# Characterization and Performance Evaluation for Proportional Delay Differentiated Services*

Matthew K. H. Leung  John C. S. Lui
Dept. of Computer Science & Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong

David K. Y. Yau
Department of Computer Science
Purdue University
West Lafayette, IN 47907

## Abstract

*In this paper, we consider a proportional delay model for Internet differentiated services. Under this model, an ISP can control the "spacing" of waiting times between different classes of traffic. Specifically, the ISP tries to ensure that the average waiting time of class $i$ traffic relative to that of class $i-1$ traffic is consistently a specifiable ratio. If the ratio is less than one, the ISP can legitimately charge users of class $i$ traffic a higher tariff rate (compared to the rate for class $i-1$ traffic), since class $i$ users consistently enjoy better performance than class $i-1$ users. We use time-dependent priority scheduling to realize the proportional delay model. We formally characterize the feasible regions in which given delay ratios can be achieved. Moreover, a set of scheduling parameters for obtaining the desired delay ratios can be determined by an efficient control algorithm. Experiments are carried out to illustrate the short-term, medium-term and long-term relative waiting time performances for different service classes.*

## 1 Introduction

The Internet is being used for many different user activities, including emails, software distribution, video and audio entertainment, e-commerce, and real-time games. Although some of these applications are designed to be adaptive to available network resources, they still expect different levels of service from the network in order to have good performance. Therefore, there is a growing need to provide an alternative Internet service model, as compared to the conventional one-size-fits-all *best-effort* service model.

The major problem about the best-effort service model is that it treats all packets from different Internet applications on an equal basis. One approach to solving this problem is the *Integrated Services (IntServ)* model proposed by the IETF. IntServ is inherently a reservation based approach. To achieve predictable performance, an application is expected to reserve resources, such as network bandwidth and buffers, using a protocol like RSVP [5]. This raises two important deployment issues. First, *all* the routers along an end-to-end network path must be RSVP-capable in order to realize IntServ benefits. Second, a router has to manage per-flow state and perform per-flow processing. This makes it difficult for IntServ to scale well to tens of thousands of network flows. Although there are proposals to alleviate these difficulties [3, 13], designing a scalable IntServ model is still an open and challenging problem.

Recently, another service model known as *Differentiated Services (DS)* is proposed by the IETF and has received a lot of attention [8]. Under the DS model, traffic flows are aggregated and identified as classes. Since the number of DS traffic classes is expected to be far fewer than the number of flows in IntServ, DS is much less susceptible to the scalability problem. Rather than providing end-to-end performance guarantees for individual flows like the IntServ, the DS service objective is to differentiate among classes of traffic using *per-hop* packet forwarding behaviors. In general, there are two approaches for delivering the DS service model: *absolute* differentiated services and *relative* differentiated services.

For absolute DS, the goal is to achieve performance measures similar to those in the IntServ model, but without keeping per-flow state within routers. Two most prominent schemes for absolute DS are premium services [9] and assured services [1]. Premium services offers to the user a performance level that is similar to that of a leased line, provided that the user's traffic satisfies an agreed upon traffic profile (e.g., the traffic is below some specified bit rate). Assured services, on the other hand, provides a way to control packet dropping for different classes of traffic. Specif-

*Research supported in part by the National Science Foundation under grant numbers EIA-9806741 and CCR-9875742 (CAREER), and by CERIAS

ically, at the network edge, packets are classified as either *in* or *out* class, according to whether they are within or outside a traffic profile. These two classes have different dropping priorities. When a router becomes congested, it starts by dropping the *out* packets first. If congestion persists, then the router will start dropping both *in* and *out* packets. Under all traffic loads, however, *out* packets are dropped with a higher probability than *in* packets. In [6, 12], the authors present some elegant mathematical models for analyzing the performance of the absolute DS service model. In [14], the authors illustrate that in order to provide service assurance with coarse spatial granularity and high network utilization, some form of route pinning is required.

Relative differentiated services is proposed in [2]. Rather than providing absolute performance guarantees for different classes of traffic, the goal is to give better performance to class $i$ traffic than class $i-1$ traffic with a fixed quality spacing. If the goal is *consistently* achieved, then class $i$ users will see better performance than class $i-1$ users. In return, the ISP can legitimately charge class $i$ traffic a higher tariff rate than class $i-1$ traffic. In [2], the authors propose two algorithms, called BPR and WTP, respectively, for implementing proportional *delay* differentiation. For WTP scheduling in particular, they claim that in order to achieve a delay ratio of $r$ between two traffic classes, the corresponding service parameters should also be set to have ratio $r$. As we will illustrate, however, the WTP control parameters should in fact depend on the distribution of traffic loads. We also formally illustrate the conditions under which given delay ratios are feasible. Specifically, our paper addresses the following questions:

- Given desired waiting time ratios for $N$ traffic classes, under what conditions (e.g., traffic load distribution for the $N$ classes) can feasible WTP control parameters be found to achieve the ratios?
- Given that the waiting time ratios are feasible, how can one obtain the WTP control parameters that will achieve the waiting time ratios?
- Given the obtained control parameters, can we maintain the waiting time spacings at different time scales?

The rest of the paper is organized as follows. In Section 2, we review proportional differentiated services as described in [2]. In Section 3, we characterize and analyze the performance of a time-dependent-priority algorithm in achieving the proportional delay differentiation. Specifically, we discuss the conditions under which control parameters for the TDP algorithm exist and how they can affect a given set of quality spacings. In addition, we present an efficient iterative method for finding the values of these control parameters when they exist. In Section 4, we present experimental results that illustrate the performance of our methods. In particular, we compare waiting time spacings achieved between different classes of traffic using the control parameters in [2] versus using those obtained by our proposed iterative method. We also present waiting time spacing results under different time scales. Lastly, conclusion is given in Section 5.

