A delay-based analytical provisioning model for a QoS-enabled service

Christos Bouras
RA Computer Technology Institute-RACTI and
Department of Computer Engineering and Informatics
University of Patras
Patras, Greece
bouras@cti.gr

Afrodite Sevasti
Greek Research and Technology Network-GRNET
Athens, Greece
sevasti@grnet.gr

Abstract— **Carefully designed and managed services are essential for quality-demanding traffic, especially in large-scale IP networks where the aggregation of flows and the variety of traffic types are extensive. Although individual mechanisms have been widely investigated, not much related work exists on integrated approaches to QoS provisioning that are also feasible to implement. This work presents a thorough approach to the design, dimensioning and provisioning of a high priority service for high-quality demanding traffic over an IP network. Our approach employs efficient scheduling and a dynamic admission control scheme while demonstrating novel characteristics in terms of the quality offered to IP flows.**

Keywords- IP QoS, high-priority service, delay-based service differentiation

I. INTRODUCTION

Despite the success of the best-effort service model of the Internet, provisioning of Quality of Service (QoS) guarantees in IP networks has been a major trend in packet-based networks' research.

As stated in [1], the main difficulty for provisioning statistical QoS for a multi-node network lies in addressing the complex correlation of traffic at downstream multiplexing points. In this work, we assume aggregated IP traffic complying with deterministic traffic envelopes imposed at the network edge and propose a set of domain-wide mechanisms operating on flow aggregates, in an effort to provide QoS guarantees and efficient service differentiation to IP packets in a scalable manner.

In [2], it is emphasized that the limiting factor for a premium QoS service is not bandwidth but end-to-end delay. Indeed, for the deployment of such a service, guaranteed service rate is feasible, by ensuring that enough capacity is provisioned in all links to serve aggregated Gold traffic. As the authors of [3] point out, multi-hop worst case delay can only be achieved with controlling and constraining the global network topology.

Based on this observation, we propose in this work the allocation of resources along the path that a flow follows in a way that guarantees a bounded worst-case perceived end-to-end delay and at the same time allows for differentiation in terms of the QoS perceived without compromising quality. As a tool for the network operator, we provide an efficient Call Admission Control (CAC) mechanism along these lines. We refer to the proposed schema as 'Gold' service.

Our approach employs some existing results of QoS related research, however it is innovative as a whole. We use Dynamic Packet State (DPS) as originally proposed in [4] to store in packet headers information for each packet's treatment in the core, at the ingress of the network. We also use a version of Earliest Deadline First (EDF) scheduling, that assigns local deadlines of packets at each node derived from time-shifting of each packet's arrival time at the ingress node, similar to deadline-curve based EDF (DC-EDF) of [5].

EDF has been shown to provide optimal delay bounds at a single node and outperform Generalized Processor Scheduling-GPS ([6],[7]) in the end-to-end case, if per node traffic shaping is exercised. Variations of EDF exist, such as Rate-Controlled EDF (RC-EDF), which employs shapers at each node, enforcing delays to packets in order to make each flow fully conformant to its traffic envelope and Service Curve based EDF scheduling, which assigns deadlines from a rate provisioning point of view but, unlike DC-EDF, doesn't provide tight statistical delay bounds. In [8], the authors use EDF in the context of coordinating packet deadlines in the core. Our approach is somewhat different. We explicitly distribute resources available to flows in proportion to the load of the nodes traversed, during the admission control phase.

In [9], Coordinated EDF (CEDF) is used to denote local deadlines of packets at each node traversed in proportion to the server speed at each hop (over the sum of speeds in the end-to-end path). As opposed to the approach of [9], our proposal does not require knowledge of the statistical characteristics of arrivals, as it is based on the traffic envelope imposed to traffic at the ingress, and at the same time provides an admission control mechanism that is straightforward and practical to apply.

The domain-wide, parameter-based flow admission control scheme of [2], which is based on GPS and uses network calculus, the theory of deterministic queuing systems, is directly comparable to our approach. This work is compared in terms of performance with our proposed schema in section III of this paper. Another work that exploits GPS for scheduling and attempts path-wide optimization of resources' allocation is presented in [10].

