Efficient Admission Control for EDF Schedulers *

Victor Firoiu
vfiroiu@cs.umass.edu

Jim Kurose
kurose@cs.umass.edu

Don Towsley
towsley@cs.umass.edu

Department of Computer Science
University of Massachusetts
LGRC, Box 34610
Amherst, MA 01003-4610, USA

CMPSCI Technical Report TR 96-46 †

June 1996, Revised April 1997

Abstract

In this paper we present algorithms for flow admission control at an EDF link scheduler when the flows are characterized by peak rate, average rate and burst size. We show that the algorithms have very low computational complexity and are easily applicable in practice. The complexity can be further decreased by introducing the notion of *discrete* admission control. We evaluate the penalty in efficiency incurred by the discretization of the EDF admission control. We find that this efficiency degradation can be made arbitrarily small and is acceptable even for a small number of discretization points.

KEYWORDS: Admission Control Algorithms, Quality of Service, EDF scheduling, Leaky Bucket, Traffic Envelope.

---

*This material is based upon work supported in part by the National Science Foundation under Grant NCR-95-08274 and CDA-95-02639. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

†ftp://ftp.cs.umass.edu/pub/techrept/techreport/1996/UM-CS-1996-046.ps

1 Introduction

The demand for real-time communication in data networks such as Internet has grown rapidly in recent years. Two important examples are voice and video communication over the Internet – applications that require timely delivery of data packets. To be able to guarantee such delay requirements, the network has to reserve resources at the links on the path of the given real-time flow. Several flow setup protocols that convey end-to-end user delay requirements to the links have been proposed and are in the process of standardization; these include RSVP [2] for the Internet, and ATM signaling [1] for ATM networks.

The problem of providing delay guarantees at a network link is the focus of much current research. Much of this work focuses on the issue of packet scheduling – determining the order in which queued packets are forwarded over outgoing links at switches and routers. This order determines the packets' waiting time in the link's queue, and ultimately the delay that the link scheduler can guarantee. Several analytical models for link scheduling have been proposed in the literature. A variant of Weighted Fair Queuing (WFQ) [4] (also known as Generalized Processor Sharing (GPS) [13]) was proposed in [14] to guarantee a maximum queuing delay by reserving a certain amount of link bandwidth for the given flow. Although simple, this policy is known to be sub-optimal. Another discipline, Earliest Deadline First (EDF) [12] associates a per-hop deadline with each packet and schedules packets in the order of their assigned deadlines. EDF has been proven to be an optimal scheduling discipline in the sense that if a set of tasks is schedulable under any scheduling discipline (i.e., if the packets can be scheduled in such a way that all of their deadlines are met), then the set is also schedulable under EDF. Also, Rate-Controlled EDF [15] was proven to outperform GPS in providing end-to-end delay guarantees in a network [7]. In the present work we adopt the Rate-Controlled model where the EDF scheduler of each link has an independent contribution to the end-to-end delay guarantee of a flow. The end-to-end admission control is thus reduced to EDF schedulability verifications at each link.

Sufficient conditions for the EDF schedulability of flows have been proposed for some particular cases of flow characterizations [8, 16]. Recently, a set of necessary and sufficient conditions for flow schedulability has been put forward by [11, 6], using a general characterization of flows. The fact that EDF is an optimal scheduling policy and that there exist necessary and sufficient conditions for schedulability makes EDF an attractive choice for providing delay guarantees for real-time flows. There are, however, two important concerns about the practicality of EDF scheduling. First, the implementation of EDF scheduling requires a search of $O(logQ)$ time in the list of packets (ordered by their deadlines) waiting in the queue of length Q for transmission. This issue has been successfully addressed in [10], where the search time is brought to constant ($O(1)$) time by discretizing the range of packet deadline values. The second issue is that, although the EDF schedulability conditions in [11] can be expressed simply, the algorithms to perform these schedulability tests can be computationally complex, or, in the general case, require an unbounded number of values that must be checked.

In this paper we consider the problem of simplifying the computation of EDF flow schedulability conditions and present simple and computationally efficient algorithm for performing flow admission at links using EDF scheduling. We take advantage of the particular flow characterization (peak rate, mean rate and burst size) proposed in the emerging standards of Internet Integrated Services [14] and ATM signaling [1]. We find that our algorithms have low complexity ($O(N)$) where N is the number of admitted flows at the EDF scheduler at the moment of the algorithm's invocation. We further simplify these algorithms significantly by discretizing the range of values for certain flow parameters (the horizontal position of the concave point of flow's envelope) to a predefined set of values. We obtain a very significant improvement in the execution time (two orders of magnitude speedup), with the additional benefit of the execution time being no longer dependent on the number of flows admitted at the scheduler. We examine the relative performance degradation (in terms of the number of flows admitted) incurred by the discretization, and find the tradeoff to be small.

The remainder of the paper is organized as follows. In Section 2 we describe the requirements imposed by IP and ATM flow setup protocols on the local (link) admission control. In Section 3 we derive simple admission control algorithms for flows characterized by peak rate, average rate, and burst size and further simplify the admission control algorithm through discretization. In Appendix C we show how to simplify further the admission control algorithms in the case that flows are characterized by (average rate, burst size) envelopes, i.e. the peak rate is not bounded. In Section 4 we evaluate by simulation the performance in computation time and flow admissibility of exact and discrete admission control algorithms. Section 5 concludes the paper.

2 Flow Admission Control in Networks: EDF Schedulers

Flow setup protocols for real-time flows such as ATM signaling and RSVP with Guaranteed Services have certain requirements for flow admission control algorithms at a link. In this section, we examine these requirements; in Section 3 we present specific admission control algorithms meeting these requirements.

Consider a source that wishes to establish a flow f to a destination using ATM signaling. It sends a SETUP message to the destination, including information such as the flow's traffic characteristics (maximum cell rate, sustained cell rate, maximum burst size [1]), and the maximum allowable end-to-end delay, d_{fm}. At each link l along the path from source to destination, the minimum delay that link l can guarantee to f, $\overline{d}_{fl}$ is computed, and added to $\overline{d}_{fc}$, the cumulative delay included in the SETUP message. If at some node the cumulative delay $\overline{d}_{fc}$ exceeds the maximum allowable delay d_{fm}, the flow cannot be accepted, and a RELEASE message is returned. Otherwise, at the end of the first pass (at the destination node), $d_{fm} \geq \overline{d}_{fc}$ and the flow is accepted. A CONNECT message is returned on the same path to the source, assigning a delay $d_{fl} \geq \overline{d}_{fl}$ to flow f at link l on path P, such that $\sum_{l \in P} d_{fl} \leq d_{fm}$ according to some delay division policy (see for example [5]).

Consider the RSVP protocol [2] in conjunction with Int-Serv "Guaranteed QoS" specification [14], protocol that is designed for real-time communication in Internet. In this protocol, the source of a real-time flow sends periodic $Path$ messages to a unicast or multicast IP address. The source includes in the $Path$ message the flow's characteristics. At each link l on the path to the receiver, the minimum delay that link l can guarantee to f, $\overline{d}_{fl}$ is computed and added to $\overline{d}_{fc}$, the cumulative delay, which is sent in the D_{tot} term of the TSpec in the $Path$ message. A receiver that requires end-to-end delay guarantee d_{fm} and receives a $Path$ message, compares d_{fm} with the minimum end-to-end delay that can be guaranteed by the network, $\overline{d}_{fc}$. If $d_{fm} < \overline{d}_{fc}$, the receiver decides that its delay requirement cannot be guaranteed. If $d_{fm} \geq \overline{d}_{fc}$, the requirement can be satisfied, and the receiver sends a $Resv$ message back to the sender including d_{fm}, its delay requirement in the delay slack term S of RSpec. On its way to source, which is the same route that $Path$ had, $Resv$ assigns a delay $d_{fl} \geq \overline{d}_{fl}$ to flow f at link l, such that $\sum_{l \in P} d_{fl} \leq d_{fm}$, according to some QoS division policy (see for example [5]).

We see that each of the above flow setup protocols requires that a local admission control procedure can be invoked at each link l with the following capabilities:

- given a flow f and its characteristics, provide the minimum delay $\overline{d}_{fl}$ that link l can guarantee to f, based on the current state (set of reserved flows) at the local scheduler;

- given a flow f, its characteristics, and a requirement $d_{fl} \geq \overline{d}_{fl}$, reserve resources at the local scheduler following the admission of f.

In the following we examine how to provide these capabilities in the case of EDF scheduling.

[11, 6] have given flow schedulability conditions at EDF schedulers for flows characterized by envelopes, or rate-controlling functions. Consider a data flow f with the amount of arrivals (measured in bits/second) in the time interval $[t_1, t_2]$ denoted by $A_f[t_1, t_2]$. The flow is characterized by an envelope A_f^*, an upper bound on the flow's arrival pattern, if:

$$A_f[t, t+\tau] \leq A_f^*(\tau) \quad \forall t \geq 0, \forall \tau \geq 0 \quad .$$

We take $A_f^*(t) = 0, \forall t < 0$, and we consider A_f^* to be non-decreasing. Note that, in order to provide a better intuition, in this paper we measure the traffic in number of data units (bits) rather than transmission time (seconds), the latter being used in [11].

Let $\mathcal{N} = \{1, 2, ..N\}$ be a set of flows, where flow $i \in \mathcal{N}$ is characterized by the envelope A_i^*. The stability condition for a work-conserving scheduler (thus including the EDF scheduler) is ([11], eq. (5)):

$$\lim_{t \to \infty} \frac{\sum_{i \in \mathcal{N}} A_i^*(t)}{ct} < 1 \tag{1}$$

where c is the constant rate of the link (bits/second). Assuming a preemptive EDF scheduler or negligible packet sizes (as in the case of ATM cells) we give the following variant of the schedulability condition proposed in [11] for the set $\mathcal{N}$ of flows:

Theorem 1 (Liebeherr,Wrege,Ferrari 1994) *Let $\mathcal{N}$ be a set of flows, stable by (1), where flow $i \in \mathcal{N}$ is characterized by the envelope A_i^* and has a maximum packet delay of d_i. The set $\mathcal{N}$ is EDF-schedulable if and only if:*

$$ct \geq \sum_{i \in \mathcal{N}} A_i^*(t - d_i), \quad \forall t \geq 0 \tag{2}$$

It is easy to show, following the proof in [11], that the above Theorem (having eq. (2) $\forall t \geq 0$) is equivalent to Theorem 1 of [11] (having eq. (2) $\forall\, 0 \leq t \leq B_1$, where B_1 is the maximum length of a busy period).

We say that the set $(A_i^*, d_i)_{i\in\mathcal{N}}$ is schedulable if (1) and (2) are satisfied. [11] provides schedulability conditions, but does not provide algorithms for schedulability testing. In the following we show that EDF schedulers have the capability to support the flow setup protocols described earlier.

Proposition 1 *If a set of flows $(A_i^*, d_i)_{i\in\mathcal{N}}$ is schedulable, then it remains schedulable if the delay for any flow is increased, $d_k' > d_k$, for any $k \in \mathcal{N}$.*

The intuition behind this result is that, by relaxing the delay requirement for a flow in a schedulable set, the set remains schedulable.

Proof . Obviously the stability condition (1) is not affected by the increase of d_k. It is easy to see that the schedulability inequalities in (2) remain true when d_k increases. Taking $d_k' > d_k$ and knowing that A_k^* is non-decreasing, we have:

$$ct \geq \sum_{i\in\mathcal{N}} A_i^*(t-d_i) = \sum_{i\in\mathcal{N}, i\neq k} A_i^*(t-d_i) + A_k^*(t-d_k) \geq \sum_{i\in\mathcal{N}, i\neq k} A_i^*(t-d_i) + A_k^*(t-d_k'), \quad \forall t \geq 0$$

∎

Corollary 1 *Given an EDF scheduler with a set $\mathcal{N}$ of admitted flows, for any new flow A_f^* there is a unique delay $\overline{d}_f(A_f^*)$ such that (A_f^*, d_f) can be admitted iff $d_f \geq \overline{d}_f(A_f^*)$.*

The delay $\overline{d}_f$ defined in Corollary 1, is the minimum (best) delay that can be guaranteed to flow f by the given EDF scheduler having the given load $\mathcal{N}$. The existence and uniqueness of the minimum delay $\overline{d}_f$ makes EDF schedulers capable of supporting the flow setup protocols described earlier.

