Efficient Admission Control of Piecewise Linear Traffic Envelopes at EDF Schedulers

Victor Firoiu, *Member, IEEE,* Jim Kurose, *Fellow, IEEE,* and Don Towsley, *Fellow, IEEE*

***Abstract*—In this paper, we present algorithms for flow admission control at an earliest deadline first link scheduler when the flows are characterized by piecewise linear traffic envelopes. We show that the algorithms have very low computational complexity and, thus, practical applicability. The complexity can be further decreased by introducing the notion of *discretized* admission control. Through discretization, the range of positions for the end points of linear segments of the traffic envelopes is restricted to a finite set. Simulation experiments show that discretized admission control can lend to two orders of magnitude decrease in the amount of computation needed to make admission control decisions over that incurred when using exact (nondiscrete) admission control, with the additional benefit that this amount of computation no longer depends on the number of flows. We examine the relative performance degradation (in terms of the number of flows admitted) incurred by the discretization and find that it is small.**

***Index Terms*—Admission control algorithms, earliest deadline first, piecewise linear traffic envelope, quality of service.**

I. Introduction

THE demand for real-time communication services in data networks such as the Internet has grown rapidly in recent years. Two important applications requiring the timely delivery of data packets are voice and video. To be able to guarantee the delay requirements of these applications, the network has to reserve resources at the links on the paths of the corresponding real-time flows. 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 ReSerVation Protocol (RSVP) [2] for Internet Protocol (IP) networks, and Asynchronous Transfer Mode (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 packet waiting times in the link's queue and, ultimately, the delays that the link scheduler can guarantee. A variant of Weighted Fair Queuing (WFQ) [6] (also known as Generalized Processor Sharing (GPS) [21]) was proposed in [24] 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 suboptimal. Another discipline, earliest deadline first (EDF) [18], associates a per-hop deadline with each packet and schedules packets in the order of their assigned deadlines. EDF has been proven to be optimal 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 [26] was shown to outperform GPS in providing end-to-end delay guarantees in a network [11]. In this paper, we use the rate-controlled EDF framework, where the end-to-end, delay-based admission control is reduced to performing EDF schedulability verifications at each link.

Sufficient conditions for the EDF schedulability of flows have been proposed for some particular cases of flow characterizations [13], [27]. Recently, a set of necessary and sufficient conditions for flow schedulability has been put forward in [17], [10], and [23], using a general characterization of flows. The optimality of EDF and the existence of necessary and sufficient conditions for schedulability make 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, an implementation of EDF scheduling requires a search of $O(\log Q)$ time in the list of packets (ordered by their deadlines) waiting in the queue of length Q for transmission. This issue has been addressed in [25] and [17], where the search time is reduced to constant $(O(1))$ time in an approximate implementation where the range of packet deadline values is discretized. The second issue is that, although the EDF schedulability conditions in [17] can be expressed simply, the algorithms to perform these schedulability tests can be computationally very complex.

In this paper, we address the second issue and present simple and computationally efficient algorithms for performing flow admission at links that use an EDF scheduler. That is, rather than considering various flow characteristics and associated admission control procedures [15], [7], [12], we take a specific traffic characterization (envelopes) and scheduling discipline (EDF) and examine the computational aspects of performing admission control in this setting.

Our proposed flow admission control algorithms are compatible with the emerging standards for flow setup protocols specified by the Internet Integrated Services [24] and the ATM signaling [1]. In the case that flows are described by

Manuscript received December 9, 1997; revised June 2, 1998; approved by IEEE/ACM Transactions on Networking Editor S. Keshav. This work was supported by the National Science Foundation under Grant NCR-95-08274, Grant NCR-96-23807, and Grant CDA-95-02639.

V. Firoiu was with the Department of Computer Science, University of Massachusetts, Amherst, MA 01003 USA. He is now with Bay Networks, Inc., Billerica, MA 01821 USA (e-mail: Victor_Firoiu@BayNetworks.com).

J. Kurose and D. Towsley are with the Department of Computer Science, University of Massachusetts, Amherst, MA 01003 USA (e-mail: jkurose@cs.umass.edu; dtowsley@cs.umass.edu).

Publisher Item Identifier S 1063-6692(98)08374-5.

envelopes characterized by piecewise linear functions with K segments, we find that our algorithms have low complexity $O(KN\log\ (KN))$ 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 restricting the range of positions for the flexion points (i.e., the points attaching two linear segments) of traffic envelopes to a finite set of values. Simulation demonstrates that we obtain a very significant improvement in the run time (two orders of magnitude speedup for the examples we consider), with the additional benefit that the execution time is independent of the number of flows N. We examine the relative performance degradation (in terms of the number of flows admitted) incurred by the discretization and find it to be small.

In this paper, we characterize flows by multiple-segment envelopes that bound the amount of generated traffic over an interval of length t. The motivation for this general characterization is that recent studies (for example, [14]) have shown that, in order to achieve high link utilization, flows must be characterized by piecewise linear functions consisting of more than two segments. Reference [14] has shown that moderately bursty traffic (e.g., some MPEG-encoded movies) achieve high link utilization (about 90%) when using envelopes with 3–4 segments. Moreover, highly bursty traffic (e.g., advertisements) need 10–15 segments in order to achieve the same result, while two-segment envelopes achieve only one-third of the maximum link utilization. Multiple-segment envelopes are easy to specify: each segment is characterized by a (rate, burst size) pair. Multiple-segment envelopes are also easy to regulate and police (see e.g., [4]). A multiple-segment regulator is constructed with a set of leaky bucket regulators in series. It has also been shown in [26] that such a tandem of regulators does not introduce any additional worst case delay for the regulated flow.

The remainder of this paper is organized as follows. In Section II, we describe the requirements imposed by IP and ATM flow setup protocols on the local (link) admission control. In Section III, we derive simple admission control algorithms for flows characterized by multiple-segment envelopes. In Section IV, we propose admission control procedures for nonpreemptive EDF schedulers with nonnegligible packet sizes. In Section V, we evaluate by simulation the performance of exact and discrete admission control algorithms. Section VI concludes the paper.

II. Flow Admission Control in Networks: EDF Schedulers

Flow setup protocols for flows with maximum end-to-end packet delay requirements, such as ATM signaling and RSVP with Guaranteed Services, impose certain requirements on flow link-level admission control algorithms. In this section, we examine these requirements; in Section III, we present specific admission control algorithms that meet 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, which includes 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_f. At each link l along the path from source to destination, the minimum delay that link l can guarantee to $f, \overline{d}_l$, is computed and added to d_c, the cumulative delay, included in the SETUP message. If, at some node, the cumulative delay exceeds the maximum allowable delay, the flow is not accepted and a RELEASE message is returned. Otherwise, at the end of the first pass (at the destination node), $d_f \geq d_c$ and the flow is accepted. A CONNECT message is returned on the same path to the source, assigning a delay $d_{f,l} \geq \overline{d}_l$ to flow f at link l on path P, such that $\Sigma_{l\in P}\ d_{f,l} \leq d_f$ according to some delay division policy (we have explored this in detail in [9]).

Consider next the RSVP protocol [2] in conjunction with the Integrated Services "Guaranteed QoS" specification [24]. The source of a real-time flow sends periodic $Path$ messages to a unicast or multicast IP address. The source includes the flow's traffic characteristics in the $Path$ message. Each link l on the path to the receiver computes the minimum delay it can guarantee to f and adds it to d_c, the cumulative delay, which is sent in the D_{tot} term of the $Path$ message. A receiver that requires an end-to-end delay guarantee d_f and receives a $Path$ message, compares d_f with the minimum end-to-end delay that can be guaranteed by the network, d_c. If $d_f < d_c$, then the receiver decides that its delay requirement cannot be guaranteed. If $d_f \geq d_c$, then the requirement can be satisfied, and the receiver sends a $Resv$ message back to the sender over the same route that $Path$ traversed. This includes d_f, its delay requirement, as part of the delay slack term S. On its return to the source, $Resv$ assigns a delay $d_{f,l} \geq \overline{d}_l$ to flow f at link l, such that $\Sigma_{l\in P}\ d_{f,l} \leq d_f$, according to some QoS division policy (studied in [9]).

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

- given a flow f and its traffic characterization (e.g., maximum and average bandwidth requirement and maximum burst size or, more generally, any traffic envelope), provide the minimum delay that link l can guarantee to $f, \overline{d}_l$, based on the current state (set of reserved flows) at the local scheduler;
- given a flow f, its traffic characterization (as above), a requirement $d_{f,l} \geq \overline{d}_l$, and a set of currently accepted flows, update the current "state" of the local scheduler to reflect the fact that a maximum packet delay $d_{f,l}$ is additionally being guaranteed to flow f.

In the following, we examine how these capabilities can be provided in the case where EDF scheduling is used to provide maximum end-to-end packet delay guarantees. Consider a flow f with the amount of arrivals (measured in bits per second) in the time interval $[\tau_1, \tau_2]$ denoted by $A_f[\tau_1, \tau_2]$. The flow is characterized by a traffic constraint function, or minimum envelope A_f^*, an upper bound on the flow's arrival pattern [5], [3]

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

which also satisfies the relation $A_f^*(t) \leq \tilde{A}_f^*(t), \forall t \geq 0$ for any envelope $\tilde{A}_f^*$ having the above property. It is easy to see that A_f^* is nondecreasing. We take $A_f^*(t) = 0, \forall t < 0$. In this paper, we measure the traffic as a number of data units (bits) rather than transmission time (seconds), as in [17].

Let us consider a set of N flows, where flow i is characterized by its envelope A_i^*. The stability condition for a work-conserving scheduler (including the EDF scheduler) is [17, eq. (5)]

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

where c is the constant rate of the server (bits per second).