## 2 Background

We further review the proportional differentiation service model as it is proposed in [2]. The model has two objectives. First, it should provide *consistent* service differentiation between classes, in that a class with higher advertised quality should consistently outperform a class with lower advertised quality. Second, it should allow the quality spacings between classes to be adjusted based on pricing and other criteria. For example, it should be possible to configure the average packet delay of a higher quality service class to be, say, 80% the delay of a lower quality class. Further, the authors stipulate that these two goals should be met even for sharing over "short" timescales.

The authors propose two scheduling algorithms for approximating the proportional DS model for *delay* differentiation under *heavy-load* conditions. The first one, called the backlog-proportional rate (BPR) scheduler, is based on GPS, but with the modification that the class service rates are dynamically adjusted so that they are ratioed proportionally to the corresponding ratios of measured class loads. The algorithm is shown to exhibit pathological *sawtooth-type* delay variations over short timescales.

The second algorithm, called the waiting-time priority (WTP) scheduler, is based on Kleinrock's Time-Dependent-Priorities algorithm [10]. Specifically, the priority of a packet in flow $i$ at time $t$ is proportional to the waiting time of the packet at time $t$, where the proportionality constant, denoted $s_i$ (using the notation in [2]), is a service parameter for $i$. Using simulations, the authors show that the relative average delay experienced by two flows, say $i$ and $j$, in a WTP server has value close to $s_i/s_j$, for monitoring timescales as short as a few tens of packet transmission times. Hence, WTP *approximates* the proportional delay differentiation model under heavy-load conditions. The reason it is an approximation is that, on one hand, it is known that consistent quality spacing cannot be achieved over timescales that are arbitrarily short. On the other hand, no conditions are given in [2] to assess feasibility given a certain value of the monitoring timescale. Hence, the notion of "short" timescales remains imprecise.

Given the lack of a complete characterization of the proportional DS model in [2], we set out to further evaluate its theoretical and experimental properties in this paper. Our objectives are (1) to further contribute to the understanding of feasibility conditions for achieving proportional delays, and (2) when it is feasible, to derive scheduling parameters that can achieve the property. We mainly focus on the WTP algorithm, which was shown to be highly effective in [2]. Since WTP is derived from Kleinrock's work [10], we follow similar analysis to further characterizing the feasibility for achieving proportional delays. We found that, independent of the timescale parameter, system utilization impacts feasibility. Further, we show that when system utilization varies significantly (i.e., the system is not always heavily loaded as assumed in [2]), the per-flow WTP service parameters should be dynamically adjusted to maintain proportional delay differentiation whenever it is feasible.

## 3 Characterization and Performance Analysis

In this section, we summarize some results for time-dependent priority (TDP) scheduling. We first characterize a necessary and sufficient condition for a given delay spacing to be feasible under TDP for two traffic classes. We then extend the characterization to TDP for $N$ classes. We also present an iterative method for obtaining the feasible control parameters.

TDP is a *non-preemptive* packet scheduling algorithm which provides a set of control variables $b_i, 1 \leq i \leq N$ where $0 \leq b_1 \leq b_2 \leq \cdots \leq b_N$ and $b_i$ dictates the instantaneous priority of class $i$ packets. Specifically, if a tagged class $i$ packet arrives at time $\tau$, then its priority at time $t$ (for $t \geq \tau$), denoted by $q_i(t)$, is

$$q_i(t) \quad = \quad (t-\tau)b_i \ . \tag{1}$$

Let $N_i(t)$ denote the number of class $i$ packets waiting in the queue at time $t$. If the server is ready to transmit a packet, it will choose a packet from class $i^*$ where

$$q_{i^*}(t) \quad = \quad \max_{1 \leq i \leq N} \{q_i(t) \text{ and } N_i(t) > 0\} \, . \tag{2}$$

Whenever a tie for the highest priority occurs, the tie is broken by an FCFS rule. If there is no packet in the system, then the server will be idle and it will be activated by any newly arriving packet. Notice that in the TDP scheduler, a class $i$ packet increases in priority at a faster rate ($b_i$) than packets of any class $j$, where $j < i$.

Assume that the arrival process of class $i$ packets is Poisson with rate $\lambda_i$ and let $\overline{x_i}$ ($\overline{x_i^2}$) be the first (second) moment of class $i$ packet service times, then the system utilization $\rho$ of a TDP server is equal to $\sum_{i=1}^{N} \rho_i$ where $\rho_i = \lambda_i \overline{x_i}$. Kleinrock derives a closed-form expression for the average long-term waiting time for class $p$ packets. The closed-form expression is given as

$$W_p \quad = \quad \frac{[W_0/(1-\rho)] - \sum_{i=1}^{p-1} \rho_i W_i [1-(b_i/b_p)]}{1 - \sum_{i=p+1}^{N} \rho_i [1-(b_p/b_i)]} \tag{3}$$

for $i = 1, 2, ..., N$, where $W_0$ is the expected residual service time, $W_0 = \frac{1}{2} \sum_{i=1}^{N} \lambda_i \overline{x_i^2}$. It is interesting to note that Kleinrock derives the above expression by assuming that packet service times are exponentially distributed. In [7], the authors illustrate that the closed-form expression in Equation (3) is also valid for any general service time distribution.

One attractive feature about the TDP scheduler is that if one wants to maintain certain proportional differentiation of waiting times between different classes of traffic, one can simply *adjust* the control parameters $b_i$'s so as to achieve the desired waiting time spacings. Let $r_{i,j}$ be the waiting time ratio between class $i$ and class $j$, that is

$$r_{i,j} = \frac{W_i}{W_j}.$$

In this paper, we will address the following important questions:

1. Given the waiting time ratio requirements for all classes $r_{i,i+1}$ (where $i = 1, 2, \ldots, N-1$), under what conditions of $\rho_i$'s does a solution for $b_i$'s exist?

