Real-Time Estimation and Dynamic Renegotiation of UPC Parameters for Arbitrary Traffic Sources in ATM Networks

Brian L. Mark, *Member, IEEE*, and Gopalakrishnan Ramamurthy, *Senior Member, IEEE*

***Abstract*— This paper presents a robust and flexible real-time scheme for determining appropriate parameter values for usage parameter control (UPC) of an arbitrary source in an asynchronous transfer mode network. In our approach, the UPC parameters are chosen as a function of the statistical characteristics of the observed cell stream, the user's tolerance for traffic shaping, and a measure of the network cost. For this purpose, we develop an approximate statistical characterization for an arbitrary cell stream. The statistical characterization is mapped to a UPC descriptor that can be negotiated with the network. The selected UPC descriptor is optimal in the sense of minimizing a network cost function, subject to meeting user-specified constraints on shaping delay. The UPC estimation scheme is extended to adapt to slow time-scale changes in traffic characteristics via dynamic renegotiation of the UPC parameters. We illustrate the effectiveness of our methodologies with examples taken from MPEG video sequences.**

***Index Terms*—Asynchronous transfer mode, quality of service, real-time estimation, resource allocation, traffic characterization.**

I. Introduction

SUPPORT FOR applications requiring quality-of-service (QoS) guarantees is an important challenge for designers of future high-speed asynchronous transfer mode (ATM) networks. Efficient resource allocation for connections requiring QoS can be performed only if reliable and accurate characterizations of the offered traffic are available. Ideally, a traffic descriptor should be able to capture the main characteristics of a cell stream that determine the amount of resource required to satisfy a specified QoS requirement. On the other hand, a traffic descriptor should be sufficiently simple such that: 1) the descriptor parameters are enforceable at the network edge and 2) admission control decisions can be made in real-time.

The ATM forum [1] has adopted an open-loop flow control mechanism for ATM networks known as usage parameter control (UPC). A UPC algorithm polices a given cell stream in accordance with a set of deterministic parameters. Each cell is determined to be either conforming or nonconforming to the UPC parameters. Nonconforming cells may be discarded or tagged as lower priority cells at the network edge. The UPC proposed by the ATM Forum [1] and the one assumed throughout this paper is the *dual leaky bucket algorithm*. The basic dual leaky bucket consists of three parameters: the peak rate, the sustainable rate, and the bucket size.

In a UPC-based network control paradigm, the user specifies its traffic to the network at connection setup time in terms of a set of UPC parameter values. The UPC parameters constitute a traffic descriptor that specifies a worst-case envelope characterizing the cell stream entering at the network edge. In practice, user applications are often ill-prepared to describe their traffic streams in terms of UPC parameters. Thus, there is a practical need to automate the process of characterizing a cell stream in terms of a UPC-based descriptor. The parameter values chosen for a cell stream should be large enough to allow the stream to pass through the UPC function with minimal distortion. However, smaller UPC values lead to more efficient resource allocation decisions and, consequently, higher network utilization.

In this paper, we develop a scheme for estimating an "optimal" UPC descriptor for an arbitrary cell stream. The choice of UPC parameters is based on the statistical characteristics of the cell stream, the user's tolerance for shaping due to UPC, and the network resource requirement associated with a UPC descriptor. The user negotiates the selected UPC parameters with the network and then shapes its traffic stream to force conformance to the negotiated UPC prior to entry at the network edge. On the network side, a connection admission control (CAC) based on the UPC descriptor ensures that quality-of-service (QoS) requirements are maintained for all admitted streams (cf. [2]).

In order to characterize a real-time cell stream, a short preamble that provides a typical traffic profile could precede the stream to permit estimation of the UPC parameters. Alternatively, an initial conservative set of UPC parameters could be negotiated during connection setup. After the UPC estimation procedure is applied to a short segment of the stream, a refined UPC descriptor could be *renegotiated* with the network to reduce the cost to the user. This concept can be extended to allow multiple renegotiations of the UPC parameters throughout the duration of a cell stream. Such a scheme would be useful for real-time video streams containing scene changes, wherein each distinct scene could be characterized by a UPC descriptor.

Our methodology for UPC estimation incorporates the ability to renegotiate UPC parameters dynamically. In our approach, a renegotiation attempt is made only if a significant change in traffic characteristics, from the user's perspective,

Manuscript received March 26, 1996; revised November 12, 1997; approved by IEEE/ACM Transactions on Networking Editor S. Li.

The authors are with C&C Research Laboratories, NEC USA, Princeton, NJ 08540 USA (e-mail: blmark@ccrl.nj.nec.com; mgr@ccrl.nj.nec.com).

Publisher Item Identifier S 1063-6692(98)09619-8.

is detected. If a renegotiation request is rejected, the network continues to police the stream according to the current negotiated UPC descriptor. The user may suffer some QoS degradation until the network can grant a renegotiation request. It is important that renegotiation requests are managed such that the probability of rejection is small (cf. [3]).

The leaky bucket was introduced by Turner [4] as a means of traffic policing. The properties of the leaky bucket have been studied extensively in the literature with various traffic model assumptions (cf. [5]). A deterministic traffic characterization based on the leaky bucket model was formalized by Cruz [6] and used to obtain deterministic network delay bounds. Related work has appeared in [7]–[9]. These works generally assume that the leaky bucket parameters are given and proceed with an analysis of the system at hand.

In contrast to these works, our paper focuses on the problem of designing a suitable set of leaky bucket parameters for usage parameter control (cf. [10]). In our approach, the input stream is viewed as a random arrival process that is characterized statistically. We remark that statistical characterizations of traffic have been studied extensively in the literature (cf. [7], [11], [12]). The contribution of this paper is to provide a sound methodology for mapping empirical observations of a random traffic stream to a deterministic UPC descriptor that is suitable for negotiation at connection setup time or for renegotiation if it is supported.

The concept of a renegotiated service has been proposed earlier in the literature. Tedijanto and Gün [13] propose a dynamic bandwidth management mechanism with leaky bucket control and traffic characterization with exponential on–off sources. Our approach to traffic control differs from theirs in that it is based only on the UPC parameters and does not assume an underlying source model. Grossglauer *et al.* [14] propose a renegotiated CBR service (RCBR), in which a single rate parameter can be renegotiated. In this paper, the renegotiation service is based on the more general three-parameter dual leaky bucket UPC for VBR service [1]. Reininger *et al.* [15] developed an earlier heuristic approach to UPC renegotiation designed specifically for MPEG streams. Our methodologies are suitable for arbitrary ATM cell streams and take into account both user constraints and network cost.

The remainder of the paper is organized as follows. Section II reviews the leaky bucket shaping and policing mechanisms. Section III develops the statistical characterization for an arbitrary cell stream. Section IV discusses the criteria for mapping the statistical characterization to an appropriate UPC descriptor. Section V describes a scheme for dynamically generating a sequence of UPC descriptors for a real-time stream in conjunction with UPC renegotiation. Numerical examples of the statistical characterization and the selection of UPC values are discussed in Section VI. Finally, Section VII concludes the paper with a discussion of future research directions.

II. Usage Parameter Control

A UPC algorithm accepts a cell stream as input and determines whether or not each cell conforms to the parameters of the algorithm. Depending on the network policy, nonconforming cells are either dropped prior to entering the network or tagged as low priority (i.e., the CLP bit in the ATM cell header is set to one). The UPC function acts as a *traffic policer* at the network edge. In contrast, a *traffic shaper* smooths out the bit-rate variations in a stream by delaying rather than dropping or tagging cells. A user can avoid possible cell loss at the network edge due to UPC by shaping its cell stream to conform to the negotiated UPC. The UPC mechanisms proposed by the ATM Forum are variants of the leaky bucket algorithm (cf. [4], [16], [17]). The leaky bucket is a simple means of shaping or policing a traffic stream and has desirable attributes in terms of network control (cf. [6], [8]).

A leaky bucket is parameterized by a leak rate μ and a bucket size B. We shall find it convenient to describe the operation of the leaky bucket in terms of a fictitious $\cdot/D/1/B$ queue served at constant rate μ with capacity B (cf. [16]). In this model, each cell arrival to the leaky bucket generates a corresponding customer arrival to the fictitious queue. At the time of a cell arrival, if the number of customers in the fictitious queue is less than B, the cell is said to be *conforming*. Otherwise, if the cell arrival sees B or more customers in the fictitious queue, the cell is *nonconforming*. Nonconforming cells are either dropped or tagged as low priority cells.

A leaky bucket shaper parameterized by (μ, B) operates in a manner similar to the leaky bucket policer, except that nonconforming cells are delayed, rather than dropped or tagged. The operation of the leaky bucket shaper can be understood in terms of a $\cdot/D/1$ queue. Upon arrival of a cell, if the number of customers in the fictitious queue is B or more, the cell is placed in an infinite capacity first-in, first-out (FIFO) buffer. Whenever a customer leaves the fictitious queue, a cell (if any) is removed and transmitted from the FIFO buffer.

The *dual leaky bucket* consists of two leaky buckets: a *peak rate* leaky bucket with parameters (λ_p, B_p) and a *sustainable rate* leaky bucket with parameters (λ_s, B_s), where $\lambda_p > \lambda_s$. A cell is conforming with respect to the dual leaky bucket if and only if it is conforming with respect to both the peak and sustainable rate buckets. The sustainable rate λ_s represents the maximum long-run average rate for conforming cell streams. The peak rate λ_p specifies a minimum intercell spacing of $1/\lambda_p$ with burst tolerance B_p. In this paper, we shall assume that $B_p = 1$, so that the intercell spacing of conforming streams is constrained to be at most $1/\lambda_p$. Values of $B_p > 1$ allow for cell delay variation (CDV) between the customer premises and the network access point (cf. [1]).

The set of three parameters $(\lambda_p, \lambda_s, B_s)$ constitutes a UPC descriptor that the user negotiates with the network. If the connection is accepted, the user applies a dual leaky bucket shaper to shape its cell stream according to the negotiated UPC. The dual leaky bucket limits the peak rate to λ_p, the long-term average rate to λ_s, and the maximum burst length (transmitted at peak rate) to $B_c = B_s\lambda_p/(\lambda_p - \lambda_s)$.

III. Traffic Characterization

A given UPC descriptor $(\lambda_p, \lambda_s, B_s)$ characterizes the set of all cell streams *conforming* to a dual leaky bucket parameterized by $(\lambda_p, \lambda_s, B_s)$. Many cell streams conform to a given set $(\lambda_p, \lambda_s, B_s)$. Conversely, a given cell stream may conform to multiple UPC descriptors. In this section, we develop a statistical characterization of a cell stream based on UPC descriptors.

A. Deterministic Characterization

We first consider a deterministic characterization based on the queueing model of the leaky bucket discussed in Section II. Suppose that the cell stream is of finite length and that the peak rate λ_p is known. The cell stream is offered as input to a queue served at constant service rate μ. Let $B_{\max}(\mu)$ be the maximum number of cells observed in the system over the duration of the cell stream. We can then characterize the stream by the peak rate and the values $B_{\max}(\mu)$ for $0 < \mu < \lambda_p$ as follows:

$$\mathcal{C}_D = [\lambda_p; B_{\max}(\mu),\, 0 < \mu < \lambda_p]. \tag{1}$$

This characterization is equivalent to the burstiness curve characterization discussed by Low and Varaiya [8].

The characterization $\mathcal{C}_D$ can be interpreted as follows. For each μ, $0 < \mu < \lambda_p$, the cell stream passes through a leaky bucket parameterized by $(\mu, B_{\max}(\mu))$ without being shaped, i.e., the cell stream conforms to the leaky bucket parameters $(\mu, B_{\max}(\mu))$. We introduce the notation $B(\mathcal{C}_D, \mu) = B_{\max}(\mu)$ to show that the choice of bucket size, for a given leak rate μ, depends on $\mathcal{C}_D$. The characterization $\mathcal{C}_D$ is *tight* in the following sense. If the cell stream conforms to a given set of leaky bucket parameters (μ, B'), then necessarily $B' \geq B(\mathcal{C}_D, \mu)$.

