A Recursive Estimator of Worst-Case Burstiness

Shahrokh Valaee, *Member, IEEE*

***Abstract*—The leaky-bucket regulator has several potential roles in the operation of future transport networks; among them, the bounding of possible source trajectories in implementations of worst-case approaches to network design. It seems plausible that there will be applications whose specific traffic characteristics are known *a priori* neither to the user nor to the network; in such cases, a *recursive* algorithm for setting the leaky-bucket parameters may prove useful. We devise such an algorithm here. The leaky-bucket parameters are computed recursively over a limited period of observation of the source behavior. We provide an explicit characterization of the dynamics of the estimator, and the results of a simulation study of performance in the case of real source trajectories.**

***Index Terms*—ATM, burstiness curve, deterministic source modeling, leaky bucket, reflection mapping, regulator.**

I. Introduction

CALL SETUP in a connection-oriented network is initiated by a source requesting a certain allotment of buffers and bandwidth. Depending on the availability of such resources, the call will be established and data will flow from source to destination. Quantifying the availability of network resources, estimating the load imposed on the network by a particular connection, and ultimately dimensioning the network so that the incidence of rejected requests for connection is acceptably small, all depend upon the characterization of the traffic streams entering the network. There are two broad approaches to the characterization problem. One, forming the basis of what we call *average-case design*, begins with *statistical* source models and expresses performance in terms of means, variances, and quantiles of appropriate probability distribution functions. The other views the source as moving arbitrarily over a range of behaviors circumscribed by a wholly deterministic constraint. It is occasionally possible in this connection to identify a particular behavior which in the class of admissible behaviors is least favorable, and to design the network assuming that the sources are all least favorable. We refer to the corresponding methodology as *worst-case design*.

The literature on worst-case design is smaller and more recent than that dealing with average-case or statistical design. It includes, for example, [1]–[4]. The main results describe the relationship among worst-case network performance, as expressed by end-to-end delay or backlog, the parameters of the transmission scheduling strategy at the switch output buffers, and the parameters of the *regulators* which are located at the various user-network interfaces for the purpose of filtering or shaping the traffic which enters the network.

The present paper focuses on the development of a *recursive* approach to the selection of the regulator parameters. It is envisioned that in the most general case, the regulator parameters will be determined by some process of negotiation between user and network which takes into account the character of the projected data flow, the required quality of service (QoS), and the ability of the network to meet its ongoing QoS commitments. We consider just part of that process—the part that ensures that the regulator parameters provide reasonably and efficiently for the particular attributes of the projected flow. Our interest in parameter estimation techniques that are recursive is motivated by the sense that there will be applications where neither the user nor the network will have had sufficient prior experience of the data flow to supply the parameters—without further measurement—at call setup.

We assume throughout that the regulators are leaky buckets [5]. Our point of departure is the burstiness curve approach to source characterization [6].

II. Background and Preliminary Results

The regulator in general, and the leaky bucket in particular, can be viewed as a vehicle for transferring congestion and traffic loss from the interior of the network to the network boundary, at which point the likelihood of loss might be anticipated and averted by action at the source. If the network is thought of as a pipe from source to sink, then the regulator defines the maximal effective cross section of that pipe. The leaky bucket itself has a couple of alternative formulations. We speak of it here as a token pool replenished by a buffered server and parameterized by the pair (σ, ρ), σ being the initial number of tokens present and ρ the instantaneous rate of replenishment when one or more tokens are absent. A packet of data emitted by the source transfers an equal amount of tokens from the pool to the server buffer and enters the network without delay; in the event that there are no such tokens available, the packet is either discarded or delayed until a proper number of tokens are generated. A token emerging from the server returns to the pool. The replenishment process turns on when the level of the pool drops below σ and shuts off when the pool is full.

The leaky bucket, as we have described it, serves merely to identify, on the basis of the history of the source to which it is attached, the nonconforming traffic (the traffic arrived in the absence of tokens). There has been much work [7]–[10] on how to select the (σ, ρ) parameters so as to achieve a prescribed data discard rate when the source is described by some canonical stochastic process (such as an On/Off process with Poisson or deterministic arrivals during the On state). Our objective here is different. We would like to avoid loss altogether, but at the

Manuscript received March 20, 2000; revised June 6, 2000 and October 27, 2000; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor G. de Veciana.

