Computing the loss differentiation parameters of the proportional differentiation service model

J. Zeng, L. Zakrevski and N. Ansari

Abstract: The proportional differentiation service model has emerged as a refined version of the DiffServ qualify of service (QoS) architecture. It relies on a series of parameters to enforce proportionally differentiated QoS criteria, such as queueing delay and packet loss. From the perspective of proportional loss differentiation, a large amount of work has been done on carrying out loss differentiation based on given parameters. Under certain network conditions, however, the loss differentiation cannot be met based on these hand picked parameters. While existing work focuses on enhancing dropping mechanisms themselves to honour the loss differentiation, the paper looks into calculating feasible differentiation parameters. By forming an optimisation problem based on multiple class blocking thresholds, the paper introduces a simple quantitative guideline to compute loss differentiation parameters. Derived closely related to network statuses and packet dropping mechanisms, these parameters ease the difficulty that dropping mechanisms may encounter when enforcing packet loss differentiation. Its finite computation time, moreover, makes practical implementation possible. Analytical and numerical results are given to substantiate the new approach and its merits.

1 Background

As compared to integrated service (IntServ), the differentiated service (DiffServ) model defines an architecture for implementing relative, scalable service differentiation in the Internet. It achieves scalability by applying per-hop behaviours (PHBs) to traffic aggregates that have been marked using the differentiated services (DS) field in Internet protocol (IP) headers [1].

The proportional differentiation service model was suggested to engineer relative quality of service (QoS). This model groups the network traffic into n classes; the service of class i is better or at least no worse than that of class $i-1$ for $1<i\leq n$, in terms of per-hop QoS metrics such as queueing delay and packet loss. It defines finer QoS differences among classes, and thus provides better differentiation granularity than the original DiffServ model with expedite forwarding (EF), assured forwarding (AF), and best effort (BE) classes.

From the packet loss perspective [2], the proportional differentiation model states that per-hop packet losses should be enforced to be proportional to the corresponding differentiation parameters chosen by network engineers, such that $l_i/l_j = \sigma_i/\sigma_j$, $1\leq i, j\leq n$, where l_i are loss related measures, and σ_i are loss differentiation parameters, ordered as $\sigma_1>\sigma_2>\sigma_3>\cdots>\sigma_n>0$. With these parameters quantifying the desired proportional differentiation, the fundamental idea of the model is to equalise the loss measures normalised by corresponding differentiation parameters, i.e., $l_i/\sigma_i = l_j/\sigma_j$, $1\leq i, j\leq n$.

The proportional loss differentiation model has been used and investigated in a number of networking scenarios [3–10], such as active buffer management, TCP throughput issue, and optical switching. Given loss differentiation parameters and involved performance criteria, diverse mechanisms have been proposed to enforce the performance differentiation among users or classified groups. These differentiation parameters are often chosen according to service pricing rules or network engineers' experiences. The loss differentiation, however, cannot be always met [2, 7] owing to its unawareness of network statuses, such as traffic load, buffer backlog, etc. Accordingly, it is difficult for network operators to know *a priori* if the chosen constraints are feasible. Coupled with the delay differentiation, this problem has been studied in existing differentiation mechanisms; for instance, dynamic class selection [11] where a user searches an appropriate class for its performance criteria; the feedback control plus delay prediction [6] that ensures the QoS algorithm to realise the desired service differentiation.

Instead of solely enforcing loss differentiation based on given parameters σ_i, $i = 1, 2, \ldots, n$, nevertheless, network operators shall benefit from a guideline on selecting these differentiation parameters. A set of parameters derived from network-related information can certainly increase the likelihood that the differentiation will eventually be enforced. Targeting for feasible differentiation from another angle, this paper then introduces a simple quantitative guideline to compute loss differentiation parameters.

2 System model and the new approach

In order to support n service classes, a buffer/queue accommodates a first-in-first-out (FIFO) module that determines which packet shall be served next, and a dropping module that decides when and which packets to

IEE Proceedings online no. 20045286

doi:10.1049/ip-com:20045286

Paper first received 20th May 2004 and in final revised form 29th September 2005

The authors are with Advanced Networking Laboratory, Electrical and Computer Engineering Department, New Jersey Institute of Technology, Newark, New Jersey, 07102, USA

E-mail: nirwan.ansari@njit.edu

be dropped. Internet service providers (ISPs) are then challenged to maintain differentiated class losses, in conformance with the differentiation parameters. The selection of these parameters should not be arbitrary. A guideline to determine these loss differentiation parameters is, therefore, called for.