In [4] Core-Jitter-VC (CJVC) is defined. CJVC distributes equally (among the hops that a packet traverses) the difference between the required end-to-end delay for a packet and the slack from the guaranteed service rate occurring due to differing packet sizes and early arrivals. The work of [4] is directly comparable to our work, therefore it is also compared in terms of performance with ours in section III of this paper.

In the next section, we present our proposed dynamic CAC scheme -based on the mechanisms DPS and arrival-times based EDF- which allows for admission control of new flows or aggregates while ensuring that the QoS provided to previously admitted flows is not compromised. In section III, we demonstrate how our CAC scheme achieves differentiation of Gold flows in terms of the end-to-end delay perceived and outperforms other approaches. The paper concludes by summarizing the benefits of our approach and describing our future work.

II. PROPOSED FRAMEWORK

We consider a network domain with N nodes. In every node i with $i \in 1..N$, a service queue $Q_{i,l}^{G}$ exclusively for Gold traffic is configured on each outgoing link l. $Q_{i,l}^{G}$ serves aggregated Gold traffic over l with priority and separately from best-effort traffic but is rate-limited to a rate of $C_{i,l}^{G}$ so that best-effort traffic is not starved. For simplicity in notation, we will consider a single outgoing link for each node in our domain, thus $Q_{i,l}^{G}$ becomes Q_{i}^{G}. We consider aggregated traffic to be a Gold flow as long as it adheres to the following definition:

Definition 1: A Gold flow f has the following properties:

- All packets of the Gold flow enter the domain at the same ingress node (I_f) and are exiting from the same egress node (E_f)
- The aggregated traffic comprising the Gold flow f is policed at the ingress node I_f with a token bucket policer (r_f, b_f), imposing a long term average rate r_f for the Gold flow and a maximum burst size b_f.

□

As demonstrated in our previous work ([11]) but also in related research work, C_i^G must always be larger than the sum of the long term average rates of the Gold flows being served by Q_i^G. Thus

$$\sum_{j \in F} r_j t \le C_i^G t \Rightarrow \sum_{j \in F} r_j t = \rho_i C_i^G t \,,\ \rho_i < 1 \text{ as } t - \propto \qquad (1)$$

where F is the set of Gold flows served through Q_i^G. This is a stability condition assuring that load does not build up in Q_i^G.

A. Load-Aware EDF scheduling (LA-EDF)

Apart from the externally observable service rate C_i^G of the Gold traffic queue, packets within $Q_{i,}^G$ are ordered (and thus served) in an EDF manner. According to our proposed policy defined in this section, each packet k of Gold flow j served through node i is assigned a local deadline $d_i^{k,j}$. In compliance to EDF principles, the packet with the smallest $d_i^{k,j}$ value is the one to be served first from Q_i^G.

In this work, we propose a novel policy for obtaining $d_i^{k,j}$ values for each hop along the packets' end-to-end path, while $d_i^{k,j}$ values still occur from a time-shifting of each packet's arrival time ($a_I^{k,j}$) in the corresponding ingress node I_j. We denote by DB^j the required end-to-end delay bound from I_j to E_j for the packets of flow j under the Gold service model. We propose Load Aware EDF (LA-EDF) as follows:

Proposition 1: **'Load Aware EDF' -** We propose a quantization of the end-to-end delay budget DB^j for the packets of flow j that is dependent upon the ratio of load of each node that the packets of flow j traverse when compared to the sum of loads of the nodes along the end to end path from I_j to E_j. The local deadline for packet k of Gold flow j served at node i is:

$$d_i^{k,j} = \overline{a_1^{k,j}} + \sum_{q=1}^{i} \sigma_q^j \qquad (2)$$

In this proposition, σ_i^j is defined as the 'local slack term' for each packet of flow j at node i, thus the maximum delay a packet of j can endure while at Q_i^G. It is defined as:

$$\sigma_i^j = (DB^j - DP_p)\frac{\rho_i}{\sum_{l \in p} \rho_l} \qquad (3)$$

with p the end-to-end path for the packets of flow j. □

It has to be stressed out at this point, that the individual values in (3) are based on the total Gold traffic subscription on each topology link, thus the sum of the accepted flows' profile rates. Thus LA-EDF does not require real-time measurements for its operation and is therefore robust against instant traffic fluctuations. Also, the term DP_p in (3) stands for the propagation delay of each packet along p. According to Proposition 1, the same local deadline d_i^j is assigned to all packets of a flow j at node i.