The author is with the Department of Electrical Engineering, Tarbiat Modares University, Tehran, Iran, and also with the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran (e-mail: valaee@modares.ac.ir).

Publisher Item Identifier S 1063-6692(01)03227-7.

same time to assign the (σ, ρ) parameters as parsimoniously as possible. These two constraints together can be summarized by saying that (σ, ρ) is to be located on the burstiness curve of the source. A similar result has independently been found in [11]. We proceed to describe the notion of the burstiness curve, first proposed in [1].

Let a_t be the instantaneous rate of the source at time t. The function $\{a_t\}$, called a "message," is a real, positive function. We assume that the rate function is bounded

$$a_\tau \le \rho_M \tag{1}$$

where ρ_M is the maximum rate. This is a valid assumption in many applications; link capacity always imposes an upper limit on rate.

The cumulative traffic in the time interval $[s, t]$ is defined by

$$A(s, t) = \int_s^t a_\tau \, d\tau. \tag{2}$$

Since the rate is bounded, the cumulative traffic forms a continuous function. We assume throughout that the message belongs to the class of *linear envelope processes* [12]; that is, that there exist $\tilde{\sigma}$ and $\tilde{\rho}$ (possibly unknown) such that

$$A(s, t) \le \tilde{\sigma} + \tilde{\rho}(t - s). \tag{3}$$

This constraint imposes an upper bound on burst length—a burst being a block of data generated concurrently and instantaneously. Constraints (1) and (3) together yield

$$A(s, t) \le \min\{\rho_M(t - s), \tilde{\sigma} + \tilde{\rho}(t - s)\}. \tag{4}$$

Furthermore, we assume there exists $\overline{\rho} < \tilde{\rho}$ such that

$$\frac{A(s, t)}{t - s} \to \overline{\rho} \quad t \to \infty \tag{5}$$

uniformly in $s \ge 0$.

Transmitting this message through a buffered server with a constant service rate ρ, the unfinished or backlogged traffic in the buffer at time t is given by

$$Q_t(\rho) = \sup_{0 \le s \le t} \int_s^t (a_\tau - \rho) \, d\tau = \sup_{0 \le s \le t} [A(s, t) - (t - s)\rho]. \tag{6}$$

Definition 1 ([1], [6]): The burstiness curve for the traffic a_t in the interval $[0, T]$, is the graph of $b_T(\rho)$ versus ρ where

$$b_T(\rho) = \sup_{0 \le t \le T} Q_t(\rho) = \sup_{0 \le t \le T} \sup_{0 \le s \le t} [A(s, t) - (t - s)\rho]. \tag{7}$$

The burstiness $b_T(\rho)$ is thus the maximum backlog when $\{a_t\}$ is submitted to a deterministic sever of rate ρ. A typical burstiness curve is depicted in Fig. 1.

The following proposition follows from the definition of the burstiness curve.

Proposition 1 [Low and Varaiya]: Given the rate function a_t, $t \in [0, T]$, the burstiness curve is a continuous convex decreasing function defined in the interval $[0, \rho_M]$, with $b_T(0) = A(0, T)$ and $b_T(\rho_M) = 0$.

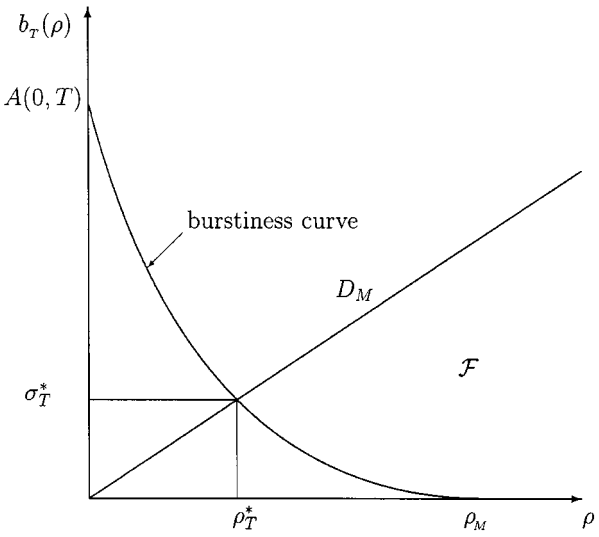

Fig. 1. Intersection of the burstiness curve and the maximum delay constraint.