For a given leak rate μ^*, the minimum bucket size for conformance is $B^* = B(\mathcal{C}_D, \mu^*)$. However, a parameter choice based on the deterministic characterization $\mathcal{C}_D$ can be overly conservative. Another drawback of the deterministic characterization is that it can only be obtained off-line, i.e., the entire cell stream must be known *a priori*.

B. Statistical Characterization

1) Ideal Characterization: Let us assume that the peak rate of the cell stream λ_p and the mean rate λ_m are known. Suppose that the cell stream is offered to a leaky bucket traffic shaper with leak rate μ, where $\lambda_m < \mu < \lambda_p$, and bucket size B. We define the *shaping probability* as the probability that an arbitrary arriving cell has to wait in the data buffer of the leaky bucket. In the $\cdot/D/1$ queueing model of the leaky bucket, this event corresponds to a customer arrival to the fictitious queue when the number in the system exceeds B.

Define $B(\mu, \epsilon)$ as the minimum bucket size necessary to ensure that the shaping probability does not exceed ϵ, where $0 < \epsilon < 1$. For fixed ϵ, define the following statistical traffic characterization:

$$\mathcal{C}_S(\epsilon) = [\lambda_p; \lambda_m; (\mu, B(\mu, \epsilon)),\, \lambda_m < \mu < \lambda_p]. \tag{2}$$

This characterization is analogous to the concept of *peakedness* in teletraffic theory [18]. In the definition of peakedness, an arrival stream is offered to an infinite group of independent servers and a statistic of the number of busy servers is recorded. Analogously, $\mathcal{C}_S(\epsilon)$ characterizes the stream in terms of its effect on a single server queue. Also note that $\mathcal{C}_S(\epsilon)$ is a statistical analog of the deterministic characterization $\mathcal{C}_D$. In fact, $\mathcal{C}_D$ may be viewed as a limiting case of $\mathcal{C}_S(\epsilon)$ as $\epsilon \to 0$. Letting ϵ range over the interval (0, 1), we define a more complete characterization by

$$\mathcal{C}_S = [\lambda_p; \lambda_m; (\mu, B(\mu, \epsilon)),\, \lambda_m < \mu < \lambda_p,\, 0 < \epsilon < 1]. \tag{3}$$

2) Approximate Characterization: The characterization $\mathcal{C}_S$ is an idealization of the information that we would like to capture from the cell stream. We now develop an approximate statistical characterization, denoted by $\hat{\mathcal{C}}_S$, that can be obtained empirically. Suppose that the cell stream is offered to a bank of constant rate queues with rates μ in the range $\lambda_m < \mu < \lambda_p$, where λ_p and λ_m are, respectively, the peak and mean rates of the offered cell stream. In practice, suitable estimates of the mean and peak rates can be obtained through measurements (see Section V). The traffic characterization records statistics of the queueing behavior observed at a bank of constant rate queues, each of which is offered the given cell stream as input.

Suppose that the cell stream is offered to a queue with constant service rate μ, where $\lambda_m < \mu < \lambda_p$. Let W denote the steady-state waiting time, in queue, for an arbitrary cell arriving to the queue. We approximate the complementary waiting time distribution with the exponential form

$$P(W > t) \approx a(\mu)e^{-b(\mu)t}, \qquad t \geq 0 \tag{4}$$

where a and b are written as functions of μ. If t is measured in seconds, b has units of seconds^{-1}. With relatively weak assumptions on the nature of the arrival process, the exponential form in (4) holds asymptotically in t, i.e.,

$$e^{bt}P(W > t) \to a \quad \text{as } t \to \infty. \tag{5}$$

This asymptotic form holds, in particular, for a fairly general class of arrival processes with Markovian structure (cf. [5], [19]). The weaker asymptotic form

$$t^{-1} \log P(W > t) \to -b \quad \text{as } t \to \infty \tag{6}$$

has been shown to hold under more general conditions than (4), i.e., no Markovian assumption is necessary (cf. [7], [20]). We note that the approximation in (4) is exact in the case of on–off Markov sources (cf. [5]). For small-to-moderate buffer sizes, Markovian models have been found to give sufficiently accurate solutions for a wide class of traffic types, including the self-similar traffic models (cf. [21]).

The constant a in (5) is often difficult to obtain. Abate *et al.* [19] propose the simple approximation $a \approx bE[W]$, which we shall also use. In many cases, the approximation in (4) with $a = 1$ tends to be conservative, i.e., the bound, $P(W > t) \leq e^{-bt}$, holds under certain conditions (see [7], [22]). The one-parameter approximation

$$P(W > t) \approx e^{-bt} \tag{7}$$

is the basis for the theory of *effective bandwidths* (cf. [5], [23] and references contained therein). This approximation has been found to be reasonably accurate (i.e., useful for resource allocation) for moderate to large values of t when the number of independent sources composing the arrival process is small. However, the approximation can deteriorate as the number of independent sources increases. We refer the reader to [20], [22] for more extensive discussions of this class of approximations. For our purposes, we shall take (4) as a reasonable approximation for an ATM cell stream modeled as a general Markovian source. Over the relatively short observation windows of interest (on the order of seconds), our approximations are fairly robust for a large class of real traffic sources, in particular, MPEG video sequences (see Section VI).

The pair of values $(a(\mu), b(\mu))$ provides a statistical characterization of the *effect* of the cell stream when offered to a queue with constant service rate μ. A more complete characterization of the cell stream records the values $(a(\mu), b(\mu))$ for all μ in the range (λ_m, λ_p) as follows:

$$\hat{\mathcal{C}}_S = [\lambda_p; \lambda_m; (a(\mu), b(\mu)) \colon \lambda_m < \mu < \lambda_p]. \tag{8}$$

We choose values of $(a(\mu), b(\mu))$ such that (4) holds *approximately* for values of t in our range of interest. Assuming that (4) holds at equality, we find that (cf. [18])

$$\frac{a(\mu)}{b(\mu)} = E[W] = \tau_r(\mu)E[S_a] + E[Q_a]/\mu \tag{9}$$

where $S_a \in \{0, 1\}$ and $Q_a \in \mathcal{Z}_+$ denote, respectively, the number of customers in service and the number of customers in queue seen by an arbitrary customer arrival. In (9), $\tau_r(\mu)$ is the average remaining service time for the customer in service as seen by an arriving customer conditional on there being a customer in service. Note that

$$E[S_a] = P(W > 0) = a(\mu). \tag{10}$$

Define $q(\mu) = \mathrm{E}[Q_a]$. Solving for $b(\mu)$ in (9), we obtain

$$b(\mu) = \frac{a(\mu)\mu}{\mu\tau_r(\mu)a(\mu) + q(\mu)}. \tag{11}$$

Practical means of obtaining the characterization $\hat{\mathcal{C}}_S$ empirically are discussed in Section V.

IV. Determining UPC Values

Having obtained the statistical characterization $\hat{\mathcal{C}}_S$ of a cell stream, the next step is to map the characterization to a suitable UPC descriptor.

A. UPC Selection Criteria

To avoid UPC violation at the network edge, typically a shaper is applied at the source to force the cell stream to conform to the negotiated UPC. In making admission control decisions, the network uses the UPC values as a worst-case descriptor of the traffic offered by a connection. No advantage is gained by smoothing the traffic stream beyond the minimal shaping imposed by the UPC. Once a UPC descriptor is negotiated for a cell stream, any smoothing of the stream which does not shape according to the UPC is suboptimal in this sense.

We shall assume that the user is able to tolerate some smoothing of its cell stream. If the user can tolerate a larger shaping delay, a UPC descriptor with smaller network cost can be chosen for the connection. We formulate the problem of selecting the UPC values for a cell stream as follows:

$$\begin{aligned} P_1 \colon & \min_{\lambda_s, B_s} c(\lambda_p, \lambda_s, B_s) \\ & \text{subject to: } P(\text{shaping}) \le \epsilon, \lambda_m \le \lambda_s \le \lambda_p. \end{aligned}$$

Here, the peak rate λ_p and the mean rate λ_m are assumed to be known *a priori* or obtained through measurements. The *cost function* $c(\lambda_p, \lambda_s, B_s)$ is an increasing function of each argument and represents the cost to the network of accepting a call characterized by the UPC values $(\lambda_p, \lambda_s, B_s)$. The shaping probability $P(\text{shaping})$, which depends on the cell stream characteristics, represents the probability that an arbitrary arriving cell will be delayed by the shaper.

In Section III-B, we discussed an idealized traffic characterization $\mathcal{C}_S(\epsilon)$ which records the values $B(\mu, \epsilon)$ for $\lambda_m < \mu < \lambda_p$, where $B(\mu, \epsilon)$ is the minimum bucket size that ensures a shaping probability less than ϵ for a leaky bucket with rate μ. From the approximate statistical characterization $\hat{\mathcal{C}}_S$, we can obtain an estimate $B(\hat{\mathcal{C}}_S, \mu, \epsilon)$ for this minimum bucket size. Substituting $B(\hat{\mathcal{C}}_S, \mu, \epsilon)$ for $B(\lambda, \epsilon)$ in P_1, the problem can be reformulated as follows:

$$\begin{aligned} P \colon & \min_{\lambda_s} c(\lambda_p, \lambda_s, B(\hat{\mathcal{C}}_S, \lambda_s, \epsilon)) \\ & \text{subject to: } \lambda_m \le \lambda_s \le \lambda_p. \end{aligned}$$

B. Shaping Probability

Consider a leaky bucket shaper with parameters (μ, B). In the $\cdot/D/1$ model of the leaky bucket, an arriving cell is delayed or shaped if and only if the number of customers seen in the fictitious queue is B or greater. Otherwise, the cell is transmitted immediately without delay. Suppose that the user imposes the constraint that the UPC descriptor $(\lambda_p, \lambda_s, B_s)$ is to be chosen such that the probability that an arbitrary cell will be shaped is less than ϵ. For a fixed leak rate μ, we would like to determine the minimum bucket size $B(\mu, \epsilon)$ required to ensure a shaping probability less than ϵ. We remark that the shaping probability is an upper bound for the *violation probability*, i.e., the probability that an arriving cell is nonconforming, in the corresponding UPC policer.

Let X_a be a random variable representing the number in the fictitious $\cdot/D/1$ queue observed by an arbitrary cell arrival. Suppose that the number of customers in the fictitious queue upon cell arrival is given by $X = M$, where $M \ge 1$ is an integer. This means that the new customer sees $M - 1$ customers in the queue and one customer currently in service. Therefore, the waiting time in the queue W' for this customer lies in the range $((M-1)/\mu, M/\mu)$. Conversely, if W' for this customer lies in this range, then $X = M$ at the arrival epoch; hence,

$$X = M \iff \frac{M-1}{\mu} < W' < \frac{M}{\mu}.$$

Therefore,

$$X \ge M \iff W' > \frac{M-1}{\mu}.$$

We can then infer that

$$P(X_a \geq B) = P(W > (B-1)/\mu).$$

The shaping probability is, therefore, given by

$$p_{\text{sh}} = P(X_a \geq B) \approx a(\mu)e^{-b(\mu)(B-1)/\mu}$$

where we have used the exponential approximation for waiting time in (4). Setting p_{sh} equal to the constant $\epsilon \in (0, 1)$ and formally solving for the bucket size yields

$$B(\hat{\mathcal{C}}_S, \mu, \epsilon) = \frac{\mu}{b(\mu)} \log\left(\frac{a(\mu)}{\epsilon}\right) + 1. \tag{12}$$

For a fixed shaping probability ϵ, $B(\hat{\mathcal{C}}_S, \mu, \epsilon)$ can be viewed as a function of μ. As ϵ increases, the required bucket size B for a given μ ($\lambda_m < \mu < \lambda_p$) decreases. Conversely, as ϵ decreases, the required bucket size B increases. Note that in (12) the bucket size grows proportionally to the logarithm of ϵ as $\epsilon \downarrow 0$. This is due to the exponential approximation in (4).