3 EDF Admission Control for (C, σ, ρ) Token Bucket Flows

3.1 Analysis of EDF Schedulability Conditions

Let us consider flows that are characterized by the following type of envelope, referred to as (C, σ, ρ) envelope, used in both IP [14] and ATM [1] networks:

$$A_i^*(t) = \begin{cases} 0 & t < 0 \\ C_i t & 0 \leq t < a_i \\ \sigma_i + \rho_i t & a_i \leq t \end{cases} \tag{3}$$

where

- C_i is the peak rate of the flow (bits/second);
- $\sigma_i \geq 0$ is the maximum burst size at time 0 (bits);

- $\rho_i > 0$ is the average rate of the flow (bits/second);

- $a_i = \sigma_i/(C_i - \rho_i)$ is the maximum duration of the flow's burst at peak rate (seconds);

- $h_i = A_i^*(a_i) = C_i a_i = \sigma_i + \rho_i a_i$ is the maximum size of the flow's burst at peak rate (bits).

Figure 1 shows an example of a (C, σ, ρ) envelope. We shall refer to the point (a, h) of A^* as the concave point of A^*.

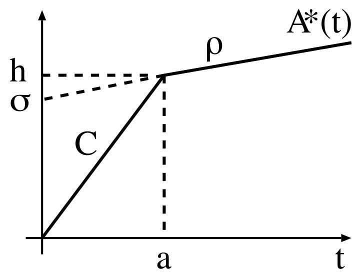

Figure 1: An illustration of (C, σ, ρ) envelope

Let $\mathcal{N}$ be a set of flows, flow i being characterized by the envelope A_i^* of the form given in (3) and having a maximum packet delay requirement of d_i. The stability condition (1) becomes:

$$\sum_{i \in \mathcal{N}} \rho_i < c \tag{4}$$

For describing the admission control algorithms of the N flows, we introduce the following sets.

Definition 1 *The sets*

$$\mathcal{P} = \{d_i + a_i | 1 \leq i \leq N\} = \{u_l | 1 \leq l \leq N\} \tag{5}$$

$$\overline{\mathcal{P}} = \mathcal{P} \cup \{-\infty, \infty\} = \{u_l | 0 \leq l \leq N+1\} \tag{6}$$

where $N = |\mathcal{P}|$, are indexed in a non-decreasing order:

$$u_0 = -\infty < u_l \leq u_{l+1} < u_{N+1} = \infty, \quad 1 \leq l \leq N-1. \tag{7}$$

Definition 2 *The sets*

$$\mathcal{Q} = \{d_i | 1 \leq i \leq N\} \tag{8}$$

$$\overline{\mathcal{Q}} = \mathcal{Q} \cup \{-\infty, \infty\} \tag{9}$$

are indexed in a non-decreasing order:

$$d_0 = -\infty < d_i \leq d_{i+1} < d_{N+1} = \infty, \quad 1 \leq i \leq N-1. \tag{10}$$

To consider the schedulability conditions (2), we give the following:

Definition 3 *The* (work) availability function $F : [0, \infty) \longrightarrow [0, \infty)$ *is defined by:*

$$F(t) = ct - \sum_{i \in \mathcal{N}} A_i^*(t - d_i) \tag{11}$$

$F(t)$ gives the amount of work (in bits) available at time t in the worst case at the EDF scheduler, while guaranteeing envelope A_i^* the maximum packet delay of d_i, for $1 \le i \le N$. This function will play a central role in the development of admission control algorithms in the rest of this paper.

The schedulability condition (2) becomes $F(t) \ge 0 \quad \forall t \ge 0$, which in turn is equivalent to $F(u) \ge 0$ for all $u \ge 0$ that are proper local minima for F. (u is a proper local minimum for F if $F(t) \ge F(u)$ in a vicinity of u and F is not constant in any vicinity of u.) Given that A_i^* has the form in (3) for all $i \in \mathcal{N}$, it is easy to see that all proper local minima of F are included in the set $\mathcal{P} \cup \{0\}$. Hence, schedulability condition (2) is equivalent to $F(d_i + a_i) \ge 0 \; i \in \mathcal{N}$ and $F(0) \ge 0$. $F(0) \ge 0$ is equivalent to $d_i \ge 0$, $i \in \mathcal{N}$. Given the form of A_i^* in (3), $F(t)$ becomes ($u_i \in \mathcal{P}$):

$$\begin{cases} 0, & t < 0 \\ ct - \sum_{\substack{i \in \mathcal{N} \\ d_i < t}} C_i(t - d_i), & 0 \le t < u_1 \\ ct - \sum_{\substack{i \in \mathcal{N} \\ d_i + a_i \le u_j}} (\sigma_i + \rho_i(t - d_i)) & \\ \quad - \sum_{\substack{i \in \mathcal{N} \\ d_i + a_i > u_{j+1}, d_i < t}} C_i(t - d_i), & u_j \le t < u_{j+1}, 1 \le j \le N - 1 \\ ct - \sum_{i \in \mathcal{N}} (\sigma_i + \rho_i(t - d_i)), & u_N \le t \end{cases} \tag{12}$$

Thus the schedulability conditions (in addition to $d_i \ge 0, i \in \mathcal{N}$) are:

$$\begin{aligned} F(d_j + a_j) = c(d_j + a_j) - &\sum_{\substack{i \in \mathcal{N} \\ d_i + a_i \le d_j + a_j}} (\sigma_i + \rho_i(d_j + a_j - d_i)) \\ - &\sum_{\substack{i \in \mathcal{N} \\ d_i < d_j + a_j < d_i + a_i}} C_i(d_j + a_j - d_i) \ge 0, \quad j \in \mathcal{N} \end{aligned} \tag{13}$$

Suppose now that a new flow, f, arrives at the EDF scheduler. Let f be characterized by (C_f, σ_f, ρ_f), having a delay requirement d_f, and let $b \in \mathcal{N}$ such that $u_b < d_f + a_f \le u_{b+1}$. By inserting $d_f + a_f$ in $\mathcal{P}$ that is ordered non-decreasingly, we obtain $\mathcal{P}' = \mathcal{P} \cup \{d_f + a_f\}$, as in Figure 2. If $d_f + a_f \le u_1$ or $d_f + a_f > u_N$, $d_f + a_f$ is inserted as the first or last element of $\mathcal{P}'$ respectively.

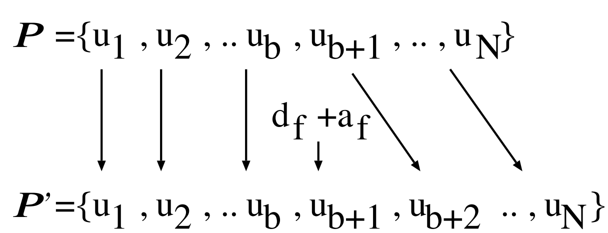

Figure 2: The mapping of $\mathcal{P}$ to $\mathcal{P}'$

Following the same reasoning as above, the scheduling conditions for the set of flows $\mathcal{N}' = \mathcal{N} \cup \{f\}$ are:

$$F'(d_j + a_j) = c(d_j + a_j) - \sum_{\substack{i \in \mathcal{N}' \\ d_i + a_i \le d_j + a_j}} (\sigma_i + \rho_i(d_j + a_j - d_i)) - $$
$$- \sum_{\substack{i \in \mathcal{N}' \\ d_i < d_j + a_j < d_i + a_i}} C_i(d_j + a_j - d_i) \ge 0 \quad j \in \mathcal{N}' \tag{14}$$

It follows that the set $\mathcal{N}'$ is schedulable iff:

$$d_i \ge 0, \quad i \in \mathcal{N} \tag{15}$$
$$d_f \ge 0 \tag{16}$$
$$F(d_i + a_i) \ge 0,$$
$$d_i + a_i \le d_f, \; i \in \mathcal{N} \tag{17}$$
$$F(d_i + a_i) - C_f(d_i + a_i - d_f) \ge 0,$$
$$d_f < d_i + a_i \le d_f + a_f, \; i \in \mathcal{N} \tag{18}$$
$$F(d_f + a_f) - C_f a_f \ge 0 \tag{19}$$
$$F(d_i + a_i) - (\sigma_f + \rho_f(d_i + a_i - d_f)) \ge 0,$$
$$d_f + a_f < d_i + a_i, \; i \in \mathcal{N} \tag{20}$$

3.2 Admission control algorithms for EDF schedulers

Let us consider the problem of computing the minimum delay $\overline{d_f}$ that can be guaranteed to a flow f characterized by (C_f, σ_f, ρ_f) at an EDF scheduler that has allocated a schedulable set $\mathcal{N}$ of flows. This reduces to the problem of computing the minimum value for d_f that will satisfy the constraints (15)-(20). In the following we explain intuitively the solution to this problem. The formal solution is given in Theorem 2.

From (12) it follows that F has the general form shown in Figure 3: it is continuous, linear on intervals, concave on the intervals $(0, u_1)$, (u_1, u_2),.. (u_{N-1}, u_N) and convex in u_1, u_2,.. u_N.

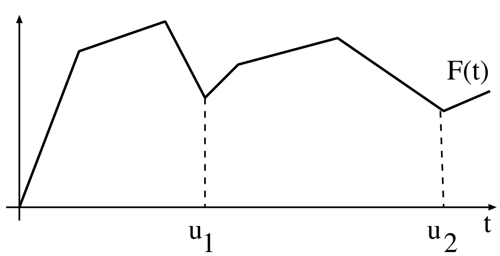

Figure 3:

Given the flow f with envelope A_f^* as in Figure 1, the problem of finding the minimum value for d_f (let us call this minimum value $\overline{d_f}$) that can be guaranteed to f reduces to determining the leftmost position for A_f^* such that it is below the graph of F for all $t \ge 0$, as in Fig.4. Three sets of constraints are imposed by F

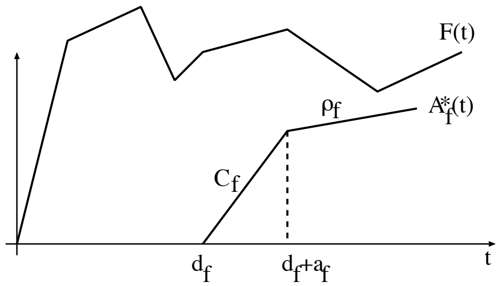

Figure 4: An example of envelope A_f^* below function F

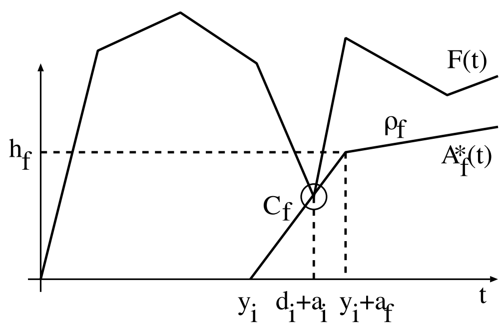

Figure 5:

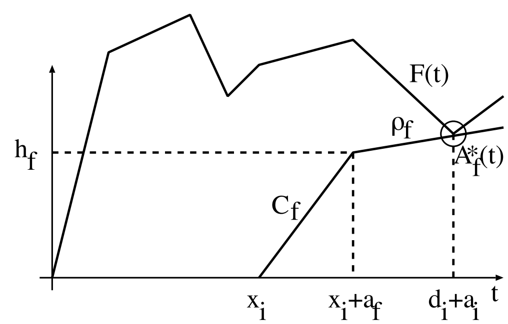

Figure 6:

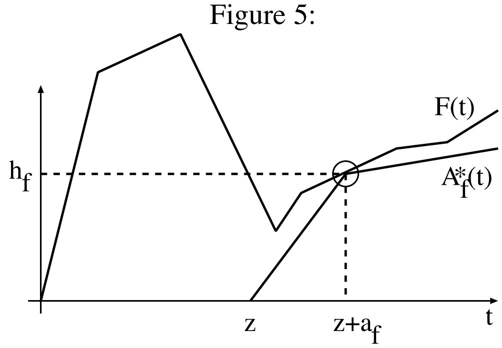

Figure 7:

on A_f^*.

- The first segment of $A_f^*(t-\overline{d}_f)$, for $t \in (\overline{d}_f, \overline{d}_f + a_f)$ must lie below any local minimum of F that is less than h_f. This is expressed by (18) and is depicted in Figure 5. By defining y to be a lower bound on $\overline{d}_f$ imposed by the local minimum in d_i+a_i of F on the first part of A_f^*, we have that $\overline{d}_f \geq \max_i y_i$.