The following proposition states that the burstiness curve is monotone.

Proposition 2: For fixed ρ, burstiness is continuous and nondecreasing in time. In other words, for given $\{a_t\}$ and ρ, and $T' \le T$

$$b_{T'}(\rho) \le b_T(\rho). \tag{8}$$

Proof: The continuity of burstiness curve resides on the continuity of $A(s, t)$ and $(t - s)\rho$. The definition of burstiness curve produces a nondecreasing $b_T(\rho)$. □

An immediate result of this proposition is that for any fixed ρ, the sequence $\{b_T(\rho)\}$ is convergent (possibly to infinity).

Proposition 3: For a traffic satisfying (1) and (5), if $\rho > \overline{\rho}$, then

$$\lim_{T \to \infty} b_T(\rho) < \infty \tag{9}$$

and if $\rho < \overline{\rho}$, then

$$\lim_{T \to \infty} b_T(\rho) = \infty \tag{10}$$

where $\overline{\rho}$ is defined in (5).

Proof: Suppose $\rho > \overline{\rho}$. Choose ϵ so that $\overline{\rho} + \epsilon - \rho < 0$. Since $A(s, t)/(t - s) \to \overline{\rho}$ uniformly in s as $t \to \infty$, there exists τ such that for $t - s > \tau$

$$\frac{A(s, t)}{t - s} < \overline{\rho} + \epsilon. \tag{11}$$

In this case

$$A(s, t) - (t - s)\rho < (\overline{\rho} + \epsilon - \rho)(t - s) < 0. \tag{12}$$

It follows that

$$\begin{aligned} b_T(\rho) &= \sup_{0 \le t \le T} \sup_{0 \le s \le t} [A(s, t) - (t - s)\rho] \\ &= \sup_{0 \le t \le T} \sup_{t - \tau \le s \le t} [A(s, t) - (t - s)\rho] \\ &\le \rho_M \tau \end{aligned} \tag{13}$$

uniformly in T.

Now suppose $\rho < \overline{\rho}$. We know $b_T(\rho) \geq A(0, T) - T\rho$. Choose τ to satisfy $A(0, T) \geq (\overline{\rho} - \epsilon)T$ for $T \geq \tau$. Then $b_T(\rho) \geq (\overline{\rho} - \rho - \epsilon)T$ for $T \geq \tau$. Choosing ϵ so that $\overline{\rho} - \rho - \epsilon > 0$ completes the proof. □

Starting from the concept of minimum envelope rate, Chang [12] arrives at a similar result. In [12], it has been proved that, for a single queue and a work-conserving server, if the minimum envelope rate of a process is smaller than the link capacity, the queuing delay is bounded; conversely, if the minimum envelope rate is larger than the capacity, the delay is unbounded. Since a uniform convergence is assumed in (5), the minimum envelope rate is identical to the mean rate [12]; hence, using a burstiness approach, Proposition 3 justifies the result of [12].

III. Leaky-Bucket Design—Problem Formulation

Our convention, in referring to the parameterization (σ, ρ) of the leaky bucket, is to reserve the first coordinate σ for the size of the token pool and the second coordinate ρ for the token generation rate. The regulator will be lossless, when applied to a source with burstiness curve $b_T(\cdot)$, if and only if for a given token replenishment rate ρ

$$\sigma \geq b_T(\rho). \tag{14}$$

Our problem, given a source whose behavior and statistics are unknown, is to set the parameters of the corresponding leaky bucket recursively. An algorithm which solves the problem is described in the next section; here we describe the additional constraints imposed upon (σ, ρ) so as to ensure that the target leaky bucket is unique.

