M}) that maps to class i.
// max_q and best_cop are call-by-reference arguments.
// Initially, call optimal_cop (0,0,...,0, 1, 0, ∅).
// The optimal COP v̂ is returned in the best_cop argument.
// avg_backlog() computes the backlog of a COP as in (4).
// Note: t_N=M and t_0 = 0.

    if (i ≤ N − 1) {
        for t_i = (t_{i−1} + 1)  to  (M − N + i)
            optimal_cop (t_1, t_2, ..., t_{N−1}, i + 1, max_q, best_cop);
    }
    else {
        q = avg_backlog (χ_{t_1}, χ_{t_2}, ..., χ_{t_{N−1}}, χ_{t_N});
        if (q > max_q) {
            max_q = q;
            best_cop = (χ_{t_1}, χ_{t_2}, ..., χ_{t_{N−1}}, χ_{t_N});
        }
    }
}
```

Figure 3. Algorithm to determine the optimal COP $\hat{\mathbf{v}}$.

acceptable COPs. In the COP of Figure 2-a, the first traffic type and a large part of the second traffic type are mapped to Class-1; the rest of the traffic is mapped to Class-2. Notice that the four traffic types meet their delay requirements with this class mapping, but there is some 'waste' of resources since the two classes provide lower delays than what the traffic types need.

In the COP of Figure 2-b, on the other hand, the average delay in each class is equal to the delay requirement of one of the traffic types. Specifically, the first two traffic types map to Class-1, which offers the delay requirement χ_2 of the second traffic type, while the two higher traffic types map to Class-4, which offers the delay requirement χ_4 of the fourth traffic type. Note that the shaded area in each COP represents the average backlog $\bar{q}_{ag}(\mathbf{v}) = \sum_{i=1}^{N} v_i h_i$. The optimal COP has to maximize the average backlog, and thus, to maximize the shaded area in Figure 2. In the previous example, the COP of Figure 2-b can be shown to be optimal.

Based on the graphical insight from the previous example, we can see that the optimal COP $\hat{\mathbf{v}}$ satisfies the following properties. First, *each optimal class delay $\hat{v}_i$ should be equal to the delay requirement of a traffic type*, i.e., for each $i = 1 \ldots N$ there is a $j \in \{1 \ldots M\}$ such that $\hat{v}_i = \chi_j$. Second, *the optimal COP should not have void classes*, because void classes always lead to an average backlog that is less than maximum. So, if $\hat{v}_i = \chi_j$, then there should be no other class k with $\hat{v}_k = \chi_j$. Third, following from the previous two properties, the target delay for the highest class should be the most stringent traffic type delay requirement, i.e., $\hat{v}_N = \chi_M$.

Putting the previous three properties together, we see that the finite set of acceptable COPs that should be examined in order to determine the optimal COP $\hat{\mathbf{v}}$ is

$$\{\mathbf{v} \in V : v_1 > v_2 > \ldots v_N,\ \forall i = 1 \ldots N, \exists j \in \{1 \ldots M\} \text{ such that } v_i = \chi_j \quad (v_N = \chi_M)\} \tag{8}$$

Note that the strict inequalities between the v_i's prevent the existence of void classes.

A recursive algorithm for selecting the optimal COP is shown in Figure 3. The run-time complexity of the algorithm is $\mathrm{O}\Big((M-N)^{N-1}\Big)$. For instance, in the case of N=3 classes and $M \geq 3$ traffic types, the algorithm examines $(M-N+1)(M-N+2)/2$ COPs. Since the provisioning methodology is performed off-line, and the number of classes and traffic types is expected to be relatively small (e.g., N=8, M=16), the run-time complexity of the algorithm is not prohibitive.

