

Proportional Delay Differentiation Employing the CBQ Service Discipline

Angelos Michalas, Paraskevi Fafali, Malamati Louta, Vassilios Loumos

School of Electrical and Computer Engineering

National Technical University of Athens

Athens, Greece

michalas@central.ntua.gr

Abstract— The DiffServ architecture provides a scalable mechanism for QoS introduction in a TCP/IP network. The idea of DiffServ is based on the aggregation of traffic flows at an ingress (or egress) point of a network and the IP packet marking for different priority flows, according to several classification criteria. Two approaches exist in the DiffServ architecture: the absolute and the relative. In absolute DiffServ, an admission control scheme is used to provide QoS guarantees as absolute bounds of specific QoS parameters. The relative DiffServ model provides QoS guarantees per class expressed with reference to guarantees given to the other classes defined. Our study targets at providing relative proportional delay differentiation service based on Class Based Queue (CBQ) scheduler. The main idea is to frequently adjust the service rates allocated to classes of a CBQ scheduler in order to achieve relative delay spacing among classes. The simulation experiments conducted show that our model can attain relative delay, provided that the required Delay Differentiation Parameters (DDPs) are feasible.

Keywords- *Quality of Service, Differentiated Services, Proportional Relative Differentiation, Scheduling, Class Based Queueing.*

I. INTRODUCTION

Service differentiation is considered to be of utmost 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 concentrated on two different techniques to provide QoS differentiation to customers of packet switched networks. First, the Integrated Services (Int-Serv) [1][2] approach. Second, the Differentiated Services (DiffServ) [3][4][5] approach. The major difference between Int-Serv and DiffServ architecture is the granularity of service differentiation. The IntServ concept lies in resource reservation. Each application requests levels of service in terms of service rate or end-to-end delay. The network accepts or rejects requests according to its resources availability. However, the Int-Serv approach faces potential problems concerning scalability and manageability, since all routers must maintain per-flow state. The main strength of DiffServ, as proposed by the IETF Differentiated Services Working Group [3], is that it allows IP traffic to be classified

into a finite number of service classes that receive different routing treatment. Routers at the network edges classify packets into predefined service classes based on the demand requirements and characteristics of the associated application. Core routers forward each packet according to its class. By this way, the model provides service differentiation on each node (Per-Hop behaviors) [6] for large aggregates of network traffic. DiffServ achieves scalability and manageability by providing quality per traffic aggregate and not per application flow. Two directions exist in the DiffServ architecture: the absolute and the relative.

In absolute DiffServ [7], an admission control scheme is used [8] to provide QoS guarantees as absolute bounds of specific QoS parameters such as bandwidth, packet transfer delay, packet loss rate, or packet delay variation (jitter). A connection request is rejected if sufficient resources are not available in the network so as to provide the desirable assurances. End to end performance requires passive or active monitoring procedures [9][10] along a specific connection before its establishment and throughout its lifetime. Thus, for any admitted user the appropriate resources are reserved and the performance level of the connection is assured.

The relative DiffServ model [11] provides QoS guarantees per class in reference to guarantees given to other classes. The only assurance from the network is that higher classes receive better service treatment than lower classes. QoS parameter values for a connection depend on the current network load since there is no admission control mechanism and resource reservation mechanisms. Relative service differentiation is a simple and easily deployed approach compared to the absolute differentiation service. Proposals for relative per class DiffServ QoS define service differentiation qualitatively [12][13], in terms that higher classes receive lower delays and losses from lower classes. Specifically research effort has focused on a qualitative relative differentiation scheme named proportional DiffServ [14][15], which controls the ratios of delays or loss rates of successive priority classes in order to be constant. In this Proportional Delay Differentiated (PDD) model, given two consecutive priority classes, it can be guaranteed that packet delay or the loss rate for the higher priority class will be a pre-specified portion of packet delay or loss rate of the lower priority class.

Considering the proportional delay differentiation model (PDD) [14] the ratios of packet delays of successive priority classes are constant and equal to the ratio of their corresponding delay differentiation parameters (DDPs).

