Proportional Delay Differentiation Provision by Bandwidth Adaptation of Class Based Queue Scheduling

Angelos Michalas, Malamati Louta, Paraskevi Fafali, George Karetsos, and Vassilios Loumos

National Technical University of Athens, Division of Computer Science,
Department of Electrical and Computer Engineering
9 Heroon Polytechneiou Str., GR-15773 Athens, Greece
tel: +30210 7721447 fax: +30210 7722538
email: michalas@central.ntua.gr

Abstract

Supporting Quality of Service (QoS) over the Internet is a very important issue and many mechanisms have been already devised or are under way towards achieving this goal. One of the most important approaches is the so called Differentiated Services (DiffServ) architecture, which provides a scalable mechanism for QoS support in a TCP/IP network. The main concept underlying DiffServ is the aggregation of traffic flows at an ingress (or egress) point of a network and the marking of the IP packets of each traffic flow according to several classification criteria. Diffserv is classified under two taxonomies: the absolute and the relative. In absolute DiffServ architecture, an admission control scheme is utilized to provide QoS as absolute bounds of specific QoS parameters. The relative DiffServ model offers also QoS guarantees per class but in reference to the guarantees given to the other classes defined. In this paper, relative proportional delay differentiation is achieved based on Class Based Queueing (CBQ) scheduler. Specifically, the service rates allocated to the classes of a CBQ scheduler are frequently adjusted in order to obtain relative delay spacing among them. The model presented can also be exploited in order to meet absolute delay constraints in conjunction with relative delay differentiation provision. Simulation experiments verify that our model can attain relative as well as absolute delay differentiation provided that the preconditions posed are satisfied.

Keywords: Quality of Service, Differentiated Services, Proportional Relative Differentiation, Scheduling, Class Based Queueing

1 INTRODUCTION

Service differentiation is considered to be of outmost importance for QoS provisioning in IP networks due to the high variations of the connection requirements posed by Internet users and the statistical in general nature of the generated traffic, which the last years is presenting an exponential increase. The research community has focused on two different techniques to provide QoS differentiation to end users: the *Integrated Services* (*Int-Serv*) [1] and the *Differentiated Services* (*DiffServ*) [2][3] approach. The major difference between Int-Serv and Diffserv architecture is the granularity of service differentiation. IntServ exploits the resource reservation concept. Each application requests levels of QoS in terms of service rate or end-to-end delay and the network grants or rejects requests according to its available resources. However, the Int-Serv model lacks in scalability and manageability, since all routers must maintain per-flow state. A new promising approach, the DiffServ, as proposed by the IETF Differentiated Services Working Group [2], allows traffic flows to be aggregated into a finite number of service classes that receive different routing treatment. The mapping from individual flows to service classes is determined at the edge routers based on the requirements and the characteristics of the associated application. Core routers forward each packet according to its class. This way, the model provides service differentiation on a per hop basis (*Per-Hop Behaviors-PHB*) [4] for large aggregates of network traffic. DiffServ provides quality assurances at traffic aggregate level and not at application flow level. There exist two directions in the DiffServ architecture: the absolute and the relative.

In absolute DiffServ [5], an admission control scheme is used [6] to provide QoS guarantees as absolute bounds of specific parameters such as bandwidth, packet delay, packet loss rate, or packet delay variation. A request is rejected if sufficient resources are not available so as to provide the desired quality. To deliver end to end performance guarantees, passive or active monitoring [7][8] is required along a specific connection before its establishment and throughout its lifetime. Thus, for any admitted request, the appropriate resources are reserved and the level of performance of the connection is assured.

The relative DiffServ model [9] supports QoS guarantees per class in reference to guarantees given to other classes. The only assurance coming from the network is that high priority classes receive better (or at least not worse) service treatment than lower priority classes. QoS parameter values of a specific connection depend on the network load since no admission control or resource reservation mechanism exists. The simplicity and ease of deployment of relative service differentiation have been viewed as its major advantages over the absolute differentiation scheme. Proposals for relative per class DiffServ application define service differentiation qualitatively [17][12], in terms that higher classes experience lower delays and losses than lower classes. Recent research studies focus on a relative service differentiation model, namely *Proportional Delay Differentiation* (*PDD*) [10][11], which controls the ratios of delays or loss rates of successive priority classes in order to be held constant.

Considering the proportional delay differentiation model, the ratios of packet delays of successive priority classes are equal to the ratio of their corresponding *Delay Differentiation Parameters* (*DDPs*). In most research efforts carried out, proportional delay DiffServ is achieved by employing Priority Based or Link Sharing Schedulers. In Section 3, the different scheduling algorithms that facilitate proportional delay differentiation will be briefly highlighted. Our work presents similar characteristics to the *Dynamic-Weighted Fair Queueing* (*D-WFQ*) scheduler [13], an extension of *Weighted Fair Queueing* (*WFQ*) scheduler [13], in which the weights of each class are dynamically adjusted so that predefined delay spacing among classes can be maintained.

A similar attempt to provide proportional delay differentiation through the *Class Based Queueing* (*CBQ*) Link Sharing scheduler [16] by dynamically adapting the bandwidth share assigned to each of the defined service classes is presented in this paper. The bandwidth adaptation values are estimated based on the current arrival rates and the queue loads of each service class. The proposed model will be henceforth called *Dynamic Bandwidth adaptation Class Based Queueing* (*DB-CBQ*). Apart from providing relative constraints, the framework presented is also capable of enforcing absolute delay bounds. Having the potential to offer strict and relative quality assurances, this model proves to be very a very significant and powerful tool in the hands of network managers for providing diversified quality of service. Additionally, our model exploits the CBQ scheduler, which is considered to be the most appealing scheduler for differentiated service provision and thus is widely supported, while modifications to the communication infrastructure, such as the architecture of routers (buffer managers, schedulers etc) are not required.

The rest of the paper is organised as follows: In section 2, the proportional delay differentiation model and its feasibility conditions are formulated, while the area of feasible delay differentiation parameters for three service classes is determined. Additionally, existing scheduling algorithms that support proportional delay differentiation are briefly reviewed. In section 3, the CBQ scheduler, the architecture of our proposed DB-CBQ module and the method adopted to provide both absolute and relative delay differentiation are presented. Section 4 describes various traffic scenarios that were simulated using the NS Network Simulator [34] and discusses the results acquired. Finally, in section 5, concluding remarks are made and future plans are highlighted.