C. Shaping Delay

If the user can tolerate a higher degree of shaping, it is more natural to specify the shaping constraint in terms of delay. Consider a cell stream passing through a leaky bucket traffic shaper with parameters (μ, B). Recall that the state counter X may be viewed as the number of customers in a fictitious $\cdot/D/1$ queue. As before, let W represent the waiting time in queue experienced by a given customer arriving to the fictitious queue served at rate μ. The *actual* delay experienced by the cell corresponding to the given fictitious customer can be expressed as

$$D = \max(0, W - (B-1)/\mu). \tag{13}$$

Using (4) for the complementary waiting time distribution, the probability that D exceeds a delay bound $D_{\max}$ is given by

$$P(D > D_{\max}) = a(\mu)e^{-b(\mu)D_{\max}}e^{-b(\mu)(B-1)/\mu}. \tag{14}$$

We can obtain an expression for the mean cell delay through the shaper as follows:

$$E[D] \approx \frac{a(\mu)}{b(\mu)} e^{-b(\mu)(B-1)/\mu}. \tag{15}$$

Setting the expected delay equal to the target mean delay $\overline{D}$ determines B as the following function of μ and $\overline{D}$

$$B(\mathcal{C}_S, \mu, \overline{D}) = \frac{\mu}{b(\mu)} \log\left(\frac{a(\mu)}{b(\mu)\overline{D}}\right) + 1. \tag{16}$$

We refer to $B(\mathcal{C}_S, \mu, \overline{D})$ as a function of μ for constant expected delay $\overline{D}$ as a *mean delay constraint curve*.

D. Cost Function

We now develop a cost function, denoted by $c(\lambda_p, \lambda_s, B_s)$, which approximately captures the amount of resource that should be allocated to a cell characterized by the UPC parameters $(\lambda_p, \lambda_s, B_s)$. Consider a source with deterministic periodic on–off sample paths and a random offset. The lengths of the on and off periods are given by

$$T_{\text{on}} = \frac{B_s}{\lambda_p - \lambda_s} \quad \text{and} \quad T_{\text{off}} = \frac{B_s}{\lambda_s} \tag{17}$$

respectively. During the on periods, the source generates cells at the constant rate λ_p and is silent during the off periods. We refer to this source as a *full rate on–off source*. For a homogeneous bufferless multiplexer model, Doshi [17] has shown that the full-rate on–off source yields the highest cell loss probability. Although this is not generally true in the buffered case, our experience with simulations of statistical multiplexing suggest that the full-rate source tends to be a sufficiently conservative model for an arbitrary source characterized by a UPC descriptor.

Consider an on–off source model in which the source alternately transmits at rate r during an on-period and is silent during an off-period. The mean on-period and off-period lengths are given by β^{-1} and α^{-1}, respectively. For mathematical tractability, we shall assume an on–off source model in which the on and off periods are distributed according to Erlang-k distributions. The *effective bandwidth* of a source is defined to be the minimum capacity required to serve the traffic source so as to achieve a specified steady-state cell loss probability ϵ_{mux}, where the multiplexer or switch buffer size is denoted by B_{mux}. Kobayashi and Ren [24] obtain an approximate effective bandwidth result via a diffusion process approximation of a general on–off fluid source. Using the result in [24], the effective bandwidth for our source model can be obtained as

$$\tilde{c}_{\text{eff}} = \frac{r\alpha}{\alpha+\beta} + \frac{r^2\alpha\beta}{(\alpha+\beta)^3}K \tag{18}$$

where

$$K = \frac{\log(1/\epsilon_{mux})}{2kB_{\text{mux}}}. \tag{19}$$

If we set the mean on and off periods as $\beta^{-1} = T_{\text{on}}$ and $\alpha^{-1} = T_{\text{off}}$, respectively, the effective bandwidth expressed in terms of the UPC parameters is obtained from (17) and (18) as follows:

$$\tilde{c}_{\text{eff}} = \lambda_s + K\frac{\lambda_s(\lambda_p - \lambda_s)B_s}{\lambda_p}. \tag{20}$$

This expression has a simple intuitive interpretation: the bandwidth allocated to a UPC-controlled source is the sustainable rate λ_s plus a component due to the burstiness of the source. Using (20), we define the following cost function for a UPC-controlled source:

$$c(\lambda_p, \lambda_s, B_s) = \min\left(\lambda_p, \lambda_s + K\frac{\lambda_s(\lambda_p - \lambda_s)B_s}{\lambda_p}\right) \tag{21}$$

where K is given in (19).

The cost function provides a measure for assessing the tradeoff between the parameters (λ_s, B_s). Observe, from (21), that $c(\lambda_p, \lambda_s, B_s)$ is monotonically increasing in each argument and in the implicit parameter K. As $\lambda_s \to \infty$, the second argument on the right-hand side of (21) grows beyond λ_p and hence we have that $c(\lambda_p, \lambda_s, B_s) \to \lambda_p$. This corresponds to allocation at the peak rate. As $B_s \to 0$, the second argument approaches the sustainable rate λ_s, and we have that $c(\lambda_p, \lambda_s, B_s) \to \lambda_s$. This corresponds to allocation at the sustainable rate. Conversely, as $B_s \to \infty$, we have $c(\lambda_p, \lambda_s, B_s) \to \lambda_p$, or allocation at the peak rate.

The implicit parameter K should be chosen such that $c(\lambda_p, \lambda_s, B_s)$ approximates the bandwidth allocation of the network CAC policy [2]. The limiting behavior of the cost function with respect to K is

$$c(\lambda_p, \lambda_s, B_s) \to \lambda_s \quad \text{as} \quad K \to 0, \tag{22}$$

$$c(\lambda_p, \lambda_s, B_s) \to \lambda_p \quad \text{as} \quad K \to \infty. \tag{23}$$

Recall from (19) that K depends on the multiplexer cell loss probability ϵ_{mux}, the multiplexer buffer size B_{mux}, and the source model burstiness parameter k. For larger values of K, the cost function curve becomes approximately flat, with values near the peak rate λ_p. For smaller values of K, the curve achieves a minimum between the mean and peak rates. As $K \to 0$, this minimum approaches the mean rate.

V. Dynamic UPC Estimation

In this section, we describe a practical methodology for performing real-time UPC estimation and shaping. In a network that allows the user to renegotiate the UPC parameters, the UPC descriptor assigned to a stream can be changed dynamically to adapt to changes in traffic characteristics.

A. Real-Time UPC Estimation

Real-time UPC estimation is based on observations of the cell stream over a time interval T. The basic procedure is outlined as follows.

1) Estimate the mean and peak cell rates.
2) Choose a set of candidate sustainable rates.
3) Obtain the characterization $\hat{\mathcal{C}}_S$ of the stream.
4) For each candidate sustainable rate, compute the minimum bucket size required to meet a user constraint on shaping delay. Finally, choose the candidate rate that minimizes the cost function.

1) Mean and Peak Rates: The peak and mean rates can be estimated from observations of the cell stream over window of length T. The interval of length T is subdivided into M smaller subintervals of equal length $\tau = T/M$. In our experiments with MPEG video sequences (see Section VI), we set τ equal to the frame period of 33 ms. Let A be the total number of cell arrivals during $I(n)$ and let n_i be the number of cell arrivals in the ith subinterval $i = 1, 2, \cdots, M$. The mean rate can be estimated as $\hat{\lambda}_m = A/T$. For some applications, the peak rate may be known *a priori*. Otherwise, the peak rate can be estimated by $\hat{\lambda}_p = \max\{n_i/\tau\colon 1 \le i \le M\}$.

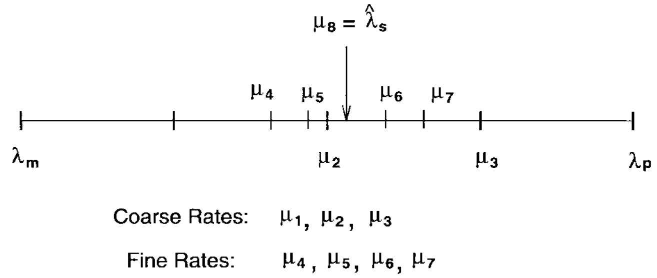


Fig. 1. Example of candidate rate assignment.

2) Choosing Candidate Rates: Candidate rates $\mu_1, \cdots, \mu_N$ are chosen between the mean and peak rates. The simplest approach is to choose N and space the candidate rates at uniform spacing between the mean and peak rates. However, with a more judicious method of assignment, the set of candidate rates can be kept small, while still including rates close to the near optimal sustainable rates. Suppose that a prior estimate of the operating sustainable rate, denoted $\hat{\lambda}_s$, is available. In the absence of prior knowledge, one could assign $\hat{\lambda}_s = (\hat{\lambda}_m + \hat{\lambda}_p)/2$. We assign the candidate rate μ_N equal to $\hat{\lambda}_s$. The remaining rates are grouped into a set of N_c *coarsely spaced rates* and N_f *finely spaced rates*, with $N = N_c + N_f + 1$ and with N_f assumed to be even. The coarse rates $\mu_1, \cdots, \mu_{N_c}$ are assigned to be spaced uniformly over the interval $(\hat{\lambda}_s, \hat{\lambda}_p)$

$$\mu_i = \hat{\lambda}_m + i\Delta_c, \qquad i = 1, \cdots, N_c \tag{24}$$

where the coarse spacing Δ_c is defined as $\Delta_c = (\hat{\lambda}_p - \hat{\lambda}_m)/(N_c + 1)$. The remaining N_f fine rates are assigned in the neighborhood of $\hat{\lambda}_s$ as follows:

$$\mu_{j+N_c} = \hat{\lambda}_s + j\Delta_f, \qquad j = 1, \cdots, \frac{N_f}{2}$$

$$\mu_{j+N_c} = \hat{\lambda}_s - \left(j - \frac{N_f}{2}\right)\Delta_f, \qquad j = \frac{N_f}{2} + 1, \cdots, N_f$$

where $\Delta_f = \Delta_c/(N_f + 1)$. Fig. 1 shows an example of the choice of candidate rates. In this example, $N = 8$, $N_c = 3$, and $N_f = 4$.

3) Traffic Characterization: The ith fictitious queue is implemented using a counter X_i that is decremented at the constant rate μ_i and incremented by one whenever a cell arrives. For each rate μ_i, estimates of the queue parameters $\hat{a}(\mu_i)$ and $\hat{b}(\mu_i)$ are obtained. Each queue is sampled at the arrival epoch of an "arbitrary" cell. This sampling is carried out by skipping a random number N_j of cells between the $(j-1)$st and jth sampled arrivals. Since all of the fictitious queues are sampled at the same arrival epochs, only one random number needs to be generated for all N queues at a sampling epoch. Simple yet robust pseudorandom number generators can be implemented straightforwardly (cf. [25]).