2. Given $\rho_i$, the traffic loads of all classes, how to obtain the $b_i$ values so that the ratios $W_i/W_{i+1}$ are equal to our target values of $r_{i,i+1}$?

To understand the problem, we start with a simple case of two traffic classes. We then go on to solve the general problem of $N$ traffic classes.

**Theorem 1** *For two classes of traffic, let $r_{1,2}$ be the required ratio of the average waiting times of class one traffic to that of class two traffic. To achieve the specified ratio $r_{1,2}$, the necessary and sufficient condition is for the system utilization $\rho$ to satisfy* $1 - \frac{1}{r_{1,2}} < \rho < 1$.

**Proof:** First, $\rho < 1$ is required so that the system is stable. According to Equation (3), packets of the two classes have average waiting times of

$$\begin{aligned} W_1 &= \frac{[W_0/(1-\rho)]}{1-\rho_2[1-(b_1/b_2)]} \\ W_2 &= [W_0/(1-\rho)] - \rho_1 W_1 [1-(b_1/b_2)]. \end{aligned}$$

By substituting $W_1$ into $W_2$, we have

$$W_2 = [W_0/(1-\rho)] \frac{1-\rho[1-(b_1/b_2)]}{1-\rho_2[1-(b_1/b_2)]}.$$

The average waiting time ratio between class one and class two is given as

$$\frac{W_1}{W_2} = \frac{1}{1 - \rho[1 - (b_1/b_2)]}. \quad (4)$$

Let us denote the target ratio of the average waiting times between class one traffic and class two traffic as $r_{1,2} = W_1/W_2$, then we have

$$\rho = \frac{b_2}{b_2 - b_1}\left(1 - \frac{1}{r_{1,2}}\right) \quad (5)$$

Since $0 < b_1 < b_2$, this implies that $\frac{b_2}{b_2 - b_1} > 1$. Therefore, $\rho > 1 - \frac{1}{r_{1,2}}$ ∎

**Remarks:** The implication for the above theorem is that in order to achieve a specified ratio $r_{1,2}$, we need to have *enough* packets arriving to the system. For example, if the requirement is $r_{1,2} = 10$, then the system has to be *at least* 90% utilized so as to achieve the desired waiting time spacing. In other words, if the system utilization is less than 90%, we cannot achieved the $r_{1,2} = 10$ spacing using WTP.

We make two observations from the above theorem. First, the ratio of the average waiting times does not *solely* depend on $b_2/b_1$, but rather, depends on the system utilization also. Only when $\rho \to 1$, will setting the control parameters $b_2/b_1 = r_{1,2}$ achieve the desired waiting time spacing. Secondly, if the system utilization is known, we can adjust $b_1$ and $b_2$ such that the resulting waiting time ratio will be equal to our target value $r_{1,2}$. Below, we will explain how to choose the appropriate values for $b_i$'s.

**Corollary 1** *If the system utilization $\rho$ satisfies the necessary and sufficient condition in Theorem 1, then $b_1 = 1$ and $b_2 = \rho/(\rho - 1 + \frac{1}{r_{12}})$ can achieve the specified ratio $r_{1,2}$.*

**Proof:** Please refer to our technical report [4]. ∎

In summary, to satisfy a specified system performance requirement given as $W_1/W_2 = r_{1,2}$, we should measure the system utilization and set the parameters $b_1$ and $b_2$ accordingly. Then the long term average waiting time ratio of class one traffic to class two traffic will be equal to the target value of $r_{1,2}$, provided that $\rho$ is within the feasibility region $(1 - 1/r_{1,2}, 1]$.

For the general case of $N$ classes, the problem becomes very complicated because to find values of the control parameters $\{b_i\}$'s, we need to solve Equation (3), which is a system of $N$ non-linear equations. Nevertheless, we can calculate $W_i$ by using the conservation law principle, provided that the configuration of system ($\rho_i$ and $W_0$) is known.

The conservation law [10] states that if a scheduling discipline is independent of the service time of jobs (which is the case for the TDP scheduler), then the weighted average of the waiting times of all classes is *invariant*, and it is equal to the average waiting time of a M/G/1 system. Mathematically, the relationship is

$$\sum_{i=1}^{N} \frac{\rho_i}{\rho} W_i = \frac{W_0}{1 - \rho}. \quad (6)$$

Let us define $s_i = r_{i,i+1} r_{i+1,i+2} \cdots r_{N-1,N} = W_i/W_N$ and express $W_i = s_i W_N$. Substituting this expression of $W_i$ in Equation (6), we have

$$\frac{W_0}{1 - \rho} = \frac{1}{\rho} \sum_{i=1}^{N} \rho_i s_i W_N.$$

Therefore, we can express

$$W_p = s_p \frac{\rho W_0}{1 - \rho} \left( \sum_{i=1}^{N} \rho_i s_i \right)^{-1} \quad \text{for } p = 1, 2 \ldots, N \quad (7)$$

From the above equations, we observe that the only unknown in Equation (3) is the vector $\boldsymbol{b} = [b_1, b_2, \cdots, b_N]$. Now, putting all $b_p$'s in Equation (3) on the left hand side, we have

$$\begin{aligned} & b_p \left( W_p \sum_{i=p+1}^{N} \frac{\rho_i}{b_i} \right) - \frac{1}{b_p} \left( \sum_{i=1}^{p-1} \rho_i W_i b_i \right) \\ & = \frac{W_0}{1 - \rho} - \sum_{i=1}^{p-1} \rho_i W_i - \left( 1 - \sum_{i=p+1}^{N} \rho_i \right) W_p. \quad (8) \end{aligned}$$

Letting

$$\begin{aligned} A(p) &= W_p \sum_{i=p+1}^{N} \frac{\rho_i}{b_i}, B(p) = \sum_{i=1}^{p-1} \rho_i W_i b_i, \\ R(p) &= \frac{W_0}{1 - \rho} - \sum_{i=1}^{p-1} \rho_i W_i - \left( 1 - \sum_{i=p+1}^{N} \rho_i \right) W_p, \end{aligned}$$

we have

$$A(p)\, b_p - \frac{B(p)}{b_p} = R(p) \qquad p = 1, 2, \ldots, N. \quad (9)$$

Now, we have a system of non-linear equations for solving the $b_p$'s. Since all the $b_p$'s have to be positive, there should be a condition for $\rho_i$ and $r_i$ such that $\{b_i\}$'s are positive. We have the following theorem.