2 PROPORTIONAL DELAY DIFFERENTIATION MODEL

2.1 Average delays per class in the PDD Model

Let us assume the existence of N service classes. d_i, δ_i denote the average queueing delay of class-i packets and the class-i DDP value, respectively. As already mentioned, the PDD model aims to control the delay ratio of packets belonging to different classes based on their DDPs. Specifically, the ratio of average delays between two classes i, j is fixed to the ratio of their corresponding DDPs. Thus, the following equation holds:

$$\frac{d_i}{d_j} = \frac{\delta_i}{\delta_j} \qquad 1 \le i, j \le N \tag{1}$$

This model applies with the same semantics to all network load conditions. The network operator for example may specify that class-i packets must experience double average delay than class-$i+1$ packets, independently of whether the delays are in the order of a few or hundreds of milliseconds.

In the rest of this section, the conventions specified in [10] are adopted. Particularly, it is assumed that higher classes experience lower queueing delays $(\delta_1 > \delta_2 > ... > \delta_N > 0)$ and class-1 is considered to be the reference class with $\delta_1 = 1$. The following equations should then hold:

$$d_i = \delta_i d_1 \qquad i = 2,...,N \tag{2}$$

In [10], the authors have shown that for a work-conserving scheduler [20] that its classes satisfy (2), the average delay in class-i should be:

$$d_i = \frac{\delta_i q_{ag}}{\sum_{n=1}^{N} \delta_n \lambda_n L_n} \qquad i = 1,...,N \tag{3}$$

where λ_n denotes the average traffic arrival rate (in terms of packets per second), L_n the average packet size, and q_{ag} the average backlog experienced in a *First Come First Served* (*FCFS*) scheduler [26] which has the same capacity and serves the same input traffic as the work conserving scheduler.

2.2 Feasibility study of the PDD Model

Up to this point it has been assumed that the PDD model is feasible. However, the formation of average packet delays in accordance with equation (1) for every set of DDPs values seems to be an impossible task. This is quite apparent, since each class has a minimum average delay, which can be estimated if the specific class is given strict priority over the rest of the generated traffic. So, d_i can not be less than this minimum average delay value.

In [10], it is shown that the PDD model is feasible, if the following $N-1$ inequalities hold. At this point it should be noted that for notation simplicity the packet size is considered equal for all classes. Notation may readily be extended.

$$\sum_{i=k}^{N} \lambda_i \delta_i \geq (S / q_{ag}) \sum_{i=k}^{N} \lambda_i d_i^{sp} \qquad k = 2,...,N \tag{4}$$

where d_i^{sp} denotes the average delay of class-i in a *Strict Priority* (*SP*) scheduler [26] which serves class m with higher priority than class n (for $m > n$), λ_i is the traffic input rate of class-i, $S = \sum_{i=1}^{N} \lambda_i \delta_i$ and $q_{ag} = \sum_{i=1}^{N} \lambda_i d_i^{sp}$.

The application of (4) for three service classes, results to the following inequalities:

$$A\delta_3 - B\delta_2 \geq \Gamma \tag{5}$$

$$\Delta\delta_3 + \mathrm{E}\delta_2 \geq \mathrm{Z} \tag{6}$$

where $A = \lambda_1 d_1^{sp} + \lambda_2 d_2^{sp}$, $B = \lambda_2 d_3^{sp}$, $\Gamma = \lambda_1 d_3^{sp}$, $\Delta = \lambda_3 d_1^{sp}$ and $\mathrm{E} = \lambda_2 d_1^{sp}$.

The area of feasible δ_2, δ_3 can be computed by graphing the inequalities (5), (6) and finding the common area which satisfies them both (Figure 1).

Table 1 depicts the Γ/B, Γ/A, Z/E and Z/Δ values simulating a SP scheduler for $\lambda_1 = \lambda_2 = \lambda_3$ and for network utilization ranging from 94% to 99%.

Considering the results depicted in Table 1 and Figure 1, we may conclude that the DDPs ratios $1/\delta_2 = 2,\ 1/\delta_3 = 4$ as well as $1/\delta_2 = 1.5,\ 1/\delta_3 = 3$ are feasible for traffic load higher than 94%. The respective DDP values will be adopted in the context of our experiments.

2.3 Schedulers for the PDD model

In this section, existing schedulers that can support proportional delay differentiation are briefly reviewed. Our aim is mainly to enumerate them and not to give a thorough picture of the underlying considerations. Recent work on proportional delay differentiation in the literature has focused on *Priority Based* and *Link Sharing* Schedulers. Regarding the first category (Priority Based Schedulers), in the *Waiting-Time Priority* (*WTP*) scheduler [10][28] packets are assigned with priorities equal to their waiting time multiplied by their DDP coefficients. The *Proportional Average Delay* (*PAD*) scheduler [10] selects for transmission the packet from the class with the maximum normalized average delay, which in essence is the average delay divided by its DDP. The *Hybrid Proportional Delay* (*HPD*) scheduler [10] chooses to transmit the packet from the class with the maximum hybrid delay, thus constituting, a combination of PAD and WTP equivalents. The *Mean Delay Proportional* (*MDP*) scheduler [29] is similar to WTP except for the fact that in order to assign priorities to packets, it utilizes an estimate of the average delay of their classes instead of the actual waiting time of each packet.

Considering the second category (Link Sharing Schedulers), the *Proportional Queue Control Mechanism* (*PQCM*) [30], the *Backlog-Proportional Rate* (*BPR)* [20], the *Joint Buffer management* and *Scheduling* (*JoBS*) [31] and the *Dynamic Weighted Fair Queueing* (*D-WFQ*) [13] are all variants of the *Generalized Processor Sharing* (*GPS*) algorithm [32]. In a WFQ scheduler classes are served according to their weights, while in D-WFQ the weights of each class are dynamically adjusted so that predefined delay differences between service classes can be achieved. The importance of D-WFQ may be attributed to the fact that it is built upon a generic service discipline, which is widely applied to QoS routers in order to obtain relative delay differentiation. All the aforementioned algorithms adjust service rate allocations to classes in order to meet relative delay requirements. Their high importance lies behind the fact that due to their nature, they may also be exploited for absolute QoS provision, such as guaranteed rate or absolute delay constraints. On the other hand, Priority based schedulers cannot provide such guarantees.