Over an interval of length T, a number of samples, say M, are taken from the queues. At the jth sampling epoch, the following quantities are recorded for each queue: S_j the number of customers in service (so $S_j \in \{0, 1\}$), Q_j the number of customers in queue, and T_j the remaining service

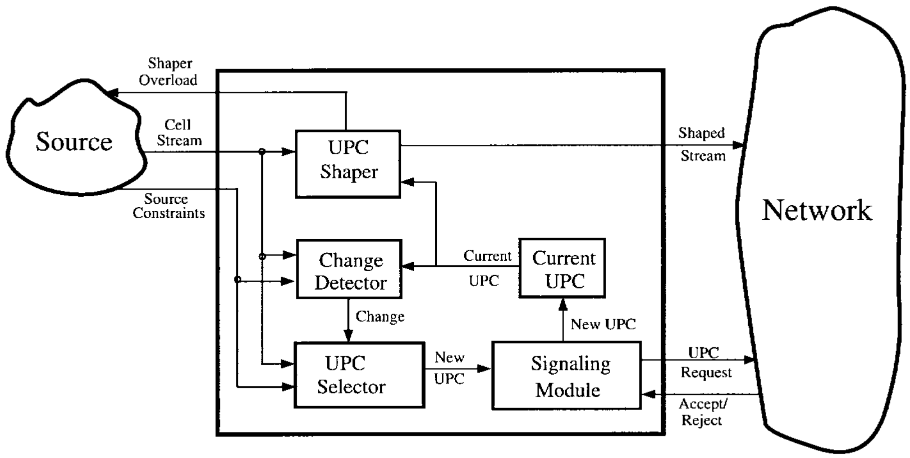


Fig. 2. System block diagram for UPC estimation/shaping module.

time of the customer in service (if one is in service). The following sample means are computed:

$$\hat{a} = \frac{1}{M}\sum_{j=1}^{M} S_j, \quad \hat{q} = \frac{1}{M}\sum_{j=1}^{M} Q_j, \quad \hat{\tau}_r = \frac{1}{\hat{a}M}\sum_{j=1}^{M} T_j. \quad (25)$$

Then, $\hat{b}$ is computed via (11)

$$\hat{b} = \frac{\hat{a}\mu}{\mu\hat{\tau}_r\hat{a} + \hat{q}} \approx \frac{\hat{a}\mu}{\hat{a}/2 + \hat{q}} \quad (26)$$

where $\hat{\tau}_r$ is approximated by $1/2\mu$. The approximate statistical characterization is given by

$$\hat{\mathcal{C}}_S = [\hat{\lambda}_p; \hat{\lambda}_m; (\hat{a}(\mu_i), \hat{b}(\mu_i)), i = 1, \cdots, N]. \quad (27)$$

4) Mapping to a UPC Descriptor: As discussed in Section IV, there are several ways to map the characterization $\mathcal{C}_S$ to a UPC descriptor. Here we focus on the case where the user specifies a maximum mean delay $\overline{D}$. The operating rate index is chosen to minimize the cost function defined in (21)

$$i^* = \arg \min_{1 \le i \le N} c(\hat{\lambda}_p, \mu_i, B_s(\hat{\mathcal{C}}_S, \mu_i, \overline{D})) \quad (28)$$

where $B_s(\hat{\mathcal{C}}_S, \mu, \overline{D})$ is defined in (16) and represents the bucket size required to achieve a mean delay of $\overline{D}$ at a leak rate μ. Then the operating UPC descriptor is assigned as follows:

$$\lambda_p = \hat{\lambda}_p, \lambda_s = \mu_{i^*}, \quad B_s = B_s(\hat{\mathcal{C}}_S, \lambda_s, \overline{D}). \quad (29)$$

We remark that if the estimated mean and peak rates are close in value, the stream can be treated as a constant bit rate (CBR) stream. In this case, the UPC descriptor can simply be assigned according to the peak rate, i.e.,

$$\lambda_p = \hat{\lambda}_p, \quad \lambda_s = \hat{\lambda}_p, \quad B_s = 1.$$

B. Dynamic Renegotiation

Some applications such as video delivery, interactive multimedia sessions, etc., cannot be adequately characterized by a static set of parameters that are expected to hold for the entire session. The ability to renegotiate parameters during a session may be a viable way of efficiently supporting VBR traffic (cf. [15]) with real-time QoS constraints. Fig. 2 depicts a block diagram of a system for UPC estimation and shaping with renegotiation. The UPC estimator generates a new UPC descriptor once every T seconds. The change detector determines whether or not there has been a significant violation of the user constraints with the current UPC parameters. A decision on whether or not to retain the current UPC parameters is made once every T seconds and is indicated to the signaling module. In case a change is detected, the signaling module updates the current UPC with the UPC descriptor estimated in the current observation interval and issues a UPC renegotiation request to the network.

The network determines whether or not the renegotiation request can be accommodated with the current available resources. Due to the processing overhead incurred by each renegotiation request, the frequency of renegotiations for each user must be limited in some manner (cf. [3]). In case a UPC request is rejected by the network, the current UPC is reset with the current negotiated UPC. This may result in an overload of the UPC shaper. In response to shaper overload, the source throttles its transmission until the overload condition subsides. For example, sources such as video or still-image transmission may increase a quantization parameter to reduce the output cell stream during the overload condition.

1) Sequential UPC Estimation: The UPC estimation procedure can be extended to generate a new UPC descriptor once every T seconds. Let us subdivide the time axis into intervals of length T, denoted by $I(n) = ((n-1)T, nT]$, $n = 1, 2, \cdots$. Let $r_m(n)$ and $r_p(n)$, respectively, denote estimates of the mean and peak rates taken during interval $I(n)$ according to the procedure discussed earlier. Smoother estimates of the mean and peak rates over $I(n)$ can be obtained via exponential averaging as follows:

$$\hat{\lambda}_m(n) = \alpha_m r_m(n) + (1 - \alpha_m)\hat{\lambda}_m(n-1) \quad (30)$$

$$\hat{\lambda}_p(n) = \alpha_p r_p(n) + (1 - \alpha_p)\hat{\lambda}_p(n-1) \quad (31)$$

where $0 < \alpha_m, \alpha_p < 1$.

During $I(n)$, a set of candidate rates $\mu_1(n), \cdots, \mu_N(n)$ are assigned values between the peak and mean rate estimates $\hat{\lambda}_p(n-1)$ and $\hat{\lambda}_m(n-1)$. The fine rates are assigned in the neighborhood of the previous sustainable rate estimate $\hat{\lambda}_s(n-1)$. The fictitious queues are then used to obtain a traffic characterization $\mathcal{C}_S(n)$. Finally, the characterization $\mathcal{C}_S(n)$ is mapped to a UPC descriptor $U(n) = [\lambda_p(n), \lambda_s(n), B_s(n)]$ according to the procedures discussed earlier.

2) UPC Change Detection: At the end of interval $I(2)$, the user negotiates the UPC descriptor $U(2)$ as part of the connection setup procedure. We denote the negotiated UPC descriptor at time $n \geq 2$ as follows:

$$U_{\text{neg}}(n) = [\lambda_p^{\text{neg}}(n), \lambda_s^{\text{neg}}(n), B_s^{\text{neg}}(n)].$$

If the connection is admitted, the negotiated UPC descriptor is assigned as $U_{\text{neg}}(2) = U(2)$. For $n > 2$, a decision is made as to whether or not $U_{\text{neg}}(n)$ should be renegotiated. If the renegotiation request is made and granted by the network, $U(n)$ is assigned as the new negotiated UPC. Otherwise, the negotiated UPC remains the same, i.e.,

$$U_{\text{neg}}(n) = \begin{cases} U(n), & \text{if renegotiation is successful} \\ U_{\text{neg}}(n-1), & \text{otherwise.} \end{cases}$$

Since the UPC estimation scheme generates a new UPC descriptor once every T seconds, T is the minimum renegotiation interval from the user's standpoint.

The UPC descriptor should be renegotiated if the traffic characteristics change such that the current UPC incurs excessive delay for the cell stream. In this case, the new UPC descriptor will impose a higher cost to the user than the current negotiated UPC. Suppose that the user specifies a target average delay $\overline{D}$. The renegotiation decision can be made on the basis of the measured peak rate and the estimated delay for a fictitious queue served at a constant rate, λ_s^{neg}. For M trials taken over an observation window of length T, the mean cell delay $\hat{d}$ can be estimated as

$$\hat{d} = \frac{1}{M} \sum_{i=1}^{M} h_s(Q_i) \left[\frac{Q_i - B_s^{\text{neg}}}{\tilde{\lambda}_s} + T_i \right]$$

where $h_s(t) = u(t - B_s^{\text{neg}})$, with $u(t)$ being the unit step function. Here, Q_i and T_i are, respectively, the number in queue and the remaining service time of the customer in service (if any) seen by the arbitrary arrival. Under moderate-to-heavy utilization, T_i can be approximated by $1/2\lambda_s^{\text{neg}}$.

Given the mean delay estimate $\hat{d}$ and the peak rate estimate $\hat{\lambda}_p$, the following criterion for renegotiating for more resource can be used:

$$\hat{d} > \overline{D} + \delta_h \quad \text{or} \quad \hat{\lambda}_p > \lambda_p^{\text{neg}} + r_h.$$

If the measured mean delay exceeds the target delay $\overline{D}$ by more than an amount $\delta_h > 0$, then a new UPC descriptor should be negotiated with the network. The quantity δ_h should be large enough to account for errors in measuring $\hat{d}$. A larger value of δ_h may lead to fewer renegotiations but could result in a higher degree of shaping. Similar considerations apply to the quantity $r_h > 0$. The quantity r_h can be expressed as a percentage of λ_p^{neg}. For example, a UPC renegotiation request could be issued whenever the measured peak rate exceeds the negotiated peak rate by 10%, i.e., $r_h = 0.1\lambda_p^{\text{neg}}$.

A renegotiated service can work only if a given user can also give up its allocated network resources from time to time, allowing other users to make use of these resources. The user's incentive is a lower cost for network usage. If the source bit-rate decreases, the user could save significantly on cost by renegotiating its UPC descriptor. A simple criterion for renegotiating for less resource is as follows:

$$\hat{d} < \overline{D} - \delta_l \quad \text{and} \quad \hat{\lambda}_p < \lambda_p^{\text{neg}} - r_l.$$

Larger values of δ_l and r_l may lead to fewer renegotiations but could result in more conservative negotiated UPC descriptors.

VI. Experimental Results

A. Characterization of Video

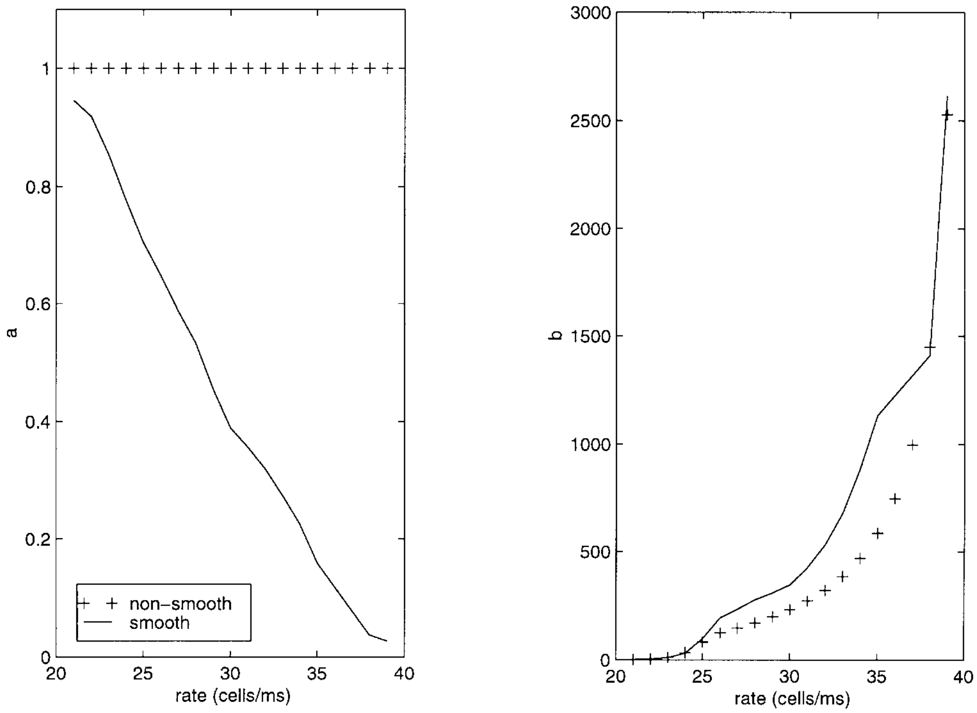
In the MPEG standard for video compression [26], video is transmitted as a sequence of frames. For transmission over an ATM network, each MPEG frame must be segmented into ATM cells. We consider two modes of cell transmission over a frame interval. In *nonsmooth mode*, the cells are transmitted at a constant rate λ_p starting at the beginning of the frame interval until the frame is fully transmitted. Then the source is silent until the beginning of the next frame interval. This results in an on–off type stream with correlated on and off periods. In *smooth mode*, the cells for the ith frame are transmitted at a constant rate $f(i)/\tau$ over the frame interval, where $f(i)$ is the total number of cells in the ith frame and τ is the interframe period. Thus, buffering is done to smooth out the transmission of cells of each frame over the corresponding frame interval.