In line with the shared buffer implementation in practice, a single server queue is assumed for the discussion in the rest of the paper. Blocking thresholds $N_1, N_2, \ldots N_n$ are adopted to distinguish service priorities. When the content of the queue reaches N_i, $i = 1, 2, \ldots, n$, the dropping module starts to block traffic from class i. Obviously, these blocking thresholds function similarly to loss differentiation parameters; both of them intend to differentiate losses among classes.

Assume that each class has a Poisson arrival with the mean rate of λ_i, $i = 1, 2, \ldots, n$, the system service rate is μ, and the queue size is m. Accordingly, the packet service times are exponentially distributed. Based on the blocking policy mentioned above, the state-transition-rate diagram of this $M/M/1$ system is depicted in Fig. 1. The probability that the content of the queue reaches k is then expressed as follows

$$p_k = \begin{cases} \rho_{1,\ldots,n}^{k} * p_0 & 0 \le k \le N_1, \\ \vdots & \\ \rho_{i+2,\ldots,n}^{k-N_{i+1}} * \ldots * \rho_{i+1,\ldots,n}^{N_{i+1}-N_i} * \ldots * \rho_{1,\ldots,n}^{N_1} * p_0 & N_i \le k \le N_{i+1}, \\ \vdots & \\ \rho_{n}^{k-N_{n-1}} * \rho_{n-1,n}^{N_{n-1}-N_{n-2}} * \ldots * \rho_{i+1,\ldots,n}^{N_{i+1}-N_i} & N_{n-1} \le k \le m, \\ * \ldots * \rho_{2,\ldots,n}^{N_2-N_1} * \rho_{1,\ldots,n}^{N_1} * p_0 & \end{cases} \tag{1}$$

where $\rho_{i,\ldots,n} = (\lambda_i + \cdots + \lambda_n)/\mu$, $i = 1, 2, \ldots, n-1$, and $\rho_n = \lambda_n/\mu$. Solving for p_0 from (1), we have

$$p_0 = \left[\frac{1 - \rho_{1,\ldots,n}^{N_1}}{1 - \rho_{1,\ldots,n} + \rho_{1,\ldots,n}^{N_1}} \frac{1 - \rho_{2,\ldots,n}^{N_2-N_1}}{1 - \rho_{2,\ldots,n}} + \cdots + \rho_{1,\ldots,n}^{N_1} \right.$$
$$* \rho_{2,\ldots,n}^{N_2-N_1} * \ldots * \rho_{i,\ldots,n}^{N_i-N_{i-1}} \frac{1 - \rho_{i+1,\ldots,n}^{N_{i+1}-N_i}}{1 - \rho_{i+1,\ldots,n}}$$
$$\left. + \rho_{1,\ldots,n}^{N_1} * \rho_{2,\ldots,n}^{N_2-N_1} * \ldots * \frac{1 - \rho_n^{M-N_{n-1}+1}}{1 - \rho_n} \right]^{-1} \tag{2}$$

Subsequently, the blocking probability of each class r_i, $i = 1, 2, \ldots, n$, can be calculated.

The above procedure shows that the blocking thresholds lead to class blocking probabilities that are actual and demonstrable values. These blocking thresholds, therefore, can be used as a good reference on selecting the differentiation parameters. To connect the blocking thresholds to loss differentiation parameters, however, a certain relationship between them is sought.

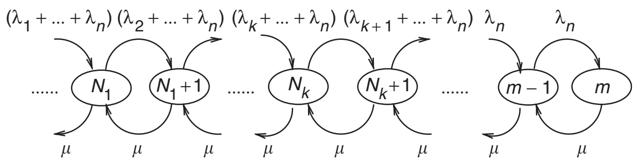

Fig. 1 *The state-transition-rate diagram*

With known class blocking probabilities r_i, $i = 1, 2, \ldots, n$, an optimisation problem minimising the system blocking probability weighted by the differentiation parameters σ_i, $i = 1, 2, \ldots, n$, is then formulated as follows

$$\min_{N_1, N_2, \ldots, N_{n-1}} (\sigma_1 r_1 + \sigma_2 r_2 + \cdots + \sigma_n r_n)$$

subject to the constraints

$$1 = \sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_n > 0,$$
$$0 \le N_1 \le N_2 \le \cdots \le N_n = m,$$
$$\sigma_1, \sigma_2, \ldots, \sigma_n \text{ are real numbers},$$
$$N_1, N_2, \ldots, N_n \text{ are integers}$$

Though a large amount of work has been done on analysing various loss systems such as the reciprocity of blocking probabilities [12] and retry blocking probabilities [13], to the best of our knowledge, none has addressed the optimisation problem discussed here. Various properties and analyses presented in this paper are thus derived based on the fundamental results of the queueing theory [14]. As a result, the blocking thresholds N_i, $i = 1, 2, \ldots, n$, and the differentiation parameters σ_i, $i = 1, 2, \ldots, n$, are coupled together.