Example of optimal COP selection:
Suppose that a certain link supports N=2 classes and M=3 traffic types. We need to consider two COPs, depending on whether the maximum traffic type that maps to Class-1 is traffic type 1 or 2. Specifically, the two COPs are:

$$\mathbf{v_1} = (\chi_1, \chi_3) \text{ with } \mathbf{h}_1 = (\zeta_1, \zeta_2 + \zeta_3) \quad \text{and} \quad \mathbf{v_2} = (\chi_2, \chi_3) \text{ with } \mathbf{h}_2 = (\zeta_1 + \zeta_2, \zeta_3)$$

The average backlog in the two COPs is:

$$\bar{q}_{ag}(\mathbf{v_1}) = \chi_1\zeta_1 + \chi_3(\zeta_2 + \zeta_3) \quad \text{and} \quad \bar{q}_{ag}(\mathbf{v_2}) = \chi_2(\zeta_1 + \zeta_2) + \chi_3\zeta_3$$

Which COP has the maximum average backlog depends on the relation between the traffic type rates and average delay requirements. If $\zeta_1(\chi_1 - \chi_2) > \zeta_2(\chi_2 - \chi_3)$, then $\bar{q}_{ag}(\mathbf{v_1}) \geq \bar{q}_{ag}(\mathbf{v_2})$ and the optimal COP is $\mathbf{v_1}$; otherwise, the optimal COP is $\mathbf{v_2}$.

5. MINIMUM LINK CAPACITY AND DELAY DIFFERENTIATION PARAMETERS

In the first part of the class provisioning methodology, the goal was to determine the mapping from traffic types to classes that leads to the minimum capacity requirement. Given this optimal COP $\hat{\mathbf{v}}$ and the corresponding expected rate vector $\mathbf{h}(\hat{\mathbf{v}})$, the second part of the provisioning methodology determines the minimum capacity requirement and the required DDPs.

The minimum capacity requirement $\hat{C}$ can be computed using the inverse backlog function, as

$$\hat{C} = C(\hat{\mathbf{v}}) = \frac{h}{\beta^{-1}\left(\bar{q}_{ag}(\hat{\mathbf{v}})\right)} = \frac{h}{\beta^{-1}\left(\sum_{i=1}^{N} \hat{v}_i \hat{h}_i\right)} \tag{9}$$

where h is the total expected rate given by (3). The required DDPs, on the other hand, are simply the ratios of the corresponding optimal average class delays, i.e.,

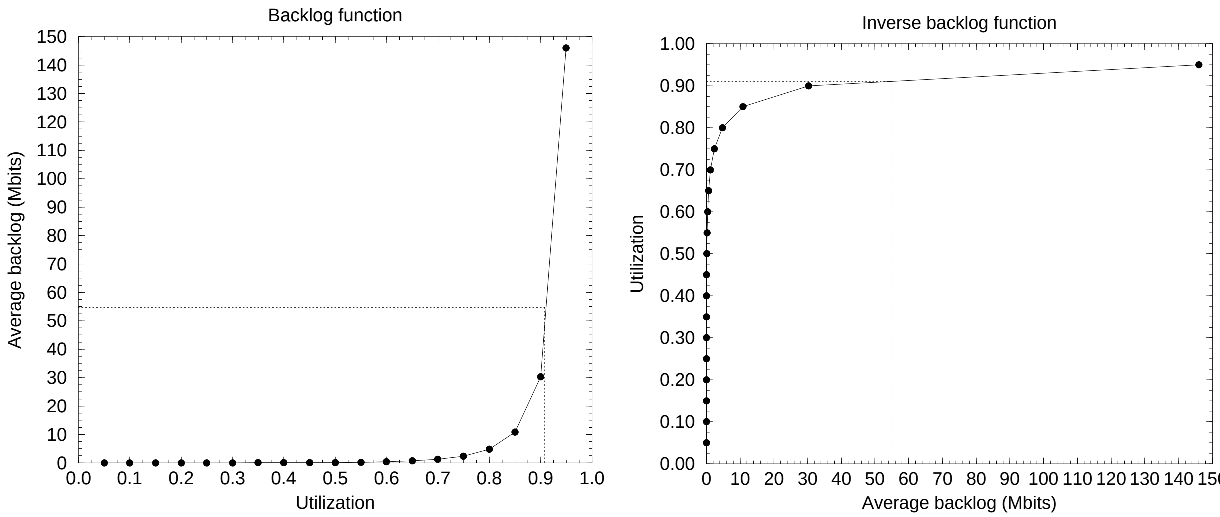
$$\frac{\hat{\delta}_i}{\hat{\delta}_1} = \frac{\hat{v}_i}{\hat{v}_1} \qquad i = 2 \ldots N \tag{10}$$

with $\hat{\delta}_1$=1.