Current work on proportional delay DiffServ model are based on Priority Based Schedulers or Link Sharing Schedulers. In the first category, the Wait-Time Priority (WTP) scheduler [16] assigns priorities to packets equal to their waiting time multiplied by their DDPs coefficients. The Proportional Average Delay (PAD) scheduler [14] 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 [14] chooses to transmit the packet from the class with the maximum hybrid delay, constituting thus, a combination of PAD and WTP equivalents. The Mean Delay Proportional (MDP) scheduler [17] is similar to the WTP except from 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.

In Link Sharing Schedulers, the Proportional Queue Control Mechanism (PQCM) [18], the Backlog-Proportional Rate (BPR)[14], the Joint Buffer management and Scheduling (JoBS)[19] and finally the Dynamic Weighted Fair Queueing (D-WFQ) [20] are all variations of the GPS algorithm [21]. They adjust service rate allocations of classes to meet relative delay QoS requirements. Their high importance is that because of their nature, they can also offer absolute services on a class such as guaranteed rate or absolute delay constraints. Priority based schedulers cannot provide such guarantees.

Our attention was drawn from D-WFQ, an extension of WFQ scheduler [22], in which the weights of each class are dynamically adjusted so that predefined delay differences between classes can be kept. 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 may be achieved. The importance of D-WFQ is that it is built upon a generic service discipline, which is widely applied to QoS routers, to achieve relative delay differentiation.

In this paper, we present a similar attempt to provide proportional delay differentiation through the Class Based Queueing (CBQ) link shared scheduler [23] by dynamically adapting the bandwidth assigned to each of the defined service classes. The bandwidth adaptation values are estimated based on the current arrival rates and the queue loads of each service class. We call this model Dynamic Bandwidth adaptation Class Based Queue (DB-CBQ).

CBQ scheduler, is based on several mechanisms that merge Priority Queueing (PQ) and fair capabilities. While CBQ internal mechanisms are quite complex, its use is quite simple. Network managers need to define the link-sharing hierarchy and assign the amount of bandwidth and priority to each class. Due to its intuitiveness, CBQ is considered the most appealing scheduler available today used to support differentiated services. We will show that by adapting appropriately the

bandwidth share of classes on a CBQ scheduler we can obtain the required relative delay spacing among them provided that the predefined DDP's are feasible.

This paper is organised as follows: In section II the proportional delay differentiation model and its feasibility conditions are presented, while the area of feasible delay differentiation parameters for three classes of service is determined. Section III describes, the CBQ scheduler, the architecture of our proposed DB-CBQ module and the method used to calculate its service rates to achieve proportional delay differentiation. In section IV, a set of results that we acquired by experimenting with the NS network simulator on selected traffic scenarios, are exposed. Finally, concluding remarks are made and future plans are discussed.

II. PROPORTIONAL DELAY DIFFERENTIATION MODEL

Let us assume that we have N service classes and the average queueing delay of class- i packets is d_i . As already mentioned, the PDD model [14] aims to control the delay ratio of packets of 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 δ_i, δ_j . Thus, the following equation holds:

$$\frac{d_i}{d_j} = \frac{\delta_i}{\delta_j} \quad 1 \leq i, j \leq N \quad (1)$$

The model applies with the same semantics to all load conditions of the network in which it is feasible. The network operator for example may specify that the average delay class- i packets are experiencing is double the average class- j packet delay, independently of whether the delays are in the order of a few or hundreds of milliseconds.

In the rest of the section, the conventions specified in [14] are adopted. Specifically, higher classes provide lower queueing delays ($\delta_1 > \delta_2 > \dots > \delta_N > 0$) and also class-1 is defined to be the reference class with $\delta_1 = 1$. The following equations should then hold:

$$d_i = \delta_i d_1 \quad i = 2, \dots, N \quad (2)$$

In the same paper, the authors have shown that for a work-conserving scheduler [24], which can enforce equation (2) in its classes, the average delay in class- i should be:

$$d_i = \frac{\delta_i q_{ag}}{\sum_{n=1}^N \delta_n \lambda_n L_n} \quad i = 1, \dots, N \quad (3)$$

where λ_n is the average arrival rate and L_n is the average packet size in class- n . The q_{ag} parameter denotes average backlog experienced in a FCFS scheduler [25] with the same capacity and the same input traffic.

However, a scheduler that can set the average delay of each class so that (1) holds for any DDPs values may not exist. This is quite apparent, since each class has a minimum average delay, which can be estimated if that class was given strict priority over the rest of the traffic. So, d_i can not be less than this minimum average delay value.