3 Analysis results

The previously formulated optimisation problem is first tackled for the two-class scenario. Its resulting solution is then accordingly extended to the n-class scenario.

3.1 The two-class scenario

From (1) and (2), the blocking probabilities of class 1 and class 2 are

$$r_1 = \frac{1 - \rho_2^{m-N_1+1}}{1 - \rho_2} \rho_{1,2}^{N_1} \left[\frac{1 - \rho_{1,2}^{N_1}}{1 - \rho_{1,2}} + \rho_{1,2}^{N_1} \frac{1 - \rho_2^{M-N_1+1}}{1 - \rho_2} \right]^{-1}$$

$$r_2 = \rho_2^{M-N_1} \rho_{1,2}^{N_1} \left[\frac{1 - \rho_{1,2}^{N_1}}{1 - \rho_{1,2}} + \rho_{1,2}^{N_1} \frac{1 - \rho_2^{M-N_1+1}}{1 - \rho_2} \right]^{-1}$$

Since there are only a few variables for the two-class scenario, the optimisation problem can be simplified as follows

$$\min_{N_1, N_2} (\sigma_1 r_1 + \sigma_2 r_2)$$

subject to the constraints

$$1 = \sigma_1 \ge \sigma_2 > 0,$$
$$0 \le N_1 \le N_2 = m,$$
$$\sigma_1, \sigma_2 \text{ are real numbers},$$
$$N_1, N_2 \text{ are integers}$$

Since the algebraic solution is not straightforward, numerical computation is proposed to solve this optimisation problem. To reasonably reduce the number of variables involved in the computation, the blocking threshold of class 2 is set to be equal to the queue size m ($N_2 = m$), and the overall system load is normalised to one (it corresponds to the scenario where the server is busy, that is, $\rho_{1,2} = \rho_1 + \rho_2 = 0.999 \rightarrow 1$). A tractable three-dimensional figure that depicts the relationship among the blocking threshold N_1, the differentiation parameter σ_2, and the utilisation factor ρ_2 is thus plotted in Fig. 2. The points where the minimum system blocking probability is achieved are aggregated in three ladders, and each ladder is associated with a blocking threshold value. To derive a formula to show this relationship, two equations illustrating the boundary curves between three ladders are expressed as follows

$$\begin{aligned} F_{\sigma_2, N_1=40}(\rho_2) &= F_{\sigma_2, N_1=39}(\rho_2) \\ F_{\sigma_2, N_1=39}(\rho_2) &= F_{\sigma_2, N_1=38}(\rho_2) \end{aligned} \tag{3}$$

Expanding (3), we have

$$\frac{\rho_{1,2} * \left[\dfrac{1-\rho_2^1}{1-\rho_2} + \sigma_2\rho_2^0\right]}{\dfrac{1-\rho_{1,2}^{40}}{1-\rho_{1,2}} + \rho_{1,2}^{40} * \dfrac{1-\rho_2^1}{1-\rho_2}} = \frac{\dfrac{1-\rho_2^2}{1-\rho_2} + \sigma_2\rho_2^1}{\dfrac{1-\rho_{1,2}^{39}}{1-\rho_{1,2}} + \rho_{1,2}^{39} * \dfrac{1-\rho_2^2}{1-\rho_2}}$$

$$\frac{\rho_{1,2} * \left[\dfrac{1-\rho_2^2}{1-\rho_2} + \sigma_2\rho_2^1\right]}{\dfrac{1-\rho_{1,2}^{39}}{1-\rho_{1,2}} + \rho_{1,2}^{39} * \dfrac{1-\rho_2^2}{1-\rho_2}} = \frac{\dfrac{1-\rho_2^3}{1-\rho_2} + \sigma_2\rho_2^2}{\dfrac{1-\rho_{1,2}^{38}}{1-\rho_{1,2}} + \rho_{1,2}^{38} * \dfrac{1-\rho_2^3}{1-\rho_2}},$$

and obtain the following two curves

$$G_{\sigma_2, N_1=(40,39)}(\rho_2) = \frac{-A_{40}(1+\rho_2) + A_{39}\rho_{12}}{(A_{40}+B_{40}) * \rho_2^1 - [A_{39} + B_{39}(1+\rho_2)] * \rho_{1,2}},$$