The CBQ scheduler, which is adopted in our study for providing proportional delay differentiation, is based on several mechanisms that merge *Priority Queueing* (*PQ*) with *Fair Queueing* (*FQ*) capabilities. While CBQ internal mechanisms are quite complex, its use is rather simple. Network managers need to define the link-sharing hierarchy and assign the amount of bandwidth and the priority attributed to each class. Due to its intuitiveness, CBQ is considered to be the most appealing scheduler available today for differentiated services support. In the following sections, indicative evidence on the efficiency of our proposed DB-CBQ module will be provided. Specifically, it will be shown that by appropriately adapting the bandwidth share of classes on a CBQ scheduler, the required relative delay spacing among them can be obtained, provided that the predefined DDP's are feasible.

3 DYNAMIC BANDWIDTH ADAPTATION CBQ (DB-CBQ) SCHEDULER

The CBQ mechanism is based on a basic scheduler, which is usually a *Weighted Round Robin* (*WRR*) scheduler [26] controlled by a link-sharing scheduler. Incoming traffic is classified into the appropriate queue according to a set of filtering rules. The basic scheduler selects packets to send in a way that guarantees that each class receives at least its allocated link sharing bandwidth. The estimator measures the departure time between successive packets of each class and characterizes the classes as over-limit, under-limit or at-limit. A class is called over-limit if it has recently consumed more than its allocated bandwidth, under-limit if it has utilised less than its bandwidth share and at-limit otherwise. The link-sharing scheduler distributes the excess bandwidth according to the link sharing hierarchy and deactivates over-limit classes so that WRR does not service them until their suspension period ends. Additionally, the link-sharing scheduler assigns priorities to queues while not allowing any class to monopolize the link.

In our proposed DB-CBQ model, a *Backlog Monitor* component is included in the CBQ model, which measures the arrival rates of each class packets and the queues size. A *Bandwidth Manager* periodically adjusts the bandwidth share (i.e., service rate) of each class according to the values observed by the Backlog Monitor so that the relative delay constraints specified in (1) are satisfied. The architecture of the DB-CBQ is illustrated in Figure 2.

The bandwidth values that should be allocated to each class are periodically computed in a manner similar to the one described in BPR [21] and D-WFQ [13] models. Specifically, both schedulers adjust periodically the service rate of each class so that the delay spacing among them is properly controlled, on the basis of an estimation of the average packet delay of each class at time t. This delay is hereby denoted by $d_i(t)$. Specifically, considering the fact that a packet arriving at time t will be dequeued after the current queue load $q_i(t)$ has been serviced, BPR approximates $d_i(t)$ by $q_i(t)/r_i(t)$, where $r_i(t)$ is the service rate assigned to class-i at time t.

Applying this approximation to (1), the following formula is acquired:

$$\frac{r_i(t)}{r_j(t)} = \frac{\delta_j}{\delta_i} \frac{q_i(t)}{q_j(t)} \tag{7}$$

Additionally, it stands that the sum of the allocated service rates must be equal to the capacity of the link, hereby denoted by C. Thus, the following equation holds:

$$\sum_{i=1}^{N} r_i(t) = C \tag{8}$$

D-WFQ attempts to perform a more accurate approximation of $d_i(t)$. More specifically, supposing that for class-i, at time instance $t-U$, its queue is already occupied by load $q_i(t)$ and p packets arrive within time interval $(t-U,t)$, then the average delay time for the p packets may be estimated by the following formula:

$$\begin{aligned} d_i(t) &= \frac{\sum_{j=1}^{p} d_i^j(t)}{p} \\ &= \frac{q_i(t)/r_i(t) + \ldots + (q_i(t) + \lambda_i(t) \cdot U)/r_i(t)}{p} \\ &\cong \frac{q_i(t) + \frac{1}{2}\lambda_i(t) \cdot U}{r_i(t)} \end{aligned} \tag{9}$$

where, $d_i^j(t)$ is the delay of the j-th packet (among the p), $\lambda_i(t)$ is the moving average approximation of the packet arriving rate for class-i at time instance t and U denotes the time between two successive rate adjustments.

By substituting the last approximation of $d_i(t)$ to (1) the following equation is acquired:

$$\frac{r_i(t)}{r_j(t)} = \frac{\delta_j (q_i(t) + 0.5\lambda_i(t) \cdot U)}{\delta_i (q_j(t) + 0.5\lambda_j(t) \cdot U)} \tag{10}$$

In the proposed DB-CBQ scheduler, the PDD model is approximated in the following manner: the service rates of the CBQ are adjusted periodically according to (7) and (8) while the instantaneous $q_i(t)$ values are replaced by their averages based on their backlog history. In other words, the $q_i(t)$ of (7) is replaced by the parameter $ave\_q_i(t)$, which is the average size in bytes of queue-i up to time t, given by the following equation:

$$ave\_q_i(t) = \frac{\sum_{k=1}^{t} q_i(k)}{n} \tag{11}$$

The load of the input traffic per class does not add much to the computation of the service rates. At this point it should be noted that U constitutes a quite important parameter to our model. It is obvious that a small U would increase the processing load in the routers while a large U would result in ratios between waiting delays of classes that would not conform to (1). We observed that for 0.001sec < U < 0.1sec the behavior of this scheme is satisfactory.

3.1 Dynamic CBQ Scheduler Meeting Absolute Constraints

In this section, the ability of the DB-CBQ scheduler to provide both absolute and relative delay differentiation is demonstrated. This capability arises from the inherent nature of CBQ to provide bandwidth guarantees to its classes. Suppose that for a set of classes, apart from the Relative Delay Constraints (RDCs) of (1), Absolute Delay Constraints (ADCs) of type $d_i \le k \;\; m\sec$ have to be satisfied. Considering three service classes, an example of relative and absolute constraints could possibly be described by the next set of equations, where the first two constraints are RDCs and the last one is ADC:

$$\frac{d_1}{d_2} = \frac{1}{\delta_2} \tag{12}$$

$$\frac{d_1}{d_3} = \frac{1}{\delta_3} \tag{13}$$

$$d_3 \le k \;\; ms \tag{14}$$

In such a case, the service rate allocation algorithm introduced in the previous section should define service rates that satisfy (7) and (8) as well as a set of inequalities of the form:

$$\frac{ave\_q_i(t)}{r_i(t)} \leq k_i \quad ms \tag{15}$$

for each class with an ADC constraint imposed. It is obvious that in some cases the absolute constraints may result in an infeasible system of constraints. In such a case, the approach adopted in our model is that the RDCs are relaxed in order to satisfy the strict absolute constraints, since it is considered of outmost importance to meet the pre-specified absolute QoS characteristics with respect to the RDCs. For example, absolute constraints may be posed to a high priority class, composed of time critical applications very sensitive to delay characteristics, such as voice-over-IP (VoIP). It is apparent that the ADC of this service class must be satisfied even at the cost of relaxing the RDCs of lower priority service classes.