According to the Gold service provisioning model, the DB^j end-to-end delay bound for the packets of flow j is guaranteed only for in-profile packets of the flow. Thus, $\overline{a_I^{k,j}}$ in (2) denotes the in-profile arrival time of packet k at node I_j, according to the policer for flow j. We also provide the following definition:

Definition 2: The 'latest arrival time' of packet k in node i is defined as

$$\overline{a_i^{k,j}} = d_{i-1}^{k,j} \tag{4}$$

thus as the deadline of the packet in the upstream node. ☐

A packet can never arrive at a node i later than its 'latest arrival time' $\overline{a_i^{k,j}}$ because otherwise it will have violated its local deadline $d_{i-1}^{k,j}$ and have been dropped at node i - 1.

B. Delay-Based Admission Control (DBAC)

With LA-EDF implemented in each Q_i^G, a domain-wide Gold service implementation must also anticipate for a requests' Admission Control scheme. The Admission Control process should provide a positive or negative result as to whether a new Gold flow f can be added to the set of flows served through a path p.

Assuming that a request for serving a new flow with arrival process A_f and a deterministic envelope $\overline{A_f} = (r_f, b_f)$ emerges, we want to examine whether the flow can be accepted along a path p where I_p is the ingress node and E_p is the egress node for all packets of the flow. As a first step, the local slack term at each node of p for the packets of f is obtained, based on (2) and (3). Our proposed Delay-Based Admission Control (DBAC) then focuses on the schedulability of f on each node i of p separately, taking into account the existing set of flows served through Q_i^G.

Assume a non-schedulable set of packets in Q_i^G, based on the actual arrival times at the i^{th} node of p and the assigned packet deadlines at the same node. All packets can be delayed until their corresponding latest arrival times in Q_i^G before being considered eligible to serve. If we can ensure that the aggregated arrival process denoted by the latest arrival times of all packets is schedulable, so will the actual aggregated arrival process be.

DBAC examines schedulability conditions at each node separately, based on the following corollary:

Corollary 1: Under the assumption of any EDF scheduling based on time-shifting of arrival times (such as LA-EDF), if the packet of a flow f is schedulable in node p_m when all existing flows' packets arrive at p_m at their latest arrival times and the packet itself arrives at p_m at its latest arrival time $\overline{a_m^{k,f}}$, then the packet is schedulable at p_m node under any circumstances. ☐

The intuition here is that in case one or more packets arrive early, they can be delayed until their latest arrival times, if needed, in order to become schedulable. At the same time, no packets can arrive later than their latest arrival times, as they would already have been dropped at the upstream node. This intuitive schedulability condition for each time interval τ and for the case of EDF schedulers, refined for the case of deadlines that are time-shifting of each packet's arrival time (like LA-EDF) in [5], is as follows:

$$\sum_{j \in F} \max\{0, D_j(v+\tau) - A_j(v)\} \le C_i^G \tau - l_i^{\max}, \forall \tau \tag{5}$$

where $D_j(t)$ is the continuous function providing the cumulative amount of traffic from flow j with a deadline up to time t. Similarly, $A_j(t)$ is the continuous function providing the amount of flow j traffic from that has arrived to Q_I^G up to time t (arrival curve of j). Finally, F is the set of Gold flows traversing node i and $l_i^{\max}$ is the maximum sized packet served in node i.