Notice that these particular DDPs provide the $N-1$ delay ratios of the N class delays in the optimal COP. Without the appropriate link capacity, however, the absolute class delays will not be as in the optimal COP. Specifically, if the capacity is $C > \hat{C}$, it is easy to see that *all* class delays will be lower, i.e., $\bar{d}_i < \hat{v}_i$ for all i. This would be an instance of *over-provisioning*. On the other hand, if the capacity is $C < \hat{C}$, *all* class delays will be larger, i.e., $\bar{d}_i > \hat{v}_i$ for all i. That would be an instance of *under-provisioning*. In practice, there is also a *well-provisioned* operating region in which the capacity is $C \in (\hat{C}_-, \hat{C}_+)$, where $\hat{C}_+ = \hat{C}$ and $\hat{C}_- = f\hat{C}$, with f being a *tolerance factor* ($f < 1$). Such a tolerance factor is necessary because of uncertainties in the workload profile and in the estimation of the average backlog function.

Also note that, throughout the provisioning methodology, we did not have to compute explicit capacity shares for each class. That would be the case if we had used a *link sharing* scheduler, such as WFQ,[10] Class Based Queueing (CBQ),[11] or Hierarchical Packet Fair Queueing (H-PFQ).[12] Unfortunately, there is no straightforward approach to compute the $N-1$ weights of such schedulers in order to meet a certain average delay in each class. Additionally, a 'trial-and-error' approach would require searching in an $(N-1)$-dimensional space, making the approach impractical even for a small number of classes.

With PDD scheduling, on the other hand, we avoid the explicit computation of a capacity allocation between classes. A PDD scheduler services backlogged packets in an appropriate order for the given delay ratios to be met. Having fixed the $N-1$ delay ratios, the absolute delay in each of the N classes depends only on the link capacity. If the average backlog function is known, the calculation of $\hat{C}$ is straightforward. If the average backlog function is unknown, we can adjust the link capacity until the class delays become as in the optimal COP. Such a trial-and-error



(a) $q_{ag} = \beta(u)$ (b) $u = \beta^{-1}(q_{ag})$

Figure 4. The backlog and the inverse backlog functions for Pareto traffic with α=1.5.

approach is simpler, because there is only one 'knob' to vary, and the relation between the link capacity and the average class delays is monotonic.

Example of DDP and capacity calculations:
Suppose that a certain link offers N=4 classes, and that we need to meet the following COP:

$$\mathbf{v} = (40, 20, 10, 5)\text{msec} \quad \text{and} \quad \mathbf{h} = (0.5, 0.5, 2.0, 1.0)\text{kpps}$$

where *kpps* stands for 'kilo-packets-per-second'. If the average packet size is 1000 bytes, the total expected rate is $h = \sum_i h_i$=4kpps, or about 32Mbps. The problem is to determine the DDPs and the minimum link capacity required to realize this COP.

From (10), the required DDPs are

$$\delta_2 = \frac{20}{40} = 0.5 \quad \delta_3 = \frac{10}{40} = 0.25 \quad \delta_4 = \frac{5}{40} = 0.125$$

With this optimal COP, the average backlog is

$$\bar{q}_{ag} = \sum_i v_i h_i = 40 \times 0.5 + 20 \times 0.5 + 10 \times 2.0 + 5 \times 1.0 = 55 \text{ packets}$$

We can now use the inverse backlog function $u = \beta^{-1}(\bar{q}_{ag})$ to compute the required utilization u. In this example, suppose that the average backlog function (and its inverse) are as in Figure 4 (these curves are generated from simulating Pareto interarrivals with α=1.5). For $\bar{q}_{ag}$=55 packets we find that the utilization is $u = \beta^{-1}(55)$ ≈92.0%, and so the capacity requirement is C=h/u=4444pps, or about 35.6Mbps.