The first such constraint is *tightness*, the leaky bucket (σ, ρ) being said to be tight relative to the source with burstiness curve $b_T(\cdot)$ if and only if $\sigma = b_T(\rho)$. Tight designs are economical in terms of the resources allocated to individual sessions in a network where such allocations are based on worst-case behavior; see also [11]. Tight designs are also responsive, in the sense that even small decrements in one of both parameters can have a favorable impact on performance when the network is congested.

We add a second constraint, in the form of an inequality. Consider, for instance, a network in which access to the nodal output buffers is mediated by generalized processor sharing (GPS) or one of its nonpreemptive variants. A session is said to be *stable* in this connection [6] if the corresponding GPS coefficient is at least as large as the token generation rate ρ in the associated leaky bucket. In this case, the connection is contemplated as a deterministic-server queue with buffer size σ (the size of the token pool) and service rate ρ; in fact, the model is approximate and conservative, the backlog in the single-server queue forming an upper bound on the end-to-end backlog in the session. This being so, it is reasonable to require that

$$\sigma \leq \rho D_M \tag{15}$$

where D_M is an upper bound on the maximal end-to-end delay deemed acceptable to the source.

The two inequality constraints taken together amount to $b_T(\rho) \leq \sigma \leq \rho D_M$. This corresponds to the region marked as $\mathcal{F}$ in Fig. 1. The solutions which are tight, in the sense defined above, form the lower boundary of that region. Within this restricted feasible domain we select, as our target, the point with minimal ρ; that is, the point with coordinates ρ_T^*, σ_T^*, corresponding to the leaky bucket (σ_T^*, ρ_T^*). It is needless to say that a sophisticated call admission control (CAC) procedure could make use of the entire feasible part of the burstiness curve, selecting a preferred operating point. Such a CAC algorithm might take either network congestion or pricing into consideration. However, we continue with the minimum bandwidth.

Our goal, then, is to devise an adaptive recursive approach to estimating (σ_T^*, ρ_T^*) for sessions with unknown traffic descriptors—T being the length of the observation interval. We assume that the peak rate ρ_M is given, and thus that

$$\sigma_T^* \leq \rho_M D_M. \tag{16}$$

This helps to limit the domain over which a search is to be performed.

Our estimate, of course, will depend upon T. Notice (by Proposition 2) that σ_T^*, ρ_T^* are nondecreasing in T and thus convergent (possibly to infinity) in the limit of large T. By Proposition 3, it is desirable that $\rho_T^* > \overline{\rho}$, where $\overline{\rho}$ is the long-run mean rate of the source. Unfortunately, this property is not satisfied in general. The following lemma asserts that if a certain lower bound to the burstiness at $\overline{\rho}$ exists, then $\rho_T^* > \overline{\rho}$ for sufficiently large values of T.

Lemma 1: If there exists τ such that $b_\tau(\overline{\rho}) > \overline{\rho} D_M$, then $\lim_{T\to\infty} \rho_T^* > \overline{\rho}$.

Proof: Since the burstiness is nondecreasing in time, $b_T(\overline{\rho}) - \overline{\rho} D_M > 0$ for $T \geq \tau$. Note that $b_T(\rho) - \rho D_M$ decreases with ρ and $b_T(\rho_T^*) - \rho_T^* D_M = 0$. □

Notice that $b_T(\rho)$, for given $\{a_t\}$, can be estimated by calculating the maximum backlog in a single-server queue with input $\{a_t\}$ and service rate ρ. Such estimates are used frequently in the numerical work reported below.

IV. An Iterative Algorithm

In this section, we propose an iterative technique to ascertain the parameters of a leaky bucket. The output of the source is stored for T seconds (the adaptation period) and the token generation rate and the token pool size are determined so that the corresponding leaky bucket is tight. The objective is to solve the equation $b_T(\rho) - \rho D_M = 0$. There exist various techniques to solve such a problem. Here, we propose an iterative algorithm based on intersecting the burstiness curve and the line $\sigma = \rho D_M$.