Here, we extend this schedulability criterion for the purposes of performing DBAC for a single additional flow f for the node i of p. We order already accepted flows in node i so that $k < l$ whenever $\sigma_i^k < \sigma_i^l$. The basic assumption is that we have a set F of Gold flows F in node i, for which the inequality of (5) holds for each τ. We wish to define DBAC to ensure that whenever a new Gold flow f is being accepted, the set $F \cup f$ remains schedulable. For the purposes of our DBAC, we exploit the property of LA-EDF provided in (2). It allows relating the deadline and arrival curves of flow j as follows

$$D_j(v) = A_j\left(v - \sigma_i^j\right) \tag{6}$$

Based on that, the following holds:

$$D_j(v+\tau) - A_j(v) = A_j(\tau - \sigma_i^j) \tag{7}$$

Also, the deterministic envelope $\overline{A_j} = (r_j, b_j)$ for each flow ensures that:

$$A_j(\tau) \leq r_j \tau + b_j, \forall \ \tau \tag{8}$$

Using (7) and (8), an alternative form of (5) can be derived such that:

$$\sum_{k=1}^{j} [r_k(\tau - \sigma_i^k) + b_k] \leq C_i^G \tau - \max_{l>j} L_l^{\max} \tag{9a}$$

with $\sigma_i^j \leq \tau < \sigma_i^{j+1}, 1 \leq j < |F|$

$$\sum_{k=1}^{|F|} [r_k(\tau - \sigma_i^k) + b_k] \leq C_i^G \tau \text{ with } \tau > \sigma_i^{|F|} \tag{9b}$$

As long as (1) is satisfied, it is straightforward to show that the schedulability condition for a flow j is:

$$\sigma_{i-\min}^f \geq \frac{b_f + \sum_{k=1}^{j-1}(b_k - r_k \sigma_k) + \max_{l>f} L_l^{\max}}{r_f} \tag{10}$$

Thus, flow f is schedulable assuming a set of already accepted flows F, with a local slack term of σ_i^f such that (10) holds and $\sigma_i^{j-1} < \sigma_i^f < \sigma_i^j$. For the case of ingress node I_p, we can set $i = I_p$ in (9) and (10).

However, DBAC also has to make sure that the addition of flow f in the set of Gold flows served in node $i = I_p$ does not cause the schedulability criterion of (10) for each flow with a local slack term such that $\sigma_i^j > \sigma_i^f$ to be violated. The schedulability condition of (10) is therefore verified for all flows with $\sigma_i^j > \sigma_i^f$, as a second step of the DBAC. Finally, the schedulability condition of (9b) is verified for $F \cup f$ and a random τ such that $\tau > \sigma_i^{|F|}$, as a last step. Based on the following lemma, this ensures that $F \cup f$ is schedulable $\forall \tau : \tau > \sigma_i^{|F|}$.

Lemma 1: If a set of flows F is schedulable for a random t such that $t > \sigma_i^{|F|}$, then F is schedulable $\forall \tau : \tau > \sigma_i^{|F|}$ □

The proof of the Lemma is straightforward and therefore omitted.

DBAC, is similarly applied in all other nodes than I_p, to provide a result on whether a Gold flow can be accepted on a single path p. However, when a new flow f can be served by more than one path, then path selection should be made so that load for Gold traffic, thus ρ_i, is balanced for each link i of the network. Thus links with a higher ρ_i value will still be able to serve some future requests. Optimal path selection, in addition to LA-EDF and DBAC, is part of our future work.

III. EXPERIMENTAL EVALUATION

For the purpose of evaluating the effectiveness of our proposed scheme in QoS provisioning and end-to-end delay differentiation, a number of experiments were conducted in a simulation environment, the components of which are analytically presented in [12]. The experiments were carried using the ns-2 simulator ([13]). The topology used is depicted in Fig. 1.

It comprises of a number of edge nodes (E_i) and three core nodes (C_i) interconnected in a topology where the backbone links' capacities are not at all times adequate to carry the accumulated traffic of all adjacent peripheral links. The topology has been designed to provide at least two alternative paths for serving a traffic flow. DBAC is used to regulate whether arriving flows are accepted or not. At each node a queue is configured to serve Gold traffic with absolute non-preemptive priority (PQ) over all other traffic. Within each Gold traffic queue, LA-EDF is implemented. Each Gold traffic flow comprises of a random number $v, 1 \leq v \leq 10$ of VoIP micro-flows with $r_j = 80Kbps$ ($j \in [1, v]$) each and a random number $m, 1 \leq m \leq 5$ of streamed MPEG video micro-flows with $r_i = 700Kbps$ each ($i \in [1, m]$).