Simulating the link with the previous DDPs, with u=92.0%, and with a WTP scheduler,[3] we get that the class average delays are $(\bar{d}_1, \bar{d}_2, \bar{d}_3, \bar{d}_4) = (35.1, 17.5, 8.8, 4.4)$msec, that are slighly less than the given maximum average delays specified in the given COP. Because the backlog curve is quite steep in the heavy load range, however, slight variations in the utilization or in the expected class rates can violate the average delay requirements. For example, if the utilization is increased to u=94.0%, the average class delays become $(\bar{d}_1, \bar{d}_2, \bar{d}_3, \bar{d}_4) = (68.2, 34.1, 17.1, 8.6)$msec, that violate the average delay requirements. The large sensitivity of the capacity requirement in the heavy load range implies that the network operator *should* use some tolerance in the computation of C. Even if the network provider provisions the link with a higher capacity than $\hat{C}$, it is still useful to know $\hat{C}$ as a *lower bound* on the required link capacity.

6. THE AVERAGE BACKLOG FUNCTION

The calculation of the capacity requirement of a COP $\mathbf{v}$ can be performed if we know the average backlog $\bar{q}_{ag}$ as a function of the link utilization u. For simple queueing models, the function $\beta(u)$ is analytically known. For instance, in the $M|M|1$ system $\beta(u) = \frac{u^2}{1-u}$ packets, while in the $M|G|1$ system $\beta(u) = \frac{u^2}{1-u}\frac{1+c_L^2}{2}$, where c_L is the coefficient of variation of the packet size distribution.[9] In the very general $G|G|1$ system, the average backlog can be approximated by the Allen-Cunneen formula $\beta(u) \approx \frac{u^2}{1-u}\frac{c_A^2+c_L^2}{2}$,[13] where c_A is the coefficient of variation of the distribution of interarrivals.

A practical alternative, instead of relying on queueing models, is to *measure* the function $\beta(u)$ directly on the router, by monitoring the actual backlog in the link. The network operator, in that case, would need to record the average backlog in different link utilizations. If the underlying traffic dynamics are stationary, it would be possible to extract an empirical curve for the average backlog function.

It is noted that the backlog function may not only depend on the utilization u, but also on the capacity C. This can occur if the statistical properties of the traffic (traffic burstiness) depend on C. For instance, with the same utilization, an OC-3 link (155Mbps) may have a larger backlog than a T-1 link (1.5Mbps), because higher capacity links attract more bursty traffic in general. In a relatively narrow range of C though, it is reasonable to assume that the traffic burstiness remains invariant, and that the backlog function depends on u, but not on C.

7. DISCUSSION AND OPEN ISSUES

The proposed class provisioning methodology is effective as long as the workload profile is valid. If the traffic types have larger arrival rates, or if the average backlog function is not accurately estimated, some traffic types may not be adequately supported. Similar problems can arise due to dynamic routing changes, link or router failures, or unexpected increases in the traffic demand. In those cases, the link may not operate in its provisioned operating point. The PDD differentiation, however, will still provide a controllable and predictable *relative differentiation* between classes, even though the absolute QoS of each class will not be known.

The class provisioning methodology can be performed over relatively long timescales (say weeks or months), depending on how simple it is to adjust the link capacity. It is noted though that it gradually becomes simpler to adjust the capacity of a link even in a few minutes or seconds, through the use of Wavelength-Division-Multiplexing (WDM).[14] Using such technologies, an ISP can lease the capacity of an additional 'wavelength' from the backbone provider that owns the network fibers, when a larger traffic demand is anticipated or encountered. Also, if the characterization of traffic types on a certain link follows different patterns through the day (e.g., many IP-telephony sessions through the day and mostly WWW sessions in the evening), the network operator can perform class provisioning for the different traffic patterns, and operate the link with a schedule of different capacities and DDPs during the day.