We consider now, and in Section III, preemptive EDF schedulers and, in Section IV, nonpreemptive EDF schedulers. In the case of a *preemptive* EDF scheduler, we state the following variant of the schedulability condition proposed in [17] for a set of N flows.

Theorem 1: (Liebeherr, Wrege, Ferrari): Consider a set of N flows, that satisfy (1), where flow i is characterized by its envelope A_i^* and has a maximum packet delay requirement at a given link of d_i. The set of flows is EDF schedulable at that link if and only if

$$ct \geq \sum_{1\leq i\leq N} A_i^*(t - d_i), \quad \forall t \geq 0. \tag{2}$$

We say that the set $\{A_i^*, d_i\}_{1\leq i\leq N}$ is schedulable if (1) and (2) are satisfied.

Note that (2) provides only a schedulability condition; it does not provide an algorithm to test this condition. In this paper, we present efficient algorithms for testing this condition. The following two properties of (1) and (2) ensure that EDF schedulers are capable of supporting the flow setup protocols described earlier.

Proposition 1: If a set of flows $(A_i^*, d_i)_{1\leq i\leq N}$ is schedulable, then it remains schedulable if the maximum tolerable delay requirement for any flow is increased from d_k to $d_k+\delta, \delta > 0$, for any $1 \leq k \leq N$.

The intuition behind this result is that, by relaxing the delay requirement for a flow in a schedulable set, the set remains schedulable. The proof is simple and we omit it.

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

The delay $\overline{d}$ 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 of N flows. The existence and uniqueness of the minimum delay $\overline{d}$ ensures that EDF schedulers are capable of supporting the flow setup protocols described earlier.

III. Admission Control Algorithms for Flows Characterized by Multiple-Segment Envelopes

In this section, we address the problem of computing the admissibility of flows characterized by multiple-segment envelopes at an EDF scheduler, when each flow has a maximum packet delay requirement. We begin by introducing the definitions and notations related to multiple-segment envelopes.

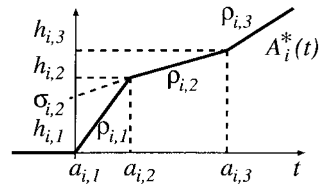


Fig. 1. A multiple-segment envelope.

Definition 1: The multiple-segment envelope A_i^* of flow i is a function $A_i^*: \mathbb{R} \to \mathbb{R}$ with the following properties.

1) A_i^* is a piecewise linear function with a finite number of segments n_i

$$A_i^*(t) = \sigma_{i,j} + \rho_{i,j}t \qquad a_{i,j} \leq t < a_{i,j+1}, \quad 0 \leq j \leq n_i \tag{3}$$

where $\sigma_{i,0} = 0, \sigma_{i,1} = 0, \rho_{i,0} = 0$ and

$$-\infty = a_{i,0} < 0 = a_{i,1} < a_{i,2} < \cdots < a_{i,n_i} < a_{i,n_i+1} = \infty.$$

2) A_i^* is a continuous function

$$\sigma_{i,j-1} + a_{i,j}\rho_{i,j-1} = \sigma_{i,j} + a_{i,j}\rho_{i,j}, \qquad 2 \leq j \leq n_i.$$

3) A_i^* is strictly increasing in $[0, \infty)$

$$\rho_{i,j} > 0, \qquad 1 \leq j \leq n_i.$$

4) Let $h_{i,j} = A_i^*(a_{i,j})$. Because A_i^* is strictly increasing, we have

$$0 = h_{i,0} = h_{i,1} < h_{i,2} < \cdots < h_{i,n_i} < h_{i,n_i+1} = \infty. \tag{4}$$

Observe that the above definition of A_i^* covers not only concave envelope functions (for example, multiple leaky buckets [14]), but also more general envelope functions (for example, D-BIND [16]). Fig. 1 shows an example of a multiple-segment envelope.

In the analysis that follows, we will use the notions of concave, convex, and flexion points that we define below.

Definition 2: Given a continuous function $f: \mathbb{R} \to \mathbb{R}$ and $a \in \mathbb{R}, (a, f(a))$ is said to be a concave (convex) point of f if f is concave (convex) in a vicinity of a

$$\begin{aligned} &\exists\epsilon > 0 \quad \forall\alpha \in (0,1) \\ &f((1-\alpha)(a-\epsilon) + \alpha(a+\epsilon)) \\ &\qquad > (1-\alpha)f(a-\epsilon) + \alpha f(a+\epsilon) \end{aligned} \tag{5}$$

("<," respectively). A flexion point is a concave or a convex point.

Observe that a is an element of the domain of f and, thus, a is the abscissa coordinate of the point $(a, f(a))$ on the graph of f. For example, in Fig. 1, $a_{i,2}$ is a concave point and $a_{i,3}$ is a convex point of A_i^*.

Finally, we introduce the following notation.

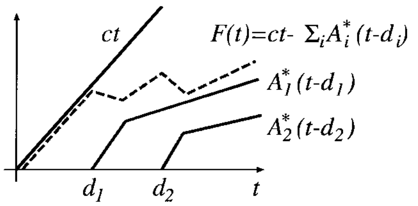


Fig. 2. A work availability function F.

Notation 1: Given a piecewise linear, continuous function f:

1) X_f^{cv} is the set of concave points of f and $Y_f^{cv} = f(X_f^{cv})$.
2) X_f^{cx} is the set of convex points of f and $Y_f^{cx} = f(X_f^{cx})$.
3) $X_f = X_f^{cv} \cup X_f^{cx}$ is the set of flexion points of f and $Y_f = f(X_f)$.

Notation 2: Given a function f: $\mathbb{R} \rightarrow \mathbb{R}$ and $S \subset \mathbb{R}$:

1) The inverse of the function f, f^{-1} is defined as

$$f^{-1}(y) \triangleq \{x | f(x) = y\}.$$

If the resulting set has one element, we write $f^{-1}(y) = x$. The inverse of the function f in $y \in \mathbb{R}$, restricted to the interval (a, b) is

$$f^{-1}_{(a,b)}(y) \triangleq \{x | x \in (a,b), f(x) = y\}. \tag{6}$$

2) The difference between a set $S \subset \mathbb{R}$ and $a \in \mathbb{R}$ is the set of differences

$$S - a \triangleq \{x - a | x \in S\}.$$

3) For any set $S, \hat{S} = S \cup \{-\infty, \infty\}$.

A. Exact Admission Control Algorithms for Multiple-Segment Envelopes

Let us consider N flows, where flow i is characterized by the multiple-segment envelope A_i^* and has a maximum packet delay requirement d_i. In order to compute the schedulability conditions (2), we introduce the following definition.

Definition 3: The $\underline{(work) availability function}$ F:$\mathbb{R} \rightarrow \mathbb{R}$ is defined by

$$F(t) = ct - \sum_{i=1}^{N} A_i^*(t - d_i). \tag{7}$$

Given this definition, the schedulability condition (2) for the set of N flows becomes $F(t) \geq 0 \; \forall t \geq 0$. $F(t)$ gives the maximum amount of work (in bits) available over an interval of length t in the worst case at the EDF scheduler, while guaranteeing for each flow i (with envelope A_i^*) its maximum packet delay of d_i, for $1 \leq i \leq N$. This function plays a central role in the development of admission control algorithms in the rest of this paper. Given that A_i^* is a piecewise linear and continuous function, $i = 1, \cdots, N$, it follows that F is also piecewise linear and continuous (Fig. 2 gives an example of F.)

We now consider the problem of admitting a new flow f, with envelope A_f^*, given that N flows (A_i^*, d_i) are scheduled at the EDF scheduler, and that the stability condition (1) is satisfied

$$\sum_{i=1}^{N} \rho_{i,n_i} + \rho_{f,n_f} < c.$$

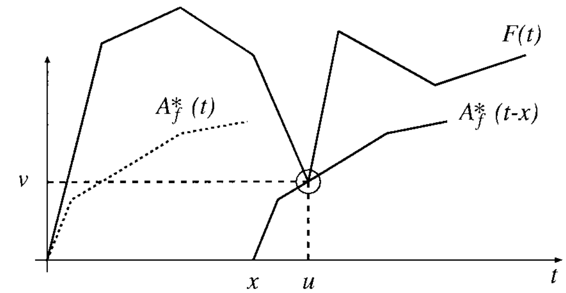


Fig. 3. Constraint by flexion point of F.

In the following, we adopt the convention that $\max \emptyset = 0$.

Theorem 2: The minimum delay $\overline{d}$ guaranteeable to flow f is

$$\overline{d} = \max(m_x, m_y)$$

where

$$m_x = \max_{u \in X_F} (u - A_f^{*\,-1}(F(u))) \tag{8}$$

$$m_y = \max \cup_{a \in X_{A_f^*}} (F^{-1}(A_f^*(a)) - a). \tag{9}$$

Observe that, in (9), $F^{-1}(A_f^*(a))$ is a set that may contain more than one element since F is not necessarily a bijective function. Thus, m_y is the maximum element in a union of sets.

We provide an intuitive explanation of the theorem; see Appendix A for its formal proof. First, note that, given an availability function F, the schedulability condition $F(t) \geq A_f^*(t - d)$ is equivalent to being able to fit the A_f^* curve "below" $F(t)$. Fig. 3 shows such an example, where the original envelope curve, $A_f^*(t)$, is translated x units to the right so that it just fits under $F(t)$. In this example, the minimum delay guaranteeable would be x. We note that this problem of fitting a multisegment curve under another differs from polygon containment problems in computational geometry [22].