A high level overview of the rate allocation algorithm used to satisfy both absolute and relative constraints is presented in Figure 3. As a fist step, an estimate of class delays $d_i(t) = \frac{ave\_q_i(t)}{r_i(t)}$ is produced by utilising the current service rate allocation. Afterwards two check stages are performed. At the first stage it is examined whether the strict ADCs can be satisfied. The second stage considers the RDCs and is performed only in case the first stage is proved to be successful, i.e., no ADC violation exists. Depending on the outcome of the two stages, the heuristic introduced distinguishes the following cases:

- **Case 1**: **No violations are predicted**

  In this case, the service rate allocations remain unchanged.

- **Case 2**: **Prediction of ADC violation**

  In the event that ADC violation is predicted, the algorithm relaxes RDCs and tries only to satisfy the strict ADCs. To this respect, our model adopts the algorithm described in [33]. Specifically, a portion of the service rates of lower priority classes that meet their ADCs are exploited in order to increase accordingly the rate of the ADC violated classes. In essence, the available resources of service classes are utilised in order to succeed in satisfying the ADCs of all service classes. The process is iterated until an ADC violation does not exist or all available resources have been redistributed. If after the completion of the reassignment process there are still classes which violate their ADCs, the extended version of the algorithm is performed, which loosens the strict ADCs of the lower priority service classes in favour of the higher order classes. Thus, the service rates of the higher order classes may still be increased by reducing accordingly the rates of lower priority classes – up to a minimum service rate – even though the ADCs of the later are violated.

- **Case 3**: **Prediction of RDC violation**

  In case RDC violation is predicted at check stage 2 (no ADC violation existence), the new service rate values are calculated on the basis of (8) and (10) equations. If the current solution (service rate assignment to service classes) violates an ADC defined for a service class, its corresponding RDC is relaxed, and the new service rate is computed using (15). The service rates of the rest classes are re-calculated based on (8) and (10) with C reduced by the capacity already allocated to the ADC violated class.

4 EXPERIMENTAL RESULTS

In this section, some indicative results are provided in order to assess the proposed framework, which offers proportional delay differentiation provision in conjunction with satisfaction of strict absolute constraints. The results given aim at the provision of indicative evidence of the efficiency and effectiveness of the proposed DB-DBQ model. In the following, three sets of experiments will be used for demonstrating these aspects. The first experiment considers only the provision of Relative Service Differentiation, while the second one incorporates the provision of Absolute Service Differentiation as well. The simulation topology adopted entails the simplest possible case, as it comprises only one router. However, this consideration will be changed in the third set of experiments, which considers the end-to-end service differentiation provision across a path of DB-CBQ schedulers. Nevertheless, the topology adopted for the realisation of the first and the second set of experiments enable the acquisition of an initial set of indicative results that show the behaviour of our DB-CBQ model. In order to test the performance of the proposed framework, we have used the NS2 Network Simulator [34] that is developed by the National Berkley Labs as the simulation platform, enhanced with the novel features of our model for the acquisition of the new service rate configuration.

4.1 Relative Service Differentiation

Figure 4 illustrates the topology adopted for the realisation of the first set of experiments aiming at Relative Service Differentiation provision. Three source nodes s_1, s_2, s_3 generate traffic to their destinations d_1, d_2, d_3, respectively. Incoming packets are classified into three classes with *class-1* having the lowest priority and *class-3* the highest. Packets from s_1 to d_1 are classified as *class-1* packets, from s_2 to d_2 as *class-2* packets and from s_3 to d_3 as *class-3* packets. All packets, in their way from their sources to their destinations pass through a DB-CBQ scheduler with output link capacity of 10Mbps. In our study we considered that the source nodes generate *Constant Bit Rate* (*CBR*) traffic with random noise in the scheduled departure times, thus eliminating the periodic input pattern effect that may have resulted otherwise. The packet length is taken equal to 1Kbyte for all service classes. This choice does not simplify or limit the model presented herein. Since the $ave\_q_i(t)$ is measured in bits (equation 11) variable or asymmetric packet length per class could have been used instead. However, this consideration enables us to acquire a set of comparable results in terms of average and individual packet delays. The DB-CBQ module adjusts the service rates regularly after U =0.1 sec in order to satisfy the required RDCs specified. The duration of the experiment was 200sec, while several runs (50 runs per experiment) have been performed to validate the results acquired. At this point, it should be noted that a major assumption is that the queue length for all service classes is infinite in order for all service class packets not to experience any losses. The experimentation with limited buffer space is considered to be a separate issue and is left as future work.

In the context of this experiment and as a first step, the DDP ratios are set to $1/\delta_2 = 2$, $1/\delta_3 = 4$ with $\delta_1 = 1$ (service *class-1* is the reference class), which are in line with the provisions specified in Section 2.2, for equal total input load rates for all classes ($\lambda_1 = \lambda_2 = \lambda_3$) and network utilisation ranging from 94% to 99%. Table 2 shows the average packet delays in msec for each service class as well as their delay ratios for total input loads higher than 94%.

By changing the DDP parameters to $1/\delta_2 = 1.5$, $1/\delta_3 = 3$ the results obtained are depicted in Table 3. All other parameters remain intact. For total load rates close or even less than 94% we observed, similarly to the D-WFQ case, small deviations from the PDD model. This should be attributed to the fact that the CBQ scheduler never idles, so, under moderate load conditions and in the event that the classes assigned with high service rate have no packets to forward, it can serve packets of low priority classes. Thus, the delay of the packets of the lower rated service classes may be reduced.

Table 4 presents the results obtained when the DDP ratios are taken equal to $1/\delta_2 = 2$, $1/\delta_3 = 4$, as in the first step, but in the current stage their input loads are varying. We observe that the total input load remains equal to 97.37%. Ideally, the DB-CBQ should be able to meet the DDP constraints independently of the class load distribution. However, in this experiment also, we experienced acceptable deviations when moderate load conditions are assumed (approximately less or equal to 15%) for high priority classes.

In the above experiments the standard deviation from the average packet delays ranges from 10.3 msec to 25.3 msec for packets of *class-1*, from 9.93 msec to 24.8 msec for packets of *class-2* and from 9.43 msec to 16.722 msec for packets of *class-3*. Thus, indicatively, for service *class-1*, the 95% confidence interval gives an acceptable range for individual packet delays ± 0.1718 msec around the average packet delay in the worst case i.e., average packet delay is equal to 25.3 msec.