Fig. 3 shows the characterization $\hat{\mathcal{C}}_S$ obtained empirically for an MPEG video stream called "mobi" under the two transmission modes. For the purposes of our simulation experiments, a transform-expand-sample (TES) model [27] of the "mobi" sequence was created to match the statistical characteristics of the original MPEG sequence. Using the TES model, a "mobi" stream of arbitrary length can be generated for simulation purposes. For this stream, the empirical mean and peak rates were determined to be 20 and 40.5 cells/ms, respectively. The two streams were offered to 19 queues running in parallel at the rates $\mu = 21, 22, \cdots, 39$ (cells/ms) and the corresponding values for $(a(\mu), b(\mu))$ were obtained according to the procedure described above. In the nonsmooth mode, $a(\mu) \approx 1$ for all values of μ.

In contrast, for smooth mode, $a(\mu)$ monotonically decreases as μ increases from λ_m to λ_p. Observe that the $b(\mu)$ curves for the nonsmooth and smooth modes are close for μ near to the extremes λ_m and λ_p. For values of μ closer to the center of the interval (λ_m, λ_p), the $b(\mu)$ curve for smooth mode lies above the curve for nonsmooth mode. Thus, the characterization $\hat{\mathcal{C}}_S$ captures the decrease in traffic burstiness due to intraframe smoothing.

B. UPC-Based Characterization

Fig. 4 shows plots of the shaping probability constraint curves $B(\mathcal{C}_S, \mu, \epsilon)$ for the "mobi" sequence transmitted in nonsmooth mode for several values of ϵ. The corresponding curves for smooth mode transmission are shown in Fig. 5. Observe that for a fixed value of shaping probability, the curve for nonsmooth mode lies below the corresponding curve for smooth mode. As we would expect, for the same leak rate and shaping probability requirement, the minimum bucket size in



(a) (b)

Fig. 3. (a) a and (b) b parameters for "mobi."

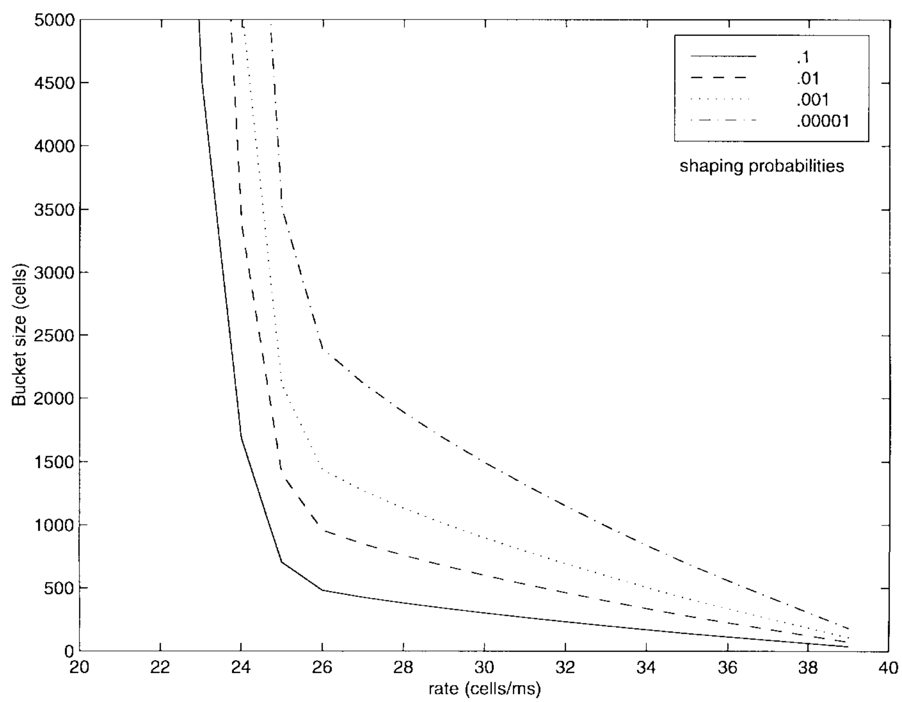


Fig. 4. Shaping probability curves for "mobi": nonsmooth mode.

the case of the smooth mode is always less than that for the nonsmooth mode.

In practice, a shaping probability in the neighborhood of 10^{-2} or 10^{-1} typically corresponds to relatively small mean cell delay through the shaper (see Section IV-C). For fixed shaping probability, the constraint curve provides, for each rate μ, $\lambda_m < \mu < \lambda_p$, the corresponding (minimum) bucket size B required to achieve a shaping probability less than ϵ. As ϵ increases, i.e., the shaping constraint is relaxed, the required B for a given μ decreases. Note that the curves for the smooth mode transmission in Fig. 5 lie below the corresponding curves for the nonsmooth mode in Fig. 4. As one would expect, presmoothing of a cell stream results in a stream characterized by smaller UPC values.

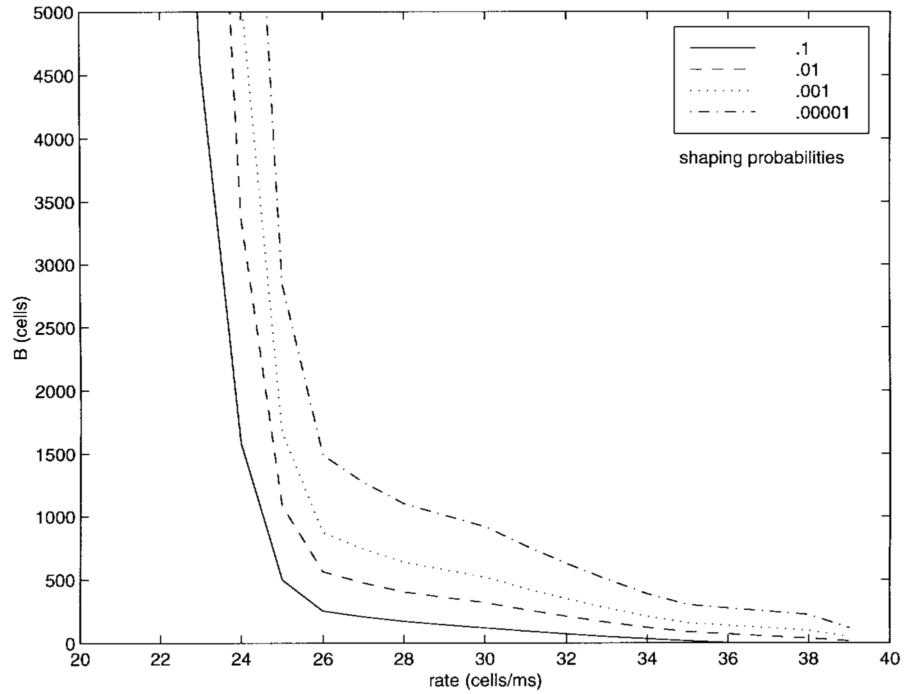


Fig. 5. Shaping probability curves for "mobi": smooth mode.

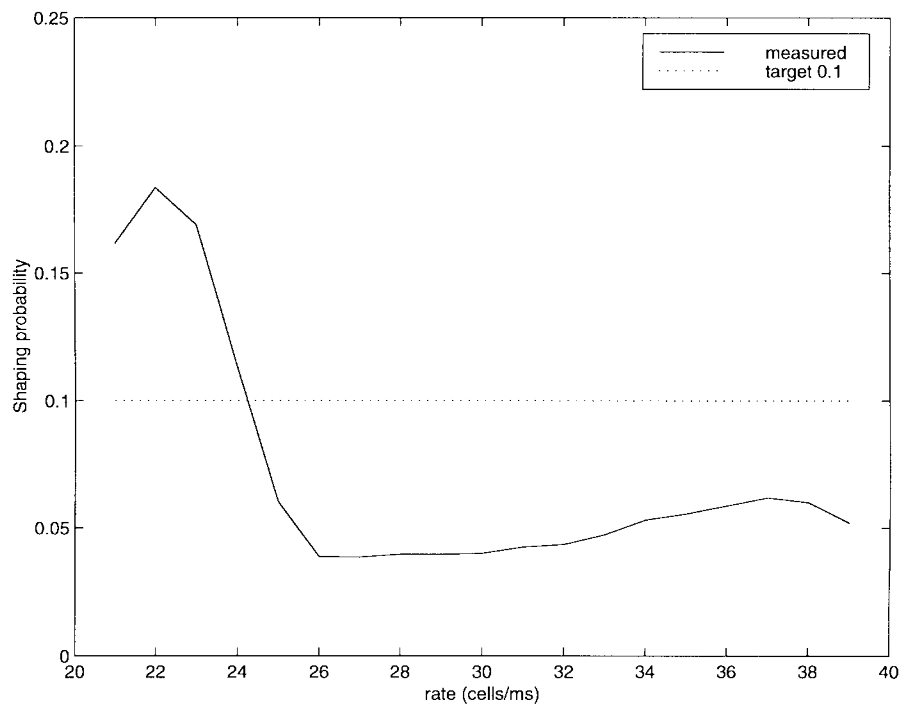


Fig. 6. Measured shaping probability for 0.1 target.

To test the accuracy of the UPC-based traffic characterization, we applied the "mobi" stream to leaky buckets with parameter values chosen from the shaping probability constraint curves and estimated the shaping probabilities experienced by the cell stream. We simulated hundreds of hours of real-time traffic from the "mobi" stream characteristics generated using the TES model. Figs. 6–8 show the measured shaping probabilities for $\epsilon = 0.1$, 0.01, 0.001, respectively. In these figures, the target shaping probabilities are met, except for rates near the mean arrival rate λ_m. In obtaining the statistical characterization $\hat{\mathcal{C}}_S$, fictitious queues with rates near the mean rate may not be stable over the observation interval. However, the operating sustainable rate is generally chosen away from the mean rate.

As ϵ decreases, the computed bucket sizes tend to overestimate the minimum required bucket sizes. For $\epsilon = 10^{-3}$, the

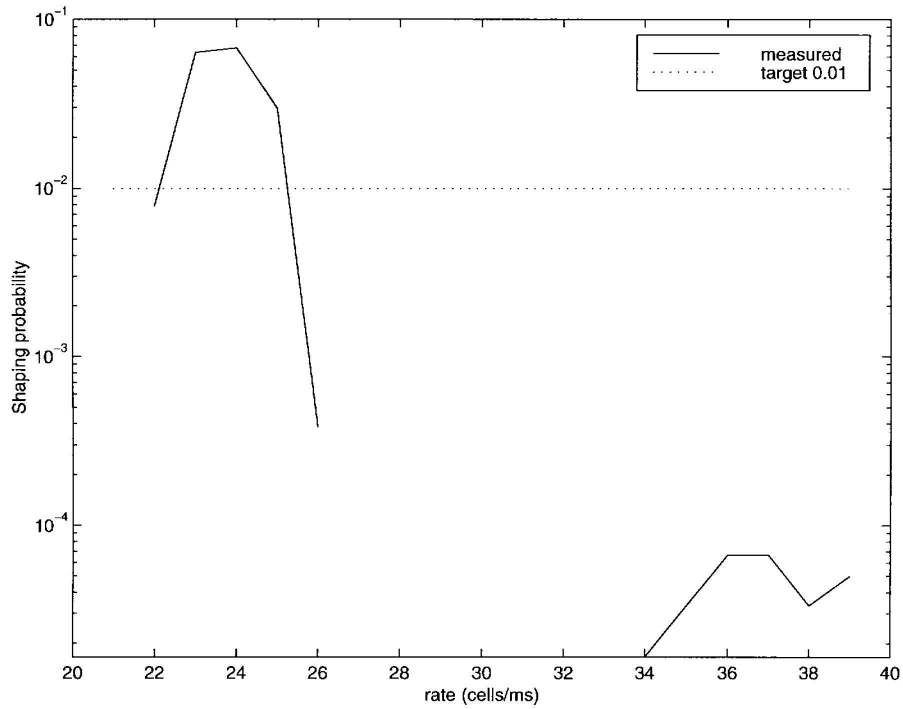


Fig. 7. Measured shaping probability for 0.01 target.