Our goal is to find the smallest value of d such that the original envelope curve, $A_f^*(t)$, translated d time units to the right, will lie completely below $F(t)$. Imagine for the moment (the algorithm we will present does not actually do this), taking the $A_f^*(t)$ curve and, starting from the right, translating the curve to the left (toward the origin) until it *first* intersects $F(t)$. $A_f^*(t)$ will either intersect $F(t)$ at a flexion point of $F(t)$ (as shown in Fig. 3, or a line segment of $F(t)$ (as shown in Fig. 4).

The equality for m_x in (8) considers the cases where flexion points of $F(t)$ just fit a translated $A_f^*(t)$. When $F(t)$ and a translated $A_f^*(t)$ are incident at such a flexion point, say at $t = u$ (as shown in Fig. 3), then the height of the translated A_f^* at u is $F(u)$. The *untranslated* A_f^* has this height at $A_f^{*\,-1}(F(u))$ and, hence, the amount of translation is $u - A_f^{*\,-1}(F(u))$. One can show (see Appendix A) that if

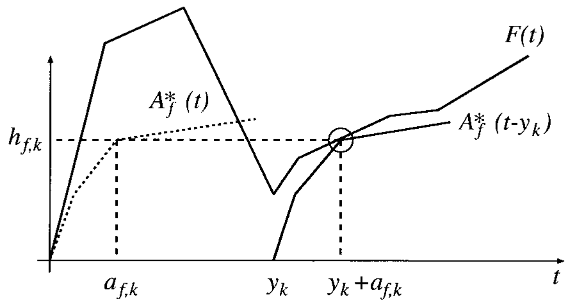


Fig. 4. Constraint by flexion point of A_f^*.

```
MINIMUM_DELAY (Input: X_{A*_f}, Y_{A*_f}, ρ_{f,n_f}, X_F, Y_F, B;
               Output: d̄)
1  for each v_l ∈ Y_F
2      and each h_{f,k} ∈ Y_{A*_f}
3      if v_l ∈ [h_{f,k}, h_{f,k+1})
4          then x_l ← u_l − A*_f^{-1}_{[h_{f,k}, h_{f,k+1})}(v_l)
5  m_x ← max x_l
6  for each h_{f,k} ∈ Y_{A*_f}
7      and each v_l ∈ Y_F
8      if h_{f,k} ∈ [v_l, v_{l+1})
9          then y_{l,k} ← F^{-1}(h_{f,k}) − a_{f,k}
10 m_y ← max y_{l,k}
11 d̄ ← max(m_x, m_y)
```

Fig. 5. An $O(K^2N)$ algorithm for computing the minimum delay for a multisegment envelope.

a translated A_f^* and $F(t)$ are incident at two flexion points of $F(t)$, with associated translations d_1 and d_2, where $d_1 < d_2$, then the A_f^* that is translated d_1 lies above $F(t)$ for some values of t. Hence, in order to find the smallest delay that guarantees schedulability, we take a max in (8).

Similarly, suppose that $F(t)$ and a translated $A_f^*(t)$ intersect at a flexion point of $A_f^*(t)$ (see Fig. 4). In this case, let a be such that the height of $F(t)$ at the incidence point is equal to the height of the untranslated $A_f^*(a)$. The amount of translation and, hence, the guaranteeable delay would be $F^{-1}(A_f^*(a))-a$.

In Fig. 5, we give an algorithm to compute the minimum guaranteeable delay $\overline{d}$, which is a direct implementation of *Theorem 2*. Here, we use the notation $h_{f,k} = A_f^*(a_{f,k}), v_l = F(u_l)$, $Y_{A_f^*} = A_f^*(X_{A_f^*}), Y_F = F(X_F)$ and $B = c - \Sigma_{i=1}^N \ \rho_{i,n_i}$.

To evaluate the computational complexity of this algorithm, we first observe that, for any piecewise linear and continuous function g, if a_1 and a_2 are consecutive points in X_g, and $v \in [g(a_1), g(a_2))$, then computing $g^{-1}(v)$ requires $O(1)$ time, since g^{-1} is linear in $[g(a_1), g(a_2))$. (More precisely, $g^{-1}(v) = a_1 + (v - g(a_1))/g'(a_1^+)$, where $g'(a_1^+) = (g(a_2) - g(a_1))/(a_2 - a_1)$ for $a_2 < \infty$, and for $a_2 = \infty, g'(a_1^+) = \lim_{x\to\infty} \ g'(x)$, which is ρ_{f,n_f} for $g = A_f^*$ and B for $g = F$.) We assume that, at the time of the algorithm's invocation, there are N flows, and that any flow envelope has $|X_{A_f^*}| = |Y_{A_f^*}| = O(K)$ flexion points. It follows that $|X_F| = |Y_F| = O(KN)$. To compute m_x (steps 1–5), a lookup in $Y_{A_f^*}$ is done for each element in Y_F. Thus, the complexity of computing m_x is $O(|Y_F||Y_{A_f^*}|) = O(K^2N)$. To compute m_y (steps 6–10), a lookup Y_F is done for each element in

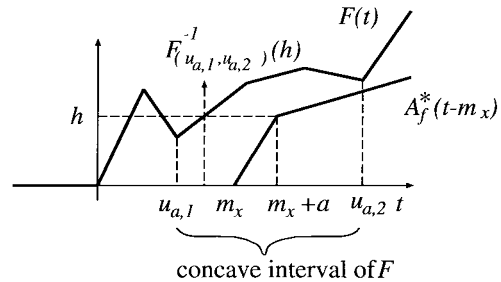


Fig. 6. Reducing the number of F^{-1} computations.

$Y_{A_f^*}$, giving a complexity of $O(K^2N)$. The complexity of the entire algorithm is, thus, $O(K^2N)$.

We can speed up the computation using two independent methods which can be combined. We begin by observing that not all points in $X_{A_f^*}$ and X_F are relevant for computing the minimum delay. It is easy to show that only the concave points of F and convex points of A_f^* impose constraints on the position of A_f^* (see Figs. 3 and 4). Thus,

$$\overline{d} = \max(m_x, m_y) \tag{10}$$

where

$$m_x = \max_{u \in X_F^{cv}} (u - A_f^{*-1}(F(u))) \tag{11}$$

$$m_y = \max \bigcup_{a \in X_{A_f^*}^{cx}} (F^{-1}(A_f^*(a)) - a). \tag{12}$$

A second method that reduces the worst case complexity of computing the minimum guaranteeable delay, reduces the number of points for which F^{-1} is computed. We can accomplish this if, in (12), for each $a \in X_{A_f^*}^{cx}$, we compute only $F^{-1}(A_f^*(a)) - a$ within the concave interval of F where $m_x + a$ is situated (see Fig. 6). In Appendix B, we show that this computation is sufficient for computing $\overline{d}$.

For a formal statement, given m_x defined in (11), we introduce $u_{a,1}, u_{a,2} \in \widehat{X^{cv}}$ (where $\widehat{X^{cv}} = X^{cv} \cup \{-\infty, \infty\}$) such that

$$u_{a,1} = \max\{x \in \widehat{X^{cv}} | x \le m_x + a\} \tag{13}$$

$$u_{a,2} = \min\{x \in \widehat{X^{cv}} | x > m_x + a\}. \tag{14}$$

Theorem 3: The minimum delay $\overline{d}$ guaranteeable to flow f is

$$\overline{d} = \max(m_x, m_z) \tag{15}$$

where

$$m_x = \max_{u \in X_F^{cv}} (u - A_f^{*-1}(F(u)))$$

$$m_z = \max_{a \in X_{A_f^*}^{cx}} (F_{(u_{a,1}, u_{a,2})}^{-1} (A_f^*(a)) - a) \tag{16}$$

where $F_{()}^{-1}()$ is defined in (6).

Observe that the set $F_{(u_{a,1}, u_{a,2})}^{-1} (A_f^*(a)) - a)$ has at most one element, since F is concave in $(u_{a,1}, u_{a,2})$.

The proof of the theorem can be found in Appendix B. In Appendix C, we present an algorithm for computing $\overline{d}$ based

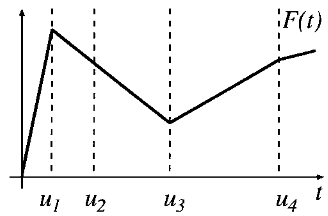


Fig. 7. A discrete work availability function.

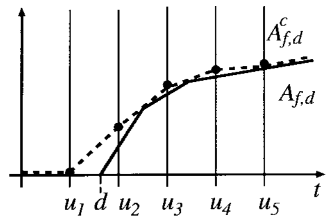


Fig. 8. A discrete cover.

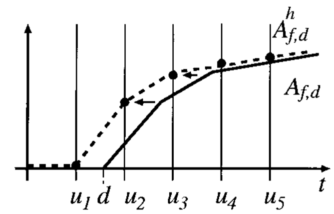


Fig. 9. A horizontal translation cover.

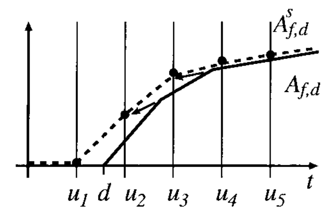


Fig. 10. A slope translation cover.

on the above theorem, having a computational complexity of $O(KN)$. Observe that this complexity is the best (lowest) possible, since all of the segments of F must be evaluated for computing d, and there are $O(KN)$ such segments.

We observe here that, although a complexity of $O(KN)$ for admission control may be acceptable when the number of flows N is small, this amount 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.