Finally, the framework considered is slightly altered by considering traffic generation that follows a Pareto distribution with shape parameter $a = 1.8$, in order to test the performance of our proposed module in an environment with highly burstly characteristics. The total input loads are taken equal to 97.37% and

99.08%. The average delays, as well as the individual delays of successive packets experienced by each service class throughout the simulation are depicted in Table 5 and Figure 5 respectively. Figure 5 exhibits large delay variations for the low and medium delay classes. The large variations illustrated are due to the silent periods and bursts of the Pareto sources considered in conjunction with the generally high service rate portion assigned to them (in relation to the lower priority class). However, the results displayed provide strong evidence of the efficiency of our model, which accomplishes proportional delay differentiation under more pragmatic traffic examples [35]. The bursts produced by Pareto are appropriately handled according to the DDPs specified per class. Finally, because of the infinite variance property of this distribution, confidence intervals are not determined for the average delay measurements.

As a next step, the effect introduced by adopting a different service rate adjustment period is examined. Specifically, the length of the update period U is reduced from 0.1 sec to 0.05 sec. Table 6 depicts the results stemming from this adjustment. From the values obtained, it can be concluded that generally the DDP ratios are more closely approximated. Furthermore, the standard deviation from the average packet delays is reduced about 1 msec for each service class. However, the complexity introduced due to the frequent service rate adjustments is highly increased, thus, not substantiating the improvement achieved.

4.2 Relative and Absolute Service Differentiation

In this section, we present the results obtained when considering relative in conjunction with absolute service differentiation provision. The same simulation topology and assumptions as in the first set of experiments are adopted, while the source nodes generate CBR traffic of equal load. The DDPs are taken equal to $1/\delta_2 = 2$, $1/\delta_3 = 4$, and the absolute constraint ADC $d_3 < 30ms$ is also taken into account. For total input load of up to 96% the results acquired were the same as those presented in Table 2. However, when considering loads greater than 96% the application of the RDCs for the acquisition of a new service rate assignment, leads to unacceptable packet delays (i.e., above 30msec) for the third service class, since its ADC is violated. Thus, for these loads, the RDC $\frac{d_1}{d_3} = \frac{1}{4}$ is relaxed and a new service rate allocation is found in order to satisfy the ADC. The average delays for input loads greater than 96% are depicted in Table 7. It is evident that in both load conditions RDC d_1/d_2 holds true, while d_3 (average value) falls from 31.5 ms to 27 ms. In this experiment the standard deviation for the mean delays gives similar values with those referenced in the CBR experiments of section 4.1. Thus, the 95% confidence intervals give resembling variations of individual packet delays from the average packet delay measurements.

4.3 End to end Service Differentiation

The aim of this set of experiments is to attain end-to-end performance measurements, in case traffic traverses a path of DB-CBQ routers. To this end, the simulation topology adopted is illustrated in Figure 6.

Three flows of equal load, one from each class, are sent from s_1, s_2, s_3, source nodes to d_1, d_2 d_3 destination nodes through *router-1* and *router-2*, which employ DB-CBQ schedulers. Each flow follows a Pareto distribution with $a = 1.8$. Multiple runs of the experiment are performed with total input loads generated from flows to router-1 equal to 97.37% and 99.08%. Three Cross-Traffic sources on *router-2* send *class-1*, *class-2* and *class-3* packets of CBR traffic each to the respective destination sink, with rate adjusted so that the input load on this router reaches the utilization level of *router-1* as well. The service class DDP ratios are taken equal to $1/\delta_2 = 2$, $1/\delta_3 = 4$. We measured the end to end packet delays originated from *router-1*. The average packet delays observed among the classes and the graphs of the individual delays of successive packets in each service class throughout the simulation are depicted in Table 8 and Figure 7, respectively. As in Figure 5, large delay variations for the low and medium delay classes are observed. The large variations illustrated are due to the silent periods and bursts of the Pareto sources considered in conjunction with the generally high service rate portion assigned to them (in relation to the lower priority class).

The obtained results demonstrate that the required end to end relative delay differentiation between classes is achieved for the average packet delays as well as for the individual delays of successive packets.

Finally, an additional DB-CBQ router is introduced to the simulation topology of Figure 6. The results acquired lead to almost the same delay differentiation ratios. Thus, we conclude that as the number of hops increases, the end-to-end delay differentiation ratios converge to the specified DDPs.

5 CONCLUSIONS

Relative service differentiation is a promising approach for building Diffserv-enabled networks due to its simplicity in deployment and management. In this paper, a DiffServ model has been presented, which may be exploited for relative as well as absolute service differentiation provision. Specifically, the DB-CBQ module is thoroughly presented, which in essence is an extension of the CBQ scheduler enhanced with the novel feature of dynamic adaptation of the service rates of its classes. The authors' choice to exploit the CBQ scheduler is based on the fact that it is considered to be the most appealing scheduler available today for differentiated services provision. Furthermore since the CBQ model is widely supported modifications to the communication infrastructure, such as the architecture of routers (buffer managers, schedulers etc) is not required.

The proposed DB-CBQ model adjusts the service rates of classes in a CBQ scheduler at specified time intervals, while the service rate allocation is provided in accordance to the queues' average load. Using feasibility inequalities and applying delay measurements of a strict priority scheduler, the area of feasible DDPs has been specified. Additionally, the manner in which absolute delay constraints are always satisfied is presented, bearing some times the cost of relaxing relative delay constraints of lower priority service classes. In the context of this study, three set of experiments have been conducted. The results obtained confirm the efficiency of our proposed model with respect to the provision of proportional delay differentiation for a set of feasible DDPs, and the satisfaction of both absolute and proportional delay constraints. Finally, a set of end to end delay measurements of traffic flows has been performed which exhibit that the end-to end delay differentiation ratios converge to the pre-specified DDPs.

There are several issues that need to be further investigated. Thus, directions for future work include, but are not limited to the following. First, the realisation of further wide scale trials using the framework presented herein, in a real networking environment. Second, the introduction of computational models for the service rates in CBQ classes, when moderate input load is considered (less than 94%). Our experiments showed that for moderate loads the best approximation of the PDD is achieved by using (8) and (10) (in which $q_i(t)$ and $\lambda_i(t)$ are replaced by their normalized averages), which however, results far from the desired values. As already stated, this should be attributed to the fact that the CBQ scheduler never idles. On account of this, under moderate load conditions and in case classes with high service rate assignment have empty queues, the CBQ may service packets of low priority classes, leading to reduction of the respective packet delay. This aspect needs more research effort despite the fact that at an underloaded link the delays experienced are very small for all classes regardless of the scheduling algorithm employed. Third, incorporation of the loss metric as a differentiation parameter (i.e., considering finite buffer space) to our model. This extension would lead to a scheme that would respond more effectively to the increasing user expectations.