$$G_{\sigma_2, N_1=(39,38)}(\rho_2) = \frac{-A_{39}(1+\rho_2+\rho_2^2) + A_{38}\rho_{12}(1+\rho_2)}{[A_{39}+B_{39}(1-\rho_2)] * \rho_2^2 - [A_{38} + B_{38}(1+\rho_2+\rho_2^2)] * \rho_{1,2} * \rho_2^1},$$

where $A_{40} = (1-\rho_{1,2}^{40})/(1-\rho_{1,2})$, $B_{40} = \rho_{1,2}^{40}$, $i = 1, 2, \ldots, n$, and $\rho_{1,2} = 0.99$. Together, these curves form a contour of the utilisation factor ρ_2 and the differentiation parameter σ_2 as depicted in Fig. 3.

To further incorporate the information of the blocking thresholds, or equivalently the queue size m, into the formulation, we extend the investigation to boundary curves among other possible ladders. In fact, with a queue

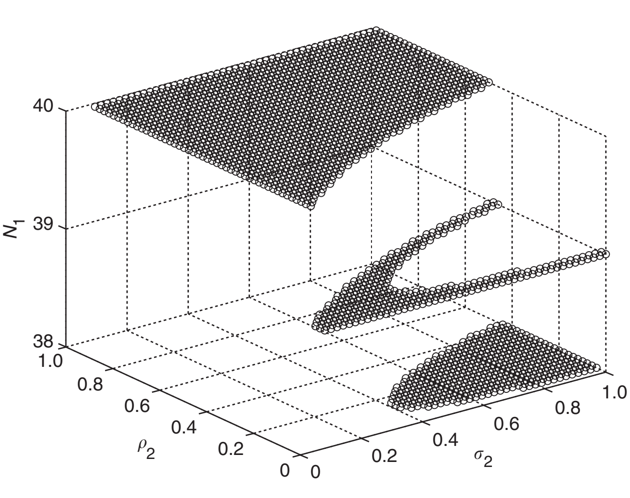

Fig. 2 *The relationship of the blocking threshold N_1, the differentiation parameter σ_2, and the utilisation factor ρ_2, where queue size $m = 40$*

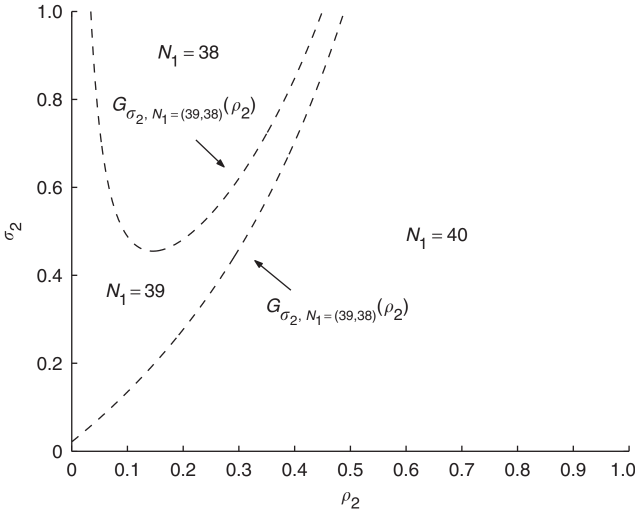

Fig. 3 *The contour of the differentiation parameter σ_2 and the utilisation factor ρ_2 from the 3-D plot in Fig. 2*

size $m = 80$, the resulting 3-D figure consists of four ladders, as compared to the previous three ladders in Fig. 3. By counting the ladders in the direction of decreasing blocking threshold N_1, the curve formula between ladder j and $j-1$, $j = 0, 1, \ldots, m-1$, that is, $G_{\sigma_2, N_2=(m-j,m-j-1)}(\rho_2)$, is induced as

$$\begin{aligned} G_{\sigma_2, N_2=(m-j,m-j-1)}(\rho_2) &= \frac{-\left(A_{m-j} + B_{m-j}\sum_{k=0}^{j}\rho_2^k\right) * \sum_{k=0}^{j+1}\rho_2^k + \left[A_{m-j-1} + B_{m-j-1}\sum_{k=0}^{j+1}\rho_2^k\right] * \rho_{1,2}\sum_{k=0}^{j}\rho_2^k}{\left(A_{m-j} + B_{m-j}\sum_{k=0}^{j}\rho_2^k\right) * \rho_2^{j+1} - \left[A_{m-j-1} + B_{m-j-1}\sum_{k=0}^{j+1}\rho_2^k\right] * \rho_{1,2} * \rho_2^j} \\ &= \frac{-A_{m-j}\sum_{k=0}^{j+1}\rho_2^k + A_{m-j-1} * \rho_{1,2}\sum_{k=0}^{j}\rho_2^k}{\left(A_{m-j} + B_{m-j}\sum_{k=0}^{j}\rho_2^k\right) * \rho_2^{j+1} - \left[A_{m-j-1} + B_{m-j-1}\sum_{k=0}^{j+1}\rho_2^k\right] * \rho_{1,2} * \rho_2^j} \end{aligned} \tag{4}$$