B. Discrete Admission Control for Multiple-Segment Envelopes

In this section, we show how the computational complexity of the admission control algorithm can be significantly reduced by ensuring that F consists of at most L linear segments. Let $\mathcal{D} = \{u_i | u_i \in \mathbb{R}^+, 1 \leq i \leq L\}$. We say that F is $\mathcal{D}$ *discrete* if the set of F's flexion points form a subset of $\mathcal{D}, X_F \subset \mathcal{D}$. Fig. 7 shows an example of a $\mathcal{D}$-discrete work availability function F where $\mathcal{D} = \{u_1, u_2, u_3, u_4\}$. Consider a $\mathcal{D}$-discrete work availability function F and a new flow f with envelope A_f^* and maximum packet delay requirement d. Consider the problem that arises when we wish to admit f having envelope A_f^* and maximum packet delay requirement d. We define f's *minimum work requirement* function as

$$A_{f,d}(t) = A_f^*(t-d). \tag{17}$$

It is important that the admission result in a new work availability function F' that is also $\mathcal{D}$ discrete and that satisfies $F'(t) \leq F(t) - A_{f,d}(t)$. We construct F' as $F'(t) = F(t) - A_{f,d}^c(t)$, where $A_{f,d}^c(t) \geq A_{f,d}(t)$ and $A_{f,d}^c(t)$ is $\mathcal{D}$ discrete. Thus, F' has the required properties of being $\mathcal{D}$ discrete and $F'(t) \leq F(t) - A_{f,d}(t)$. $A_{f,d}^c$ is called a $\mathcal{D}$-discrete cover for the minimum work function $A_{f,d}$. An example of a $\mathcal{D}$-discrete cover for a work function $A_{f,d}$ is shown is Fig. 8.

We make two observations. First, it is clear that reserving a cover that is larger than the required work function implies that more resources (work) will be reserved than are needed to accommodate the request. This will lead to lower resource utilization at the link scheduler and potentially fewer flows being admitted at the link, compared to what is possible with the exact admission control. We investigate this tradeoff in Section V. The second observation is that, in general, there are many choices of discrete covers for a given work function. Among these choices, a cover that is "closer" to the original request is preferred because the amount of over-reservation is smaller. Unfortunately, it is not possible to establish a total order among the covers of a given envelope and, thus, there is no one cover that would minimize the over-reservation. We have developed and analyzed several heuristics to choose covers, and we present two of them in the following.

The *horizontal translation* policy (Fig. 9) constructs the cover $A_{f,d}^h$ by translating the concave points of the work function $A_{f,d}^*$ horizontally to the left. The *slope translation* policy (Fig. 10) constructs the cover $A_{f,d}^s$ by translating each concave point of the work function $A_{f,d}^*$ to its left, following the slope of the envelope's segment that begins at that concave point of $A_{f,d}$. It is easy to show that $A_{f,d}^s(t) \leq A_{f,d}^h(t)$, i.e., the slope translation results in a smaller over-reservation than the horizontal translation. Henceforth, we only consider the slope translation policy.

A potential problem with the slope translation policy is that, although the work function is schedulable $(A_{f,d}(t) \leq F(t), \forall t)$, it may be possible that the cover is not schedulable (i.e., $A_{f,d}^s(t) > F(t)$ for some t). Hence, there is a minimum value of $d, d' \geq d$, such that $A_{f,d'}^s(t) \geq F(t)$. In [8], we have considered another approach where we construct a $\mathcal{D}$-discrete cover $A_{f,d}^c$ that is schedulable $(A_{f,d}^c(t) \leq F(t), \forall t)$ whenever $A_{f,d}$ is schedulable $(A_{f,d}(t) \leq F(t), \forall t)$. We have found that this procedure does not perform significantly better than slope translation. Because it is a more complicated procedure and due to lack of space, we do not present it here; the interested reader can find its description in [8]. In the remainder of this section, we will focus on the slope translation policy and present an algorithm for its implementation.

We begin by deriving a method for computing the minimum delay $\overline{d}^s$ guaranteeable by the slope translation policy, given

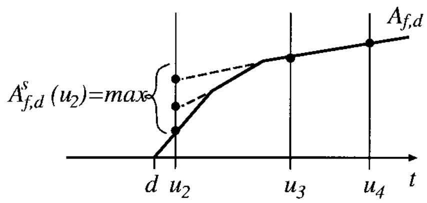


Fig. 11. Construction of slope translated cover.

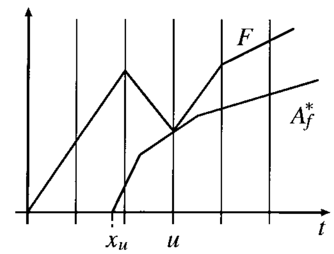


Fig. 12. F constrains points "on" original envelope.

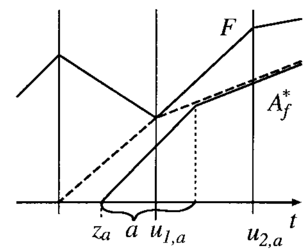


Fig. 13. F constrains slope translating points.

that the stability condition (1) is satisfied. We introduce the following notation:

$$A^*_{f,a_{f,j}}(t) = \sigma_{f,j} + \rho_{f,j}t, \quad t \in \mathbb{R}, 1 \le j \le n_f \tag{18}$$

thus,

$$A^*_f(t) = A^*_{f,a_{f,j}}(t), \quad \text{for} \quad t \in [a_{f,j}, a_{f,j+1}). \tag{19}$$

Given an envelope A^*_f and a maximum delay request d, the slope translation cover envelope $A^s_{f,d}$ is a piecewise linear, continuous function defined by its flexion points

$$A^s_{f,d}(u) = \max(A_{f,d}(u), A^+_{f,d}(u)), \quad u \in \mathcal{D} \tag{20}$$

where

$$A^+_{f,d}(u_i) = \max\{0\} \cup \{A^*_{f,a}(u_i - d) | a \in X^{cx}_{A^*_f} \\ u_i \le a + d < u_{i+1}\}, \quad u_i \in \mathcal{D}. \tag{21}$$

In Fig. 11, we illustrate the construction of such a cover. Observe that the second set in (21) is empty when there is no $a \in X^{cx}_{A^*_f}$ such that $u_i \le a + d < u_{i+1}$; in this case, $A^+_{f,d}(u_i) = 0$ and, thus, $A^s_{f,d}(u_i) = A^*_f(u_i - d)$. This corresponds to $A_{f,d}$ having no flexion points in the interval $[u_i, u_{i+1})$. This case is exemplified in Fig. 11 for u_3.

Theorem 4: The minimum delay $\overline{d}^s$ guaranteeable to flow f using the discrete admission control algorithm coupled with the slope translation cover policy is

$$\overline{d}^s = \max(m_x, m_z), \tag{22}$$

where

$$m_x = \max_{u \in \mathcal{D}} \{u - {A^*_f}^{-1}(F(u))\} \tag{23}$$

$$m_z = \max_{a \in X^{cx}_{A^*_f}} \{\min(u_{i_a} - {A^*_{f,a}}^{-1}(F(u_{i_a})), \\ u_{i_a+1} - a)\} \tag{24}$$

$$u_{i_a} \le m_x + a < u_{i_a+1} \quad u_{i_a}, u_{i_a+1} \in \hat{\mathcal{D}}. \tag{25}$$

We provide some intuition behind the above result; the formal proof can be found in Appendix D. Given a $\mathcal{D}$-discrete availability function F, the schedulability condition $F(t) \ge A^s_{f,d}(t)$ for all $t \in \mathbb{R}^+$ reduces to $F(u) \ge A^s_{f,d}(u)$ for all $u \in \mathcal{D}$, since the slope translation cover $A^s_{f,d}$ is by construction $\mathcal{D}$ discrete. The minimum delay $\overline{d}^s$ guaranteeable to f is the leftmost position of $A_{f,d}$, while its cover $A^s_{f,d}$ is below F. $A^s_{f,d}$ is defined as the maximum of two components, $A_{f,d}$ and $A^+_{f,d}$, which are both increasing functions. m_x, the leftmost position of $A_{f,d}$, accounts for all points "on" the original envelope A^*_f being constrained by F, as in Fig. 12.

Minimum_Delay_Slope_Tr (Input: $X^{cx}_{A^*_f}, Y_{A^*_f}, \rho_{f,n_f}$, $\mathcal{D}, F(\mathcal{D}), B$;
Output: $\overline{d}^s$)

1 **for** each $v \in F(\mathcal{D})$
2 find $h_{f,i} \in Y_{A^*_f}, h_{f,i} \le v < h_{f,i+1}, h_{f,i} = A^*_f(a_{f,i})$
3 $x_u \leftarrow u - {A^*_{f,a_{f,i}}}^{-1}(v)$
4 $m_x \leftarrow \max_u x_u$
5 **for** each $a \in X^{cx}_{A^*_f}$
6 find $u_j \in \mathcal{D}, u_j \le m_x + a < u_{j+1}$
7 $z_a \leftarrow \min(u_j - {A^*_{f,a}}^{-1}(F(u_j)), u_{j+1} - a)$
8 $m_z \leftarrow \max_a z_a$
9 $\overline{d}^s \leftarrow \max(m_x, m_z)$

Fig. 14. An $O(L+K)$ algorithm for computing the minimum delay for the slope-translating policy.

m_z, the leftmost position of $A^+_{f,d}$, accounts for all slope translated points of A^*_f being constrained by F, as in Fig. 13. $\overline{d}^s$ is the smallest delay (leftmost position) admissible for both components, which is the largest (rightmost) of the two individual positions, $\overline{d}^s = \max(m_x, m_z)$.