Choose ρ_1 and ρ_2 with $\rho_1 > \rho_2$. For a server with the service rate ρ_1, find the maximum backlog σ_1. Repeat it for the service rate ρ_2 and obtain σ_2. The line crossing the two points (ρ_1, σ_1) and (ρ_2, σ_2) in the ρ–σ plane intersects the line $\sigma = \rho D_M$ at a point with rate ρ_3. Next, inject the stored data into a queue with the service rate ρ_3 and find the corresponding σ_3. Continue this process by connecting (ρ_2, σ_2) and (ρ_3, σ_3) and intersecting with $\sigma = \rho D_M$. This procedure converges to the point

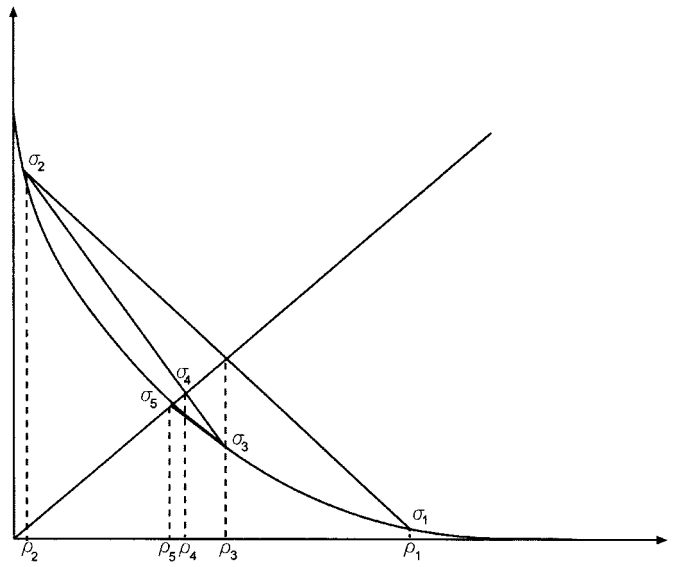

Fig. 2. Iterative procedure converging to (ρ_T^*, σ_T^*).

(ρ_T^*, σ_T^*), the cross section of the burstiness curve and the delay line in Fig. 2. The algorithm is specified by the iterations

$$\rho_i = \frac{\sigma_{i-1}\rho_{i-2} - \sigma_{i-2}\rho_{i-1}}{D_M(\rho_{i-2} - \rho_{i-1}) - (\sigma_{i-2} - \sigma_{i-1})} \tag{17}$$

$$\sigma_i = \sup_{0\le t\le T} \sup_{0\le s\le t} [A(s,t) - (t-s)\rho_i]. \tag{18}$$

Define

$$\mathcal{R} \triangleq \{\rho \mid \rho_T^* \le \rho \le \rho_M\} \tag{19}$$

$$\mathcal{L} \triangleq \{\rho \mid 0 \le \rho \le \rho_T^*\}. \tag{20}$$

$\mathcal{R}$ and $\mathcal{L}$ are the intervals to the right and to the left of the optimum point ρ_T^*.

Lemma 2: Let ρ_1 and ρ_2 be the initializing service rates; then for $i = 2, 3, \ldots$

1) if $\rho_{i-1} \in \mathcal{L}$ and $\rho_i \in \mathcal{R}$, then $\rho_{i+1} \in \mathcal{R}$ and $\rho_{i+1} \le \rho_i$;
2) if $\rho_{i-1} \in \mathcal{R}$ and $\rho_i \in \mathcal{L}$, then $\rho_{i+1} \in \mathcal{R}$ and $\rho_{i+1} \le \rho_{i-1}$;
3) if $\rho_{i-1}, \rho_i \in \mathcal{L}$, then $\rho_{i+1} \in \mathcal{L}$ and $\max\{\rho_{i-1}, \rho_i\} \le \rho_{i+1}$;
4) if $\rho_{i-1}, \rho_i \in \mathcal{R}$, then $\rho_{i+1} \in \mathcal{L}$.