6 REFERENCES

[1] P. White, "RSVP and Integrated Services in the Internet: A Tutorial", In *IEEE Communications Magazine*, Vol. 35, no. 5, May 1997, pp. 100-106.

[2] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[3] S. Blake, et al., "An Architecture for Differentiated Services", RFC 2475, December 1998.

[4] J. Heinanen, F. Baker, W. Weiss, J. Wrockawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.

[5] B. Teitelbaum, "QBone Architecture (v1.0)", Internet2 QoS Working Group Draft, http://www.internet2.edu/qos/wg/papers/qbArch/1.0/draft-i2-qbone-arch-1.0.html, August 1999.

[6] I. Stoika and H. Zhang, "LIRA: An approach for Service Differentiation in the Internet", In Proc. *NOSS-DAV*, 1998.

[7] ITU-T Rec. X.739, "Information Technology - Open Systems Interconnection, Systems Management Functions - Metric Objects and Attributes", 1992.

[8] ITU-T Rec. X.738, "Information Technology - Open Systems Interconnection, Systems Management Functions - Summarization Function", 1993.

[9] C. Dovrolis, D. Stiliadis, "Relative Differentiated Services in the Internet: Issues and Mechanisms", In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 27, No 1, June 1999.

[10] C. Dovrolis, D. Stiliadis, P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling", In *IEEE/ACM Transactions in Networking*, February 2002.

[11] C. Dovrolis, D. Stiliadis, Parmesh Ramanathan, "Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping", In Proc. of *International Workshop on Quality of Service* (*IWQoS*), Pittsburgh PA, June 2000.

[12] D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service", In *IEEE/ACM Transactions on Networking*, Vol. 6 (4), August 1998.

[13] 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 Proc. *IEEE International Conference on Computer Communications and Networks* (*ICCCN*), October 2000.

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

[15] A. Michalas, P. Fafali, M.Louta, and V. Loumos, "Proportional Delay Differentiation Employing the CBQ Service Discipline", In Proc. of *7th International Conference on Telecommunications* (*ConTEL*), Zagreb, Croatia, June 2003.

[16] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks", In IEEE/ACM Transactions on Networking, Vol. 3 (4) pp. 365–386, August 1995.

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

[18] J. Heinanen, F. Baker, W. Weiss, J. Wrockawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.

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

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

[21] C. Dovrolis, D. Stiliadis, P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling", In Proc. of *ACM SIGCOMM*, September 1999.

[22] E. G. Coffman and I. Mitrani, "A Characterization of Waiting Time Performance Realizable by Single-Server Queues", Operational Results, Vol.28, May 1980, pp. 810–21.

[23] J. C. R. Bennett and H. Zhang, "Hierarchical Packet Fair Queuing Algorithms", In *IEEE/ACM Transactions on. Networking*, Vol. 5, October 1997, pp. 675–89.

[24] I. Stoika, H. Zhang, and T. Ng, "A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services", In Proc. of *ACM SIGCOMM*, September 1997.

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

[26] L. Kleinrock, "Queueing Systems", Volume II, John Wiley and Sons, 1976.

[27] J. Regnier, "Priority Assignment in Integrated Services Networks", Technical Report, LIDS-TH-1565, LIDS-MIT, December 1986.

[28] K. H. Leung, J. Lui, and D. Yau, "Characterization and Performance Evaluation for Proportional Delay Differentiated Services", In Proc. of *International Conference on Network Protocols* (*ICNP*), October 2000.

[29] T. Nandagopal, N.Venkitaraman, R.Sivakumar, and V.Bharghavan, "Delay Differentiation and Adaptation in Core Stateless Networks", In Proc. of *IEEE INFOCOM*, March 2000.

[30] Y. Moret and S.Fdida, "A Proportional Queue Control Mechanism to Provide Differentiated Services", In Proc. of *International Symposium on Computer and Information Systems* (*ISCIS*), October 1998.

[31] J. Liebeherr and N.Christin, "Buffer Management and Scheduling for Enhanced Differentiated Services", Technical Report CS-2000-24, University of Virginia, August 2000.

[32] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks", In *IEEE Proceedings*, Vol. 83, pp. 1374-1399, October 1995.

[33] M. Louta, A. Michalas, E. Loutas and V. Loumos, "Quality of Service Management in IP networks", In Proc. of *7th International Conference on Telecommunications* (*ConTEL*), Zagreb, Croatia, June 2003.

[34] S. McCanne and S. Floyd, http://www-mash.cs.berkeley.ed u/ns/ns.html Network Simulator, 1996.

[35] W. Willinger, M.S.Taqqu, R.Sherman, and D.V.Wilson, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level", In Proc. of *ACM SIGCOMM*, pp. 100-113, September 1995.

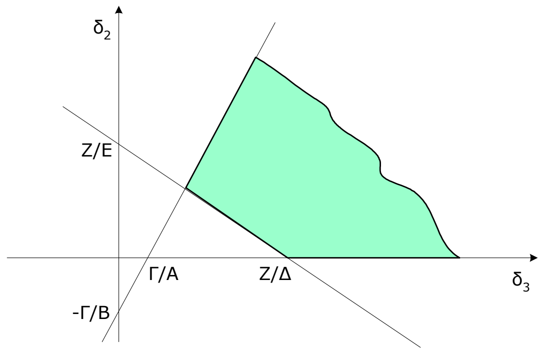

Figure 1: Area of feasible δ_2 and δ_3

Total Input Load %	Γ/B	Γ/A	Z/E	Z/Δ
94.23	1	0.189	0.560	0.560
95.08	1	0.195	0.140	0.140
95.37	1	0.012	0.031	0.031
95.65	1	0.012	0.031	0.031
95.94	1	0.011	0.030	0.030
96.23	1	0.012	0.031	0.031
96.51	1	0.012	0.030	0.030
96.8	1	0.012	0.032	0.032
97.08	1	0.012	0.032	0.032
97.37	1	0.013	0.032	0.032
97.65	1	0.013	0.034	0.034
97.94	1	0.013	0.034	0.034
98.23	1	0.014	0.035	0.035
98.51	1	0.015	0.038	0.038
98.79	1	0.016	0.039	0.039
99.08	1	0.015	0.039	0.039