- The second segment of $A_f^*(t-\overline{d}_f)$, for $t \in (\overline{d}_f + a_f, \infty)$ must lie below any local minimum of F that is greater than h_f. This is expressed by (20) and is depicted in Figure 6. By defining x_i to be a lower bound on $\overline{d}_f$ imposed by the local minimum in $d_i + a_i$ of F on the second part of A_f^*, we have that $\overline{d}_f \geq \max_i x_i$.

- Finally, the concave point $(\overline{d}_f + a_f, h_f)$ of $A_f^*(t-\overline{d}_f)$ must lie below F within any concavity interval of F. This is expressed by (19) and is depicted in Figure 7. By defining z to be a lower bound on $\overline{d}_f$ imposed by F on the concave point of A_f^*, we have that $\overline{d}_f \geq z$.

Theorem 2 gives the formal solution for computing the minimum delay that can be guaranteed to a flow, by stating the above three sets of constraints. The proof can be found in Appendix A. In the following we use the notation $(x)^+ = \max(x, 0)$.

Theorem 2 *Let $(C_i, \sigma_i, \rho_i, d_i)_{i\in\mathcal{N}}$ be a schedulable set of envelopes and let (C_f, σ_f, ρ_f) characterize a new flow f such that the stability condition $\sum_{i\in\mathcal{N}} \rho_i + \rho_f < c$ is satisfied.*

1. *Define $(y_i)_{i\in\mathcal{N}}$ by*
$$F(u_i) = C_f(u_i - y_i), \quad u_i \in \mathcal{P} \tag{21}$$
and let $m_y \triangleq \max_{\substack{i\in\mathcal{N} \\ F(u_i)<h_f}} (y_i)^+$.
Define $(x_i)_{i\in\mathcal{N}}$ by
$$F(u_i) = \sigma_f + \rho_f(u_i - x_i), \quad u_i \in \mathcal{P} \tag{22}$$
and let $m_x \triangleq \max_{\substack{i\in\mathcal{N} \\ F(u_i)\geq h_f}} (x_i)^+$.
Let $m = \max(m_x, m_y)$ and b, $0 \leq b \leq N$, such that
$$u_b < m + a_f \leq u_{b+1} \tag{23}$$
where $u_b, u_{b+1} \in \overline{\mathcal{P}}$. Then b exists and is unique.

2. *Define z as follows.*
   - *If*
   $$F(u_b) < h_f < F(u_{b+1}) \tag{24}$$
   z satisfies
   $$u_b < z + a_f < u_{b+1} \tag{25}$$
   and
   $$F(z + a_f) = h_f \tag{26}$$
   - *Otherwise ($F(u_b) \geq h_f$ or $F(u_{b+1}) \leq h_f$), define $z = 0$.*

   Then z exists and is unique.

3. *Define $\overline{d}_f$ by*
$$\overline{d}_f = \max(m, z) \tag{27}$$
Then $\overline{d}_f$ is the minimum delay that can be guaranteed to flow f.

Based on Theorem 2 we present in Figures 9 and 10, a set of algorithms for admission control of (C, σ, ρ) flows that support the flow setup protocol described in Section 2. MINIMUM_DELAY takes as inputs the characteristics and delay guarantees for the existing flows in $\mathcal{N}$ and the characteristics for the new flow f. It outputs the minimum delay guarantee-able for f. First (lines 1-2) it checks whether there is sufficient capacity to accept the new flow. If there is (lines 3-7), it computes a lower bound m on $\overline{d}_f$ based on the constraints imposed by the local minima of F on the first and second parts of A^*_f, as described in Theorem 2.1. It then determines (line 8) the concavity interval (u_b, u_{b+1}) of F where $m + a_f$ is situated. If condition (24) is met in this interval, another lower bound on $\overline{d}_f$, z is computed (line 9-11) based on the constraint imposed by the concave point on A^*_f, as described in Theorem 2.2. The final value of $\overline{d}_f$ is the maximum of all the lower bounds computed so far (line 12).

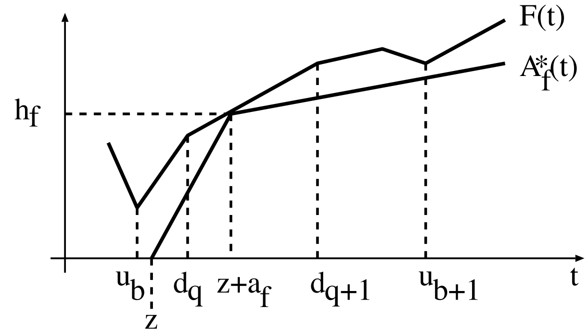

Figure 8: Computing z

MINIMUM_DELAY(input: $(C_i, \sigma_i, \rho_i, d_i)_{i \in \mathcal{N}}$, (C_f, σ_f, ρ_f); output: $\overline{d}_f$)

1 **if** $c \leq \sum_{i \in \mathcal{N}} \rho_i + \rho_f$
2   **then exit** "cannot accept flow f"
3 **for** $i = 1$ **to** N **do**
4   **if** $F(u_i) \geq h_f$
5   **then** $x_i \leftarrow u_i - \frac{F(u_i) - \sigma_f}{\rho_f}$
6   **else** $y_i \leftarrow u_i - \frac{F(u_i)}{C_f}$
7 $m \leftarrow \max_{1 \leq i \leq N}((x_i)^+, (y_i)^+)$
8 **find** b such that $u_b < m + a_f \leq u_{b+1}$ with $u_b, u_{b+1} \in \overline{\mathcal{P}}$
9 **if** $F(u_b) < m + a_f < F(u_{b+1})$
10   **then** COMPUTE_z(b, $(C_i, \sigma_i, \rho_i, d_i)_{i \in \mathcal{N}}$, (C_f, σ_f, ρ_f); z)
11   **else** $z \leftarrow 0$
12 $\overline{d}_f \leftarrow \max(m, z)$

Figure 9: An $O(N^2)$ algorithm for computing the minimum delay for (C, σ, ρ) flow at EDF scheduler

COMPUTE_z(Input: $b, (C_i, \sigma_i, \rho_i, d_i)_{i\in\mathcal{N}}, (C_f, \sigma_f, \rho_f)$;

Output: z)

find $q, 0 \le q \le N$ such that ($d_q \le u_b$ or $F(d_q) \le h_f$) and ($d_{q+1} \ge u_{b+1}$ or $F(d_{q+1}) > h_f$)

$$z \leftarrow \frac{h_f - ca_f + \sum_{\substack{i=1\\ d_i+a_i\le u_b}}^N (\sigma_i + \rho_i(a_f - d_i)) + \sum_{\substack{i=1\\ d_i+a_i\ge u_{b+1}\\ d_i\le d_q}}^N C_i(a_f - d_i)}{c - \sum_{\substack{i=1\\ d_i+a_i\le u_b}}^N \rho_i - \sum_{\substack{i=1\\ d_i+a_i\ge u_{b+1}\\ d_i\le d_q}}^N C_i}$$

Figure 10: Auxiliary $O(N^2)$ algorithm for computing z

To compute z, we use an auxiliary algorithm in Fig 10. z is the solution of $F(z + a_f) = h_f$ given that $F(u_b) < h_f < F(u_{b+1})$. To compute z we need to determine the segment of F in the interval $t \in (u_b, u_{b+1})$ that crosses the horizontal $y = h_f$ (see Fig 8). The segment is defined by that $q, 0 \le q \le N$ such that ($d_q \le u_b$ or $F(d_q) \le h_f$) and ($d_{q+1} \ge u_{b+1}$ or $F(d_{q+1}) > h_f$), (again, see Fig 8). On that segment, $t \in (\max(d_q, u_b), \min(d_{q+1}, u_{b+1}))$ and F has the form

$$F(t) = c(t) - \sum_{\substack{i=1\\ d_i+a_i\le u_b}}^N (\sigma_i + \rho_i(t - d_i)) - \sum_{\substack{i=1\\ d_i+a_i\ge u_{b+1}\\ d_i\le d_q}}^N C_i(t - d_i) \tag{28}$$

Consequently, we have that the equation $F(z + a_f) = h_f$ has the solution:

$$z = \frac{h_f - ca_f + \sum_{\substack{i=1\\ d_i+a_i\le u_b}}^N (\sigma_i + \rho_i(a_f - d_i)) + \sum_{\substack{i=1\\ d_i+a_i\ge u_{b+1}\\ d_i\le d_q}}^N C_i(a_f - d_i)}{c - \sum_{\substack{i=1\\ d_i+a_i\le u_b}}^N \rho_i - \sum_{\substack{i=1\\ d_i+a_i\ge u_{b+1}\\ d_i\le d_q}}^N C_i} \tag{29}$$

Thus, we obtain the algorithm in Fig 10 for computing z.

Observing that if the computation of $F(t)$ is done as in (12) in $O(N)$ time, then MINIMUM_DELAY has complexity $O(N^2)$, thus having a limited practical applicability. In the following we will reduce this complexity in several steps.

The first step in reducing the complexity of the admission control computation is to maintain partial results for $F(d_i + a_i)$ and $F(d_i)$ in order to reduce their complexity to $O(1)$. For each $l, 1 \le l \le N$ define W_l as:

$$W_l = F(u_l), \quad u_l \in \mathcal{P} \tag{30}$$

and $W_0 = 0, W_{N+1} = \infty$. For $1 \le i \le N$ define $V_i = F(d_i)$, and $V_0 = 0, V_{N+1} = \infty$. Define B as

$$B = c - \sum_{i=1}^N \rho_i \tag{31}$$

The new version of MINIMUM_DELAY and COMPUTE_z in Fig 11 and Fig 12 are identical to MINIMUM_DELAY and COMPUTE_z in Fig 9 and Fig 10, with the exception of the computation of F, replaced by W_l and V_i. This new algorithms have complexity $O(N)$, since the computation of F is no longer needed.

In order to update the values of W_l and V_i upon the admission of flow f, (which can be viewed as a resource reservation), the algorithm in Fig 13 updates in step 1 the asymptotic slope of F, and in steps 2-10 and 14-21 the values of F in $(u_l)_{1\leq l\leq N}$, $(d_f + a_f)$, $(d_i)_{1\leq i\leq N}$, and d_f respectively:

$$W_l \leftarrow W_l - A_f^*(u_l - d_f) \quad 1 \leq l \leq N \tag{32}$$

$$W_f \leftarrow F(d_f + a_f) - A_f^*(a_f) \tag{33}$$

$$V_i \leftarrow V_i - A_f^*(d_i - d_f) \quad 1 \leq i \leq N \tag{34}$$

$$V_f \leftarrow F(d_f) \tag{35}$$

In steps 11-13 and 22-24, the algorithm augments the sets $\mathcal{P}$ and $\mathcal{Q}$ with the new values $d_f + a_f$ and d_f respectively, while preserving the sets' non-decreasing order. The ordering of $\mathcal{P}$ is important for an efficient implementation of step 8 of the MINIMUM_DELAY algorithm in Fig 11, and the ordering of $\mathcal{Q}$ is important for an efficient implementation of step 1 of the COMPUTE_z algorithm in Fig 12.

In order to release the resources reserved for a flow upon its termination, the RELEASE algorithm in Fig 14 performs all the operations of the RESERVE algorithm, with an opposite sign.

$$W_l \leftarrow W_l + A_f^*(u_l - d_f) \quad 1 \leq l \leq N \tag{36}$$

$$V_i \leftarrow V_i + A_f^*(d_i - d_f) \quad 1 \leq i \leq N \tag{37}$$

It also eliminates the variables W_l and V_i associated with the terminating flow.

We can see that the algorithms for admission control and resource and release of reservations have $O(N)$ complexity. Although this may be acceptable for small values of N, even this level of computation can be problematic when the number of flows reserved at a link is large (e.g., thousands of flows on an OC12 link). In the next section we explore a technique for further reducing the computation time for flow admission.