Proof: Immediate from the fact that the burstiness curve is a decreasing convex function defined in $[0, \rho_M]$ and that $D_M > 0$. □

Theorem 1: The iterative technique proposed in (17) and (18) converges to (ρ_T^*, σ_T^*).

Proof: It is assumed in the algorithm that $\rho_1 > \rho_2$. We consider two cases.

Case 1: $\rho_1 \in \mathcal{L}$: Since ρ_1 is larger than ρ_2, from part 3 of Lemma 2, for $i > j \ge 2$, we have $\rho_j < \rho_i \le \rho_T^*$, and hence $\{\rho_i\}$ is convergent. We now prove that, in fact, the convergence is to ρ_T^*.

Define $b_T'(\rho) = \partial b_T(\rho)/\partial\rho$. Since the burstiness curve is a continuous convex decreasing function of ρ, we have

$$\max_{\rho} |b_T'(\rho)| = |b_T'(0)| = T. \tag{21}$$

Thus, for any ρ_i and ρ_{i-1}

$$-\frac{\sigma_i - \sigma_{i-1}}{\rho_i - \rho_{i-1}} \le T. \tag{22}$$

From the iteration (17), we arrive at

$$|\rho_{i+1} - \rho_i| = \left| \frac{\sigma_i - D_M\rho_i}{D_M - \dfrac{\sigma_i - \sigma_{i-1}}{\rho_i - \rho_{i-1}}} \right|. \tag{23}$$

Using (22) yields

$$|\rho_{i+1} - \rho_i| \ge \frac{|\sigma_i - D_M\rho_i|}{D_M + T}. \tag{24}$$

Since the sequence $\{\rho_i\}$ is convergent, $|\sigma_i - D_M\rho_i| \to 0$, and hence $\rho_i \to \rho_T^*$, $\sigma_i \to \sigma_T^*$.

Case 2: $\rho_1 \in \mathcal{R}$: In this case, ρ_2 belongs to either $\mathcal{L}$ or $\mathcal{R}$. Here, the set of rate indices is separated into two subsets containing the indices of the rates in $\mathcal{R}$ and $\mathcal{L}$. If $\rho_2 \in \mathcal{R}$, then all the rates with the indices $i = 3K$, $K = 1, 2, \ldots$ are in $\mathcal{L}$. If $\rho_2 \in \mathcal{L}$, then all the rates with the indices $i = 3K - 1$, $K = 1, 2, \ldots$ are in $\mathcal{L}$. In both cases, the indices of two adjacent rates in $\mathcal{L}$ are separated by 3.

From parts 1 and 2 in Lemma 2, for $\rho_i, \rho_j \in \mathcal{R}$ and $i > j$ we have $\rho_i < \rho_j$ and therefore the subsequence $\{\rho_i \mid \rho_i \in \mathcal{R}\}$ is convergent. The limit is represented by ρ_u.

Let $\rho_i, \rho_{i+3} \in \mathcal{L}$. Note that ρ_i is formed from $\rho_{i-1} \in \mathcal{R}$ and $\rho_{i-2} \in \mathcal{R}$, and similarly ρ_{i+3} is formed from $\rho_{i+1} \in \mathcal{R}$ and $\rho_{i+2} \in \mathcal{R}$. Since the burstiness curve is convex and decreasing, the slope of the line connecting $(\rho_{i+1}, \sigma_{i+1})$ and $(\rho_{i+2}, \sigma_{i+2})$ is larger than the slope of the line connecting $(\rho_{i-2}, \sigma_{i-2})$ and $(\rho_{i-1}, \sigma_{i-1})$. This means that $\rho_i < \rho_{i+3} \le \rho_T^*$. Thus, we conclude that if $\rho_i, \rho_j \in \mathcal{L}$ with $i > j$, then $\rho_j < \rho_i \le \rho_T^*$ and hence the subsequence $\{\rho_i \mid \rho_i \in \mathcal{L}\}$ is convergent. The limit is represented by ρ_l.