In Fig. 14, we give an algorithm to compute the minimum guaranteeable delay $\overline{d}^s$, which is a direct implementation of *Theorem 4*. To evaluate the computational complexity of this algorithm, we first observe that the computation of m_x in lines 1–3 requires a lookup in $Y_{A^*_f}$ and $X_{A^*_f}$, which can be done in tandem if both sets are sorted. Thus, the complexity of computing m_x is $O(K + L)$, since $|Y_F| = |\mathcal{D}| = L$ and $|Y_{A^*_f}| = |X_{A^*_f}| = K$. A similar analysis applies to the computation of m_z in lines 4–7, where the sets $X^{cx}_{A^*_f}$ and $\mathcal{D}$ can be looked up in tandem if sorted, giving a complexity of $O(K_2 + L)$, since $|X^{cx}_{A^*_f}| = K_2$. Thus, the total complexity of computing $\overline{d}_s$ is $O(K + L)$. This is an important improvement over the $O(KN)$ complexity of exact admission control algorithm when the number of flows N is large (thousands) and the number L of discretization points is small (tens). We compare the performace of the two algorithms through simulation in Section V.

IV. Admission Control at a Nonpreemptive EDF Scheduler

So far, we have studied admission control procedures for EDF schedulers assuming that the scheduling discipline is preemptive or that the packet transmission time is negligible. While preemptive schedulers are usually not of practical applicability, the above assumption is valid for scheduling ATM cells, since the cell transmission time (on the order of microseconds) is much smaller than practical delay requirements (on the order of miliseconds or greater).

In this section, we remove this assumption and develop admission control algorithms for nonpreemptive EDF (NPEDF) schedulers. This is important in the case that packet transmission times are nonnegligible, as might be the case for IP packets. We present a simple correction to the results proposed in previous sections for preemptive EDF (PEDF) that accounts for nonzero packet transmission times.

There are two conditions required for a set of envelopes $(A_i^*, d_i)_{1\le i\le N}$ to be schedulable at an NPEDF scheduler with capacity c [17]. The stability condition

$$\lim_{t\to\infty} \sum_{i=1}^{N} A_i^*(t)/ct < 1$$

is identical to the stability condition of the corresponding PEDF scheduler. The schedulability condition includes a term accounting for the packet sizes

$$ct \ge \sum_{i=1}^{N} A_i^*(t-d_i) + \left(\max_{\substack{1\le i\le N\\ d_i > t}} p_i\right)\mathbf{1}(d_1 \le t < d_N) \quad \forall t \ge 0 \qquad (26)$$

where $d_1 \le d_2 \le \cdots \le d_N$ are the delay requirements, p_i is the maximum packet size of flow i, and $\mathbf{1}(P)$ is defined as

$$\mathbf{1}(P) = \begin{cases} 1 & \text{if predicate } P \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

In [8], we show that *Proposition 1* no longer holds in the context of the schedulability condition (26), i.e., it is possible to guarantee a delay d but not d', even if $d < d'$. Consequently, (26) cannot be used for admission control in the context of the flow setup protocols described in Section II. In the following, we propose a sufficient scheduling condition for NPEDF that eliminates the above-mentioned problem.

Let us assume that the NPEDF scheduler has an upper bound on the maximum packet size $p_{\max}$, such that $p_i \le p_{\max}$ for all i. This assumption is consistent with all current IP router architectures. For ATM switches, all flows have the same packet size, $p_i = p_{\max} = 1$ cell. Also, this assumption has been used in other studies (see e.g., the study of packetized generalized processor sharing in [21]).

Our proposed schedulability condition is

$$ct \ge \sum_{i=1}^{N} A_i^*(t-d_i) + p_{\max}, \quad \forall t \ge \min_{1\le i\le N} (d_i). \qquad (27)$$

It is easy to see that (27) ⇒ (26). We further transform (27)

$$c\left(t - \frac{p_{\max}}{c}\right) \ge \sum_{i=1}^{N} A_i^*(t-d_i), \quad \forall t \ge \min_{1\le i\le N} (d_i)$$

$$ct \ge \sum_{i=1}^{N} A_i^*\left(t - d_i - \frac{p_{\max}}{c}\right), \quad \forall t \ge 0.$$

The last inequality is in a form that permits us to extend the admission control algorithms developed for PEDF schedulers to NPEDF schedulers.

In order to compute $\overline{d}_f^{NP}$, the minimum delay for A_f^* to be schedulable at an NPEDF, we first use the algorithm developed in the previous sections to determine a minimum delay $\overline{d}_f$, and then $\overline{d}_f^{NP} = \overline{d}_f + p_{\max}/c$. Conversely, to reserve resources for (A_f^*, d_f^{NP}) at an NPEDF scheduler, we use the algorithms for reserving resources for $(A_f^*, d_f^{NP} - p_{\max}/c)$ as if the EDF scheduler were preemptive.

V. Evaluation of Admission Control Algorithms Through Simulations

We are interested in answering two questions regarding the efficacy of the discrete admission control algorithms developed in Section III-B. First, we are interested in empirically assessing the computational gains obtained by the discrete admission control over the exact algorithm. This is done by comparing the running times of an implementation of the exact and discrete admission control. Second, we wish to determine the performance degradation of the discrete admission control. This is done by comparing the link blocking probability yielded by the implementations of the exact and discrete algorithms.

We consider a link that forwards ATM traffic according to the EDF scheduling policy. The traffic characteristics of the flows to be serviced at this link are chosen randomly from the set of flow characterizations displayed in Table I. Each row represents a four-segment characterization (σ_i, ρ_i) of a movie trace, where ρ_1 is the peak rate and ρ_4 is the mean rate. These characterizations have been derived as four-segment covers of the empirical envelopes of traces of MPEG-1 coded movies from [20] and [19]. To account for possible variations in bandwidths associated with different encodings (MPEG-1, H.261, H.263, RealVideo, and Vxtreme), the (σ, ρ) characterizations from Table I are scaled with a random parameter 10^{θ}, where θ is uniformly distributed in $[-2, 0]$.

Flow arrivals are generated according to a Poisson process with parameter α and their durations are 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 which is uniformly distributed in [50 ms, 3 s]. After a flow is generated with the above parameters, its EDF schedulability is verified by our admission control algorithms. We generate 100 000 flows in each 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

TABLE I
FOUR-SEGMENT CHARACTERIZATION FOR SIX MPEG-CODED MOVIE TRACES

Movie	σ_1	ρ_1	σ_2	ρ_2	σ_3	ρ_3	σ_4	ρ_4
Advertisements	0	1600.0	800.0	800.0	1333.0	600.0	1600.0	533.0
Jurassic	0	4000.0	133.3	1054.0	400.0	853.3	1066.0	761.9
Mtv	0	6000.0	266.6	2356.5	933.3	1973.3	1866.6	1866.6
Silence	0	4000.0	266.6	666.5	533.0	600.0	1133.0	500.0
Soccer	0	5000.0	266.6	2500.0	1000.0	1238.0	2133.3	1066.6
Terminator	0	3400.0	133.3	787.8	266.6	586.6	800.0	366.6

TABLE II
COMPARISON OF COMPUTATION TIMES (IN μs) FOR EXACT AND DISCRETE ADMISSION CONTROL ALGORITHMS

		Exact	Discrete
MINIMUM_DELAY	T3	77.63	13.39
	OC3	283.71	14.99
	OC12	1267.60	14.79
RESERVE	T3	129.85	7.03
	OC3	490.21	6.71
	OC12	2357.54	6.96
RELEASE	T3	84.06	7.30
	OC3	335.63	7.27
	OC12	1727.95	7.71

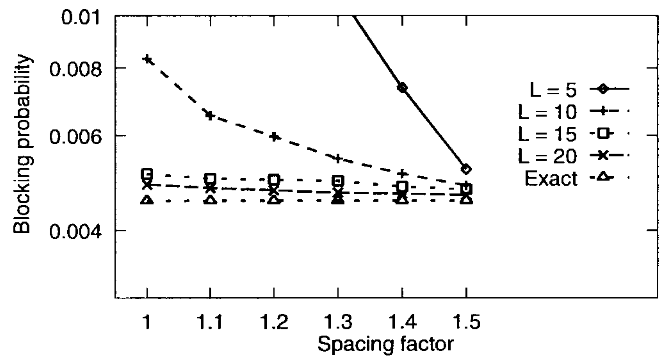


Fig. 15. Impact of spacing of discretization points on admission control performance.

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.

In the first experiment, we compare the computational performance of discrete admission control algorithms (having 15 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 45 Mb/s (a T3 link) and offered load 120 flows; link capacity 155.52 Mb/s (an OC3 link) and offered load 414 flows; link capacity 622.08 Mb/s (an OC12 link) 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. We measure the computation time of the following algorithms: MINIMUM_DELAY (Fig. 17), RESERVE (Fig. 18) and RELEASE for exact admission control, and MINIMUM_DELAY_SLOPE_TR (Fig. 14), RESERVE and RELEASE for discrete admission control using slope translation. The average computation time has been measured with the GNU code profiler $gprof$ on a 266-MHz DECAlpha system.

First, observe that the average run times of the exact algorithms in Table II increase as a linear function of the average number of reserved flows N, which is consistent with the $O(KN)$ complexity found in Section III-A. Second, observe that the run times of the discrete algorithms are independent of the number of reserved flows N, which was predicted by the $O(K+L)$ complexity found in Section III-B. Also, observe that the run times on all "OC12" lines in the table show a gain of about two orders of magnitude in computation time for the discrete admission control. Most important, the discrete algorithms have run times around 10 μs/call, which makes them a practical solution for flow admission control.

For the rest of our simulations, we consider a T3 link (45 Mb/s).