For the simulation of VoIP flows an exponential ON/OFF distribution was used, with an average duration of ON periods equal to 1.004 sec, average duration of OFF periods (idle-time) equal to 1.587 sec, packet size of 188 bytes (8 byte UDP header+20 byte IP header+160 byte voice data) and transmission rate during the 'on' period equal to 80Kbps. For the simulation of MPEG flows, a number of MPEG video traces were used, with a packet size of 200 bytes. In this way, each Gold flow comprises of a realistic mix of micro-flows in order to simulate burstiness and statistical properties of delay-sensitive traffic in IP networks.

Gold traffic flows' arrivals are uniformly distributed with a mean of 10 ms, the duration of each flow has a uniform distribution with a mean of 10 sec. For each flow that is accepted to be served, a policer is configured at the ingress node with a rate equal to the sum of the individual VoIP and MPEG flows' average rates ($\sum_1^v j + \sum_1^m i$) and a burst size of:

$$a \times (v \times L_{VoIP} + m \times L_{MPEG}) \tag{11}$$

where L_{VoIP} and L_{MPEG} are the sizes of the VoIP and MPEG traffic packets. The value of a (an integer) is varied in some of the experiments in order to observe how the different mechanisms react when increasing the burstiness of each admitted flow.

The simulated duration of each experiment was 120 minutes, while each experiment was repeated 10 times.

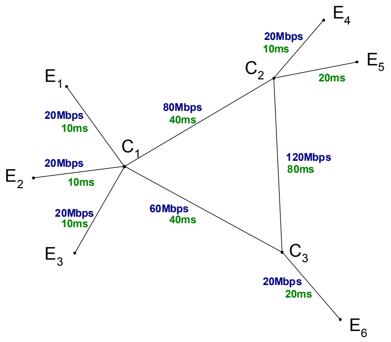

Figure 1. Topology of simulation experiments

A. LA-EDF and DBAC performance in achievable end-to-end queuing delay

In order to evaluate the effectiveness of LA-EDF as a scheduling mechanism and DBAC as an admission control mechanism in provisioning rate and end-to-end delay guarantees in a backbone network, we are comparing how our mechanisms perform when compared with equivalent QoS provisioning schemes. More specifically, we compare LA-EDF and DBAC with the Core-Jitter-Virtual Clock (CJVC) of [4] and the approach of [2], based on Network Calculus. These two approaches can, to the best of our knowledge, be directly compared with our work, as they provide QoS mechanisms and admission control for serving quality demanding traffic in a network serving aggregated IP traffic.

In the first set of experiments, the efficiency of LA-EDF and DBAC in terms of achieving minimal queuing delay for QoS-demanding traffic when compared to CJVC and the Calculus scheme is demonstrated by observing how the three different schemes behave when topology results in high burstiness of aggregated traffic. We have observed and measured the queuing delay perceived by a Gold flow F (a mix of VoIP and MPEG traffic) generated at E_5 and destined to E_2, as the capacity of the E_5 - C_2 link varies from 20 to 50 Mbps. During the experiments, flow F co-exists in the simulated topology with tens of other flows, generated between random pairs of E_i nodes, using the principles provided in the beginning of this section.

A higher capacity value for the E_5 - C_2 link results in bursts of aggregated Gold traffic from E_5 reaching C_2 with a much higher rate than the capacity of the queue serving Gold traffic on the C_2 - C_1 link. As a result, all three schemes demonstrate an increase in the end-to-end queuing delay perceived by the packets of F (see Fig. 2), as the capacity of the E_5 - C_2 link increases. However, LA-EDF and DBAC demonstrate the lowest worst-case queuing delay among the three, even as burstiness increases. CJVC demonstrates the poorest performance among the three approaches. However, this outcome is somewhat justified. CJVC introduces extra delay as it adds to the allowable delay for a packet on each queue (along the hops that a packet traverses) the difference between the required end-to-end delay for a packet and the slack expected under the guaranteed service rate.

A second set of experiments was conducted along the same lines. In this case, the capacity of the E_5 - C_2 link was held constant while the a factor in (11) was varied in order to introduce increased burstiness in the output of the policer applied to F. As demonstrated in Fig. 3, the combination of LA-EDF and DBAC continuously outperforms CJVC and the Calculus scheme in terms of the worst-case queuing delay perceived by the packets of F. DBAC ensures that end-to-end queuing delay for Gold packets remains low even when traffic is highly bursty, while the Calculus scheme results in a queuing delay that increases at a higher rate. CJVC demonstrates again the worst performance and seems to be much more vulnerable to burstiness of traffic.