The paper [14] proves that the PDD model is feasible, if the following $N-1$ inequalities hold. It is noted that the packet size is taken equal for all classes:

$$\sum_{i=k}^N \lambda_i \delta_i \geq \frac{S}{q_{ag}} \sum_{i=k}^N \lambda_i d_i^{sp} \quad k = 2, \dots, N \quad (4)$$

where d_i^{sp} is the average delay of class- i in a strict priority scheduler [25] which serves class m with higher priority than class n for $m > n$, λ_i is the input rate of class- i ,

$$S = \sum_{i=1}^N \lambda_i \delta_i \quad \text{and} \quad q_{ag} = \sum_{i=1}^N \lambda_i d_i^{sp}.$$

For three classes by applying (4) we get:

$$A\delta_3 - B\delta_2 \geq \Gamma \quad (5)$$

$$\Delta\delta_3 + E\delta_2 \geq Z \quad (6)$$

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 $E = \lambda_2 d_1^{sp}$

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 (Fig. 1). For $\lambda_1 = \lambda_2 = \lambda_3$ and for network utilization from 94% to 99% the simulation of a strict priority scheduler provided the following values for Γ/B , Γ/A , Z/E and Z/Δ :

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

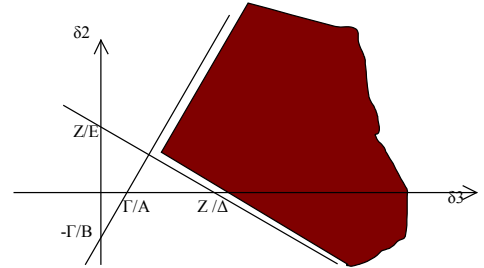


Figure 1. Area of feasible δ_2 and δ_3

Considering and Fig. 1, the DDPs $1/\delta_2 = 2$, $1/\delta_3 = 4$, as well as $1/\delta_2 = 1.5$, $1/\delta_3 = 3$ are feasible for traffic load higher than 94%. These DDPs will be used in our experiments.

III. DYNAMIC CBQ SCHEDULER

The CBQ component is based on a basic scheduler, which is usually a Weighted Round Robin scheduler (WRR) [25] 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 from queues 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 each class as over-limit, under-limit or at-limit. A class is called over-limit if it has recently used more than its allocated bandwidth, under-limit if it has used less than its allocated bandwidth and at-limit otherwise. The link-sharing scheduler distributes the excess bandwidth according to the link sharing structure and also makes over-limit classes inactive so that the WRR does not service them until their suspension period ends. Additionally, the link-sharing scheduler provides priorities to queues, while not allowing any class to monopolize the link.

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

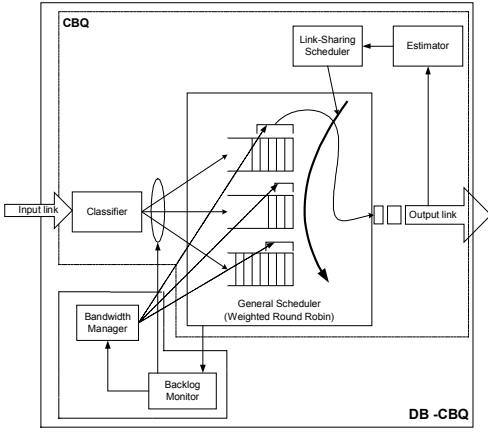


Figure 2. DB-CBQ Modules

In order to compute the bandwidth values to be assigned to each class periodically, we experimented with the service rate estimates specified in BPR [14] and D-WFQ [20] models. Both schedulers adjust periodically the service rate of each class so that the delay spacing among classes is properly controlled and (1) is satisfied. To achieve this they estimate the average packet delay of each class at time t . We denote this delay $d_i(t)$. Specifically, considering that a packet arriving at time instance t is serviced after the current queue load $q_i(t)$ has been serviced, BPR approximates $d_i(t)$ by $\frac{q_i(t)}{r_i(t)}$, where $r_i(t)$ is the service rate assigned to class- i at time t . Applying this approximation to (1) we acquire the following constraint:

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

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

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

D-WFQ attempts to perform a more accurate approximation of $d_i(t)$. Suppose that for class- i its buffer is already occupied by $q_i(t)$ at time $t-U$ and p packets are arriving within time interval $(t-U, t)$. Denoting by $d_i^j(t)$ the delay of the j -th packet (among these p) and $\lambda_i(t)$ the moving average approximation of the amount of incoming packets for class- i at time t , the average delay time of the p packets may be estimated by the following formula:

$$d_i(t) = \frac{\sum_{j=1}^p d_i^j(t)}{p} = \frac{q_i(t)/r_i(t) + \dots + (q_i(t) + \lambda_i(t) \cdot U)/r_i(t)}{p} \cong \frac{q_i(t) + \frac{1}{2} \lambda_i(t) \cdot U}{r_i(t)}$$

By applying the last approximation of $d_i(t)$ to (1) we get the following equation:

$$\frac{r_i(t)}{r_j(t)} = \frac{\delta_j (q_i(t) + 0.5 \lambda_i(t) U)}{\delta_i (q_j(t) + 0.5 \lambda_j(t) U)} \quad (9)$$

In our scheduler, we show that the PDD model is approximated if the service rates of the CBQ are adjusted periodically according to (7) and (8) by replacing the instantaneous $q_i(t)$ values by their averages based on their backlog history. In other words the $q_i(t)$ of (7) is substituted by the parameter $ave_q_i(t)$, given by the following equation:

$$ave_q_i(t) = \frac{\sum_{k=1}^t q_i(k)}{n} \quad (10)$$

which is the average size in bytes of queue- i up to time t . The load of input traffic per class does not add much to the computation of the service rates. Another parameter is the time between two successive service rate adjustments of the CBQ which is defined as U . It is obvious that a small U would increase processing load in the routers. A large U would result in ratios between waiting delays of classes that will not conform to (1). As in PQCM [18] we find that for $0.001\text{sec} < U < 0.1\text{sec}$ the behavior of the scheme is good and the influence of this parameter remains low.

IV. RESULTS

In this section, some indicative results are presented in order to assess the proposed framework, which allows for proportional delay differentiation service provision by adopting a CBQ scheduler. Specifically, the results attained indicate the efficiency and the effectiveness of the proposed DB-DBQ module.

In order to evaluate the performance of the proposed framework of this paper we used NS2 network simulator [26] developed by National Berkley Labs as the simulation platform. Fig. 3 shows the topology used in the simulations of this section. 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.

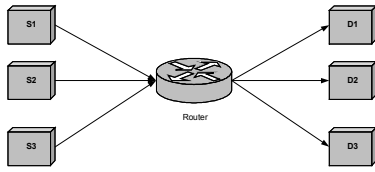


Figure 3. Relative differentiation simulation

All packets are passed from their sources to their destinations through a DB-CBQ scheduler with output capacity link of 10Mbps. The source nodes generate Constant Bit Rate (CBR) traffic. The packet length of incoming traffic is taken equal to 1Kbyte for all classes and the service rates are adjusted to DB-CBQ routers every 0.1 seconds. All experiments lasted 200 seconds.

TABLE II. Delay differentiation ratio = 1:2:4

Total Input Load %	d_1	d_2	d_3	d_1/d_2	d_1/d_3
94.23	102.177	53.833	26.837	1.8980	3.8073
95.08	103.272	54.342	27.122	1.9004	3.8077
95.37	106.885	55.282	27.717	1.9334	3.8563
95.65	113.873	56.528	29.557	2.0144	3.8526
95.94	116.233	59.126	29.914	1.9659	3.8855
96.23	117.718	60.020	30.278	1.9613	3.8879
96.51	118.909	59.757	30.777	1.9899	3.8635
96.8	119.979	61.023	30.542	1.9661	3.9282
97.08	120.922	60.827	30.265	1.9879	3.9954
97.37	121.814	61.115	30.775	1.9932	3.9582
97.65	122.746	60.879	30.784	2.0162	3.9872
97.94	123.583	62.001	31.496	1.9932	3.9237
98.23	124.447	63.279	32.117	1.9666	3.8747
98.51	125.277	63.358	32.511	1.9773	3.8533
98.79	126.18	63.806	32.041	1.9775	3.9380
99.08	127.031	64.266	32.813	1.9766	3.8713

In our first experiment, the DDPs are assumed to be $1/\delta_2 = 2$, $1/\delta_3 = 4$ and the input loads per class are equal ($\lambda_1 = \lambda_2 = \lambda_3$). TABLE II shows the average packet delays in msec of each service class as well as their delay ratios for total input loads above 94%.