Table 1: δ_2 and δ_3 intercepts of graphs

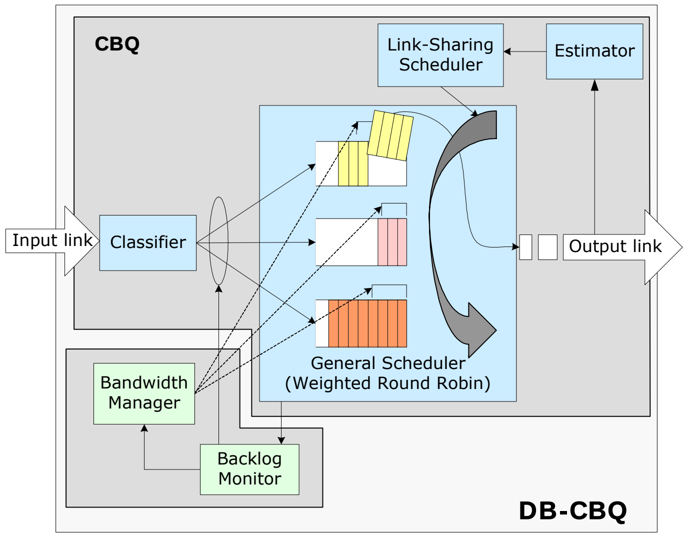

Figure 2: DB-CBQ Modules

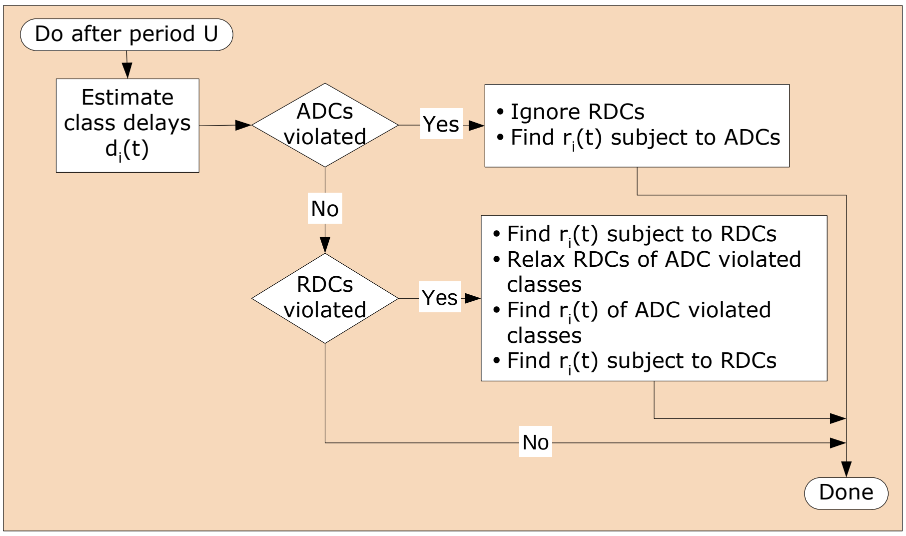

Figure 3: Service rates computation algorithm for absolute and relative delay differentiation

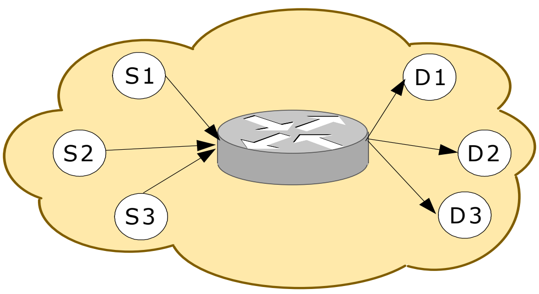

Figure 4: Simulation topology

Total Input Load %	d_1 (msec)	d_2 (msec)	d_3 (msec)	d_1/d_2	D_1/d_3
94.23	77.543	35.958	18.476	2.156	4.197
95.08	80.634	41.456	20.987	1.945	3.842
95.37	88.498	45.208	23.689	1.958	3.736
95.65	104.323	51.703	27.342	2.018	3.815
95.94	110.428	54.891	29.026	2.012	3.805
96.23	113.768	56.792	29.410	2.003	3.868
96.51	116.046	57.778	29.935	2.009	3.877
96.8	117.684	58.646	29.618	2.007	3.973
97.08	119.063	59.599	30.158	1.998	3.948
97.37	120.234	60.368	30.469	1.992	3.946
97.65	121.309	60.941	30.724	1.991	3.948
97.94	122.374	61.677	30.825	1.984	3.970
98.23	123.381	62.215	31.045	1.983	3.974
98.51	124.303	62.723	31.329	1.982	3.968
98.79	125.214	63.301	31.455	1.978	3.981
99.08	126.155	63.701	31.636	1.980	3.988

Table 2: Delay differentiation ratio = 1:2:4

Total Input Load %	d_1 (msec)	d_2 (msec)	d_3 (msec)	d_1/d_2	d_1/d_3
94.23	74.547	39.864	21.848	1.870	3.412
95.08	79.495	42.824	28.773	1.856	2.763
95.37	85.570	46.549	29.875	1.838	2.864
95.65	104.212	67.846	35.778	1.536	2.913
95.94	110.185	73.372	38.702	1.502	2.847
96.23	113.641	74.666	38.787	1.522	2.930
96.51	115.665	76.635	39.784	1.509	2.907
96.8	117.574	77.314	39.551	1.521	2.973
97.08	118.965	78.319	40.281	1.519	2.953
97.37	120.089	77.951	39.318	1.541	3.054
97.65	121.086	80.079	40.981	1.512	2.955
97.94	122.074	81.039	41.235	1.506	2.960
98.23	123.024	81.316	41.328	1.513	2.977
98.51	123.912	82.186	41.732	1.508	2.969
98.79	124.842	83.175	42.033	1.501	2.970
99.08	125.722	83.639	42.078	1.503	2.988

Table 3: Delay differentiation ratio = 1:1.5:3

Input Load Per Class % $(\lambda_1,\lambda_2,\lambda_3)$	d_1 (msec)	d_2 (msec)	d_3 (msec)	d_1/d_2	d_1/d_3
(60,20,20)	760.397	328.195	176.44	2.3169	4.3097
(70,15,15)	665.147	258.98	155.966	2.5683	4.2647
(50,25,25)	856.752	416.91	212.91	2.055	4.024
(25,50,25)	4311.76	2157.79	1140	1.9982	3.7822
(25,25,50)	1160.84	581.523	303.9	1.9962	3.8198