MINIMUM_DELAY (Input: $(W_l)_{1\leq l\leq N}, (V_i)_{1\leq i\leq N}, B, (C_i, \sigma_i, \rho_i, d_i)_{1\leq i\leq N}, (C_f, \sigma_f, \rho_f)$;
Output: $\overline{d}_f$)

if $B \leq \rho_f$
  then exit "cannot accept flow f"
for $i = 1$ to N do
    if $W_i \geq h_f$
      then $x_i \leftarrow u_i - \frac{W_i - \sigma_f}{\rho_f}$
      else $y_i \leftarrow u_i - \frac{W_i}{C_f}$
$m \leftarrow \max_{1\leq i\leq N}((x_i)^+, (y_i)^+)$
find b such that $u_b < m + a_f \leq u_{b+1}$, with $u_b, u_{b+1} \in \overline{\mathcal{P}}$
if $W_b < h_f < W_{b+1}$
  then COMPUTE_z(b, $(V_i)_{1\leq i\leq N}$, $(C_i, \sigma_i, \rho_i, d_i)_{1\leq i\leq N}$, (C_f, σ_f, ρ_f); z)
  else $z \leftarrow 0$
$\overline{d}_f \leftarrow \max(m, z)$

Figure 11: An $O(N)$ algorithm for computing the minimum delay for (C, σ, ρ) flow at EDF scheduler

COMPUTE_z(Input: $b, (V_i)_{i\in\mathcal{N}}, (C_i, \sigma_i, \rho_i, d_i)_{i\in\mathcal{N}}, (C_f, \sigma_f, \rho_f)$;
Output: z)

find $q, 0 \leq q \leq N$ such that ($d_q \leq u_b$ or $V_q \leq h_f$) and ($d_{q+1} \geq u_{b+1}$ or $V_{q+1} > h_f$)

$$z \leftarrow \frac{h_f - ca_f + \sum_{\substack{i=1\\ d_i + a_i \leq u_b}}^{N} (\sigma_i + \rho_i(a_f - d_i)) + \sum_{\substack{i=1\\ d_i + a_i \geq u_{b+1}\\ d_i \leq d_q}}^{N} C_i(a_f - d_i)}{c - \sum_{\substack{i=1\\ d_i + a_i \leq u_b}}^{N} \rho_i - \sum_{\substack{i=1\\ d_i + a_i \geq u_{b+1}\\ d_i \leq d_q}}^{N} C_i}$$

Figure 12: Auxiliary $O(N)$ algorithm for computing z

RESERVE (Input: $(W_l)_{1\leq l\leq N}, (V_i)_{1\leq i\leq N}, B, (C_f, \sigma_f, \rho_f, d_f)$;

Output: $(W_l)_{1\leq l\leq N}, (V_i)_{1\leq i\leq N}, B)$

$B \leftarrow B - \rho_f$

/* update of $(W_i)_i$ */

for $i = 1$ to N do

  if $(d_f + a_f \leq d_i + a_i)$

    then $W_i \leftarrow W_i - (\sigma_f + \rho_f(d_i + a_i - d_f))$

    else if $(d_f < d_i + a_i)$

      then $W_i \leftarrow W_i - C_f(d_i + a_i - d_f)$

create variable W_f

let b such that $u_b < d_f + a_f \leq u_{b+1}$

$W_f \leftarrow c(d_f + a_f) - \sum^N_{\substack{i=1 \\ d_i+a_i\leq u_b}} (\sigma_i + \rho_i(d_f + a_f - d_i)) - \sum^N_{\substack{i=1 \\ d_i+a_i\geq u_{b+1} \\ d_i<d_f+a_f}} C_i(d_f + a_f - d_i) - C_f a_f$

insert $d_f + a_f$ in ordered set $\mathcal{P}$

let l be the index of $d_f + a_f$ in $\mathcal{P}$

relabel W_f to W_l

/* update of $(V_i)_i$ */

for $i = 1$ to N do

  if $(d_f + a_f \leq d_i)$

    then $V_i \leftarrow V_i - (\sigma_f + \rho_f(d_i - d_f))$

    else if $(d_f < d_i)$

      then $V_i \leftarrow V_i - C_f(d_i - d_f)$

create variable V_f

$V_f \leftarrow c d_f - \sum^N_{\substack{i=1 \\ d_i+a_i\leq d_f}} (\sigma_i + \rho_i(d_f - d_i)) - \sum^N_{\substack{i=1 \\ d_i<d_f<d_i+a_i}} C_i(d_f - d_i)$

insert d_f in ordered set $\mathcal{Q}$

let l be the index of d_f in $\mathcal{Q}$

relabel V_f to V_l

Figure 13: An algorithm for reservation for flow f

RELEASE (Input: $(W_l)_{1\leq l\leq N}, (V_i)_{1\leq i\leq N}, B, (C_f, \sigma_f, \rho_f, d_f)$;
Output: $(W_l)_{1\leq l\leq N}, (V_i)_{1\leq i\leq N}, B)$

$B \leftarrow B + \rho_f$
/* update of $(W_i)_i$ */
for $i = 1$ to N do
    if $(d_f + a_f \leq d_i + a_i)$
        then $W_i \leftarrow W_i + (\sigma_f + \rho_f(d_i + a_i - d_f))$
        else if $(d_f < d_i + a_i)$
            then $W_i \leftarrow W_i + C_f(d_i + a_i - d_f)$
let l be the index of $d_f + a_f$ in $\mathcal{P}$
destroy variable W_l
/* update of $(V_i)_i$ */
for $i = 1$ to N do
    if $(d_f + a_f \leq d_i)$
        then $V_i \leftarrow V_i + (\sigma_f + \rho_f(d_i - d_f))$
        else if $(d_f < d_i)$
            then $V_i \leftarrow V_i + C_f(d_i - d_f)$
let l be the index of d_f in $\mathcal{Q}$
destroy variable V_l

Figure 14: An algorithm for updating parameters after a flow leave

INIT_STATE(input: c; output: B)

$B \leftarrow c$

Figure 15: State initialization at an empty EDF scheduler

3.3 Discrete admission control algorithms

We have seen in Section 3.2 that in order to verify the schedulability conditions for N flows, we must verify that $F(d_i + a_i) \geq 0$ for $i \in \mathcal{N}$, i.e., that N distinct evaluations of $F()$ must be made. This is why the complexity of algorithms in Section 3.2 cannot be lower that $O(N)$. In this section, we explore a way to reduce this complexity by reducing the number of convex points of F, thus requiring a smaller number of $F \geq A_f$ inequality computation. We achieve this by discretizing the range of horizontal positions of the concave point of the flow's envelope.

Let us define

$$\mathcal{P} = \{e_l | 1 \leq l \leq L\} \tag{38}$$

the set of positions on the "x" axis of the convex points of F. We assume that $\mathcal{P}$ and the extended set $\overline{\mathcal{P}} = \mathcal{P} \cup \{-\infty, 0, \infty\}$ are indexed in a non-decreasing order:

$$e_{-1} = -\infty < e_0 = 0 < e_l \leq e_{l+1} < e_{L+1} = \infty, \quad 1 \leq l \leq L-1. \tag{39}$$

For each flow i with envelope A_i^* given by (C_i, σ_i, ρ_i) and with delay requirement d_i, we reserve a "cover" envelope $A_i^o(t)$, with

$$A_i^o(t) = A_i^*(t - (e_{b_i} - a_i)) \tag{40}$$

where b_i is such that $e_{b_i} \leq d_i + a_i < e_{b_i+1}$. A_i^o is a translation of $A_i^*(t - d_i)$ such that its concave point matches the nearest discretization point at its left, as in Fig. 16. It follows that all the concave points of A_i^o have their "x" coordinates in $\mathcal{P}$, and thus all convex points of F have their "x" coordinates in $\mathcal{P}$.

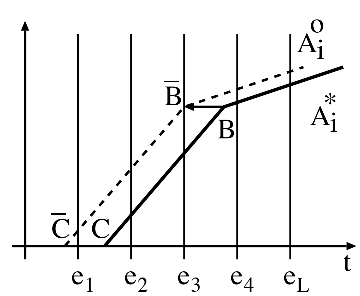

Figure 16: Cover envelope A_i^o constructed from A_i^*

For a given set of envelopes $(A_i^*, d_i)_{1 \leq i \leq N}$, after reserving A_i^o, F becomes:

$$F(t) = ct - \sum_{i=1}^{N} A_i^o(t) \tag{41}$$

Since all A_i^o are determined by their values in points in $\mathcal{P}$ and by their slope $\rho_i = \lim_{t \to \infty} dA_i^o(t)/dt$, then the same is true for F. We define

$$W_l \triangleq F(e_l) = ce_l - \sum_{i=1}^{N} A_i^o(e_l), \quad 1 \leq l \leq L \tag{42}$$

and

$$B \triangleq c - \sum_{i=1}^{N} \rho_i \tag{43}$$

and we observe that F is uniquely defined by $(W_l)_{1 \leq l \leq L}$ and B.

The minimum delay guarantee-able to flow f with envelope A_f^* is the smallest value $\overline{d}_f$ that satisfies the following conditions:

1. $F(t) \geq A_f^*(t - \overline{d}_f)$ for all convex points of F

2. $F(t) \geq A_f^*(t - \overline{d}_f)$ for the concave point of A_f^*

3. $\lim_{t \to \infty} dF(t)/dt > \rho_f$ (the stability condition)

In the following we give a more precise definition of the algorithm for computing the minimum delay that can be guaranteed to a flow using the discretization method described above. We have seen that all local minima of F, that are included in the set $(d_i + a_i)_{i \in \mathcal{N}}$, are also included in the set $\mathcal{P}$. Consequently the schedulability conditions in (13) become:

$$\begin{aligned} F(e_j) = c(e_j) - \sum_{\substack{i \in \mathcal{N} \\ d_i + a_i \leq e_j}} (\sigma_i + \rho_i(e_j - d_i)) \\ - \sum_{\substack{i \in \mathcal{N} \\ d_i < e_j < d_i + a_i}} C_i(e_j - d_i) \geq 0 \quad 1 \leq j \leq L \end{aligned} \tag{44}$$

Given the set $\mathcal{N}$ of flows and a new flow f, the constraints imposed on any guarantee-able delay d_f for f (besides $d_f \geq 0$) are (from (17)-(20)):

$$F(e_i) \geq 0 \quad 1 \leq i \leq L,\ e_i \leq d_f \tag{45}$$

$$F(e_i) - C_f(e_i - d_f) \geq 0 \quad 1 \leq i \leq L,\ d_f < e_i \leq d_f + a_f \tag{46}$$

$$F(e_i) - (\sigma_f + \rho_f(e_i - d_f)) \geq 0 \quad 1 \leq i \leq L,\ d_f + a_f < e_i \tag{47}$$

Observe that there is no inequality corresponding to (19), as the constraint is included in (46) since the discretization method mandates $d_f + a_f \in \mathcal{P}$. Thus the computation of z in point 2 of Theorem 2 is no longer needed. In the following theorem we give the formal solution for computing the minimum delay that can be guaranteed to a new flow.

Theorem 3 *Let an EDF scheduler have capacity c and a set of flex point positions $\mathcal{P} = \{e_l | 1 \leq l \leq L\}$. Let $(C_i, \sigma_i, \rho_i, d_i)_{1 \leq l \leq N}$ with $d_i + a_i \in \mathcal{P}$ be a schedulable set of envelopes, and let (C_f, σ_f, ρ_f) characterize a new flow f such that the stability condition $\sum_{i \in \mathcal{N}} \rho_i + \rho_f < c$ is satisfied.*

1. Let $(y_i)_{1 \leq i \leq L}$ such that

$$F(e_i) = C_f(e_i - y_i) \quad e_i \in \mathcal{P} \tag{48}$$

and let $m_y = \max_{\substack{1 \le i \le L \\ F(e_i) < h_f}} (y_i)^+$.
Let $(x_i)_{1 \le i \le L}$ *such that*

$$F(e_i) = \sigma_f + \rho_f (e_i - x_i) \quad e_i \in \mathcal{P} \tag{49}$$

and let $m_x = \max_{\substack{1 \le i \le L \\ F(e_i) \ge h_f}} (x_i)^+$.
Let $m = \max(m_x, m_y)$ *and* b*,* $0 \le b \le L$*, such that*

$$e_b < m + a_f \le e_{b+1} \tag{50}$$

Then b *exists and is unique.*

2. *Define* $\overline{d}_f = e_{b+1} - a_f$*. If* $0 \le \overline{d}_f < \infty$*, then* $\overline{d}_f$ *is the minimum delay that can be guaranteed to flow* f *by the discrete admission control. Otherwise,* f *cannot be scheduled.*

The proof follows from Theorem 2 and can be found in Appendix B.