**Theorem 2** *A necessary condition to have positive solutions of the $b_i$'s is $R(1) > 0$ and $R(N) < 0$.*

**Proof:** Since $b_i > 0$ for $i = 1, 2, \ldots, N$, this implies $A(p) > 0$ and $B(p) > 0$. However, $R(p)$ can be positive or negative. Let us consider three cases.

**Case 1:** For $p = 1$, we have $B(1) = 0$, which implies that $b_i = \frac{R(1)}{A(1)}$. Since $b_1 > 0$, this implies that $R(1) > 0$.

**Case 2:** For $1 < p < N$, we use the result from Equation (9) and we have

$$A(p)\, b_p^2 - R(p)\, b_p - B(p) = 0.$$

Since we want $\{b_i\}$'s to be positive, we have

$$b_p = \frac{R(p) + \sqrt{R(p)^2 + 4A(p)B(p)}}{2A(p)}.$$

Because $R(p)^2 + 4A(p)B(p) > R(p)^2$, we have $\sqrt{R(p)^2 + 4A(p)B(p)} > |R(p)|$. Hence

$$b_p > \frac{R(p) + |R(p)|}{2A(p)} \geq 0.$$

Therefore, for $1 < p < N$, $b_p$ is always greater than zero even when $R(p)$ is negative.

**Case 3:** For $p = N$, we have $A(N) = 0$, which implies that

$$b_N = -\frac{B(N)}{R(N)}$$

Since $b_N > 0$ and $B(N) > 0$, this implies that $R(N) < 0$. ∎

**Remarks:** The implication of the above theorem is that a necessary condition for a feasible region (i.e., a region wherein a positive solution of $b_i$'s exist) is $R(1) > 0$ and $R(N) < 0$. If the system configuration $(\rho_i, r_i)$ falls outside this region, it is possible that there exist no positive values of $b_i$'s for the TDP scheduler to obtain the target waiting time ratios.

The first condition, $R(1) > 0$, implies

$$\frac{W_0/(1-\rho)}{W_1} \quad > \quad 1 - \sum_{i=2}^{N} \rho_i \tag{10}$$

where $W_0/(1-\rho)$ is the average waiting time of the aggregate traffic. If we want a large waiting time differentiation, $W_1$ has to be large. Since $W_1 \geq W_2 \geq \cdots \geq W_N$, this implies the fraction on the left hand side of Equation (10) has to be small. Thus, $\sum_{i=2}^{N} \rho_i$ should be close to one to make the inequality hold. The *physical meaning* is that to have a large waiting time differentiation, there should be *sufficient* amount of higher class packets to keep the server busy so that the lower class packets are delayed adequately.

The second condition is $R(N) < 0$, which implies

$$\frac{W_0}{1-\rho} \quad < \quad \sum_{i=1}^{N-1} \rho_i W_i + W_N. \tag{11}$$

---

**Procedure: Iterative Algorithm**
**Input:** $\lambda_i, \overline{x_i}, \overline{x_i^2}$ for $i = 1, .., N$.
/*arrival rates, $1^{st}$, $2^{nd}$ moments of service times*/
**Output:** $\boldsymbol{b} = [b_1, b_2, \ldots, b_N]$.

**begin**
    $k = 0$ and $b_p^{(0)} = \frac{1}{W_p}$ for $p = 1, \ldots, N$; /* initialize */
    /* test for convergence */
    **while** (($\sum_{p=1}^{N} |f_p(\boldsymbol{b}^{(k)}| > \epsilon$) and $k <$ MAX_ITERATION_COUNT )
        **begin**
            **for** ($p = 1; p <= N; p = p + 1$)
/* update the value of $b_i^{(k)}$ */
                $b_p^{(k+1)} = \phi_p(b_1^{(k+1)}, b_2^{(k+1)}, \ldots, b_{p-1}^{(k+1)}, b_p^{(k)}, \ldots, b_N^{(k)})$;
            $k = k + 1$;
        **end**
**end**

---

**Figure 1. Iterative algorithm**

To make the inequality hold, the value of the left hand side in Equation (11) should be large. Therefore, $\rho_i$ should be large, especially for the lower traffic classes. The *physical meaning* is that in order to have a large waiting time differentiation, the server has to delay packets of the lower classes. If their load is high, many of them will be backlogged and their waiting times will increase.

Last, but not least, another important implication of the above necessary conditions is that even though the system utilization $\rho$ remains unchanged, it is still possible that certain distributions of $\rho_i$'s will not lead to a positive solution of $b_i$'s. In such cases, the system cannot achieve the target waiting time ratios.

Let us now present an efficient algorithm for computing the values of $b_i$'s, provided that the necessary condition is satisfied. In general, we have to find a solution for the set of non-linear equations in (8). To achieve this, the following iterative algorithm is proposed. Note that this iterative algorithm is based on the Gauss-Seidel iteration method [11], which has a well-known condition for convergence.

First, let $\phi_p$ be the *functional evaluation operator* for $b_p$ where $b_p = \phi_p(b_1, b_2, \ldots, b_N)$ for $p = 1, 2, \ldots, N$ where

$$\phi_p(b_1, b_2, \ldots, b_N) \quad = \quad \begin{cases} \frac{R(p) + B(p)/R(p)}{A(p)} & \text{for } p \neq N, \\ \sqrt{\frac{b_p B(p)}{-R(N)}} & \text{for } p = N. \end{cases} \tag{12}$$

and

$$f_p(\boldsymbol{b}) \quad = \quad A(p)\, b_p - B(p)/b_p - R(p) \qquad \text{for } p = 1, 2, \ldots, N. \tag{13}$$

The iterative algorithm is shown in Figure 1.