As stated in the following lemma, (4) has a lower bound. When the ladder index j increases, that is, the blocking threshold N_1 decreases, this lower bound increases. Since the differentiation parameter $\sigma_2 < 1$, when $G_{\sigma_2, N_1=(m-j,m-j-1)}(\rho_2) > 1$, the resulting curve will fall out of the contour; this is where the search for the curves in the contour stops.

Lemma 1: The general boundary equation as shown in (4) has a lower bound that is bigger than zero. Moreover, when the ladder index j increases, that is, the blocking threshold N_1 decreases, this lower bound increases.

Proof: Knowing that $1<A_j<A_m$, $0<B_m<B_j<1$, $0<\rho_2^{j+1}<\rho_2^j<1$, and $\rho_2^m \leq \rho_2^{m-j} \leq (m-j+1)\ \rho_2^{m-j} \leq \sum \leq_{k=0}^{m-j} m-j+1 \leq m$, $j = 1, \ldots, m$, the lower bound of (4) is obtained as follows

$$
\begin{aligned}
&G_{\sigma_2, N_2=(m-j, m-j-1)}(\rho_2) \\
&= \frac{A_{m-j} \sum_{k=0}^{j+1} \rho_2^k - A_{m-j-1} * \rho_{1,2} \sum_{k=0}^{j} \rho_2^k}{-\left(A_{m-j} + B_{m-j} \sum_{k=0}^{j} \rho_2^k\right) * \rho_2^{j+1} + \left(A_{m-j-1} + B_{m-j-1} \sum_{k=0}^{j+1} \rho_2^k\right) * \rho_{1,2} * \rho_2^j} \\
&\geq \frac{(A_{m-j} - A_{m-j-1} * \rho_{1,2}) \sum_{k=0}^{j} \rho_2^k}{-\left(A_{m-j} + B_{m-j} \sum_{k=0}^{j} \rho_2^k\right) * \rho_2^{j+1} + \left(A_{m-j-1} + B_{m-j-1} \sum_{k=0}^{j+1} \rho_2^k\right) * \rho_{1,2} * \rho_2^{j+1}} \\
&= \frac{\sum_{k=0}^{j} \rho_2^k}{\rho_2^{j+1}\left[-\left(A_{m-j} + B_{m-j} \sum_{k=0}^{j} \rho_2^k\right) + \left(A_{m-j-1} + B_{m-j-1} \sum_{k=0}^{j+1} \rho_2^k\right) * \rho_{1,2}\right]} \\
&\geq \frac{\sum_{k=0}^{j} \rho_2^k}{\rho_2^{j+1}[-(1 + B_m * \rho_2^m) + (A_m + m) * \rho_{1,2}]} > 0
\end{aligned}
\tag{5}
$$

As observed in (5), when the search procedure continues, that is, the ladder index j increases, $\sum_{k=0}^{j} \rho_2^k$ increases and ρ_2^{j+1} decerases; subsequently, the value of this lower bound increases with respect to j. Since the differentiation parameter $\sigma_2 < 1$, when $G_{\sigma_2, N_1=(m-j, m-j-1)}(\rho_2) > 1$, the resulting curve will fall out of the contour; it is the stopping point of the curve search. Therefore, for any queue size m, all the curves in the contour can be obtained from (4). Given such a contour, the value range of the differentiation parameter σ_2 can be drawn directly, based on the known utilisation factor ρ_2 and the appropriately chosen blocking threshold N_2.

3.2 The n-class scenario

To directly apply the previous results to the n-class scenario is not a trivial task, owing to the dramatic increase in the number of variables, that is, differentiation parameters σ_i, blocking thresholds N_i, and utilisation factors ρ_i. The previous analysis is thus applied recursively. First, classes $1, 2, \ldots, n-1$, are aggregated into one class. Applying the results for the two-class scenario to this class aggregate and class n, the contour showing the relationship between ρ_n, σ_n, and N_{n-1} consists of the following curves