Using Theorem 3 we can give the algorithms for discrete admission control for (C, σ, ρ) flows. In Figure 17, we give an algorithm to compute the minimum delay that can be guaranteed to a new flow (C_f, σ_f, ρ_f), using a set of pre-computed parameters $W_i = F(e_i)$ and $B = \sum_{i \in \mathcal{N}} \rho_i$. The algorithms in Figure 18 and Figure 19 update $(W_i)_{1 \le i \le L}$ and B to reserve and release resources for flow $(C_f, \sigma_f, \rho_f, e_f)$. State initialization at an empty scheduler is given in 20). We can easily see that we have an overall $O(L)$ complexity algorithm for discrete admission control of (C, σ, ρ) flows.

MINIMUM_DELAY(Input: $(W_i)_{1 \le i \le L}, B, (e_i)_{1 \le i \le L}, (C_f, \sigma_f, \rho_f)$;
Output: $\overline{d}_f$)

if $B \le \rho_f$
  then exit "cannot accept flow f"
for $i = 1$ **to** L **do**
    if $W_i \ge h_f$
      then $x_i \leftarrow e_i - \frac{W_i - \sigma_f}{\rho_f}$
      else $y_i \leftarrow e_i - \frac{W_i}{C_f}$
$m \leftarrow \max_{1 \le i \le L}((x_i)^+, (y_i)^+)$
find b **such that** $e_b < m + a_f \le e_{b+1}$
if $b = L$ or $e_{b+1} - a_f < 0$
  then exit "cannot accept flow f"
  else $\overline{d}_f \leftarrow (e_b - a_f)$

Figure 17: An $O(L)$ algorithm for computing the minimum delay for (C, σ, ρ) flow, guaranteed by a discrete admission control

RESERVE(Input: $(W_i)_{1\leq i\leq L}, B, (e_i)_{1\leq i\leq L}, (C_f, \sigma_f, \rho_f, e_f)$;
Output: $(W_i)_{1\leq i\leq L}, B$)

$B \leftarrow B - \rho_f$
/* update of $(W_i)_i$ */
for $i \leftarrow 1$ **to** L **do**
  if $(e_f \leq e_i)$
    then $W_i \leftarrow W_i - (\sigma_f + \rho_f(e_i - (e_f - a_f)))$
    else if $(e_f - a_f < e_i)$
      then $W_i \leftarrow W_i - C_f(e_i - (e_f - a_f))$

Figure 18: Reservation algorithm for discrete admission control

RELEASE(Input: $(W_i)_{1\leq i\leq L}, B, (e_i)_{1\leq i\leq L}, (C_f, \sigma_f, \rho_f, e_f)$;
Output: $(W_i)_{1\leq i\leq L}, B$)

$B \leftarrow B + \rho_f$
/* update of $(W_i)_i$ */
for $i \leftarrow 1$ **to** L **do**
  if $(e_f \leq e_i)$
    then $W_i \leftarrow W_i + (\sigma_f + \rho_f(e_i - (e_f - a_f)))$
    else if $(e_f - a_f < e_i)$
      then $W_i \leftarrow W_i + C_f(e_i - (e_f - a_f))$

Figure 19: Release algorithm for discrete admission control

INIT_STATE(input: $c, (e_i)_{i\in\mathcal{L}}$; output: $(W_i)_{i\in\mathcal{L}}, B$)

$B \leftarrow c$
/* Init of $(W_i)_i$ */
for $i \leftarrow 1$ **to** L **do**
  $W_i \leftarrow ce_i$

Figure 20: State initialization at an empty EDF scheduler with discrete admission control

4 Evaluation of admission control algorithms through simulations

We are interested in two directions of evaluation for the admission control algorithms. One is to asses the benefit of discrete admission control over the exact algorithm by comparing their respective runnning times in a simulation environment. The other direction concerns the shortcoming of the discrete admission control to potentially admitting less flows due to a conservative resource reservation. We measure the link blocking probability yielded by the exact and discrete algorithms through simulation.

We consider a link that forwards ATM traffic according to the EDF scheduling policy. The characteristics of the flows to be serviced at this link are generated randomly and are intended to cover a wide range of traffic patterns. In our simulations we take $\rho = 10^p Kb/s$ where p is uniformly distributed in $[1, 3]$, that makes ρ cover the range $[10Kb/s, 1Mb/s]$. From multiple video and audio traces we have observed that both C and σ are correlated with ρ. In our simulations we take $C = q * \rho Kb/s$, where q is uniformly distributed in $[2, 5]$. Similarly, $\sigma = r * \rho Kb$ where r is uniformly distributed in $[0.8, 1.6]$. Observe that the range of generated traffic patterns include a typical MPEG video source (sequence of advertisements presented in [9]) with peak rate $C = 1Mb/s$, mean rate $\rho = 500Kb/s$, burst size $\sigma = 500Kb$, and a typical packetized voice source (see e.g., [3]) with peak rate $C = 32Kb/s$, mean rate $\rho = 10Kb/s$, burst size $\sigma = 8Kb$. Flows are created according to a Poisson process with parameter α and their duration is exponentially distributed with mean $1/\beta$. The ratio α/β characterizes the load offered to the link, i.e., the average number of flows that would exist at any time at a link with no capacity limitation. Each flow has a delay requirement $d = 10^s * 30ms$, where s is uniformly distributed in $[0, 1.52]$, thus d ranging in $[30ms, 1s]$. After a flow is generated with the above parameters, its EDF schedulability is verified by our admission control algorithms. We generate 100000 flows in one simulation run, and we are interested in the link blocking probability, i.e., the ratio between the number of rejected flows and the total number of generated flows. We take the link blocking probability for an admission control algorithm as an indication of its performance. In our simulations, we use the method of independent replications to generate 90% confidence intervals for the link blocking probability.

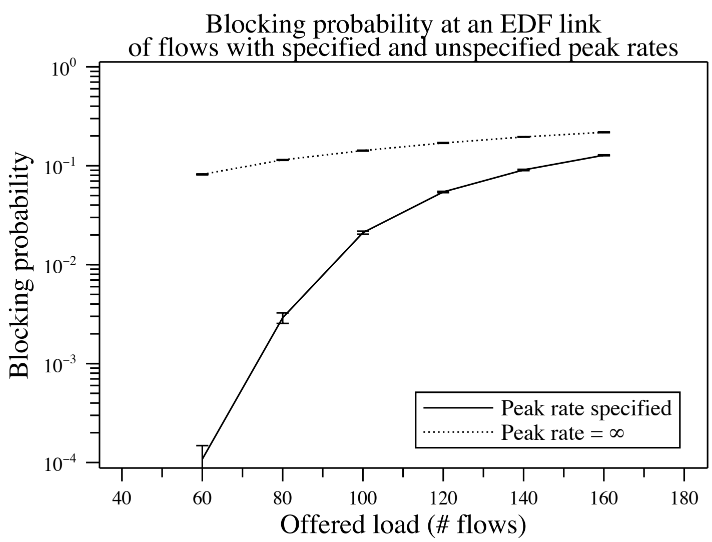

Figure 21:

In the first experiment (Figure 21), we evaluate the impact of not specifying the flow's peak rate (i.e. assuming peak rate infinite). We note that the peak rate is an optional parameter in both ATM and Internet "Int-Serv Guaranteed Service" specifications. Avoiding peak rate specification will result in simpler admissibility tests (see Appendix C). However, the relaxed constraints on source behavior (by not specifying its peak) will result in fewer flows being admitted to the network. Figure 21 quantitatively shows this tradeoff.

We see that the performance degradation is quite severe (orders of magnitude increase in blocking probability). Moreover, the complexity of admission control algorithms is improved when ignoring the peak rate by only a constant factor, having the same asymptotic complexity as the algorithms that consider the peak rate (see Appendix C for details). We conclude here that the use of peak rate in flow characterizations is highly desirable for achieving good link utilization. In the remainder of our simulation experiments we will include the peak rate in the characterization of flows.

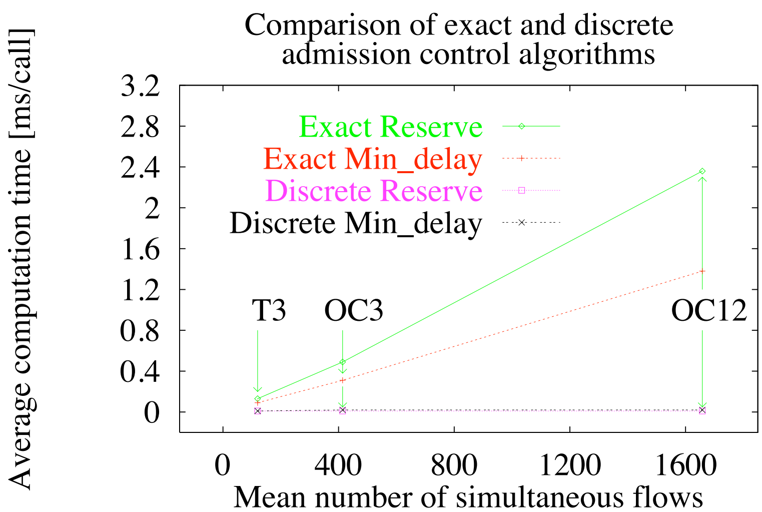

Figure 22:

In the following we compare the computational performance of discrete admission control algorithms (having 13 discretization points) with the exact algorithm when both operate in the same environment. Both algorithms input the same series of flows under three scenarios: link capacity $45Mb/s$ (T3) and offered load 120 flows; link capacity $155.52Mb/s$ (OC3) and offered load 414 flows; link capacity $622.08Mb/s$ (OC12) and offered load 1658 flows. The offered loads have been chosen to incur the same blocking probability (0.05) in all three scenarios. Given this low rejection probability, the average number of flows N reserved at the link at any time is approximately equal to the offered load. The average computation time has been measured with the GNU code profiler $gprof$ on a DECalpha 347. We know that the exact admission control algorithm has an asymptotic computation complexity of $O(N)$, which is confirmed by the linear shape of the plots of the exact algorithms for MIN_DELAY, RESERVE in Figure 22. The figure confirms also that the execution time (0.01ms/function call) of discrete admission control is independent of the number of flows N. Most importantly, the figure shows the very large gain in computation time for the discrete admission control: 240 times faster for an OC12 link having an average load of 1658 flows.

For the rest of our simulations we consider a T3 link ($45Mb/s$). In the following we evaluate the penalty in link performance when using the discrete admission control. Recall that the discrete algorithms in Section 3.3 take their discretization point values from a finite set $\mathcal{P} = \{e_i | 1 \leq i \leq L\}$. A large spacing between discretization points implies a significant over-reservation for a flow, that would translate in fewer flows being admitted (higher blocking probability). A small spacing between discretization points, on the other hand, results in a large number of points and consequently a higher overhead for the admission control algorithms. In the following we address two issues. First, for a fixed number of points, what is a good policy for choosing the spacing between points? Second, given that we have found a good spacing policy, what is a number of points that is sufficient for good link performance and small enough for low computational overhead.

One possibility for spacing of discretization points is equal (linear) spacing:

$$e_2 - e_1 = e_3 - e_2 = .. = e_L - e_{L-1}$$

Another possibility is to have the points geometrically spaced:

$$\frac{e_3 - e_2}{e_2 - e_1} = \frac{e_4 - e_3}{e_3 - e_2} = .. = \frac{e_L - e_{L-1}}{e_{L-1} - e_{L-2}} = \text{spacing factor}$$

This latter spacing policy is expected to result in a smaller over-reservation for a small distance between discretization points compared to the linear policy, due to a smaller space the request falls in.