## 4 Experiments

In this section, we report results from several experiments. In the first experiment, we compare performance results using control parameters taken from [2] versus control parameters obtained using our iterative algorithm. We also investigate the long-term average waiting time spacings under different system utilizations. Lastly, we study waiting time spacings among different traffic classes given different monitoring window sizes. Please refer to [4] for more experimental results (e.g., non-Poisson traffic) and the convergence conditions of the proposed iterative algorithm.

**Experiment 1 (Comparisons with [2]):** In this experiment, we want to compare the achievable waiting time spacings, using the control parameters in [2] and the control parameters that we obtain in Section 3. We consider three classes of traffic. The arrival process of class $i$ ($i = 1, 2, 3$) is Poisson with a rate of $\lambda_i$. The packet length distribution is the same for all classes where 40% of the packets are 40 bytes, 50% are 550 bytes, and 10% are 1500 bytes. The output link capacity is 441 bytes/unit time (the time unit can be normalized to achieve an arbitrary link speed). In each run of the experiment, we generate at least 50,000 packets for each class. Then, we average the waiting times for each class and compare the ratios with the target waiting time spacings. In part one of the experiment, we set $\lambda_1 = 0.35, \lambda_2 = 0.3, \lambda_3 = 0.3$ (since the service time requirement is normalized to one, the system utilization is $\rho = 0.95$) and we consider the target waiting time spacings of $r_{i,i+1} = 4.0$. Table 1 illustrates the achievable spacings, using the control parameters in [2] and our proposed method. We observe that the proposed control parameters in [2] cannot achieve the target spacings; however, our proposed algorithm can find the appropriate values of the control parameters such that the spacings are achieved.

In the second part of the experiment, we set the arrival rates as $\lambda_1 = 0.2, \lambda_2 = 0.2, \lambda_3 = 0.2$ (or $\rho = 0.6$) and $r_{i,i+1} = 2.0$. Table 2 illustrates the achievable spacings. The table shows that our algorithm can determine that it is *not possible* to achieve the target waiting time spacings ($r_{i,i+1} = 2$) for the given loading distribution. Indeed, using the proposed control parameter values in [2], we can only achieve spacing values around 1.3.

**Experiment 2 (Long-term waiting time spacing):** In the first part of the experiment, we want to test whether we can achieve the target waiting time ratios under different system utilizations. We consider three classes of traffic. All arrival processes are Poisson. For a low system utilization ($\rho = 0.2$) case, the arrival rates are $\lambda_1 = 0.05, \lambda_2 = 0.1, \lambda_3 = 0.05$. For a medium utilization ($\rho = 0.6$) case, the arrival rates are $\lambda_1 = 0.2, \lambda_2 = 0.2, \lambda_3 = 0.2$. For a high utilization ($\rho = 0.9$) case, the arrival rates are $\lambda_1 = 0.3, \lambda_2 = 0.3, \lambda_3 = 0.3$. The packet length distribution is similar to the one in Experiment 1. Again, for each run of the experiment, we generate at least 50,000 packets for each class and then average the waiting times for each class and compare the ratios with the target waiting time spacings. The experimental results are summarized in Tables 3,4 and 5. We observe that our iterative algorithm is very efficient (less than 20 iterations are executed) and we are able to find the values for the control parameters so as to achieve the long-term average waiting time spacings.

We also evaluate system performance under different class load distributions. We consider three classes of traffic with target spacings of $r_{i,i+1} = 1.1$. In all the cases considered, the system utilization is $\rho = 0.9$. The results are shown in Table 6. We can observe that under different traffic distributions, our proposed algorithm is highly effective in achieving the specified waiting time ratios.

We conclude from these experiments that the proposed algorithm can accurately determine the control parameter values under different operating conditions (e.g., different system utilizations, different traffic load distributions, etc) so as to achieve the long term waiting time spacings.

**Experiment 3 (Short-term waiting time spacing):** Besides long-term analysis, we study the short-term behavior of our packet scheduling algorithm. In these experiments, we measure the ratios of the average waiting times between successive classes in consecutive time intervals. We measure the average waiting times of all the packets that get served within a specified time window. The length of the time window is varied to be 10, 100, 500, 1000, 3000, 7000 and 10,000 p-units, where a p-unit is the average packet transmission time (the average service time is one time unit.) We average all the data collected from all the three traffic classes because the target ratios are the same between two successive classes and this simplifies the presentation of our results.

Figures 2 and 3 give histogram plots for the waiting time spacings under system utilizations of 0.6 and 0.9, respectively, and with various target waiting time spacings and window sizes. The x-axis of each plot shows the possible range of waiting time spacings. From the figures, we can observe that if the system is highly utilized ($\rho = 0.9$), then the achievable waiting time ratios can be kept close to our target spacings even in *short* timescales. Moreover, the variance of the ratios is small (which implies that most of the probability mass is within the target spacings). However, if the system utilization is low ($\rho = 0.6$), the variance of the waiting time spacings is large. For really short time scales (e.g., window size of 100), only a small percentage of the data points lies within our target region. The reason is that when the utilization is low, it takes longer for a sufficient number of packets to be serviced so that the waiting times can reach their equilibrium values. This in turn requires a larger window size in order for the target waiting times to be achieved.

| Method | target spacing ($r_{i,i+1}$) | control parameters | achievable spacing ($\frac{W_1}{W_2}$) | achievable spacing ($\frac{W_1}{W_2}$) |
|---|---|---|---|---|
| [2] | 4 | $b_1 = 1, b_2 = 4, b_3 = 16$ | 3.366 | 3.030 |
| our approach | 4 | $b_1 = 1, b_2 = 5.11, b3 = 35.937$ | 3.990 | 3.890 |