$$
G_{\sigma_n, N_{n-1}=(m, m-1)}(\rho_n) = \frac{-A_m * \sum_{k=0}^{1} \rho_n^k + A_{m-1} * \rho_1, \ldots, n \sum_{k=0}^{1} \rho_n^k}{(A_m + B_m) * \rho_n^1 - \left(A_{m-1} + B_{m-1} \sum_{k=0}^{1} \rho_n^k\right) * \rho_{1,\ldots,n}}
$$

$$
G_{\sigma_n, N_{n-1}=(m-1, m-2)}(\rho_n) = \frac{-A_{m-1} * \sum_{k=0}^{2} \rho_n^k + A_{m-2} * \rho_1, \ldots, n \sum_{k=0}^{1} \rho_n^k}{\left(A_{m-1} + B_{m-1} \sum_{k=0}^{1} \rho_n^k\right) * \rho_n^2 - \left(A_{m-2} + B_{m-2} \sum_{k=0}^{2} \rho_n^k\right) * \rho_{1,\ldots,n} * \rho_n}
$$

$\vdots$

$$
G_{\sigma_n, N_{n-1}=(m-j, m-j-1)}(\rho_n) = \frac{-A_{m-j} * \sum_{k=0}^{j+1} \rho_n^k + A_{m-j-1} * \rho_1, \ldots, n \sum_{k=0}^{j} \rho_n^k}{\left(A_{m-j} + B_{m-j} \sum_{k=0}^{j} \rho_n^k\right) * \rho_n^{j+1} - \left(A_{m-j-1} + B_{m-j-1} \sum_{k=0}^{j+1} \rho_n^k\right) * \rho_{1,\ldots,n} * \rho_n^j}
$$

where $G_{\sigma_n, N_{n-1}=(m-j, m-j-1)}(\rho_n) > 1$. With the known ρ_n and a selected blocking threshold N_{n-1}, the value range for the differentiation parameter σ_n can be drawn from the contour.

Next, class n is excluded from the system. By aggregating classes $1, 2, \ldots$, and $n-2$, there are again two classes, that is, this aggregate and class $n-1$, in the system. The contour of ρ_{n-1}, σ_{n-1}, and N_{n-2} is then solved as

$$
G_{\sigma_{n-1}, N_{n-2}=(N_{n-1}, N_{n-1}-1)}(\rho_{n-1}) = \frac{-A_{N_{n-1}} * \sum_{k=0}^{1} \rho_{n-1}^k + A_{N_{n-1}-1} * \rho_{1,\ldots,(n-1)}}{(A_{N_{n-1}} + B_{N_{n-1}}) * \rho_{n-1}^1 - \left(A_{N_{n-1}-1} + B_{N_{n-1}-1} \sum_{k=0}^{j} \rho_{n-1}^k\right) * \rho_{1,\ldots,(n-1)}}
$$

$\vdots$

$$
G_{\sigma_{n-1}, N_{n-2}=(N_{n-1}-j, N_{n-1}-j-1)}(\rho_{n-1}) = \frac{-A_{N_{n-1}} * \sum_{k=0}^{j+1} \rho_{n-1}^k + A_{N_{n-1}-j-1} * \rho_{1,\ldots,(n-1)} \sum_{k=0}^{j} \rho_{n-1}^k}{\left(A_{N_{n-1}} + B_{N_{n-1}} \sum_{k=0}^{j} \rho_{n-1}^k\right) * \rho_{n-1}^{j+1} - \left(A_{N_{n-1}-j-1} + B_{N_{n-1}-j-1} \sum_{k=0}^{j+1} \rho_{n-1}^k\right) * \rho_{1,\ldots,(n-1)} * \rho_{n-1}^j}
$$

where $G_{\sigma_{n-1}, N_{n-2}=(N_{n-1}-j, N_{n-1}-j-1)}(\rho_{n-1}) > 1$. Note that in this iteration, the queue size has been updated by the previously chosen blocking threshold N_{n-1}, and the utilisation factor after excluding that of class n becomes $\rho_{1,2,\ldots,n-1} \to 1 - \rho_n$. Again, with the known ρ_{n-1} and an appropriate blocking threshold N_{n-2} $(N_{n-2} \geq N_{n-1})$, the value range for the differentiation parameter σ_{n-1} can be found in the contour.