We show that $\rho_u = \rho_l = \rho_T^*$. Consider the contrary. Let $\rho_u > \rho_l$. If so, there is no point inside the interval $[\rho_l, \rho_u]$ that can be reached by (17) and (18). We show that this is not true.

Since $\{\rho_i \mid \rho_i \in \mathcal{R}\}$ is convergent, for any fixed ϵ, there exists $\rho_{i-1} \in \mathcal{R}$ and $\rho_i \in \mathcal{L}$ such that $\rho_{i-1} \le \rho_u + \epsilon$. Define $\delta_1 = \rho_{i-1} - \rho_i$ and $\delta_2 = \sigma_i - \sigma_{i-1}$ and let $\rho_{i-1}D_M - \sigma_{i-1} = \beta$. Notice that $\delta_1 \to \rho_u - \rho_l$, for $i \to \infty$. Similarly, $\delta_2 \to \sigma_l - \sigma_u$, for $i \to \infty$, and $\beta \to \rho_u D_M - \sigma_u$, $i \to \infty$, where σ_u and σ_l are the source burstiness at the rates ρ_u and ρ_l, respectively. Using (17), it can be shown that

$$\rho_{i+1} = \rho_{i-1} - \frac{\beta\delta_1}{D_M\delta_1 + \delta_2} \le \rho_u + \epsilon - \frac{\beta\delta_1}{D_M\delta_1 + \delta_2}. \tag{25}$$

Let $\epsilon \to 0$ to arrive at $\rho_{i+1} < \rho_u$. From part 2 of Lemma 2, since $\rho_{i-1} \in \mathcal{R}$ and $\rho_i \in \mathcal{L}$, in fact, $\rho^* \le \rho_{i+1} \le \rho_u$, which is contradictory.

Thus $\rho_u = \rho_l$, and since $\rho_u \in \mathcal{R}$ and $\rho_l \in \mathcal{L}$, we have $\rho_u = \rho_l = \rho_T^*$. □

In the present section, we have assumed that T seconds of data is stored and repeatedly used to estimate the crossing point of the burstiness curve and the line $\sigma = \rho D_M$. During this phase of adaptation, traffic transmission must be frozen. Although it might work for nonreal-time data, it will cause data clipping in

real-time traffics. Thus, an issue of concern is the method that one should use to select an appropriate T. On the one hand, T should be large enough to capture the whole burstiness behavior of traffic. On the other hand, large T introduces delay in the burstiness estimator. Hence, a tradeoff should be placed into order; one might set T to a minimal value and enlarge it as the burstiness surpasses a prescribed threshold, or use an elaborate cost function to arrive at an appropriate resolution. Furthermore, T is also related to the maximum delay D_M. We will discuss shortly that a large D_M requires a large T for convergence; in fact, the burstiness curve converges faster along a line with a small D_M.

Nonetheless, choosing T is a critical issue that will affect the performance of the algorithm. In the following section, we will propose an on-line estimator of burstiness which can be applied to both real-time and nonreal-time traffics. The on-line structure of the algorithm simplifies the selection of T and averts the necessity of storing a frozen window of traffic.

V. Dynamics of the Optimum Point

In the present section, we study the evolution of the optimum point (ρ_T^*, σ_T^*) as a function of T. The optimum point at time T is the solution of the following set of equations

$$\sigma_T^* = \sup_{0\le t\le T} \sup_{0\le s\le t} [A(s,t) - (t-s)\rho_T^*] \tag{26}$$

$$\sigma_T^* = \rho_T^* D_M. \tag{27}$$

In the following proposition, we show that the optimum rate ρ_T^* and the buffer size σ_T^* are nondecreasing functions of time.