**Table 1. Comparison between achievable waiting time spacing**

| Method | target spacing ($r_{i,i+1}$) | control parameters | achievable spacing ($\frac{W_1}{W_2}$) | achievable spacing ($\frac{W_1}{W_2}$) |
|---|---|---|---|---|
| [2] | 2 | $b_1 = 1, b_2 = 2, b_3 = 4$ | 1.39 | 1.36 |
| our approach | 2 | cannot pass feasibility test | —— | —— |

**Table 2. Determination of non-achievable waiting time spacing**

## 5 Conclusion

In this paper, we consider a WTP scheduler so as to achieve delay proportional differentiated services. The scheduler tries to ensure that the average waiting time of class $i$ traffic relative to that of class $i-1$ traffic is consistently a specifiable ratio. This way, an ISP can legitimately charge users of class $i$ traffic a higher tariff rate (compared to the rate of class $i-1$ traffic) because class $i$ users consistently enjoy better performance than class $i-1$ users.

For two-class WTP, we obtain a necessary and sufficient condition for a given delay spacing to be feasible. For general $N-$class WTP, we present a set of necessary conditions, and give their physical meanings. Using these conditions, we can easily determine if a given delay proportional differentiation is impossible. We also present an efficient algorithm for finding WTP control parameter values that will realize a set of specified waiting time spacings when these parameters exist. Experiments are carried to illustrate that using our control parameter values, we can obtain waiting time spacings that are closer to the given target waiting time ratios, when compared with results in [2]. We also illustrate waiting time spacing performance under short, medium and long timescales. Future work includes a dynamic measurement technique for tracking system loads and adjusting control parameters in real-time to achieve consistent waiting time ratios.

**Acknowledgment:** We wish to acknowledge helpful comments by Professor E. V. Krishnamurthy on the iterative numerical solution, as well as insightful comments by the referees.

## References

[1] D. Clark, W. Fang. *Explicit Allocation of best Effort Packet Delivery Service*. IEEE/ACM Transactions on Networking, Vol. 6, pp. 362-373. August, 1998.

[2] C. Dovrolis, D. Stiliadis, P. Ramanathan. *Proportional Differentiated Services: Delay Differentiation and Packet Scheduling*. ACM SIGCOMM'99, pp. 109-119, August, 1999.

[3] R. Guerin, S. Kamat, V. Peris, R. Rajan. *Scalable QoS Provision Through Buffer Management.* ACM SIGCOMM, Sept, 1998.

[4] K. H. Leung, John C. S. Lui, David K. Y. Yau. *Characterization and Performance Evaluation for Proportional Delay Differentiated Services*. Technical Report CU-2000-03, Chinese University of Hong Kong, April, 2000

[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 Guildlines on Deployment*. September, 1997. IETF RFC 2208.

[6] M. May, J.C. Bolot, C. Diot and A. Jean-Marie. *Simple Performance Models of Differentiated Services Schemes for the Internet*. IEEE INFOCOM, March, 1999.

[7] A. Netterman, I. Adiri. *A Dynamic Priority Queue with General Concave Priority Functions*. Operation Research, 27(6), pp. 1088-1100, 1979.

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

[9] K. Nichols, V. Jacobson and K. Poduri. *Expedited Forwarding PHB Group*. Internet RFC 2598, June 1999.

[10] L. Kleinrock. *Queueing Systems: Vol 2.* Wiley-interscience, 1976.

[11] E. V. Krishnamurthy, S. K. Sen. *Numerical Algorithms Computations in Science and Engineering.* Affiliated East-West Press, 1991.

[12] S. Sahu, D. Towsley, J. Kurose. *A Quantitative Study of Differentiated Services for the Internet*. Proc. IEEE Global Internet'99, Rio de Janeiro, Brazil, December 1999.

[13] I. Stoica, S. Shenker, H. Zhang. *Core-Stateless Fair-Queueing: Achieving Approximately Fair Bandwidth Allocations in High Speed Network.* ACM SIGCOMM, September, 1998.

[14] I. Stoica, H. Zhang. *LIRA: An Approach for Service Differentiation in the Internet*. Proceedings NOSSDAV, 1998.

| Utilization $\rho$ | Low ($\rho = 0.2$) | Medium ($\rho = 0.6$) | High ($\rho = 0.9$) |
|---|---|---|---|
| Target ratio $r_{12}$ | 1.1 | 1.1 | 1.1 |
| Target ratio $r_{23}$ | 1.1 | 1.1 | 1.1 |
| $[b_1, b_2, b_3]$ | $[1, 2.03, 4.17]$ | $[1, 1.18, 1.4]$ | $1, 1.11, 1.24]$ |
| Loops executed | 5 | 11 | 12 |
| achieved $W_1/W_2$ | 1.10 | 1.09 | 1.10 |
| achieved $W_2/W_3$ | 1.12 | 1.09 | 1.10 |

**Table 3. Long Term Waiting Time Spacings with $r_{i,i+1} = 1.1$.**

| Utilization $\rho$ | Low ($\rho = 0.2$) | Medium ($\rho = 0.6$) | High ($\rho = 0.9$) |
|---|---|---|---|
| Target ratio $r_{12}$ | 1.5 | 1.5 | 1.5 |
| Target ratio $r_{23}$ | 1.5 | 1.5 | 1.5 |
| $[b_1, b_2, b_3]$ | Outside feasible region | $[1, 2.6, 7.4]$ | $[1, 1.6, 2.58]$ |
| Loops executed | —— | 11 | 17 |
| achieved $W_1/W_2$ | —— | 1.54 | 1.49 |
| achieved $W_2/W_3$ | —— | 1.48 | 1.51 |