Table 4: Variable Class Loads. Delay differentiation ratio = 1:2:4

Total Input Load %	d_1 (msec)	d_2 (msec)	d_3 (msec)	d_1/d_2	d_1/d_3
97.37	119.733	59.711	30.267	2.0052	3.9560
99.08	125.999	61.894	31.689	2.0357	3.9761

Table 5: Pareto Input Loads. Delay differentiation ratio = 1:2:4

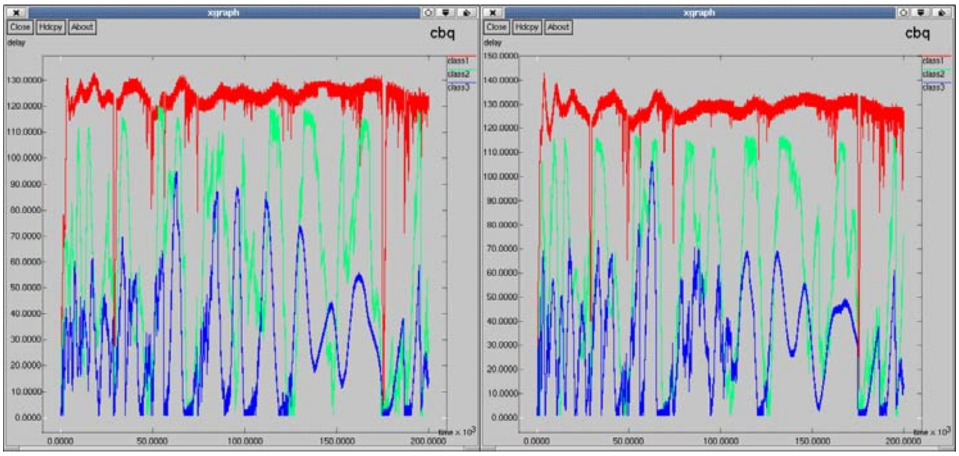

Figure 5: Delays of successive packets under 97.37% and 99.08% input loads

Total Input Load %	d_1 (msec)	d_2 (msec)	d_3 (msec)	d_1/d_2	d_1/d_3
94.23	76.164	35.697	23.058	2.134	3.303
95.08	81.254	40.987	20.952	1.982	3.878
95.37	89.163	45.480	23.824	1.961	3.743
95.65	103.575	50.834	26.969	2.038	3.841
95.94	110.623	54.649	28.915	2.024	3.826
96.23	113.723	56.721	29.526	2.005	3.852
96.51	116.058	57.782	29.655	2.009	3.914
96.8	117.739	58.544	29.723	2.011	3.961
97.08	119.044	59.524	30.117	2.000	3.953
97.37	120.264	60.319	30.420	1.994	3.953
97.65	121.397	61.042	30.623	1.989	3.964
97.94	121.371	61.614	30.810	1.970	3.939
98.23	123.345	62.121	30.981	1.986	3.981
98.51	124.300	62.649	31.381	1.984	3.961
98.79	125.221	63.199	31.279	1.981	4.003
99.08	126.136	63.680	31.500	1.981	4.004

Table 6: Rate adlustment period 0.05sec. Delay differentiation ratio = 1:2:4

Total Input Load 100%	d_1 (msec)	d_2 (msec)	d_3<30 (msec)	d_1/d_2
96.23	113.671	56.601	27.378	2.008
96.51	115.775	57.674	27.733	2.007
96.8	117.522	58.216	27.368	2.019
97.08	118.992	59.206	27.770	2.010
97.37	120.109	59.990	27.352	2.002
97.65	121.188	60.703	27.636	1.996
97.94	122.198	61.273	27.484	1.994
98.23	123.197	62.094	27.355	1.984

98.51	124.142	62.371	27.485	1.990
98.79	125.116	62.919	27.341	1.989
99.08	126.050	63.413	27.248	1.988

Table 7: Delay differentiation ratio = 1:2:4 and d_3< 30ms

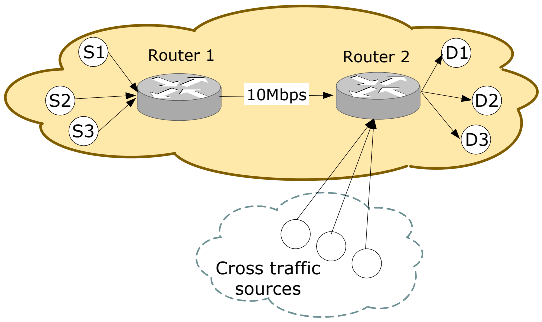

Figure 6: Traffic traversing two DB-CBQ enabled routers

Total Input Load %	d_1 (msec)	d_2 (msec)	d_3 (msec)	d_1/d_2	d_1/d_3
97.37	210.322	105.088	52.7307	2.0014	3.9886
99.08	219.685	106.643	55.0483	2.0600	3.9908

Table 8: End to end delay differentiation

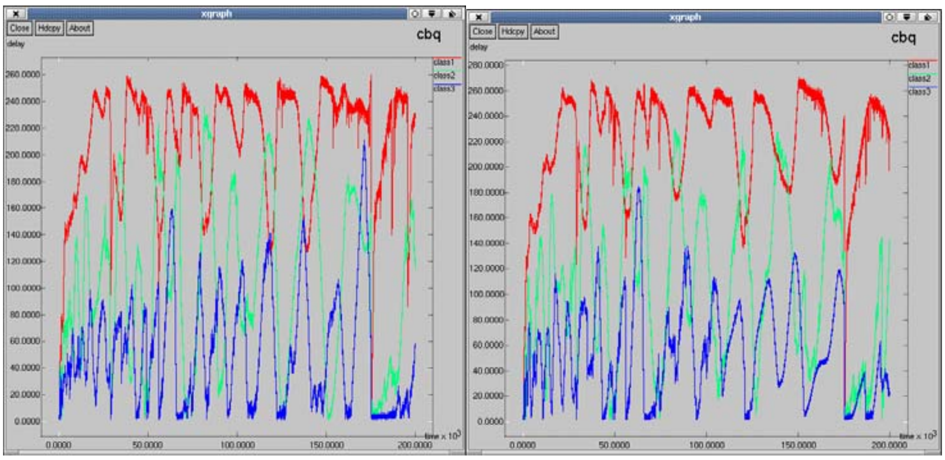

Figure 7: End to end packet delays under 97.37% and 99.08% input loads