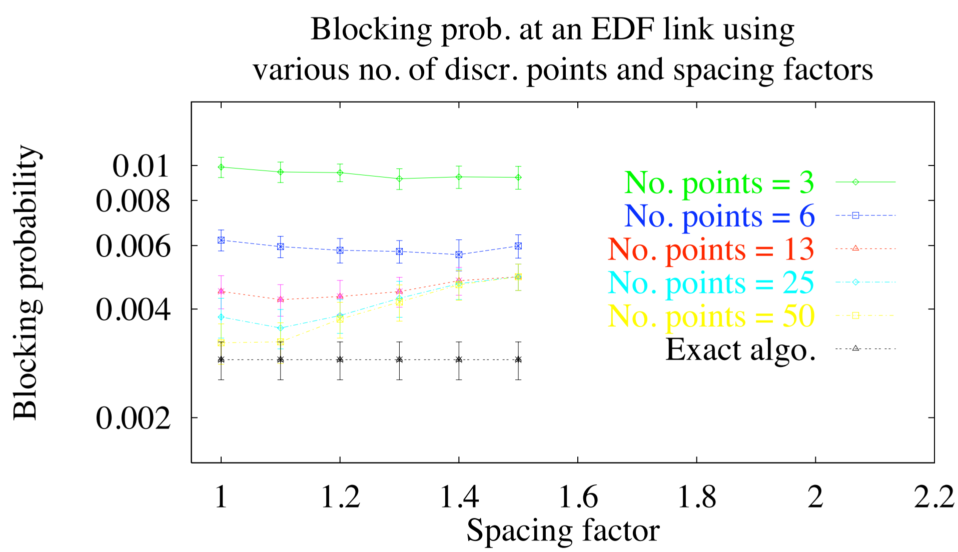

Figure 23:

In Figure 23 we plot the results of our simulations for values of spacing factor between 1 and 1.5, value 1 corresponding to linear spacing. The graph "Exact algo." corresponds to the exact admission control algorithm, which forms the base case for our comparison. First, we note that with less that 13 points, the blocking probability is unacceptably high, compared to the base case. For the rest (more than 13 points), we see that the linear spacing policy can provide link performance close to or better than that given by the geometric spacing policy, with any spacing factor. For this scenario, the linear spacing is the solution of choice due to its simplicity and near optimal performance.

In Figure 24 we plot the results of simulation experiments with algorithms using linear spacing and various number of discretization points. We can see that 13 points are sufficient to provide link utilization within 10% off the optimal (compare the offered load for the same link blocking probability). This result confirms our estimation in Appendix D that about 10 points are sufficient to have a performance tradeoff within 10%.

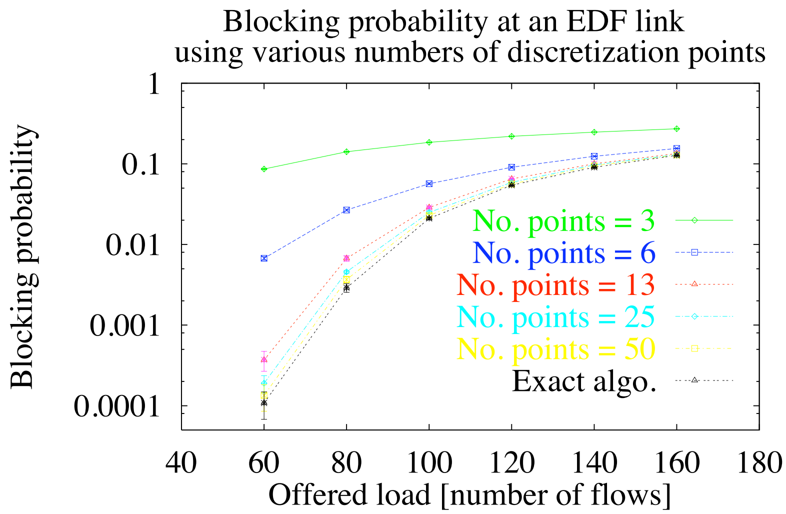

Figure 24:

5 Conclusion and future work

In this paper we have proposed practical solutions to the problem of admission control for real-time flows with delay guarantees at an EDF scheduler, as a part of end-to-end flow admission control in IP and ATM networks. We applied the admission control conditions put forward by [11] to flows characterized by peak rate, mean rate and burst size. We developed a first set of algorithms with a computation complexity of $O(N)$, where N is the number of flows admitted in the EDF scheduler at the time of algorithm invocation. A second set of algorithms places the horizontal position of concave points of flow envelopes into a predefined set of values (discretization points), thus reducing the computational complexity of admission control to $O(L)$, where L is the number of predefined discretization points. A set of simulation experiments showed that the performance improvement achieved by the discrete admission control is indeed very important (240 times faster for an OC12 link) and that the algorithms' execution time is independent of the number of flows admitted. Moreover, we have seen that the link performance degradation of the discrete admission control relative to the exact admission control is less that 10%, while using a small number of discretization points (13). Taken together, these results suggest that the algorithms we have studied in this paper form the basis of a practical and highly efficient solution for the problem of admission control of real-time flows at EDF schedulers.

Our present work can be extended in several ways. First, we can generalize our results to take into consideration packet sizes at non-preemptive EDF schedulers. Second, both exact and discrete admission control algorithms can be extended to flows characterized by multiple (σ, ρ) pairs (i.e. envelopes consisting of multiple linear segments). This characterization has the potential to increase link utilization in comparison to the two-segment characterization used in the present work.

A Proof of Theorem 2

1. By definition, $\overline{\mathcal{P}} = (u_i)_{0 \le i \le N+1}$ is non-decreasing with i. Since $m > u_0 = -\infty$ and $m < u_{N+1} = \infty$ we conclude that there is a unique $b, 0 \le b \le N$ such that (23) is satisfied.

2. Assume (24), $F(u_b) < h_f < F(u_{b+1})$.

For $t \in (u_b, u_{b+1})$ we have:

$$F(t) = ct - \sum_{\substack{i=1 \\ u_i \le u_b}}^{N} (\sigma_i + \rho_i(t - d_i)) - \sum_{\substack{i=1 \\ u_i \ge u_{b+1} \\ d_i < t}}^{N} C_i(t - d_i)$$

and thus:

$$\frac{dF}{dt}(t) = c - \sum_{\substack{i=1 \\ u_i \le u_b}}^{N} \rho_i - \sum_{\substack{i=1 \\ u_i \ge u_{b+1} \\ d_i < t}}^{N} C_i$$

Hence, dF/dt is non-increasing, since more C_i terms are possibly subtracted as t increases. Consequently, F is concave on (u_b, u_{b+1}). Given that F is continuous and concave, and $F(u_b) < h_f < F(u_{b+1})$ by (24), we conclude that $F(t) = h_f$ has a unique solution in (u_b, u_{b+1}). Let z such that $z + a_f \in (u_b, u_{b+1})$ and $F(z + a_f) = h_f$. Observe that:

$$F(t) < h_f \quad t \in (u_b, z + a_f) \tag{51}$$

$$F(t) > h_f \quad t \in (z + a_f, u_{b+1}) \tag{52}$$

which follow from F having the above mentioned properties.

3. We prove that $\overline{d}_f$ is the minimum delay guarantee-able to f.

First, observe that

$$\overline{d}_f \ge 0 \tag{53}$$

since $\overline{d}_f \ge m \ge m_x \ge 0$. Also, observe that

$$u_b < \overline{d}_f + a_f \le u_{b+1} \tag{54}$$

which follows from (23) and (25).

3.1. We prove that $\overline{d}_f$ can be guaranteed to flow f.

Below are the scheduling conditions for the set $\mathcal{N} \cup \{f\}$ ((15)-(20)):

$$d_i \ge 0 \quad i \in \mathcal{N} \tag{55}$$

$$\overline{d}_f \ge 0 \tag{56}$$

$$F(u_i) \ge 0 \quad i \in \mathcal{N},\ u_i \le \overline{d}_f \tag{57}$$

$$F(u_i) - C_f(u_i - \overline{d}_f) \ge 0 \quad i \in \mathcal{N},\ \overline{d}_f < u_i \le \overline{d}_f + a_f \tag{58}$$

$$F(\overline{d}_f + a_f) - h_f \ge 0 \tag{59}$$

$$F(u_i) - (\sigma_f + \rho_f(u_i - \overline{d}_f)) \ge 0 \quad i \in \mathcal{N},\ \overline{d}_f + a_f < u_i \tag{60}$$

3.1.1 Since we assume that the set $\mathcal{N}$ is schedulable, (55) and (57) are true. Relation (56) follows from (53).

3.1.2 Consider (60). We begin by establishing

$$F(u_i) \geq h_f \ , \quad \forall i \in \mathcal{N},\ i \geq b+1 \tag{61}$$

by contradiction. Suppose that there exists $k \geq b+1$ such that $F(u_k) < h_f = C_f a_f$. Then

$$m \geq m_y \geq y_k = u_k - F(u_k)/C_f > u_k - a_f \geq u_{b+1} - a_f \quad .$$

Thus $m + a_f > u_{b+1}$, which contradicts (23). Hence (61) holds. Consequently, $\overline{d}_f \geq m \geq m_x \geq x_i$ for $\forall i \geq b+1$. Thus,

$$F(u_i) \geq \sigma_f + \rho_f(u_i - \overline{d}_f), \quad \forall i \in \mathcal{N},\ i \geq b+1\ . \tag{62}$$

Since $\overline{d}_f + a_f \geq m + a_f > u_b$ (from (23)) we have that $\overline{d}_f + a_f < u_i, i \in \mathcal{N}$ implies $u_i \geq u_{b+1}, i \in \mathcal{N}$, and with (62) we conclude (60).

3.1.3 Let us establish (58). Let $u_i \leq \overline{d}_f + a_f$. If $F(u_i) \geq h_f$ then, since $h_f = C_f a_f \geq C_f(u_i - \overline{d}_f)$ for $u_i \leq \overline{d}_f + a_f$, we have $F(u_i) - C_f(u_i - \overline{d}_f) \geq 0$. Conversely, if $F(u_i) < h_f$ then $\overline{d}_f \geq m \geq m_y \geq y_i$ and from (21) we have $F(u_i) \geq C_f(u_i - \overline{d}_f)$ and we conclude (58).

3.1.4 Let us prove (59). Given that $\overline{d}_f + a_f \leq u_{b+1}$ by (54), we have three cases.

3.1.4.1 $\mathbf{\overline{d}_f + a_f = u_{b+1}}$. Then (59) follows from (58) with $u_i = u_{b+1}$.

3.1.4.2 $\mathbf{F(u_b) \geq h_f}$. Since $F(u_{b+1}) \geq h_f$ (from (61)), and because $F(t)$ is continuous and concave for $t \in (u_b, u_{b+1})$, we have $F(t) \geq h_f, \forall t \in [u_b, u_{b+1}]$. Thus $F(\overline{d}_f + a_f) \geq h_f$ from (54).

3.1.4.3 $\mathbf{\overline{d}_f + a_f < u_{b+1}}$ **and** $\mathbf{F(u_b) < h_f}$. We have $F(u_{b+1}) > \sigma_f + \rho_f a_f = h_f$:

$$\begin{aligned} F(u_{b+1}) &= \sigma_f + \rho_f(u_{b+1} - x_{b+1}) \quad \text{by (22)} \\ &\geq \sigma_f + \rho_f(u_{b+1} - \overline{d}_f) \quad \text{by } \overline{d}_f \geq x_{b+1} \\ &> \sigma_f + \rho_f a_f \quad \text{by } \overline{d}_f + a_f < u_{b+1} \end{aligned}$$

According to point 2 of Theorem 2, z exists and by (52),

$$h_f \leq F(t), \quad t \in [z + a_f, u_{b+1}] \tag{63}$$

It follows from the fact that $\overline{d}_f \geq z$ and from (54) that $\overline{d}_f + a_f \in [z + a_f, u_{b+1}]$, which, with (63), implies (59).

By establishing that (55)-(60) are satisfied by $\overline{d}_f$, we have proved that $\overline{d}_f$ can be guaranteed to flow f.

3.2. We prove by contradiction that $\overline{d}_f$ is the minimum delay that can be guaranteed to f.

Suppose there is a delay $\overline{d}'_f \geq 0, \overline{d}'_f < \overline{d}_f$, that can be guaranteed to f. We have three cases:

3.2.1 $\mathbf{\overline{d}_f = m_x}$ **and** $\mathbf{0 \leq \overline{d}'_f < m_x}$. For any i such that $F(u_i) > h_f$ we have $u_i - x_i > a_f$ from (22). Thus, for any $i \leq b$, $x_i + a_f < u_i \leq u_b < m + a_f$ and so $x_i < m_x$ for $i \leq b$ (since $\overline{d}_f = m_x = m$). From this and (61) we have $m_x = \max_{i \geq b+1}(x_i)^+$. Thus $0 \leq \overline{d}'_f < m_x$ implies that there is $k, b+1 \leq k \leq N$ such that $\overline{d}'_f < x_k$. But $\overline{d}'_f + a_f < m + a_f \leq u_{b+1} \leq u_k$, and thus, for $\mathcal{N}' = \mathcal{N} \cup \{f\}$ to be schedulable it is necessary that (20):

$$F(u_k) - (\sigma_f + \rho_f(u_k - \overline{d}'_f)) \geq 0$$

From $\overline{d}'_f < x_k$ and from (22) with $i = k$ we have

$$F(u_k) = \sigma_f + \rho_f(u_k - x_k) < \sigma_f + \rho_f(u_k - \overline{d}'_f)$$

which contradicts the assumption that $\overline{d}'_f$ can be guaranteed.