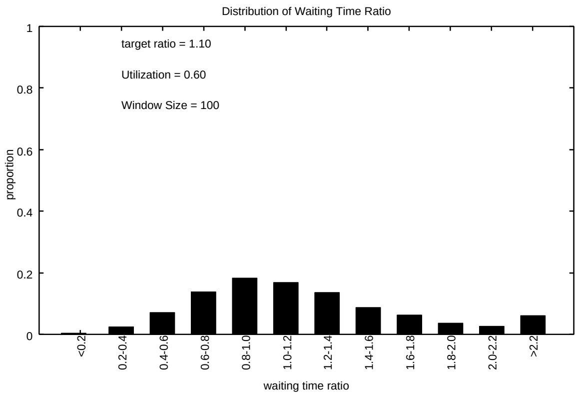
**Table 4. Long Term Waiting Time Spacings with $r_{i,i+1} = 1.5$.**

| Utilization $\rho$ | Low ($\rho = 0.2$) | Medium ($\rho = 0.6$) | High ($\rho = 0.9$) |
|---|---|---|---|
| Target ratio $r_{12}$ | 2.0 | 2.0 | 2.0 |
| Target ratio $r_{23}$ | 2.0 | 2.0 | 2.0 |
| $[b_1, b_2, b_3]$ | Outside feasible region | Outside feasible region | $[1, 2.32, 5.554]$ |
| Loops executed | —— | —— | 20 |
| achieved $W_1/W_2$ | —— | —— | 2.01 |
| achieved $W_2/W_3$ | —— | —— | 1.99 |

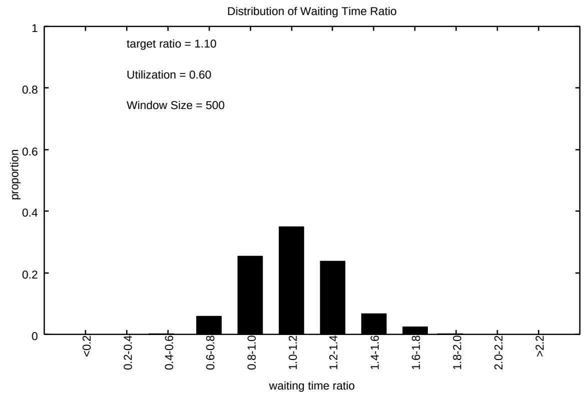
**Table 5. Long Term Waiting Time Spacings with $r_{i,i+1} = 2.0$.**

| load distribution (%) | loops | $b_1$ | $b_2$ | $b_3$ | $W_1/W_2$ | $W_2/W_3$ |
|---|---|---|---|---|---|---|
| 33.3-33.3-33.3 | 11 | 1 | 1.113 | 1.238 | 1.10 | 1.10 |
| 30-20-50 | 9 | 1 | 1.113 | 1.238 | 1.10 | 1.10 |
| 20-30-50 | 6 | 1 | 1.113 | 1.238 | 1.10 | 1.10 |
| 10-45-45 | 4 | 1 | 1.113 | 1.238 | 1.10 | 1.11 |
| 45-10-45 | 51 | 1 | 1.113 | 1.239 | 1.10 | 1.10 |
| 45-45-10 | 36 | 1 | 1.113 | 1.238 | 1.10 | 1.10 |

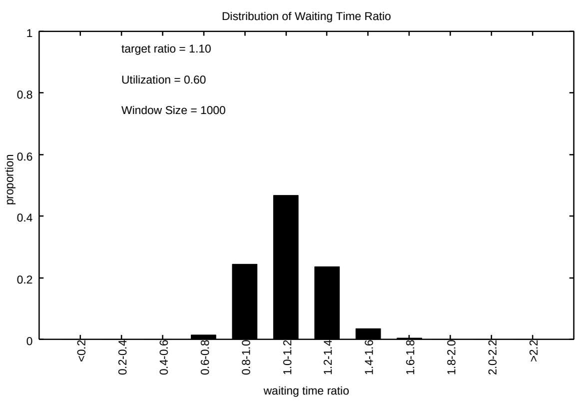
**Table 6. Waiting Time Spacings under different traffic loading distributions ($r_{i,i+1} = 1.1$).**