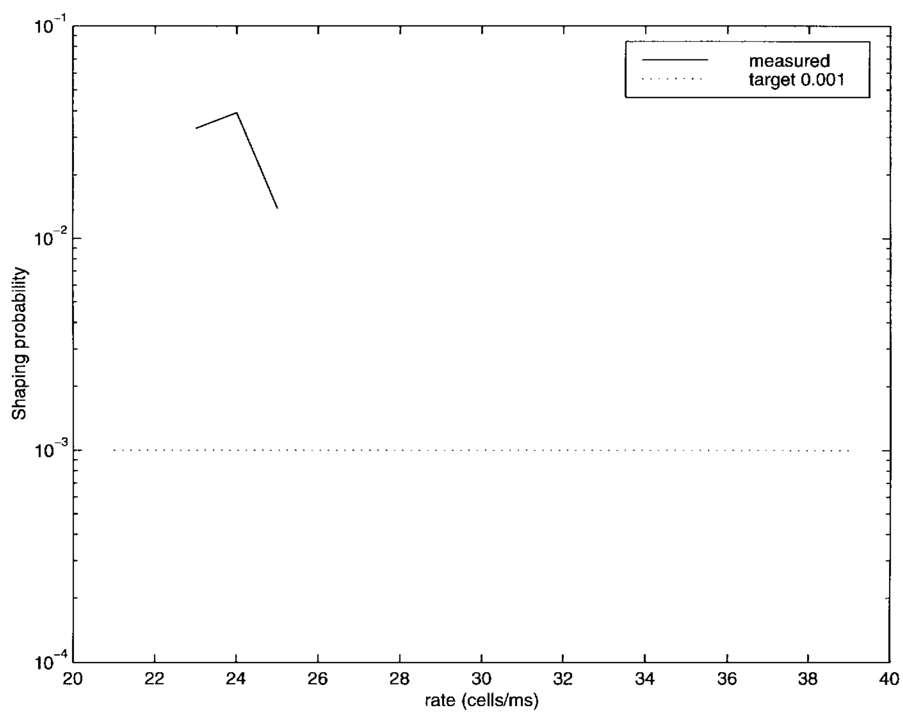


Fig. 8. Measured shaping probability for 0.001 target.

measured shaping probabilities were found to be negligible (i.e., $<10^{-7}$), except for rates near the mean. In practice, to achieve near negligible shaping probability with our method, it suffices to perform the mapping to the UPC descriptor using a value for ϵ on the order of 10^{-3}–10^{-4}; the use of smaller values of ϵ may lead to overly conservative UPC values.

Fig. 9(a) displays the mean delay constraint curve for a constant mean delay of 5 ms. In Fig. 9(b), the probabilistic delay bound curves are shown for several values of $D_{\max}$. Fig. 10 plots a graph of the measured mean delay when the traffic stream is offered to leaky buckets with leak rate and bucket size parameters taken from Fig. 9(a). Observe that the mean delay constraint of 5 ms is met for rate values away from the mean rate λ_m. For rates closer to the mean rate, the corresponding bucket sizes from Fig. 9 are insufficiently large to keep the mean delay near 5 ms. This was also observed

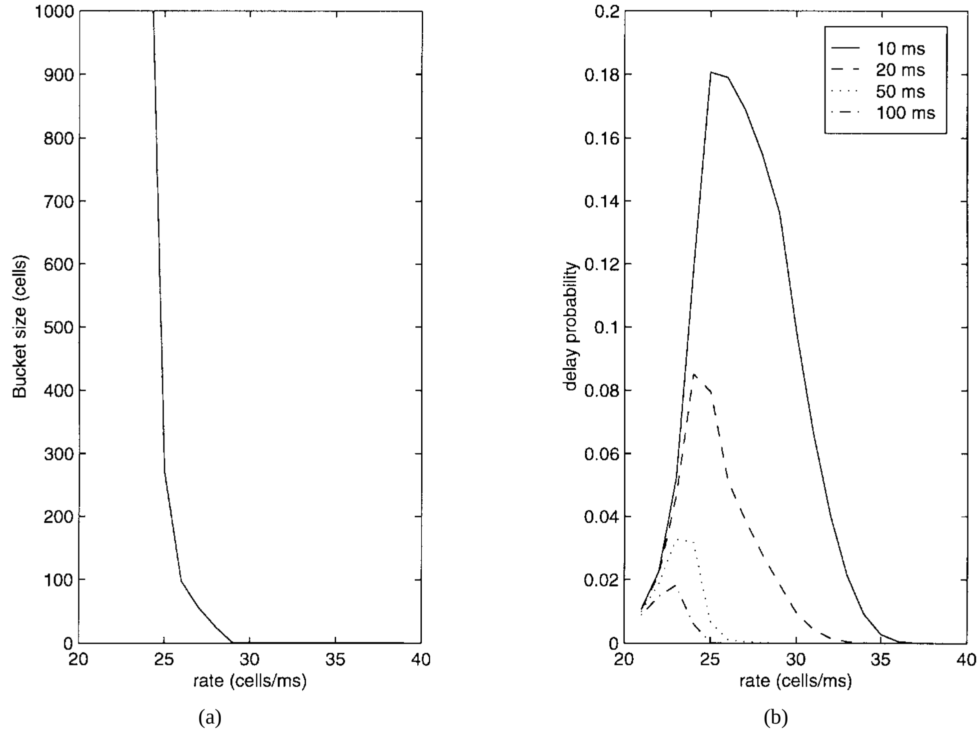


Fig. 9. (a) Mean delay constraint curve and (b) $P(D > D_{\max})$.

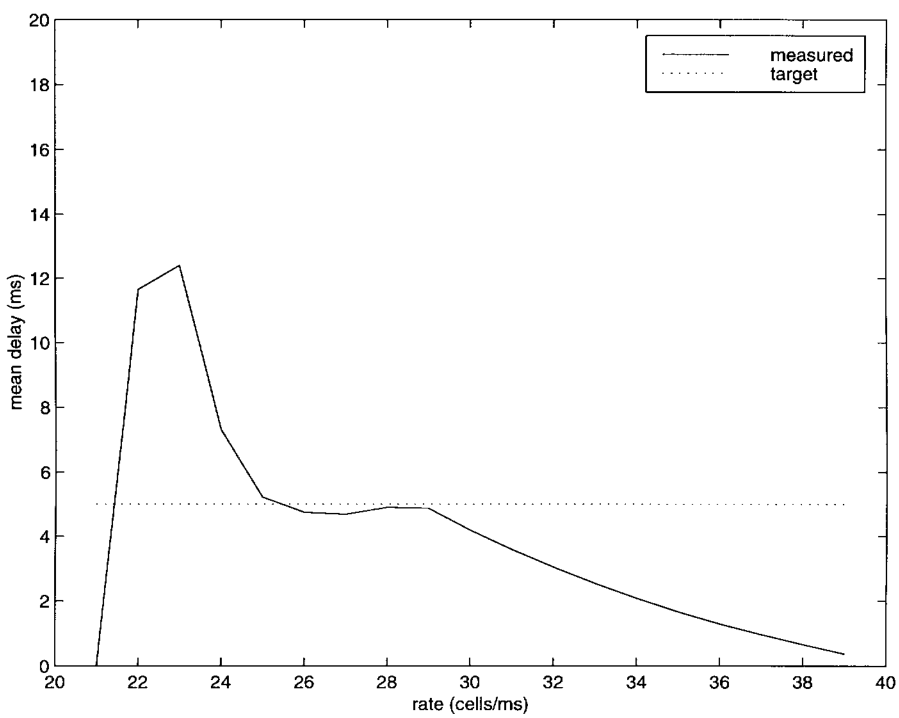


Fig. 10. Measured mean delay with leaky bucket parameters chosen for 5–ms target mean delay.

in Fig. 6, where the leaky bucket parameters were designed to achieve a shaping probability of 0.1. Comparing Fig. 9(a) with Fig. 4, it is clear that the mean delay constraint of 5 ms corresponds to a shaping probability larger than 0.1. For comparison with Fig. 10, Fig. 11 plots the measured mean delay when the target mean delay is 10 ms. Observe that as the target mean delay is increased the minimum bucket size to achieve the target delay reduces to the minimum value 1 for smaller leak rate values. In this case, the actual mean delay will lie below the target mean delay for a larger range of rates near the peak rate.

Fig. 12(a) plots the mean delay constraint curve as a function of μ for the "mobi" sequence with mean delay $\overline{D} = 5$ ms. Fig. 12(b) plots the cost function curves

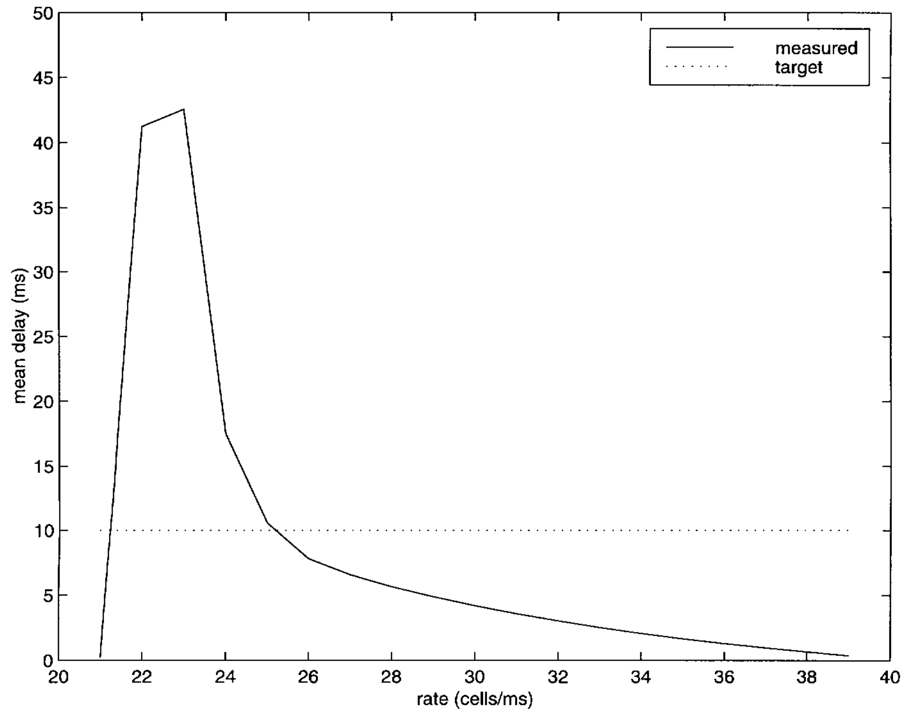


Fig. 11. Measured mean delay with leaky bucket parameters chosen for the 10-ms target mean delay.