This parameter search procedure ends when only classes 1 and 2 are left in the system. The contour used to reach the value range of the differentiation parameter σ_2 consists of the following curves

$$G_{\sigma_2, N_1=(N_2, N_2-1)}(\rho_2) = \frac{-A_{N_2} * \sum_{k=0}^{1} \rho_2^k + A_{N_2-1} * \rho_{1,2}}{(A_{N_2} + B_{N_2}) * \rho_2^1 - \left[A_{N_2-1} + B_{N_2-1} \sum_{k=0}^{1} \rho_{n-1}^k\right] * \rho_{1,2}}$$

$$\vdots$$

$$G_{\sigma_2, N_1=(N_2-j, N_2-j-1)}(\rho_2) = \frac{-A_{N_2-j} * \sum_{k=0}^{j+1} \rho_2^k + A_{N_2-j-1} * \rho_{1,2} \sum_{k=0}^{j} \rho_2^k}{\left(A_{N_2-1} + B_{N_2-1} \sum_{k=0}^{j} \rho_2^k\right) * \rho_2^{j+1} - \left[A_{N_2-j-1} + B_{N_2-j-1} \sum_{k=0}^{j+1} \rho_2^k\right] * \rho_{1,2} * \rho_2^j}$$

where $G_{\sigma_2, N_1=(N_2-j, N_2-j-1)}(\rho_2) > 1$.

Having completed the above recursive procedure, the differentiation parameters $\sigma_2, \ldots, \sigma_n$ and the blocking thresholds $N_1, N_2, \ldots, N_{n-1}$ can thus be solved. Note that $\sigma_1 = 1$ and $N_n = m$.

4 Numerical examples

The previously stated iterations are validated by a three-class scenario. Assume the utilisation factor for each class, that is, $\rho_i = (\lambda_i/\mu)$, $i = 1, 2, 3$, are given as $\rho_1 = 0.5$, $\rho_2 = 0.3$, and $\rho_3 = 0.19$. The queue size is assumed as $m = 30$. First, we apply the previous two-class solution to class 3 and the aggregate of classes 1 and 2. A contour similar to that of Fig. 3 is then obtained to illustrate the relationship of differentiation parameter σ_3, utilisation factor ρ_3, and blocking threshold N_2 (note that $N_3 = m$). From this contour, given $\rho_3 = 0.19$, one chooses $N_2 = 28$ and obtains $\sigma_3 \in [0.553564, 1]$. Next, by applying the same solution to class 1 and class 2, the contour relating differentiation parameter σ_2, utilisation factor ρ_2, and blocking threshold N_1 is obtained. Given $\rho_2 = 0.3$, one chooses $N_1 = 26$ and finds $\sigma_2 \in [0.699037, 1]$. The differentiation parameters are then solved as $\sigma_1 = 1$, $\sigma_2 \in [0.699037, 1]$, and $\sigma_3 \in [0.553564, 1]$.

The algorithm is applicable to practical scenarios since the range of the algorithm variables are all finite. The number of classes n is limited by traffic aggregates supported by a network; referring to the DiffServ service model, the value of n will not go beyond 8. The queue size m, another important parameter of the algorithm, is normalised as packet units. Its value is restricted by the memory space.

The computational complexity of the algorithm depends on two major factors. One is the number of iterations N; given the number of classes as n, the computation needs $N = n - 1$ iterations. The other factor is the computation time of each iteration; it is the time required to find all curves of the two-class scenario as explained in Section 3.1 and Fig. 3. Given the queue size m, the algorithm searches through ladder pairs of $(m - j, m - j - 1)$, $j = 0, 1, \ldots, m - 1$, that is, curves in a contour, as illustrated in (4) and Fig. 3. Assume that the calculation time of each curve (4) is one unit. In the worst case, each iteration takes m units of time to search m ladder pairs and obtain all m curves. Given the value range of both utilisation factor ρ_i and differentiation parameter σ_i as $[0, 1]$, nevertheless, the number of curves accommodated in a contour are far less than m. As explained in Section 3.1, for a queue size $m = 40$, only two curves are located in the contour (see Fig. 3); for $m = 80$, the resulting 3-D plot has four ladders, that is, three curves in the contour. Therefore, the computational complexity of obtaining n differentiation parameters is bounded by $O(nm)$, and it can be significantly less than $O(nm)$.

From the perspective of achieving the minimum system blocking probability, the exhaustive search that checks all combinations of variables, such as the number of classes n, queue size m, utilisation factor ρ_i, differentiation parameters σ_i, and blocking thresholds N_i, is brought in as a reference, to show the merits of this new approach. First, as observed from Table 1, the minimum blocking probabilities found by the new approach are close enough to those of the exhaustive search, although the difference between these two values increases with the number of classes. This value difference is resulted from the iterations that accumulate approximation errors. Since the number of classes n supported by DiffServ is limited, nevertheless, this tendency has no considerably negative effects on approximating the minimum blocking probabilities. Second, the new approach significantly shortens the search time for the minimum probability, by reducing the search space based on the contours. To limit the computation time of the exhaustive search to an acceptable level, the example described above adopts queue size m and the maximum number of classes n as 30 and 4, respectively. The new approach itself, nevertheless, can accommodate much larger values and is thus potentially feasible for on-line computation.