In the following, we evaluate the penalty in link performance when using discrete admission control coupled with the slope translation policy. Recall that the discrete algorithms in Section III-B take their discretization point values from a finite set $\mathcal{D} = \{u_i | 1 \leq i \leq L\}$. 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 questions. First, for a fixed number of discretization points, what is a good policy for choosing the spacing between points? Second, given that we have found a good spacing policy, what number of points is sufficient for good link performance, yet small enough for low computational overhead?

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

$$u_2 - u_1 = u_3 - u_2 = \cdots = u_L - u_{L-1}.$$

Another possibility is to have the points geometrically spaced

$$\frac{u_3 - u_2}{u_2 - u_1} = \frac{u_4 - u_3}{u_3 - u_2} = \cdots = \frac{u_L - u_{L-1}}{u_{L-1} - u_{L-2}} = \mathcal{S}$$

where $\mathcal{S}$ is a 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. In Fig. 15,

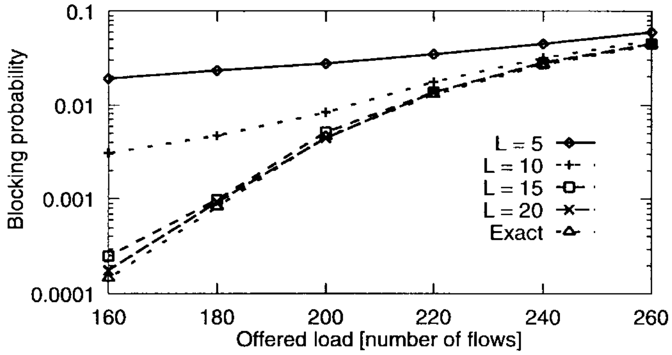


Fig. 16. Impact of number of discretization points on admission control performance.

we plot the results of our simulations for spacing factors $\mathcal{S} = 1, 1.1, \cdots, 1.5$, where a value of 1 corresponds to linear spacing. All of the half-widths of the 90% confidence intervals are within 5% of the point value. The graph "Exact" corresponds to the exact admission control algorithm, which forms the base case for our comparison. First, we note that with less than ten points, the blocking probability is unacceptably high compared to the base case. If 15 or more discretization points are used, then the linear spacing policy provides a link performance close to that given by the geometric spacing policy, regardless of spacing factor. For this scenario, linear spacing is the solution of choice due to its simplicity and near optimal performance.

In Fig. 16, we plot the results of simulation experiments with algorithms using linear spacing and different numbers of discretization points as a function of offered load. All of the half-widths of the 90% confidence intervals are within 5% of the point value. We observe that the blocking probability achieved by the discrete algorithm with 15 points is, indeed, quite close to the one achieved by the exact algorithm.

VI. Conclusion

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 [17] to flows characterized by multiple-segment envelopes. We developed a first set of algorithms having a computation complexity of $O(KN\log(KN))$, where N is the number of flows admitted in the EDF scheduler at the time of algorithm invocation and K is the number of segments per envelope. 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 $O(K+L)$, where L is the number of predefined discretization points. A set of simulation experiments showed that the improvement in execution time achieved by the discrete admission control is, indeed, significant (two orders of magnitude faster for the examples we consider) and that the algorithm's execution time is independent of the number of flows admitted. The computation time of our admission control algorithm has been around 10 μs per flow on a 266-MHz DECAlpha system. Moreover, we have seen that the link performance degradation of the discrete admission control relative to the exact admission control is small, while using a small number of discretization points (15). Taken together, these results suggest that the algorithms we have studied in this paper form the basis for a practical and highly efficient solution to the problem of admission control of real-time flows at EDF schedulers.

Appendix A
Proof of Theorem 2

By *Theorem 1*, a delay $d \in \mathbb{R}$ can be guaranteed to A_f^* iff (2), i.e.,

$$G(t) \triangleq F(t) - A_f^*(t-d) \geq 0, \quad \forall t \geq 0$$

where F is defined in (7). Given that F and A_f^* are piecewise linear and continuous, G has the same properties. Since a segment is above 0 if its ends are above 0, it follows that the schedulability condition (2) is equivalent to

$$G(u) \geq 0, \quad \forall u \in X_G. \tag{28}$$

By observing that $X_G = X_F \cup (X_{A_f^*} + d)$, (2) is further equivalent to

$$G(u) \geq 0, \quad \forall u \in X_F \tag{29}$$
$$G(u) \geq 0, \quad \forall u \in X_{A_f^*} + d. \tag{30}$$

To prove *Theorem 2*, it is sufficient to prove that (29) and (30) are equivalent to

$$d \geq u - A_f^{*\,-1}(F(u)), \quad \forall u \in X_F \tag{31}$$
$$d \geq v - a, \quad \forall v \in F^{-1}(A_f^*(a)), \quad \forall a \in X_{A_f^*}. \tag{32}$$

First, we observe that (29) $\Leftrightarrow$ (30), since A_f^* is invertible on $[0, \infty)$ (because it is strictly increasing and, thus, bijective), and since $F(u) \in [0, \infty) \quad \forall u \in \mathbb{R}$ (we assumed the set of N flows to be schedulable).

We prove the rest of the equivalence in two parts.

(29), (30) $\Rightarrow$ (32): We know that (29), (30) $\Rightarrow$ $G(t) \geq 0, \forall t \in \mathbb{R}$, so it is sufficient to show $G(t) \geq 0, \forall t \in \mathbb{R}$ implies (32). We prove this by contradiction. Assume that there is $a \in X_{A_f^*}$ and $v \in F^{-1}(A_f^*(a))$ such that $d < v - a$. Then, $v - d > a$ and, thus,

$$A_f^*(v-d) > A_f^*(a) = F(v)$$

which contradicts the statement $G(t) \geq 0 \quad \forall t \in \mathbb{R}$.

(32) $\Rightarrow$ (30): We prove this by contradiction. Assume that there exists $u \in X_{A_f^*} + d$ such that $G(u) < 0$. Then, there exists $a \in X_{A_f^*}$ such that $G(a+d) < 0$, which is $F(a+d) < A_f^*(a)$. But, $\lim_{t\to\infty} F(t) = \infty$ (otherwise $\lim_{t\to\infty} F(t) = 0$, contradicting the stability condition (1)). Since F is continuous, there exists $v \in (a+d, \infty)$ such that $F(v) = A_f^*(a)$, or, equivalently,

$$\exists \quad v \in F^{-1}(A_f^*(a)) \text{ such that } a + d < v$$

which contradicts (32).

APPENDIX B
PROOF OF THEOREM 3

We prove that $\overline{d}$ computed in *Theorem 3* is equal to the one computed in (10)–(12). Since

$$\forall a, \quad F^{-1}_{(u_{a,1}, u_{a_2})}(A^*_f(a)) \subset F^{-1}(A^*_f(a))$$

we have that $m_y \geq m_z$. It suffices to show that, if $m_y > m_x$, then $m_y = m_z$, where m_y is defined in (12).

First, we show that the set $F^{-1}_{(u_{a,1}, u_{a_2})}(A^*_f(a))$ is either empty or has one element. We have

$$\begin{aligned} u_{a,2} &> m_x + a && \text{by (14)} \\ &\geq a + u_{a,2} - {A^*_f}^{-1}(F(u_{a,2})) && \text{by (16).} \end{aligned}$$

It follows that ${A^*_f}^{-1}(F(u_{a,2})) > a$, and, thus, $F(u_{a,2}) > A^*_f(a)$ since A^*_f is strictly increasing. Since F is continuous and concave on $(u_{a,1}, u_{a,2})$ (there is no convex point of F between $u_{a,1}$ and $u_{a,2}$ by definition of $u_{a,1}$ and $u_{a,2}$), we have that exactly one of the following holds:

$$\begin{aligned} &\text{if } F(u_{a,1}) > A^*_f(a) \text{ then } F^{-1}_{(u_{a,1}, u_{a_2})}(A^*_f(a)) = \emptyset \\ &\text{if } F(u_{a,1}) \leq A^*_f(a) \text{ then } F^{-1}_{(u_{a,1}, u_{a_2})}(A^*_f(a)) \\ &\qquad \text{has one element.} \end{aligned}$$

Let us define

$$y_a = \max F^{-1}(A^*_f(a)) - a \tag{33}$$

$$z_a = F^{-1}_{(u_{a,1}, u_{a_2})}(A^*_f(a)) - a. \tag{34}$$

It is then sufficient to prove that, for any $a \in X^{cx}_{A^*_f}$, if $y_a > m_x$ then $y_a = z_a$. Let us assume $y_a > m_x$. By the definition of $u_{a,1}$ it follows that

$$y_a + a > u_{a,1}. \tag{35}$$

We have

$$F(u) > A^*_f(a) \quad \forall u \in \widehat{X^{cv}}, u \geq u_{a,2} \tag{36}$$

because

$$\begin{aligned} u - a &\geq u_{a,2} - a \quad \text{by (36)} \\ &> m_x \quad \text{by (14)} \\ &\geq u - {A^*_f}^{-1}(F(u)) \quad \text{by (16)} \end{aligned}$$

and, thus, ${A^*_f}^{-1}(F(u)) > a$, or $F(u) > A^*_f(a)$ since A^*_f is strictly increasing. It follows that $F(t) > A^*_f(a) \quad \forall t \geq u_{a,2}$, since $[u_{a,2}, \infty)$ can be partitioned into intervals $[u_i, u_{i+1}), [u_{i+1}, u_{i+2}), \cdots$, for i such that $u_i = u_{a,2}$, and u_i, u_{i+1} consecutive in $\widehat{X^{cv}}$. Then, for any $t \in (u_i, u_{i+1}), F$ is concave and continuous in (u_i, u_{i+1}), and, thus, $F(t) > \min(F(u_1), F(u_2)) > A^*_f(a)$. From $F(t) > A^*_f(a) \quad \forall t \geq u_{a,2}$, we have that $F^{-1}_{(u_{a,2}, \infty)}(A^*_f(a))$ is an empty set, and, thus,

$$y_a + a < u_{a,2}. \tag{37}$$

From (35) and (37), it follows that $y_a \in F^{-1}(A^*_f(a)) - a$. Then, $F^{-1}(A^*_f(a))$ has one element and, since $z_a \in F^{-1}(A^*_f(a)) - a$, it follows that $y_a = z_a$.