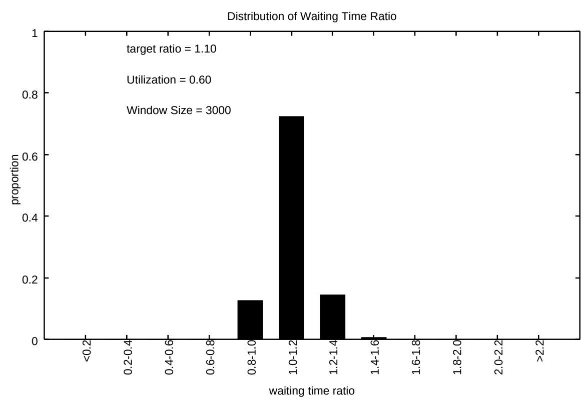

window=100



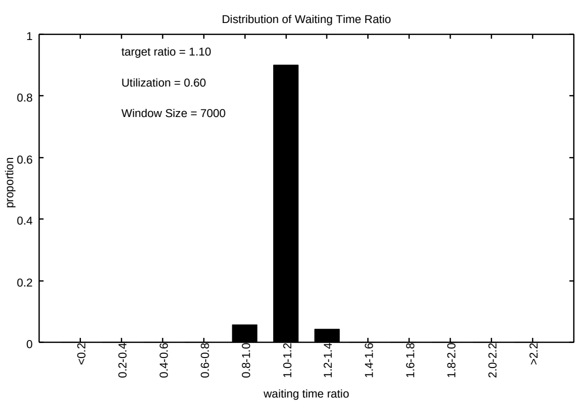

window=500



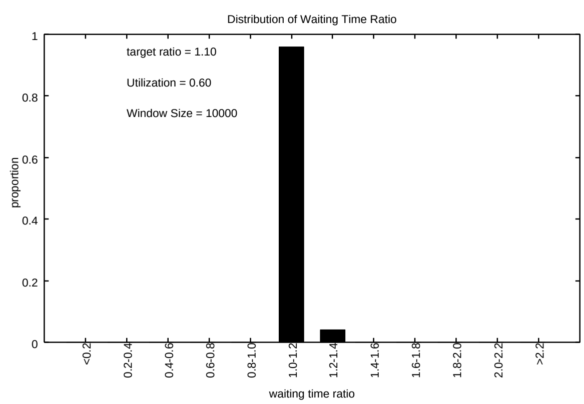

window=1000




window=3000

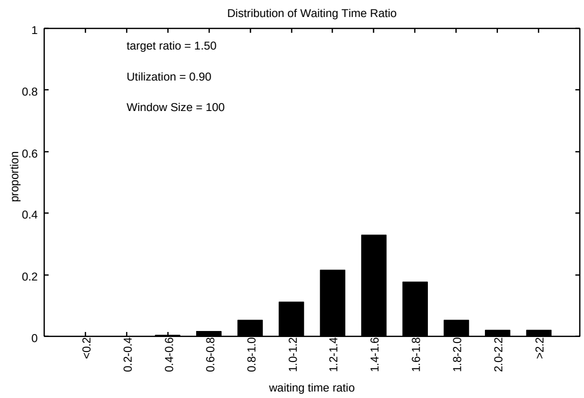



window=7000

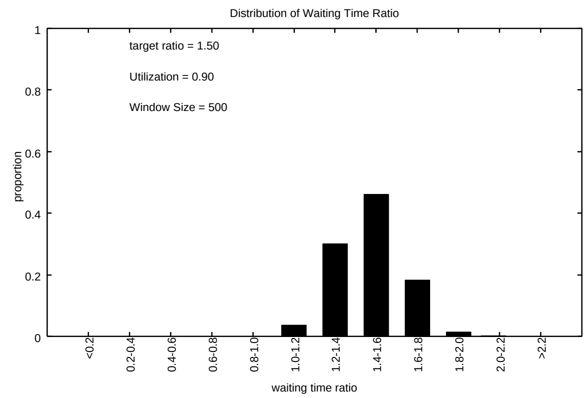



window=10000

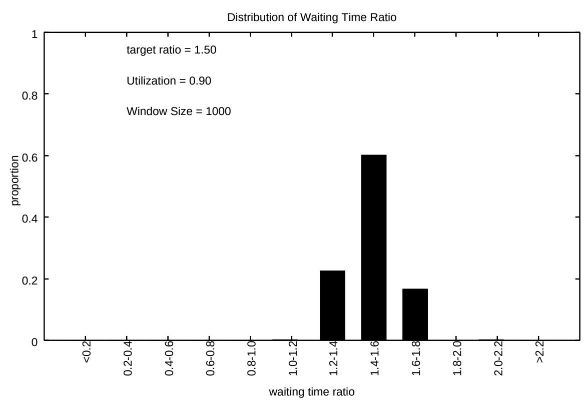
**Figure 2. System utilization $\rho = 0.6$, target ratio $r_{i,i+1} = 1.1$**



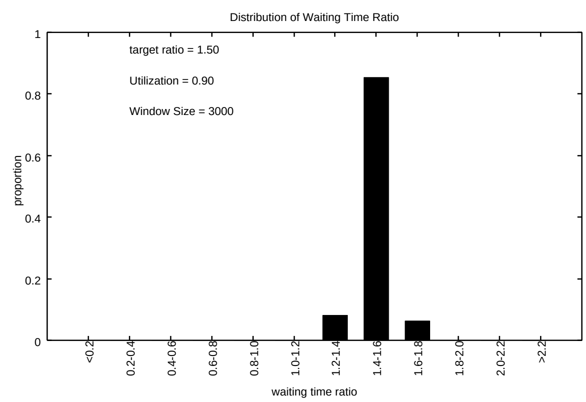

window=100



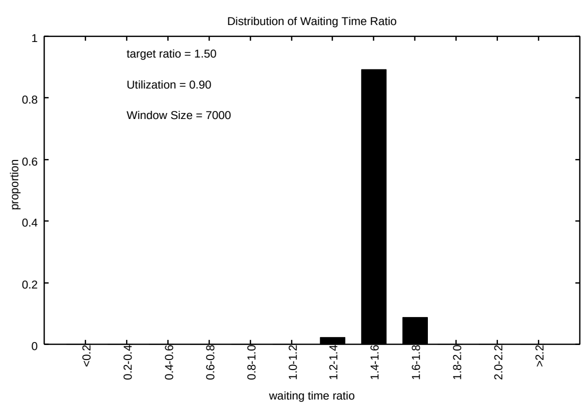

window=500



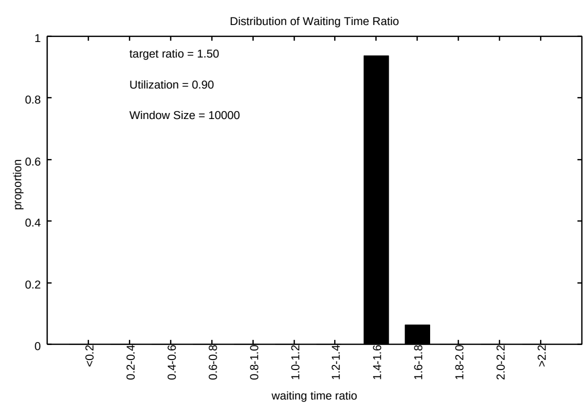

window=1000




window=3000




window=7000




window=10000

**Figure 3. System utilization $\rho = 0.9$, target ratio $r_{i,i+1} = 1.5$**