TABLE III. Delay differentiation ratio = 1:1.5:3

Total Input Load %	d_1	d_2	d_3	d_1/d_2	d_1/d_3
94.23	102.171	72.077	36.096	1.4175	2.8305
95.08	103.796	72.209	36.572	1.4374	2.8381
95.37	105.691	72.344	36.864	1.4610	2.8671

95.65	113.177	76.487	39.709	1.4797	2.8501
95.94	115.773	78.693	39.435	1.4712	2.9358
96.23	117.355	78.846	40.419	1.4884	2.9035
96.51	118.733	79.140	39.923	1.5003	2.9741
96.8	119.737	79.520	40.780	1.5058	2.9362
97.08	120.773	80.691	40.187	1.4967	3.0053
97.37	121.653	80.942	41.241	1.5030	2.9498
97.65	122.547	81.124	41.350	1.5106	2.9636
97.94	123.405	83.332	42.225	1.4809	2.9226
98.23	124.325	83.973	43.097	1.4805	2.8848
98.51	125.127	83.071	42.535	1.5063	2.9418
98.79	126.058	84.761	42.515	1.4872	2.9651
99.08	126.938	84.372	43.027	1.5045	2.9502

In TABLE III the DDP parameters are changed to $1/\delta_2 = 1.5$, $1/\delta_3 = 3$. The rest parameters remain intact. For total load close or even less than 94% we have observed similarly to D-WFQ case small deviations from the PDD model. This may be attributed to the fact that CBQ scheduler never idles, so, under moderate load conditions and in case classes assigned with high service rate have no packets to send, it may serve packets of low priority classes. Thus, the delay of low rate service class packets may be reduced.

In TABLE IV the DDPs are taken equal to $1/\delta_2 = 2$, $1/\delta_3 = 4$, as in the first experiment but their input loads vary. The total input load remains the same and equal to 97.37%. Ideally, the DB-CBQ must meet the DDP constraints independent of the class load distribution.

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

Input Load Per Class % ($\lambda_1, \lambda_2, \lambda_3$)	d_1	d_2	d_3	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

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.

Finally, we performed the first experiment with the source nodes now generating traffic following a Pareto distribution with $\alpha = 1.8$, providing total input loads 97.37% and 99.08%.

TABLE V. Pareto Input Loads. Delay differentiation ratio = 1:2:4

Total Input Load %	d_1	d_2	d_3	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

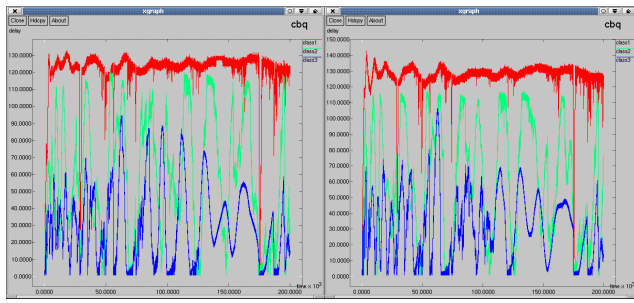


Figure 4. Queueing delays per packet under 97.37% and 99.08%

The average, as well as the queueing packet delays experienced by each service class throughout the simulation are depicted in TABLE V and Fig. 4, respectively. The results displayed show that our model accomplishes the proportional delay differentiation even under more realistic traffic models [27]. The bursts produced by Pareto can be consumed appropriately according to the DDPs specified per class.