It is important to mention that in the scenarios presented in this section, all schemes operated under congestion, as the rejection rate of arriving flows throughout the whole range of experiments was non-zero, or, in other words, Gold flows were generated with a higher rate than that any of the three schemes was able to accommodate. For each experiment, LA-EDF and DBAC were evaluated first and the experiment provided a subset of accepted flows, from the total of arriving flows. This subset was then fed to the experiments using CJVC and the Calculus scheme. In this way, it was ensured that CJVC and the Calculus schemes were evaluated against LA-EDF and DBAC under the same set of accepted flows and thus that the observed robustness of LA-EDF and DBAC in providing minimal queuing delay was not due to a higher flow rejection ratio but rather due to the nice properties of these mechanisms.

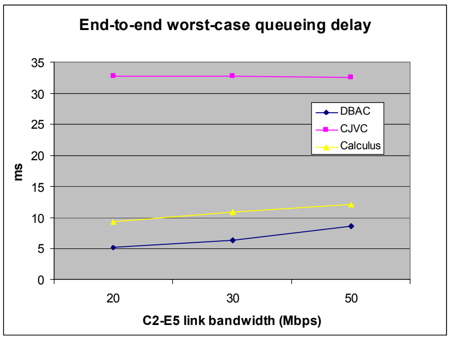

Figure 2. Performance in terms of end-to-end delay guarantees

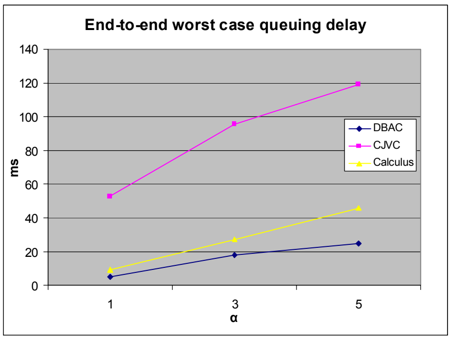

Figure 3. End-to-end queuing delay vs. increasing burstiness

B. *Using DBAC for queuing delay differentiation*

In this set of experiments, the efficiency of DBAC in differentiation of the end-to-end queuing delay perceived by Gold traffic flows is evaluated. Differentiation is based on the end-to-end delay budget DB^j for the packets of each flow j. Thus, DBAC is evaluated against its efficiency in ensuring for a flow an end-to-end queuing delay that is directly proportional to the flow's DB^j value when compared to the DB values of all other flows with which flow j shares the same path. For this experiment, we have monitored the perceived worst case queuing delay by two different flows in the topology of Fig. 1. In Table 1, the queuing delay tolerance ($DB^j - DB_p$) of each of the flows in a series of 5 experiments is provided.

As can be seen from Fig. 4, LA-EDF and DBAC are very successful in achieving fair differentiation in terms of the queuing delay perceived by Gold traffic flows. When the two flows have the same queuing delay tolerance value (in the case of Experiment 3), the worst case queuing delay perceived by the packets of each flow is identical for the two flows. As the difference between $DB^j - DB_p$ for each flow increases, so is the difference in the queuing delay for the packets of each flow.

IV. CONCLUSIONS – FUTURE WORK

In this work, we propose an EDF scheduling algorithm as well as an Admission Control scheme for implementing a QoS service for delay-sensitive IP traffic. The service offers guaranteed service rate, bounded end-to-end delay and no packet loss to aggregated IP traffic across a backbone IP network, while at the same time demonstrating efficiency in differentiation of the worst-case end-to-end delay perceived by flows with different requirements. The proposed mechanisms are also scalable as all the information required for their operation can be kept at the traffic ingress node and encoded in the packet headers. They are fully based on a theoretical scheme and therefore do not require any kind of measurements for their operation. Moreover, no state information or configuration changes are required in core nodes.