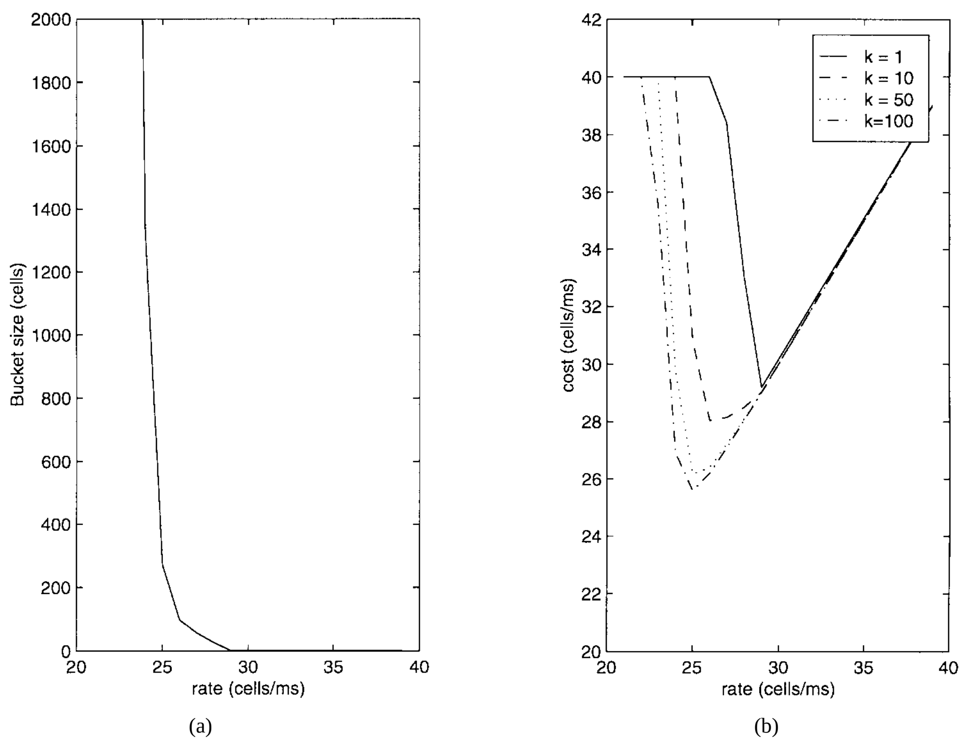


Fig. 12. (a) Mean delay constraint and (b) cost curves.

$c(\lambda_p, \mu, B(\mathcal{C}_S, \mu, \overline{D}))$ for various values of the parameter k. Note that $k = 1$ corresponds to an exponential on–off source model. As k increases, the variability in the on and off periods decreases. Here we have set $B_{\rm mux} = 500$ (cells) and $\epsilon_{\rm mux} = 10^{-5}$, which are typical values for an ATM multiplexer. Observe that each of the cost function curves achieves a minimum giving a suitable choice for the operating sustainable rate λ_s. As k increases (K decreases), the minimum shifts closer to the mean rate λ_m.

We applied our procedure for selecting UPC parameters to the "mobi" sequence with observations taken over time windows of four different lengths: 600, 60, 5, and 2 s. The proximity of the four curves in each of Fig. 13(a)–(d) indicate that the statistical characterization developed in Section III-B

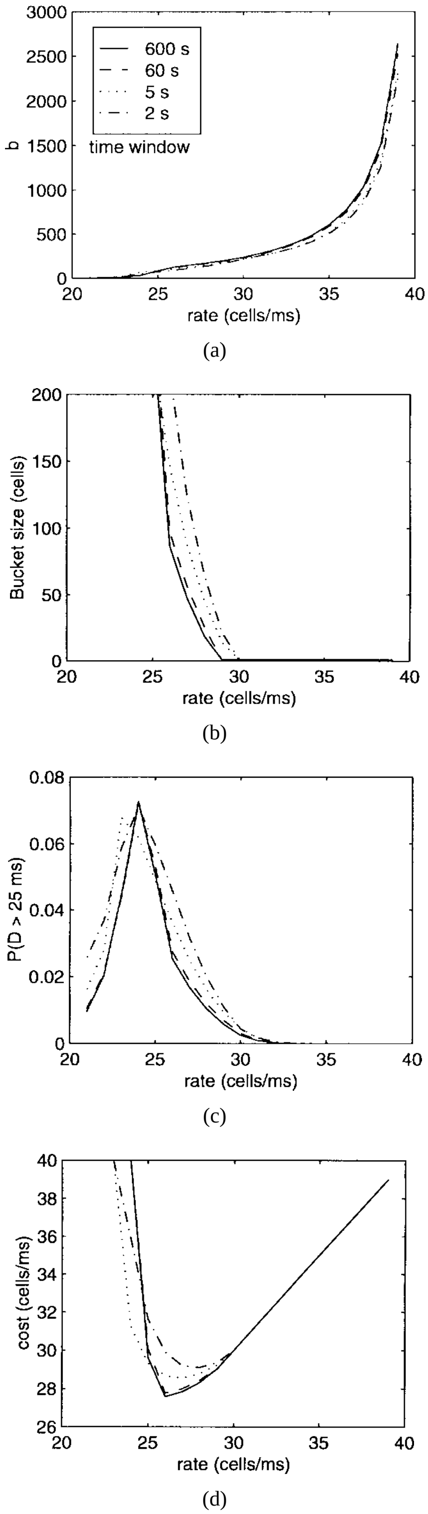


Fig. 13. Curves for UPC estimation: "mobi." (a) b-exponent. (b) Mean delay constraint $E[D] = 5$ ms. (c) Probabilistic delay constraint. (d) Cost function.

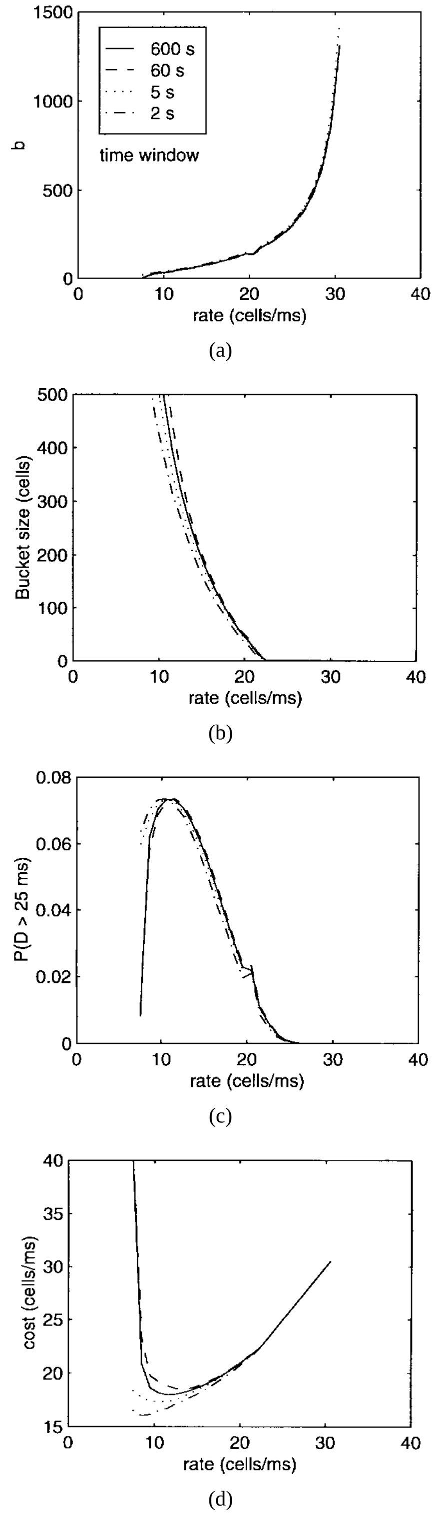


Fig. 14. Curves for UPC estimation: "flower." (a) b-exponent. (b) Mean delay constraint $E[D] = 5$ ms. (c) Probabilistic delay constraint. (d) Cost function.

is quite robust to the length of the observation window. In Fig. 13(a), the exponent $b(\mu)$ is plotted for $\lambda_m < \mu < \lambda_p$. Fig. 13(b) shows the mean delay constraint curves for an expected delay of $\overline{D} = 5$ ms. Fig. 13(c) shows the curves for $P(D > 25 \text{ ms})$ corresponding to the expected delay constraint curves. Fig. 13(d) shows the cost function curves with the parameter K in (19) set to 2×10^{-3}.

Fig. 14 shows analogous results for another MPEG sequence, "flower," transmitted in nonsmooth mode. This sequence comes from a video clip of a flower garden scene which has significantly less activity than the "mobi" clip. The mean and peak rates for "flower" are approximately 7.35 cells/ms and 32.8 cells/ms, respectively. In each of Fig. 14(a)–(d), the curves show good agreement for different window lengths.

3) Dynamic UPC Renegotiation: To demonstrate our algorithms for combined dynamic UPC estimation and renegotiation, we interleaved the "mobi" and "flower" MPEG sequences. The interleaved sequence consists of the first 10 s of "mobi" followed by the first 10 s of "flower" followed by the next 10 s of "mobi," etc. We applied our algorithms to the interleaved sequence with parameters specified as follows:

$$\overline{D} = 5 \text{ ms}, \; D_{\max} = 25 \text{ ms}, \; \epsilon_D = 0.05$$
$$\delta_h = 25 \text{ ms}, \; \delta_l = 2 \text{ ms}$$
$$r_h = 3 \text{ cells/ms}, \; r_l = 5 \text{ cells/ms}.$$

Here, the UPC parameters are chosen to minimize the cost function with $K = 2 \times 10^{-3}$ subject to the constraints $P(D > D_{\max}) \leq \epsilon_D$ and $E[D] \leq \overline{D}$.

Fig. 15 displays the results over a time period of 50 s and with an observation window length of $T = 1.25$ s. Fig. 15(a) plots the sequences of the *estimated* peak, sustainable and mean rates, $\{\hat{\lambda}_p(n)\}$, $\{\hat{\lambda}_s(n)\}$, and $\{\hat{\lambda}_m(n)\}$, respectively. Fig. 15(b) plots the sequence of estimated bucket sizes $\{\hat{B}_s(n)\}$ corresponding to the sustainable rate sequence for the given delay requirements and cost function. Fig. 15(c) shows the sequences of negotiated peak and sustainable rates, $\{\lambda_p^{neg}(n)\}$ and $\{\lambda_s^{neg}(n)\}$, respectively. Fig. 15(d) plots the sequence of negotiated bucket sizes, $\{B_s^{neg}(n)\}$. Observe that the sequence of negotiated UPC parameters remains approximately constant over the 10-s windows corresponding to either the "mobi" or "flower" sequences. For this sequence, a renegotiation request would be issued to the network about once every 10 s. For comparison purposes, Fig. 16 displays the results with the same parameters as in Fig. 15, except that the observation window length is set to $T = 2.5$ s.

VII. Conclusions

We have developed a real-time estimation scheme for characterizing an arbitrary ATM cell stream in terms of UPC descriptors that can be negotiated and renegotiated dynamically. The problem is important because users of ATM networks need some way of determining UPC parameters to negotiate for their traffic streams. We introduced an approximate statistical characterization of the cell stream that maps naturally to a UPC descriptor. The characterization does not assume a particular source model for the cell stream. The choice of UPC descriptor optimizes a cost function subject to user-defined constraints on the degree of UPC shaping that can be tolerated. The cost function approximates the network resource allocated to a stream characterized by a given UPC descriptor.

Our experimental results indicate that the UPC descriptors selected by our scheme are relatively robust with respect to the observation window size. The UPC estimation procedure can be applied off-line, for stored video or data applications, as well as on-line, for real-time applications such as live video. For applications with relatively stationary traffic characteristics, a single UPC descriptor may be adequate to characterize the user's traffic stream. Applications which have significant slow time-scale traffic variation are better characterized by a sequence of UPC descriptors. By allowing the user to give up resource when it is not needed, the network should be able to achieve higher utilization under a UPC renegotiation scheme. The question of how the network can best support renegotiation is an important issue to be investigated further.