V. CONCLUSIONS

The relative differentiated service architecture is a promising approach for differentiated service provision in IP networks due to its ease of deployment and management. In this paper, a proportional delay differentiation model has been presented by enhancing the CBQ scheduler with the novel feature of dynamic adaptation of its service classes' rates. A fundamental assumption taken is that the queue length of classes is infinite, so service class packets do not experience any losses. 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 in accordance to the queues' average load. Using feasibility inequalities and applying delay measurements of a strict priority scheduler we specified the area of feasible DDPs. We then tested our model for achieving proportional delay differentiation for a set of feasible 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 introduction of computation models for the service rates in CBQ classes, when moderate input load are considered (less than 94%). Our experiments showed that for moderate loads the best approximation of the PDD is achieved by using (9) and (8) (in which $q_i(t)$ and $\lambda_i(t)$ are replaced by their normalized averages), which however, is far from the required values. As already stated, this may be attributed to the fact that the CBQ never idles, so under moderate load conditions and in case classes with high service rate assigned have no packets to send, it may serve packets of low priority service classes, thus, the respective packet delay may be reduced. This aspect needs more research effort even though at an under-loaded link the delays experienced are very small for all classes regardless the scheduling algorithm employed. Second, we intend to include in our model the loss metric as differentiation parameter. This attempt would lead to a model, that would respond more effectively to increasing user expectations.

REFERENCES

- [1] P. White, "RSVP and Integrated Services in the Internet: A Tutorial", In IEEE Communications Magazine, May 1997.
- [2] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview", IETF RFC1633, July 1994.
- [3] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, Dec. 1998.
- [4] K. Nichols, V. Jacobson, and L. Zhang, "Two-bit differentiated services architecture for the Internet", IETF RFC2638, July 1999.
- [5] S. Blake, et al., "An Architecture for Differentiated Services", RFC 2475, Dec. 1998.
- [6] J. Heinanen, F. Baker, W. Weiss, J. Wrockawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.
- [7] 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>, Aug. 1999
- [8] I. Stoika and H. Zhang, "LIRA: An approach for Service Differentiation in the Internet", In Proceedings NOSS-DAV, 1998
- [9] ITU-T Rec. X.739, "Information Technology - Open Systems Interconnection, Systems Management Functions - Metric Objects and Attributes", 1992.
- [10] ITU-T Rec. X.738, "Information Technology - Open Systems Interconnection, Systems Management Functions - Summarization Function", 1993
- [11] 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.
- [12] A. Odlyzko, "Paris Metro Pricing: The Minimalist Differentiated Services Solution", In Proceedings IEEE/IFIP International Workshop on Quality of Service, June 1999.
- [13] D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service", In IEEE/ACM Transactions on Networking, Vol. 6 (4), Aug. 1998.
- [14] C. Dovrolis, D. Stiliadis, P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling", In IEEE/ACM Transactions in Networking, Feb. 2002.
- [15] C. Dovrolis, D. Stiliadis, P. Ramanathan, "Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping", In Proceedings of the 2000 International Workshop on Quality of Service (IWQoS), Pittsburgh PA, June 2000.
- [16] M. Leung, J. Lui, and D. Yau, "Characterization and Performance Evaluation for Proportional Delay Differentiated Services", In Proceedings International Conference on Network Protocols (ICNP), Oct. 2000.
- [17] T. Nandagopal, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "Delay Differentiation and Adaptation in Core Stateless Networks", In Proceedings of IEEE INFOCOM, Mar. 2000.
- [18] Y. Moret and S. Fdida, "A Proportional Queue Control Mechanism to Provide Differentiated Services", International Symposium on Computer and Information Systems (ISCIS), Oct. 1998.
- [19] J. Liebeherr and N. Christin, "Buffer Management and Scheduling for Enhanced Differentiated Services", Technical Report CS-2000-24, University of Virginia, Aug. 2000.
- [20] C. Li, S. Tsao, M. Chen, Y. Sun, and Y. Huang, "Proportional Delay Differentiation Service Based on Weighted Fair Queueing", In Proceedings IEEE International Conference on Computer Communications and Networks (ICCCN), Oct. 2000.
- [21] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks", In IEEE Proceedings, vol. 83, Oct. 1995.
- [22] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm", In Internetworking: Research and Experience, 1990.
- [23] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks", In IEEE/ACM Transactions on Networking, Vol. 3 (4), Aug. 1995.

- [24] G. Bolch, S. Greiner, H. Meer, and K. Trivedi, "Queueing Networks and Markov Chains", John Wiley and Sons, 1999.
- [25] L. Kleinrock, "Queueing Systems", Vol. II, John Wiley and Sons, 1976.
- [26] S. McCanne and S. Floyd. [http://www-mash.cs.berkeley.edu u/ns/ns.html](http://www-mash.cs.berkeley.edu/ns/ns.html) Network Simulator, 1996.
- [27] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level", In Proceedings of ACM SIGCOMM, Sept. 1995..