5 Conclusions

This paper introduces a simple approach to compute the loss differentiation parameters for the proportional differentiation service model. The quantitative guideline based on the principles of optimisation and queueing has been presented and validated. The intrinsic characteristics of the approach also guarantee that the resulting system blocking probability is minimised with respect to the chosen blocking thresholds.

Table 1: The comparison between the exhaustive search and the new approach

	Search approach	2-class scenario	3-class scenario	4-class scenario
Minimum value	Exhaustive	0.0328520171	0.0328520171	0.0328520171
	New	0.0329166398	0.0340719454	0.0353427972
Simulation time (s)	Exhaustive	0.22	122.29	26277.72
	New	0.01	0.42	8.03

Since the differentiation parameters are selected without considering the network or system conditions, the QoS differentiation may not be enforced owing to the dynamical nature of the network traffic. The computation guideline proposed in this paper, however, incorporates the network statuses and dropping mechanisms into the parameter computation procedure, and thus eliminates this problem.

From the practical implementation perspective, this parameter selection procedure can help network engineers define feasible loss differentiation parameters, without resorting to extra control mechanisms in curbing the service differentiation violation. The algorithm, as explained in Section 4, can be implemented in real time within the framework of the DiffServ service model.

6 Acknowledgments

This work has been supported in part by the New Jersey Commission on Higher Education via the NJI-TOWER project, and the New Jersey Commission on Science and Technology via NJWINS.

7 References

1 Blake, S., Black, S., Carlson, M., Davies, E., Wang, Z., and Weiss, W.: 'An architecture for Differentiated Services', IETF RFC 2457, December 1998

2 Dovrolis, C., and Ramanathan, P.: 'Proportional differentiated services, part II: loss rate differentiation and packet dropping'. Proc. 8th Int. Workshop on Quality of Service (IWQoS'00), Pittsburgh, USA, June 2000, pp. 53–61

3 Striegel, A., and Manimaran, G.: 'Packet scheduling with delay and loss differentiation', *Comput. Commun.*, 2002, **25**, pp. 21–31

4 Selvaraj, M., Lazarou, G.Y., and Hu, R.: 'Proportional bandwidth, delay, and loss differentiation'. Proc. 2nd IASTED Int. Conf. on Communications, Internet, & Information Technology (CIIT), Scottsdale, Arizona, November 2003, pp. 291–296

5 Kumar, A., Kaur, J., and Vin, H.M.: 'End-to-end proportional loss differentiation', Technical report TR-01-33, University of Texas at Austin, September 2001

6 Christian, N.: 'Quantifiable service differentiation for packet networks', PhD thesis, University of Virginia, 2003

7 Aweya, J., Quellette, M., and Montuno, D.Y.: 'Weighted proportional loss rate differentiation of TCP traffic', *Int. J. Netw. Manage.*, 2004, **4**, (4), pp. 257–272

8 Chen, Y., Qiao, C., Hamdi, M., and Tsang, D.H.K.: 'Proportional differentiation: a scalable QoS approach', *IEEE Commun. Mag.*, 2003, **14**, (6), pp. 52–58

9 Zeng, J., and Ansari, N.: 'An enhanced dropping scheme for proportional differentiated services'. Proc. Int. Conf. on Communications (ICC'03), Alaska, USA, May 2003, pp. 11–15

10 Tan, C., Gurusamy, M., and Lui, J.: 'Achieving proportional loss differentiation using probabilistic preemptive burst segmentation in optical burst switching WDM networks'. IEEE Global Telecommunications Conf. (GLOBECOM'04), Dallas, Texas, November 2004, Vol. 3, pp. 1754–1758

11 Dovrolis, C., and Ramanathan, P.: 'Dynamic class selection and class provisioning in proportional differentiated services', *Comput. Commun. J.*, 2003, **26**, (3), pp. 204–221

12 Virtamo, J.T.: 'Reciprocity of blocking probabilities in multiservice loss systems', *IEEE Trans. Commun.*, 1988, **36**, (10), pp. 1174–1175

13 Kaufman, J.S.: 'Blocking in a completely shared resource environment with state dependent resource and residency requirements'. Proc. Eleventh Annual Joint Conf. of the Computer and Communications Societies (INFOCOM'92), Florence, Italy, May 1992, pp. 2224–2232

14 Kleinrock, L.: 'Queueing systems, Volume I: Theory' (Wiley-Interscience, New York, 1975)