3.2.2 $\overline{\mathbf{d}}_\mathbf{f} = \mathbf{m_y}$ and $\mathbf{0} \le \overline{\mathbf{d}}'_\mathbf{f} < \mathbf{m_y}$. From (61) follows that $m_y = \max_{\substack{i \le b \\ F(u_i) < h_f}} (y_i)^+$. Thus $0 \le \overline{d}'_f < m_y$ implies that there is k, $1 \le k \le b$ such that $\overline{d}'_f < y_k$ and $F(u_k) < h_f$. If $u_k > \overline{d}'_f + a_f$ then for $\mathcal{N}'$ to be schedulable it is necessary that (20):

$$F(u_k) - (\sigma_f + \rho_f(u_k - \overline{d}'_f)) \ge 0$$

But $F(u_k) < h_f$ and $\sigma_f + \rho_f(u_k - \overline{d}'_f) > \sigma_f + \rho_f a_f = h_f$ which leads to contradiction.

Conversely, let us consider $u_k \le \overline{d}'_f + a_f$. From (21) taking $i = k$, and since $F(t) \ge 0$ $\forall t$ ($\mathcal{N}$ is schedulable), we have $y_k \le u_k$. So, $\overline{d}'_f < u_k$. Thus, for $\mathcal{N}'$ to be schedulable it is necessary that (18):

$$F(u_k) - C_f(u_k - \overline{d}'_f) \ge 0$$

But from (21) with $i = k$ and $\overline{d}'_f < y_k$ we have

$$F(u_k) = C_f(u_k - y_k) < C_f(u_k - \overline{d}'_f)$$

which leads to a contradiction.

3.2.3 $\overline{\mathbf{d}}_\mathbf{f} = \mathbf{z}$ and $\mathbf{0} \le \overline{\mathbf{d}}'_\mathbf{f} < \mathbf{z}$. Then $z > 0$ and $F(u_b) < h_f < F(u_{b+1})$ from point 2 of Theorem 2, and

$$\overline{d}'_f + a_f < z + a_f \tag{64}$$

If $\overline{d}'_f + a_f \le u_b$ then for $\mathcal{N}'$ to be schedulable it is necessary that (20):

$$F(u_b) - (\sigma_f + \rho_f(u_b - \overline{d}'_f)) \ge 0$$

But $F(u_b) < h_f$ and $\sigma_f + \rho_f(u_b - \overline{d}'_f) \ge \sigma_f + \rho_f a_f = h_f$, which leads to contradiction.

Conversely, let us consider $u_b < \overline{d}'_f + a_f$. For $\mathcal{N}'$ to be schedulable it is necessary that (19):

$$F(\overline{d}'_f + a_f) - h_f \ge 0$$

But from $u_b < \overline{d}'_f + a_f$, (64) and (51) we have

$$F(\overline{d}'_f + a_f) < h_f$$

which leads to a contradiction.

Since in all the above cases we have found contradictions, we conclude that $\overline{d}_f$ is the minimum delay that can be guaranteed to f.

B Proof of Theorem 3

From hypothesis, all envelopes reserved at the EDF scheduler have their concave points $d_i + a_i \in \mathcal{P}$, so all convex points of F are in $\mathcal{P}$. It follows that schedulability equations (17)-(20) are equivalent to (45)-(47). Let $\overline{d}'_f = \min(m, z)$ be the minimum delay guarantee-able to f by the exact (non-discretized) EDF admission control, as given by Theorem 2. It is easy to see that point 1 of Theorem 3 is equivalent to point 1 of Theorem 2. It follows that m and b defined by Theorem 2 are identical to m and b defined by Theorem 3. From Theorem 2 we have that

$$e_b < \overline{d}'_f + a_f \leq e_{b+1} \ . \tag{65}$$

But the discrete admission control mandates that any delay d_f guaranteed to f should have $d_f + a_f \in (e_i)_{1 \leq i \leq L} = \mathcal{P}$. So, the smallest delay guarantee-able to f by the discrete admission control is $\overline{d}_f = e_{b+1} - a_f$. Point 2 of Theorem 3 then follows.

C EDF Admission Control for (σ, ρ) Token Bucket Flows

In this section we derive simpler admission control algorithms for the particular case, often used in practice, of flows that are token bucket (σ, ρ) constrained and are assumed to have infinite peak rate. The flows are characterized by the following type of envelope:

$$A^*(t) = \begin{cases} 0 & t < 0 \\ \sigma + \rho t & 0 \leq t \end{cases} \tag{66}$$

where

- $\sigma \geq 0$ is the maximum burst size (bits);
- $\rho > 0$ is the average rate of the flow (bits/second).

See Figure 25 for an example of (σ, ρ) envelope.

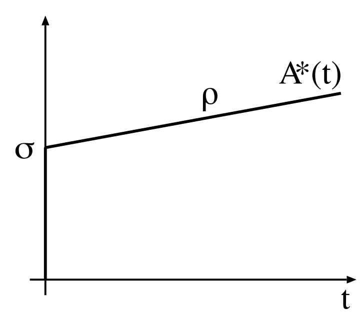

Figure 25: An illustration of (σ, ρ) envelope

Let $\mathcal{N}$ be a set of flows, flow i being characterized by the envelope A_i^* of the form given in (3) and having a maximum packet delay requirement of d_i. The stability condition (1) becomes:

$$\sum_{i \in \mathcal{N}} \rho_i < c \tag{67}$$

In the following we derive schedulability conditions from (2) that are simple to compute. Let $F(t) = ct - \sum_{i\in\mathcal{N}} A_i^*(t-d_i)$. Then the schedulability condition (2) is equivalent to $F(u) \geq 0 \; \forall u \in (d_i)_{i\in\mathcal{N}} \cup \{0\}$. Let us assume, without loss of generality, that the flows in $\mathcal{N} = \{1, 2, ..N\}$ are ordered by (d_i):

$$i < j \Rightarrow d_i \leq d_j \quad \forall i, j \in \mathcal{N} \tag{68}$$

Since the form of A_i^* is given in (3), F becomes:

$$F(t) = \begin{cases} 0 & t < 0 \\ ct & 0 \leq t < d_1 \\ ct - \sum_{\substack{i\in\mathcal{N} \\ i\leq j}} (\sigma_i + \rho_i(t-d_i)) & d_j \leq t < d_{j+1} \\ ct - \sum_{i\in\mathcal{N}} (\sigma_i + \rho_i(t-d_i)) & d_N \leq t \end{cases} \tag{69}$$

Thus the schedulability conditions become (besides $d_i \geq 0, i \in \mathcal{N}$):

$$F(d_j) = cd_j - \sum_{\substack{i\in\mathcal{N} \\ i\leq j}} (\sigma_i + \rho_i(d_j - d_i)) \geq 0 \quad j \in \mathcal{N} \tag{70}$$

Let a new flow f be characterized by (σ_f, ρ_f) and have a delay guarantee d_f, and assume that there is $b \in \mathcal{N}$ such that $d_{b-1} < d_f \leq d_b$. The scheduling conditions for the set $\mathcal{N}' = \mathcal{N} \cup \{f\}$ are:

$$F'(d_j) = cd_j - \sum_{\substack{i\in\mathcal{N}' \\ i\leq j}} (\sigma_i + \rho_i(d_j - d_i)) \geq 0 \quad j \in \mathcal{N}' \tag{71}$$

This is equivalent to the following set of inequalities:

$$d_i \geq 0 \quad i \in \mathcal{N} \tag{72}$$

$$d_f \geq 0 \tag{73}$$

$$F(d_i) \geq 0 \quad d_i \leq d_f, \; i \in \mathcal{N} \tag{74}$$

$$F(d_f) - \sigma_f \geq 0 \tag{75}$$

$$F(d_i) - (\sigma_f + \rho_f(d_i - d_f)) \geq 0 \quad d_f < d_i, \; i \in \mathcal{N} \tag{76}$$

C.1 Exact admission control algorithms for (σ, ρ) flows at EDF schedulers

Theorem 4 *Let $\mathcal{N}$ be a schedulable set of flows $(\sigma_i, \rho_i, d_i)_{i\in\mathcal{N}}$ sorted in increasing order of (d_i), and let (σ_f, ρ_f) characterize a new flow f such that the stability condition $\sum_{i\in\mathcal{N}} \rho_i + \rho_f < c$ is satisfied.*

1. Let $(x_i)_{i\in\mathcal{N}}$ such that

$$F(d_i) = \sigma_f + \rho_f(d_i - x_i) \quad i \in \mathcal{N} \tag{77}$$

and let $m_x = \max_{\substack{i\in\mathcal{N} \\ F(d_i)\geq\sigma_f}} (x_i, 0)$.
Let $m_y = \max_{\substack{i\in\mathcal{N} \\ F(d_i)<\sigma_f}} d_i$.
Let $m = \max(m_x, m_y)$ and b, $1 \leq b \leq N+1$, such that

$$d_{b-1} < m \leq d_b \tag{78}$$

where $d_0 = -\infty$ *and* $d_{N+1} = \infty$. *Then* b *exists and is unique.*

2. If

$$F(d_b) > \sigma_f \tag{79}$$

and

$$F(d_{b-1}) < \sigma_f \tag{80}$$

let z *such that*

$$d_{b-1} < z < d_b \tag{81}$$

and

$$F(z) = \sigma_f \tag{82}$$

Then z *exists, and is unique.*

3. Let $\overline{d}_f$ *such that:*

$$\overline{d}_f = \begin{cases} \max(m, z) & \textit{if (79) and (80)} \\ m & \textit{otherwise} \end{cases} \tag{83}$$

Then $\overline{d}_f \geq 0$ *is the minimum delay that can be guaranteed to flow* f.

MINIMUM_DELAY(input: $(W_i)_{i \in \mathcal{N}}, B, (\sigma_i, \rho_i, d_i)_{i \in \mathcal{N}}, (\sigma_f, \rho_f)$; output: $\overline{d}_f$)

if $B \leq \rho_f$
  then exit "cannot accept flow f"
$m_x \leftarrow 0; m_y \leftarrow 0$
for $i = 1$ **to** N **do**
    if $W_i \geq \sigma_f$
      then $x_i \leftarrow d_i - \frac{W_i - \sigma_f}{\rho_f}$
        $m_x \leftarrow \max(m_x, x_i)$
      else $m_y \leftarrow \max(m_y, d_i)$
$m \leftarrow \max(m_x, m_y)$
$b \leftarrow 1$
while $(b \leq N)$ **and** $(m > d_b)$ **do**
    $b \leftarrow b + 1$
if $[(b = N + 1)$ **or** $(W_b > \sigma_f)]$ **and** $[(b = 1)$ **or** $(W_{b-1} < \sigma_f)]$
  then $z \leftarrow \frac{\sigma_f + \sum_{i=1}^{b-1}(\sigma_i - \rho_i d_i)}{c - \sum_{i=1}^{b-1} \rho_i}$
    $\overline{d}_f \leftarrow \max(m, z)$
  else $\overline{d}_f \leftarrow m$
return $\overline{d}_f$

Figure 26: An $O(N)$ algorithm for computing the minimum delay for (σ, ρ) flow at EDF scheduler

JOIN_UPDATE(input: $(W_i)_{i \in \mathcal{N}}, B, (\sigma_g, \rho_g, d_g)$; output: $(W_i)_{i \in \mathcal{N}}, B$)

$B \leftarrow B - \rho_g$
/* update of $(W_i)_i$ */
for $i \leftarrow 1$ **to** N **do**
    if $(d_g \leq d_i)$
        then $W_i \leftarrow W_i - (\sigma_g + \rho_g(d_i - d_g))$
create variable W_g
let b such that $d_{b-1} < d_g \leq d_b$
$W_g \leftarrow c d_g - \sum_{i=1}^{b-1}(\sigma_i + \rho_i(d_g - d_i)) - \sigma_g$
insert flow g with d_g in ordered set $(d_i)_{i \in \mathcal{N}}$

Figure 27: An algorithm for updating parameters after a flow join

LEAVE_UPDATE(input: $(W_i)_{i \in \mathcal{N}}, B, (\sigma_g, \rho_g, d_g)$; output: $(W_i)_{i \in \mathcal{N}}, B$)

$B \leftarrow B + \rho_g$
/* update of $(W_i)_i$ */
for $i \leftarrow 1$ **to** N **do**
    if $(d_g \leq d_i)$
        then $W_i \leftarrow W_i + (\sigma_g + \rho_g(d_i - d_g))$
destroy variable W_g
extract flow g from the ordered set $(d_i)_i$

Figure 28: An algorithm for updating parameters after a flow leave

INIT_STATE(input: c; output: B)

$B \leftarrow c$

Figure 29: State initialization at an empty EDF scheduler

C.2 Discrete admission control algorithms for (σ, ρ) flows at EDF schedulers

In this section we introduce a set $\mathcal{L} = \{1, 2, ..L\}$ of classes for the delays d_f at the EDF scheduler. If flow f is admitted in delay class k it is given delay guarantee $d_f = e_k$, where $e_1 < e_2 < ... < e_L$ are predefined values. We define c_f to be the class of flow f, $c_f = k$ if $d_f = e_k$.