MINIMUM_DELAY (Input: $X^{cx}_{A^*_f}, Y_{A^*_f}, \rho_{f,n_f}, X_F, Y_F, X^{cv}_F,$
Y^{cx}_F, B;
Output: $\overline{d}$)

1 **for** each $W_l \in Y^{cx}_F$
2   and each $h_{f,k} \in Y_{A^*_f}$
3   **if** $W_l \in [h_{f,k}, h_{f,k+1})$
4   **then** $x_l \leftarrow u_l - {A^*_f}^{-1}(W_l)$
5 $m_x \leftarrow \max x_l$
6 **for** each $u_l \in X^{cv}_F$
7   and each $a_{f,k} \in X^{cx}_{A^*_f}$
8   **if** $u_l \leq m_x + a_{f,k} < u_{l+1}$ and $W_l < h_{f,k} < W_{l+1}$
9   **then** find $v_i \in X_F \cap [u_l, u_{l+1})$
          such that $V_i < h_{f,k} < V_{i+1}$, where $V_i \in Y_F$
10           $z_k \leftarrow F^{-1}_{(v_i, v_{i+1})}(h_{f,k}) - a_{f,k}$
11 $m_z \leftarrow \max z_k$
12 $\overline{d} \leftarrow \max(m_x, m_z)$

Fig. 17. An $O(KN)$ algorithm for computing the minimum delay for a multisegment envelope.

APPENDIX C
AN ADMISSION CONTROL ALGORITHM BASED ON THEOREM 3

In Fig. 17, we present an algorithm to compute the minimum guaranteeable delay, based on *Theorem 3*. We use the notation $h_{f,k} = A^*_f(a_{f,k}), W_l = F(u_l)$, for $u_l \in X^{cv}_F, V_q = F(v_q)$, for $v_q \in X_F$, and $B = c - \Sigma_{i=1}^N \rho_{i,n_i}$.

We assume that any flow envelope has $|X^{cv}_{A^*_i}| = |Y^{cv}_{A^*_i}| = O(K_1)$ convex points, $|X^{cx}_{A^*_i}| = |Y^{cx}_{A^*_i}| = O(K_2)$ concave points, and $|X_{A^*_i}| = |Y_{A^*_i}| = O(K)$ total points. Since $X^{cx}_F = \cup_i X^{cv}_{A^*_i}$ and $X^{cv}_F = \cup_i X^{cx}_{A^*_i}$, we have $|X^{cx}_F| = |Y^{cx}_F| = O(K_1 N)$ and $|X^{cv}_F| = |Y^{cv}_F| = O(K_2 N)$. To compute m_x, a lookup in both Y^{cx}_F and $Y_{A^*_f}$ is needed. Observe that, if we assume $Y_{A^*_f}$ is sorted in increasing order, then

$$\forall v \in \mathbb{R}^+ \quad \text{there is a unique } h_{f,k} \in Y_{A^*_f} \text{ such that } v \in [h_{f,k}, h_{f,k+1}).$$

If, in addition, Y^{cx}_F is also sorted, the lookups in steps 1 and 2 and the test in step 3 can be done in tandem, with two pointers that advance in Y^{cx}_F and $Y_{A^*_f}$, one at a time, without ever returning. It follows that the complexity for computing m_x is $O(|Y^{cx}_F| + |Y_{A^*_f}|) = O(K_1 N + K)$. To compute m_z, a search in X^{cv}_F and $X^{cx}_{A^*_f}$ is needed. Observe that, if X^{cv}_F sorted in increasing order,

$$\forall a_{f,k} \in X^{cx}_{A^*_f} \quad \text{there exists a unique } u_l \in X^{cv}_F \text{ such that } a_{f,k} \in [u_l, u_{l+1}).$$

If, in addition, $X^{cx}_{A^*_f}$ is also sorted, the lookups in steps 6 and 7 and the test in step 8 can be done in tandem, with two pointers that advance in X^{cv}_F and $X^{cx}_{A^*_f}$, one at a time, without ever returning, giving a complexity of $O(K_2 N + K_2) = O(K_2 N)$. Line 9 requires a search in X_F between consecutive convex points u_l and u_{l+1}, i.e., a search among the concave points of F in that interval. Since, in the loop 6–10, there is one lookup over all convex points of F, it follows that there is a total of one lookup over all concave points of F in the same loop, giving an aggregate complexity of $O(K_1 N)$. The total complexity of the algorithm is then $O(K_1 N + K + K_2 N + K_1 N) = O(KN)$. We observe here that this is also a lower

RESERVE (Input: $d, X_{A_f^*}, \rho_{f,n_f}, X_F, Y_F, X_F^{cv}, Y_F^{cx}, B$;
Output: $X_F, Y_F, X_F^{cv}, Y_F^{cx}$)
1 **for** each $v_l \in X_F$
2 **find** $a_{f,k} \in X_{A_f^*}$ such that $v_l - d \in [a_{f,k}, a_{f,k+1})$
3 $W_l' \leftarrow W_l - A^*_{f[a_{f,k}, a_{f,k+1}]}(v_l - d)$
4 **for** each $a_{f,k} \in X_{A_f^*}$
5 **find** $v_l \in X_F$ such that $d + a_{f,k} \in [v_l, v_{l+1})$
6 $W_{f,k} \leftarrow F_{[v_l, v_{l+1})}(d + a_{f,k}) - A_f^*(a_{f,k})$
7 $Y_F^{cx} \leftarrow \{W_l' | u_l \in X_F^{cv}\} \cup \{W_{f,k} | a_{f,k} \in X_{A_f^*}^{cx}\}$; sort Y_F^{cx}
8 $Y_F \leftarrow \{W_l'\} \cup \{W_{f,k}\}$; sort Y_F
9 $X_F^{cv} \leftarrow X_F^{cv} \cup (X_{A_f^*}^{cx} + d)$; sort X_F^{cv}
10 $X_F \leftarrow X_F \cup (X_{A_f^*} + d)$; sort X_F

Fig. 18. An $O(KN\log(KN))$ algorithm for resource reservation for flow f.

bound on the algorithm's complexity, since all F's segments have to be considered in computing d, and the number of segments in F is $O(KN)$. We conclude that the computation of $\overline{d}$ cannot be further simplified.

To complete the admission control algorithm at EDF schedulers, we show how the sets X_F, Y_F, X_F^{cv}, and Y_F^{cx} are updated when a flow is admitted (reservation), and a flow is terminated (release). The reservation algorithm in Fig. 18 updates the availability function, $F(t) \leftarrow F(t) - A_f^*(t-d)$, after a flow is admitted. Specifically, a new value of F is computed at each existing flexion point of F (lines 1–3). Next, all flexion points of A_f^* become new flexion points for F, and the value of F at these points is computed (lines 4–6). Finally (lines 7–10), the sets X_F, Y_F, X_F^{cv}, and Y_F^{cx}, containing the new values, are sorted, as required by the MINIMUM_DELAY algorithm. To determine the complexity of this RESERVE algorithm, we observe that loops 1–3 and 4–6 can be performed in $O(KN)$ time if X_F and $X_{A_f^*}$ are sorted. Sorting X_F^{cv} and Y_F^{cx} requires $O(K_2 N\log(K_2 N))$, and sorting X_F and Y_F requires $O(KN\log(KN))$. It follows that the total complexity of RESERVE is $O(KN\log(KN))$. A complementary but similar algorithm updates the same sets upon a flow termination, by changing all values of F in its flexion points according to $F(t) \leftarrow F(t) + A_f^*(t-d)$, where f is the terminating flow.

APPENDIX D
PROOF OF THEOREM 4

We need to prove

$$d \ge \overline{d}^s \Leftrightarrow A^s_{f,d}(t) \le F(t), \qquad \forall t \in \mathbb{R}. \tag{38}$$

It suffices to restrict ourselves to $t \in \mathcal{D}$, since, by definition, the cover envelope has all its flexion points in the fixed set $\mathcal{D}$. Given (22) and (20), the following two statements yield (38) for $t \in \mathcal{D}$:

$$d \ge m_x \Leftrightarrow A_{f,d}(u) \le F(u), \qquad \forall u \in \mathcal{D}. \tag{39}$$

Given $d \ge m_x$,

$$d \ge m_z \Leftrightarrow A^+_{f,d}(u) \le F(u), \quad \forall u \in \mathcal{D}. \tag{40}$$

Proof of (39): From (23) and the definition of $A_{f,d}$, (17) it is sufficient to prove

$$d \ge u - A_f^{*-1}(F(u)) \Leftrightarrow A_{f,d}(u) \le F(u), \qquad \forall u \in \mathcal{D}.$$

This follows from

$$A_f^{*-1}(F(u)) \ge u - d \Leftrightarrow F(u) \ge A_f^*(u-d) = A_{f,d}(u)$$

since A_f^* is increasing and bijective and from (17).