As part of our future work, we aim to investigate the use of statistical models (such as effective envelopes), rather than the deterministic envelopes used here for modeling traffic arrivals. These models will be then used by DBAC to provide admission control in, hopefully, a more efficient way, thus achieving higher schedulability. We also aim to define ways for partitioning the delay budget of a flow along a path other than in proportion to the service load at each node, by examining different correlations of the queuing delay and the load imposed on each Gold queue. We also aim to investigate domain-wide path selection optimization methodologies, which, in addition to the benefits of LA-EDF and DBAC, will provide higher schedulability and load balancing.

TABLE I. QUEUING DELAY TOLERANCE OF FLOWS

	Exp1	**Exp2**	**Exp3**	**Exp4**	**Exp5**
Flow 1	25	30	55	80	130
Flow 2	130	80	55	45	40

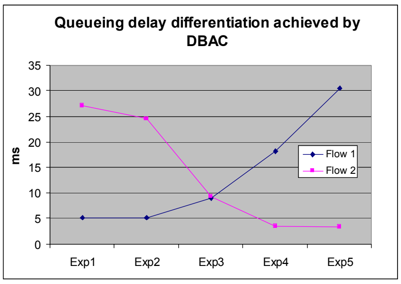

Figure 4. LA-EDF and DBAC effectiveness in differentiating queuing delay

REFERENCES

[1] S. D. Patek, J. Liebeherr, and E. Yilmaz, 'Tradeoffs in Designing Networks with End-to-End Statistical QoS Guarantees', Telecommunications Systems, vol. 23, no. 1, 2003, pp. 9-34.

[2] M. Fidler, V. Sander, "A Parameter Based Admission Control for Differentiated Services Networks", Elsevier Computer Networks Journal, Special issue on QoS in Multiservice IP Networks, vol. 44, no. 4, 2004, pp. 463-479.

[3] B. Davie et al., RFC 3246, "An Expedited Forwarding PHB (Per-Hop Behavior)", March 2002 (Obsoletes RFC2598).

[4] I. Stoica, H. Zhang, "Providing guaranteed services without per flow management", In Proc. of ACM SIGCOMM'99, Boston, MA, 1999, pp. 81 – 94.

[5] K. Zhu, Y. Zhuang, Y. Viniotis, "Achieving end-to-end delay bounds by EDF scheduling without traffic shaping", in Proc. of IEEE INFOCOM '01, no. 1 (2001), pp. 1493-1501.

[6] V. Sivaraman, F. M. Chiussi, M. Gerla, "End-to-End Statistical Delay Service under GPS and EDF Scheduling: A Comparison Study", in Proc. of IEEE INFOCOM '01, no. 1 (2001), pp. 1113-1122.

[7] N. Dukkipati, J. Kuri, H. Jamadagni. "Optimal Call Admission Control in Generalized Processor Sharing (GPS) Schedulers", In Proc. of IEEE INFOCOM 2001, pp. 468-477.

[8] C. Li, E. Knightly, "Coordinated Multihop Scheduling: a framework for end-to-end services", IEEE/ACM Transactions on Networking (TON), Volume 10 , Issue 6 (December 2002), pp: 776 – 789.

[9] M. Andrews, "Probabilistic end-to-end delay bounds for earliest deadline first scheduling", in Proc. of IEEE INFOCOM 2000, vol. 2, Tel Aviv, Israel, pp. 603-612.

[10] T. Chiueh and Y. Lin, "Delay Budget Partitioning to Maximize Network Resource Usage Efficiency", in Proc. of IEEE INFOCOM '04, vol. 23, no. 1 (2004), pp. 2061-2072.

[11] C. Bouras, A. Sevasti, "Analytical approach and verification of a DiffServ-based priority service", In Proc. of 6th IEEE International Conference on High Speed Networks and Multimedia Communications-HSNMC 2003, Estoril, Portugal, pp. 11 – 20.

[12] C. Bouras, D. Primpas, A. Sevasti, A. Varnavas, "Enhancing the DiffServ architecture of a simulation environment", in Proc. of 6th IEEE International Workshop on Distributed Simulation and Real Time Applications (2002), pp. 108-118.

[13] S. McCanne and S. Floyd, 'ns Network Simulator', available at: http://www.isi.edu/nsnam/ns