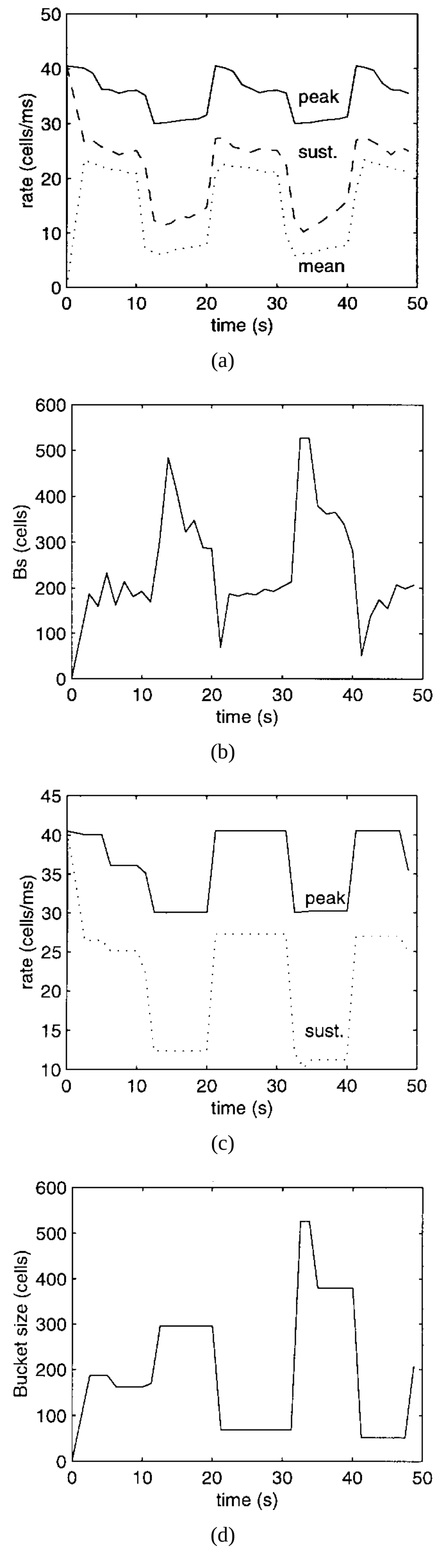


Fig. 15. Dynamic UPC with $T = 1.25$ s. (a) Estimated rates. (b) Estimated bucket size. (c) Negotiated rates. (d) Negotiated bucket size.

Our scheme for UPC estimation and renegotiation is intended to be applied on the user side, e.g., as part of a network interface card handling a moderate number of streams. We

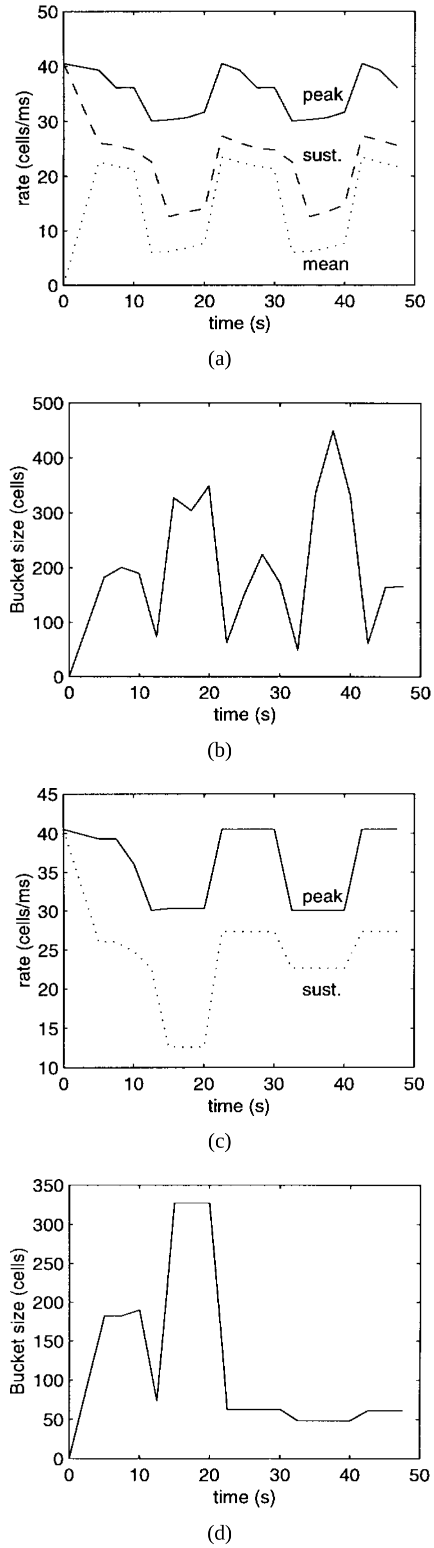


Fig. 16. Dynamic UPC with $T = 2.5$ s. (a) Estimated rates. (b) Estimated bucket size. (c) Negotiated rates. (d) Negotiated bucket size.

assume that the user can tolerate shaping of its cell stream to a certain degree. The UPC estimation procedure is relatively simple to implement, yet provides a good characterization of a stream in terms of UPC parameters.

In principle, it is also possible to apply the scheme on the network side, e.g., as an edge switch or access multiplexer function. In this case, the edge switch handles the selection and renegotiation of UPC parameters on behalf of the user. The switch chooses UPC parameters that result in very low violation probability and informs the user when the UPC parameters are changed. Since the switch may handle on the order of thousands of streams, the per-stream processing overhead is an important issue. At the expense of suboptimality in choosing the sustainable rate giving the minimum cost, the implementation complexity of our scheme can be reduced significantly by employing a smaller number of fictitious queue counters per stream. Fast and efficient UPC-based traffic characterization for a large set of streams is a potentially important research area requiring further investigation.

REFERENCES

[1] ATM Forum Technical Committee, *Traffic Management Specification, Version 4.0,* ATM Forum, 1996.

[2] G. Ramamurthy and Q. Ren, "Multi-class connection admission control policy for high speed ATM switches," in *Proc. IEEE INFOCOM'97,* Kobe, Japan, Apr. 1997.

[3] B. L. Mark and G. Ramamurthy, "Joint source-channel control for real-time VBR over ATM via dynamic UPC renegotiation," in *Proc. IEEE Globecom'96,* London, U.K., Nov. 1996, pp. 1726–1731.

[4] J. S. Turner, "New directions in communications (or which way to the information age?)," *IEEE Commun. Mag.,* pp. 8–15, Oct. 1986.

[5] A. I. Elwalid and D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks," *IEEE/ACM Trans. Networking,* vol. 1, pp. 323–329, June 1993.

[6] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Trans. Inform. Theory,* vol. 37, pp. 114–131, Jan. 1991.

[7] C. Chang, "Stability, queue length and delay of deterministic and stochastic queueing networks," *IEEE Trans. Automat. Contr.,* vol. 39, pp. 913–931, May 1994.

[8] S. Low and P. Varaiya, "A simple theory of traffic and resource allocation in ATM," in *Proc. IEEE Globecom'91,* Dec. 1991, pp. 1633–1637.

[9] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Networking,* vol. 1, pp. 344–357, June 1993.

[10] B. L. Mark and G. Ramamurthy, "UPC-based traffic descriptors for ATM: How to determine, interpret and use them," *Telecommun. Syst.,* vol. 5, pp. 109–122, 1996.

[11] S. Chong and S. Li, "(σ, ρ)-characterization based connection control for guaranteed services in high speed networks," in *Proc. IEEE INFOCOM'95,* Boston, MA, Apr. 1995, pp. 835–844.

[12] O. Yaron and M. Sidi, "Performance and stability of communication networks via robust exponential bounds," *IEEE/ACM Trans. Networking,* vol. 1, pp. 372–385, June 1993.

[13] T. Tedijanto and L. Gün, "Effectiveness of dynamic bandwidth management in ATM networks," in *Proc. INFOCOM'93,* San Francisco, CA, Mar. 1993, pp. 358–367.

[14] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A simple and efficient service for multiple time-scale traffic," in *Proc. ACM SigCom'95,* Boston, MA, Aug. 1995, pp. 219–230.

[15] D. Reininger, G. Ramamurthy, and D. Raychaudhuri, "VBR MPEG video coding with dynamic bandwidth renegotiation," in *Proc. ICC'95,* Seattle, WA, June 1995, pp. 1773–1777.

[16] A. I. Elwalid and D. Mitra, "Analysis and design of rate-based congestion control of high speed networks—I: Stochastic fluid models, access regulation," *Queueing Syst.,* vol. 9, pp. 29–63, 1991.

[17] B. T. Doshi, "Deterministic rule based traffic descriptors for broadband ISDN: Worst case behavior and connection acceptance control," in *Proc. IEEE GlobeCom'93,* Dec. 1993, pp. 1759–1764.

[18] A. E. Eckberg, "Approximations for bursty and smoothed arrival delays based on generalized peakedness," in *Proc. 11th Int. Teletraffic Congress,* Kyoto, Japan, 1985.

[19] J. Abate, G. L. Choudhury, and W. Whitt, "Asymptotics for steady-state tail probabilities in structured Markov queueing models," *Stochastic Models,* vol. 10, pp. 99–143, 1994.

[20] P. W. Glynn and W. Whitt, "Logarithmic asymptotics for steady-state tail probabilities in a single-server queue," in *Studies in Applied*

Probability, Papers in Honor of Lajos Takaćs, J. Galambos and J. Gani, Eds., Applied Probability Trust, 1994, pp. 131–156.
[21] D. P. Heyman and T. V. Lakshman, "What are the implications of long-range dependence for VBR-video traffic engineering?" *IEEE/ACM Trans. Networking*, vol. 4, pp. 301–317, June 1996.
[22] G. L. Choudhury, D. M. Lucantoni, and W. Whitt, "Squeezing the most out of ATM," *IEEE Trans. Commun.*, vol. 44, pp. 203–217, Feb. 1996.
[23] W. Whitt, "Tail probabilities with statistical multiplexing and effective bandwidths in multi-class queues," *Telecommun. Syst.*, vol. 2, pp. 71–107, 1993.
[24] H. Kobayashi and Q. Ren, "A diffusion approximation analysis of an ATM statistical multiplexer with multiple types of traffic, Part I: Equilibrium state solutions," in *Proc. 1993 IEEE Int. Conf. on Communications* Geneva, Switzerland, May 1993, vol. 2, pp. 1047–1053.
[25] D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms, Second Addition.* Reading, MA: Addison-Wesley, 1981.
[26] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *CACM*, vol. 34, pp. 46–58, Apr. 1991.
[27] B. Melamed, "TES: A class of methods for generating autocorrelated uniform variates," *ORSA J. Comput.*, vol. 3, no. 4, pp. 317–329, 1991.

Brian L. Mark (M'91) received the B.A.Sc. degree in computer engineering with an option in mathematics from the University of Waterloo in 1991 and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1995.

In 1995, he joined the C&C Research Laboratories, NEC USA, in Princeton, where he is a Research Staff Member in the Advanced Switching Design and Analysis Department. His current research interests are in broadband network architecture and traffic management.

Gopalakrishnan Ramamurthy (SM'95) received the Masters degree in electrical communication engineering from the Indian Institute of Science and the Ph.D. degree in electrical engineering from the University of Aston, Birmingham, U.K.

He was with the Indian Telephone Industries, Bangalore, India, between 1974 and 1980. He was a member of the Technical Staff at AT&T Bell Laboratories between 1984 and 1991. In 1991, he joined the C&C Research Laboratories, NEC USA, in Princeton, NJ, where he heads the Advanced Switching Design and Analysis Department. His current interests are in broadband network architecture and control, design of large-capacity multiclass switching systems, multiprotocol switching, multiclass scheduling, resource and QoS management.