Proof of (40): Given $d \ge m_x$, it is sufficient to prove

$$d \ge m_z \Leftrightarrow A^+_{f,d}(u) \le F(u), \quad \forall u \in \mathcal{D}$$
$$\text{such that } A^+_{f,d}(u) > 0.$$

Based on (24) and (21), it is sufficient to show that

$$d \ge \min(u_{i_a} - A_{f,a}^{*-1}(F(u_{i_a})), u_{i_a+1} - a)$$
$$\Leftrightarrow A^*_{f,a}(u_j - d) \le F(u_j), \quad \forall a \in X^{cx}_{A_f^*}, \tag{41}$$
$$\text{where} \quad u_j \le a + d < u_{j+1}. \tag{42}$$

Let us consider an arbitrary $a \in X^{cx}_{A_f^*}$. We are given $d \ge m_x$, so $d + a \ge m_x + a$, and, from (25) and (42), it follows that $u_j \le u_{i_a}$. It follows that we have two cases, $u_j = u_{i_a}$ and $u_j > u_{i_a}$, which are considered separately in the following.

Case $u_j = u_{i_a}$: We have to prove

$$d \ge \min(u_j - A_{f,a}^{*-1}(F(u_j)), u_{j+1} - a) \tag{43}$$
$$\Leftrightarrow A^*_{f,a}(u_j - d) \le F(u_j). \tag{44}$$

From (42), we have $d < u_{j+1} - a$. It follows that (43) $\Leftrightarrow$ $d \ge u_j - A_{f,a}^{*-1}(F(u_j))$. But, the latter is equivalent to (44), since $A^*_{f,a}$ is increasing.

Case $u_j > u_{i_a}$: We have to prove

$$d \ge \min(u_{i_a} - A_{f,a}^{*-1}(F(u_{i_a})), u_{i_a+1} - a) \tag{45}$$
$$\Leftrightarrow A^*_{f,a}(u_j - d) \le F(u_j). \tag{46}$$

From the hypothesis of this case, $u_j \ge u_{i_a+1}$. Also, from (42), $u_j \le a + d < u_{j+1}$, we have $u_{i_a+1} \le a + d$, and, thus, (45) is true. Thus, to prove (45) $\Leftrightarrow$ (46), it is sufficient to prove that (46) is true for any $d, d \ge u_{i_a+1} - a$. Given that A_f^* is increasing, we have

$$\begin{aligned} F(u_j) &\ge A_f^*(u_j - m_x) \ \text{ by } m_x \ge u_j - A_f^{*-1}(F(u_j)), \ (23) \\ &\ge A_f^*(u_{i_a+1} - m_x) \quad \text{by } u_j \ge u_{i_a+1} \\ &\ge A_f^*(a) \quad \text{by } m_x + a < u_{i_a+1}, (25) \\ &= A^*_{f,a}(a) \quad \text{by } (19) \\ &\ge A^*_{f,a}(u_j - d) \quad \text{by } u_j \le a + d \text{ by } (42) \end{aligned}$$

REFERENCES

[1] *ATM User–Network Interface Specification, Version 3.1*, ATM Forum, Sept. 1994.

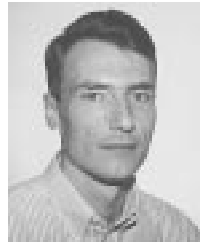
[2] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, *Resource ReSerVation Protocol (RSVP)—Version 1, Functional Specification*, IETF Proposed Standard RFC 2205, Sept. 1997.

[3] C.-S. Chang, "Stability, queue length and delay of deterministic and stochastic queuing networks," *IEEE Trans. Automat. Contr.*, vol. 39, May 1994.

[4] R. Cruz, "Quality of service guarantees in virtual circuit switched networks," *IEEE J. Select. Areas Commun.*, vol. 13, Aug. 1995.

[5] ——, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, Jan. 1991.

[6] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queuing algorithm," *Comput. Commun. Rev.*, vol. 19, no. 4, 1989.
[7] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.*, vol. 8, Apr. 1990.
[8] V. Firoiu, "Network support for applications requiring quality of service in heterogeneous environments," Ph.D. dissertation, Dep. Comput. Sci., Univ. Massachusetts, Amherst, 1998.
[9] V. Firoiu and D. Towsley, "Call admission and resource reservation for multicast sessions," in *Proc. IEEE INFOCOM*, 1996, pp. 94–101.
[10] L. Georgiadis, R. Guérin, and A. Parekh, "Optimal multiplexing on a single link: Delay and buffer requirements," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1518–1535, Sept. 1997.
[11] L. Georgiadis, R. Guérin, V. Peris, and K. N. Sivarajan, "Efficient network QoS provisioning based on per node traffic shaping," *IEEE/ACM Trans. Networking*, vol. 4, pp. 482–501, Aug. 1996.
[12] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A simple and efficient service for multiple time-scale traffic," in *Proc. ACM SIGCOMM*, 1995.
[13] D. Kandlur, K. Shin, and D. Ferrari, "Real-time communications in multihop networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 5, Oct. 1994.
[14] E. Knightly, D. Wrege, J. Liebeherr, and H. Zhang, "Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic," in *Proc. ACM SIGMETRICS'95*, 1995.
[15] E. Knightly and H. Zhang, "Traffic characterization and switch utilization using a deterministic bounding interval dependent traffic model," in *Proc. IEEE INFOCOM*, 1995.
[16] E. Knightly and H. Zhang, "D-BIND: An accurate traffic model for providing QoS guarantees to VBR traffic," *IEEE/ACM Trans. Networking*, vol. 5, Apr. 1997.
[17] J. Liebeherr, D. Wrege, and D. Ferrari, "Exact admission control for networks with bounded delay services," *IEEE/ACM Trans. Networking*, vol. 4, Dec. 1996.
[18] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. Assoc. Comput. Mach.*, vol. 20, no. 1, Jan. 1973.
[19] O. Rose, "MPEG coded video traces," ftp://ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG/.
[20] ——, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," University of Würzburg, Würzburg, Germany, Technical Report TR 101, Feb. 1995.
[21] A. K. Parekh, "A generalized processor sharing approach to flow control in integrated services networks," Ph.D. dissertation, Dep. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, Feb. 1992.
[22] F. Preparata and M. Shamos," *Computational Geometry*. Berlin, Germany, Springer-Verlag, 1985.
[23] S. Sahu, V. Firoiu, D. Towsley, and J. Kurose, "Traffic models and admission control for variable bit rate continuous media transmission with deterministic service," in *Proc. SPIE Symp. Voice, Video and Data Communications*, 1998.
[24] S. Shenker, C. Partridge, and R. Guérin, *Specification of Guaranteed Quality of Service*, IETF Proposed Standard RFC 2212, Sept. 1997.
[25] D. Wrege and J. Liebeherr, "A near-optimal packet scheduler for QoS networks," in *Proc. IEEE INFOCOM*, 1997.
[26] H. Zhang and D. Ferrari, "Rate-controlled service disciplines," *J. High Speed Networks*, vol. 3, no. 4, 1994.
[27] Q. Zheng and K. Shin, "On the ability of establishing real-time channels in point-to-point packet-switched networks," *IEEE Trans. Commun.*, vol. 42, 1994.



Victor Firoiu (S'95–M'98) received the Diploma of Engineer degree from the Department of Computer Science, Polytechnic Institute of Bucharest, Bucharest, Romania, in 1987 and the M.S. and Ph.D. degrees from the Department of Computer Science, University of Massachusetts, Amherst, in 1995 and 1998, respectively.

He was a Research Intern with the Broadband Networking Group, IBM T. J. Watson Research Center, Yorktown Heights, NY, in 1995 and 1996. He is currently a member of the Internetworking Architecture Laboratory, Bay Networks, Inc., Billerica, MA. His current research interests include differentiated services for the Internet, network support for multicast applications, and congestion control for non-TCP and multicast flows.

Dr. Firoiu is a member of the Association for Computing Machinery.

Jim Kurose (S'81–M'84–SM'91–F'97) received the B.A. degree in physics from Wesleyan University, Middletown, CT, and the M.S. and Ph.D. degrees in computer science from Columbia University, New York, NY.

He is currently a Professor of Computer Science and Department Chair at the University of Massachusetts, Amherst. He has been a Visiting Scientist at IBM Research, INRIA Sophia ANtipolis, and EURECOM. His research interests include real-time and multimedia communication, network and operating system support for servers, and modeling and performance evaluation. For many years, he has served on the the program committees for the IACM SIGCOMM and ACM SIGMETRICS Conferences.

Dr. Kurose is the past Editor-in-Chief of the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE/ACM TRANSACTIONS ON NETWORKING. He has been active in the Program Committee for IEEE INFOCOM for a number of years. He has received several awards for his teaching.

Don Towsley (M'78–SM'93–F'95) received the B.A. degree in physics and the Ph.D. degree in computer science from the University of Texas at Austin in 1971 and 1975, respectively.

From 1976 to 1985, he was a member of the faculty of the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. He is currently a Distinguished University Professor at the University of Massachusetts in the Department of Computer Science. During 1982–1983, he was a Visiting Scientist at the IBM T. J. Watson Research Center, Yorktown Heights, NY, and, during the years 1989–1990, he was a Visiting Professor at the Laboratoire MASI, Paris, France. His research interests include high-speed networks and multimedia systems. He currently serves on the Editorial Board of *Performance Evaluation*. He was a Program Co-Chair of the joint ACM SIGMETRICS and PERFORMANCE '92 Conference. He is a member of the IFIP Working Groups 6.3 and 7.3.

Prof. Towsley has previously served on the Editorial Boards of the IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE/ACM TRANSACTIONS ON NETWORKING. He is a member of the Operations Research Society of America. He has been elected a Fellow of the Association for Computing Machinery (ACM) and has received two Best Conference Paper Awards from ACM SIGMETRICS.