Theorem 5 *Let a class-based EDF scheduler have a set $\mathcal{L} = \{1, 2, ..L\}$ of classes of delays, $e_1 < e_2 < ..e_L$. Let $\mathcal{N}$ be a schedulable set of flows $(\sigma_i, \rho_i, d_i)_{i \in \mathcal{N}}$, sorted in increasing order of d_i and $d_j \in (e_i)_{i \in \mathcal{L}}$, and let (σ_f, ρ_f) characterize a new flow f such that the stability condition $\sum_{i \in \mathcal{N}} \rho_i + \rho_f < c$ is satisfied.*

1. Let $(x_i)_{i \in \mathcal{L}}$ such that

$$F(e_i) = \sigma_f + \rho_f(e_i - x_i) \quad i \in \mathcal{L} \tag{84}$$

and let $m_x = \max_{\substack{i \in \mathcal{L} \\ F(e_i) \geq \sigma_f}} (x_i, 0)$.
Let $m_y = \max_{\substack{i \in \mathcal{L} \\ F(e_i) < \sigma_f}} (e_i, 0)$.
Let $m = \max(m_x, m_y)$ and b, $1 \leq b \leq L + 1$, such that

$$e_{b-1} < m \leq e_b \tag{85}$$

where $e_0 = -\infty$ and $e_{L+1} = \infty$. Then b exists and is unique.

2. If $b = L + 1$ then the flow f cannot be scheduled by the class-based EDF scheduler. Otherwise f can be scheduled and $\overline{d}_f = e_b$ is the minimum delay that can be guaranteed to flow f by the class-based EDF scheduler.

MINIMUM_DELAY_CLASS(input: $(W_i)_{i\in\mathcal{L}}, B, (e_i)_{i\in\mathcal{L}}, (\sigma_f, \rho_f)$; output: $\overline{d}_f$)

if $B \le \rho_f$
  then exit "cannot accept flow f"
$m_x \leftarrow 0; m_y \leftarrow 0$
for $i = 1$ **to** L **do**
    if $W_i \ge \sigma_f$
      then $x_i \leftarrow e_i - \frac{W_i - \sigma_f}{\rho_f}$
          $m_x \leftarrow \max(m_x, x_i)$
      else $m_y \leftarrow \max(m_y, e_i)$
$m \leftarrow \max(m_x, m_y)$
$b \leftarrow 1$
while $(b \le L)$ **and** $(m > e_b)$ **do**
      $b \leftarrow b + 1$
if $b = L + 1$
  then exit "cannot accept flow f"
  else $\overline{d}_f \leftarrow e_b$
       return $\overline{d}_f$

Figure 30: An $O(L)$ algorithm for computing the minimum delay for (σ, ρ) flow at class-based EDF scheduler

JOIN_UPDATE(input: $(W_i)_{i\in\mathcal{L}}, B, (e_i)_{i\in\mathcal{L}}, (\sigma_g, \rho_g, e_p)$; output: $(W_i)_{i\in\mathcal{L}}, B$)

$B \leftarrow B - \rho_g$
/* update of $(W_i)_i$ */
for $i \leftarrow 1$ **to** L **do**
    if $(e_p \le e_i)$
      then $W_i \leftarrow W_i - \rho_g(e_i - e_p) - \sigma_g$

Figure 31: An algorithm for updating parameters after a flow join

LEAVE_UPDATE(input: $(W_i)_{i\in\mathcal{L}}, B, (e_i)_{i\in\mathcal{L}}, (\sigma_g, \rho_g, e_p)$; output: $(W_i)_{i\in\mathcal{L}}, B$)

$B \leftarrow B + \rho_g$
/* update of $(W_i)_i$ */
for $i \leftarrow 1$ **to** L **do**
    if $(e_p \leq e_i)$
        then $W_i \leftarrow W_i + \rho_g(e_i - e_p) + C_g a_g$

Figure 32: An algorithm for updating parameters after a flow leave

INIT_STATE(input: $c, (e_i)_{i\in\mathcal{L}}$; output: $(W_i)_{i\in\mathcal{L}}, B$)

$B \leftarrow c$
/* Init of $(W_i)_i$ */
for $i \leftarrow 1$ **to** L **do**
    $W_i \leftarrow ce_i$

Figure 33: State initialization at an empty class-based EDF scheduler

D An Estimation of Performance Tradeoff for class-based EDF Schedulers

We have seen the computational benefits of discrete admission control in reducing the complexity of admission control. However, there is a tradeoff in that (Section 3.3) we schedule the envelope A°_f rather than A^*_f for flow f, which is an over-allocation of resources to flow f. Hence a discrete admission control may not be able to admit as many flows as an exact admission control algorithm. In the following we estimate this penalty by comparing the maximum number of flows accepted by the exact and discrete admission controls, under a set of assumptions.

Let us consider two EDF schedulers one having an exact and the other a discrete admission control, and both having throughput c. All flows to be scheduled at both servers, have the same characterization (σ, ρ) ($C = \infty$, i.e. $a = 0$) and the delay requirements fall in the interval $[d_m, d_M)$. Let N_g be the maximum number of flows that can be admitted by the exact algorithm, where the deadlines associated with the flows are equally spaced within $[d_m, d_M)$, $d_k = d_m + \frac{d_M - d_m}{N_g} k, 0 \leq k \leq N_g - 1$. Let N_c be the maximum number of flows that can be admitted by a discrete admission control with L equally spaced discretization points ($e_k = d_m + \frac{d_M - d_m}{L} k, 0 \leq k \leq L - 1$), the flows having delay requirements equally spaced in $[d_m, d_M)$ $d_k = d_m + \frac{d_M - d_m}{N_c} k$, $0 \leq k \leq N_c - 1$. Since both flow delays $(d_i)_i$ and discretization points $(e_i)_i$ are equally spaced, the number N_i of flows that are admitted in the same point e_i (i.e. such that $e_i \leq d_k < e_{i+1}$) is $N_i = N_u$ or $N_i = N_u + 1$ where $N_u = \lfloor N_c / L \rfloor$. Defining $N_d = N_u L$, we have $N_d \leq N_c < N_d + L$. In the following we derive an estimate on the upper bound $\frac{N_g - N_d}{N_d}$ of the relative penalty in performance $\frac{N_g - N_c}{N_c}$ for the discrete admission control.

Using Theorem 2, the exact admission control having reserved N_g flows can accept more flows with delays less than d_M if and only if $F(d_M) > \sigma$. Thus, if N_g is the maximum number of flows we have $F(d_M) = \sigma$, or:

$$cd_M - \sum_{k=0}^{N_g - 1} (\sigma + \rho(d_M - d_m)) = \sigma$$

which yields:

$$N_g + 1 \approx \frac{cd_M}{\sigma + \frac{\rho}{2}(d_M - d_m)}$$

where the approximation comes from the fact that N_g is an integer. Similarly, using Theorem 3, we have that the discrete admission control can accept more flows if and only if $x_{L-1} < e_{L-1}$. Thus, for the maximum number of flows $N_d = N_u L$ we have $x_{L-1} = e_{L-1}$, which yields:

$$N_u \approx \frac{-\sigma + cd_M - \frac{1}{L}\rho(d_M - d_m)}{L\sigma + \frac{L+1}{2}\rho(d_M - d_m)}$$

Since we assumed $N_d = N_u L$ we have:

$$N_d + 1 = N_u L + 1 \approx \frac{Lcd_M + \frac{L-1}{2}\rho(d_M - d_m)}{L\sigma + \frac{L+1}{2}\rho(d_M - d_m)} \approx \frac{Lcd_M}{L\sigma + \frac{L+1}{2}\rho(d_M - d_m)}$$

where the last approximation holds because we assume that all N_d flows are accepted, thus $\sum_{i=0}^{N_d - 1} \rho < c$ and assuming $N_d > 10$ we have $\rho \ll c$ and thus $\frac{L-1}{2}\rho(d_M - d_m) \ll Lcd_M$.

We can now compare N_g and N_d:

$$\frac{N_g - N_d}{N_d + 1} \approx \frac{cd_M}{\sigma + \frac{\rho}{2}(d_M - d_m)} \frac{L\sigma + \frac{L+1}{2}\rho(d_M - d_m)}{Lcd_M} - 1 = \frac{1}{L(1 + \frac{2\sigma}{\rho(d_M - d_m)})} < \frac{1}{L}$$

or

$$N_g - N_d \approx \frac{N_d}{L}$$

We conclude that having L discretization points reduces the average number of accepted flows with an estimated factor of $1/L$. For example, a discrete admission control with 10 points accepts on average around 10% less flows than the exact admission control with the same throughput. We consider that this penalty is acceptable. In Section 4 we evaluate the link performance penalty by simulation which confirms this estimation.

References

[1] ATM Forum. ATM User-Network Interface Specification. Version 3.1, September 1994.

[2] R. Braden, L. Zhang, D. Estrin, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. Internet draft-ietf-rsvp-spec-12.ps, May 1996.

[3] P. T. Brady. A statistical analysis of on-off patterns in 16 conversations. *Bell System Technical Journal*, 47:73–91, January 1968.

[4] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queuing Algorithm. *Computer Communication Review (ACM SIGCOMM'89)*, 19(4), 1989.

[5] V. Firoiu and D. Towsley. Call Admission and Resource Reservation for Multicast Sessions. In *IEEE INFOCOM*, 1996.

[6] L. Georgiadis, R. Guérin, and A. Parekh. Optimal multiplexing on a single link: Delay and buffer requirements. In *IEEE INFOCOM*, 1994.

[7] L. Georgiadis, R. Guérin, V. Peris, and K. Sivarajan. Efficient Network QoS Provisioning Based on per Node Traffic Shaping. In *IEEE INFOCOM*, 1996.

[8] D. Kandlur, K. Shin, and D. Ferrari. Real-time Communications in Multi-hop Networks. In *Proceedings of the 11-th Int'l Conference on Distributed Computing Systems*, 1991.

[9] E. Knightly, D. Wrege, J. Liebeherr, and H. Zhang. Fundamental Limits and Tradeoffs of Providing Deterministic Guarantees to VBR Video Traffic. In *ACM SIGMETRICS'95*, 1995.

[10] J. Liebeherr and D. Wrege. Design and Analysis of a High-Performance Packet Multiplexer for Multiservice Networks with Delay Guarantees. Technical Report CS-94-30, University of Virginia, 1994.

[11] J. Liebeherr, D. Wrege, and D. Ferrari. Exact Admission Control for Networks with Bounded Delay Services. Technical Report CS-94-29, University of Virginia, 1994.

[12] C. Liu and J. Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *Journal of ACM*, 20(1), January 1973.

[13] A. K. Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, February 1992.

[14] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. Internet draft-ietf-intserv-guaranteed-svc-06.txt, August 1996.

[15] H. Zhang and D. Ferrari. Rate-Controlled Service Disciplines. *Journal of High Speed Networks*, 3(4), 1994.

[16] Q. Zheng and K. Shin. On the Ability of Establishing Real-Time Channels in Point-to-Point Packet-Switched Networks. *IEEE Transactions on Communications*, 42(2/3/4), 1994.