It will be interesting to extend this class provisioning methodology in the case of coupled delay and loss differentiation. The Proportional Loss Differentiation model,[15] with the corresponding packet droppers PLR(∞) and PLR(M), can be used to differentiate the loss rates between classes. It is not clear, though, how to jointly provision a link for a per-class maximum average delay *and* a maximum loss rate at the same time.

Another open issue is to examine the effectiveness of PDD scheduling compared to other differentiation models, such as the link sharing schedulers (e.g., WFQ). Is PDD the *optimal* differentiation model for the provisioning of average class delays? The optimality criterion here is: can a PDD scheduler provide a certain set of target average class delays with the minimum possible capacity among all work-conserving and non-preemptive schedulers? Some work in this direction has been done in the context of loss rate provisioning using PLD droppers. Specifically, Yang and Pan showed that the PLR(∞) dropper, jointly with a FCFS scheduler, is optimal, in the sense that it requires *the minimum capacity for a certain loss rate in each class.*[16]

REFERENCES

1. 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.
2. C. Dovrolis and P.Ramanathan, “A Case for Relative Differentiated Services and the Proportional Differentiation Model,” *IEEE Network* , Oct. 1999.
3. C. Dovrolis, D.Stiliadis, and P.Ramanathan, “Proportional Differentiated Services: Delay Differentiation and Packet Scheduling,” in *ACM SIGCOMM*, Sept. 1999.
4. C. C. Li, S.-L. Tsao, M. C. Chen, Y. Sun, and Y.-M. Huang, “Proportional Delay Differentiation Service Based on Weighted Fair Queueing,” in *Proceedings IEEE International Conference on Computer Communications and Networks (ICCCN)*, Oct. 2000.
5. M. K. H. Leung, J. Lui, and D. Yau, “Characterization and Performance Evaluation for Proportional Delay Differentiated Services,” in *Proceedings International Conference on Network Protocols (ICNP)*, Oct. 2000.
6. Y. Moret and S.Fdida, “A Proportional Queue Control Mechanism to Provide Differentiated Services,” in *International Symposium on Computer and Information Systems (ISCIS)*, Oct. 1998.
7. J. Liebeherr and N.Christin, “Buffer Management and Scheduling for Enhanced Differentiated Services,” in *Proceedings IWQoS*, June 2001.
8. C. Dovrolis, D. Stiliadis, and P. Ramanathan, “Proportional Differentiated Services: Delay Differentiation and Packet Scheduling,” tech. rep., University of Delaware, Apr. 2001. Submitted for publication to the IEEE/ACM Transactions on Networking.
9. G. Bolch, S.Greiner, H.Meer, and K.S.Trivedi, *Queueing Networks and Markov Chains*, John Wiley and Sons, 1999.
10. A. Demers, S.Keshav, and S.Shenker, “Analysis and Simulation of a Fair Queueing Algorithm,” in *Internetworking: Research and Experience*, pp. 3–26, 1990.
11. S. Floyd and V. Jacobson, “Link-Sharing and Resource Management Models for Packet Networks,” *IEEE/ACM Transactions on Networking* **3**, pp. 365–386, Aug. 1995.
12. J. Bennett and H.Zhang, “Hierarchical Packet Fair Queueing Algorithms,” *IEEE/ACM Transactions on Networking* **5**, pp. 675–689, Oct. 1997.
13. O. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications*, Academic Press, 2nd edition, 1990.
14. B. Rajagopalan, D. Pendarakis, D. Saha, R. Ramamoorthy, and K. Bala, “IP over Optical Networks: Architectural Aspects,” *IEEE Communications Magazine* , pp. 94–102, Sept. 2000.
15. C. Dovrolis and P.Ramanathan, “Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping,” in *IEEE/IFIP International Workshop on Quality of Service (IWQoS)*, June 2000.
16. T. Yang and J.Pan, “A Measurement-Based Loss Scheduling Scheme,” in *Proceedings INFOCOM*, 1996.