Proposition 4: For $T' \le T$ we have

$$\rho_{T'}^* \le \rho_T^* \tag{28}$$

$$\sigma_{T'}^* \le \sigma_T^*. \tag{29}$$

Proof: Using Proposition 2, for all $\rho \in [0, \rho_M]$, we have

$$b_T(\rho) \ge b_{T'}(\rho). \tag{30}$$

Note that both $b_T(\rho)$ and $b_{T'}(\rho)$ are decreasing functions of ρ (see Proposition 1). Represent by (σ_T^*, ρ_T^*) the intersection of $b_T(\rho)$ and the line $\sigma = \rho D_M$. Define $(\sigma_{T'}^*, \rho_{T'}^*)$ similarly. Since $b_T(\rho)$ is an upper bound to $b_{T'}(\rho)$, the proof is complete. □

Remark: Since $\rho_T^* \le \rho_M$, the sequence $\{\rho_T^*\}$ is convergent. Let $Q_T(t;\rho)$ be defined as the queue length for a single server queue with service rate ρ, i.e.

$$Q_T(t;\rho) = \sup_{0\le s\le t} [A(s,t) - (t-s)\rho], \qquad \text{for } t \le T. \tag{31}$$

Then

$$\sigma_T^* = \sup_{0\le t\le T} Q_T(t;\rho_T^*). \tag{32}$$

Using this definition, we have the following proposition.

Proposition 5: For $T' \le T$, we have

$$Q_{T'}(t;\rho_{T'}^*) \ge Q_{T'}(t;\rho_T^*). \tag{33}$$

Proof: The proof is straightforward from Proposition 4 and (31). □

This proposition indicates that for $T \ge T'$, $(\sigma_{T'}^*, \rho_{T'}^*)$ carries all the necessary information contained in $[0, T']$ for computing (σ_T^*, ρ_T^*). To see this, note that

$$Q_T(t;\rho_T^*) = Q_{T'}(t;\rho_T^*) + Q_{T'T}(t;\rho_T^*) \tag{34}$$

where $Q_{T'T}(t;\rho_T^*)$ for $T' \le t \le T$ is defined as

$$Q_{T'T}(t;\rho_T^*) = \sup_{0\le s\le t} [A(s,t) - (t-s)\rho]. \tag{35}$$

To get the burstiness σ_T^* at time T, we should solve

$$\begin{aligned}\sigma_T^* &= \sup_{0\le t\le T} Q_T(t;\rho_T^*) \\ &= \sup_{0\le t\le T'} Q_{T'}(t;\rho_T^*) \vee \sup_{T'\le t\le T} Q_{T'T}(t;\rho_T^*)\end{aligned} \tag{36}$$

where $a \vee b$ is defined as

$$a \vee b = \max\{a, b\}. \tag{37}$$

Using Proposition 5 and (31), we have

$$\sup_{0\le t\le T} Q_T(t;\rho_T^*) \le \sup_{0\le t\le T'} Q_{T'}(t;\rho_{T'}^*) \vee \sup_{T'\le t\le T} Q_{T'T}(t;\rho_T^*). \tag{38}$$

We now distinguish two cases.

Case I:

$$\sup_{T'\le t\le T} Q_{T'T}(t;\rho_T^*) \le \sup_{0\le t\le T'} Q_{T'}(t;\rho_{T'}^*). \tag{39}$$

From (38), we get

$$\sup_{0\le t\le T} Q_T(t;\rho_T^*) \le \sup_{0\le t\le T'} Q_{T'}(t;\rho_{T'}^*). \tag{40}$$

Using Proposition 4, we have

$$\sup_{0\le t\le T'} Q_{T'}(t;\rho_{T'}^*) \le \sup_{0\le t\le T} Q_T(t;\rho_T^*). \tag{41}$$

Inequalities (40) and (41) yield

$$\sup_{0\le t\le T'} Q_{T'}(t;\rho_{T'}^*) = \sup_{0\le t\le T} Q_T(t;\rho_T^*). \tag{42}$$

This means that the maximum backlog of data in the interval $[T', T]$ is smaller than the burstiness computed in $[0, T']$. Thus, the burstiness curve is constant in the interval $[T', T]$.

Case II:

$$\sup_{T'\le t\le T} Q_{T'T}(t;\rho_T^*) > \sup_{0\le t\le T'} Q_{T'}(t;\rho_{T'}^*). \tag{43}$$

This means that the trajectory of traffic, generated in the interval $[T', T]$, has increased the maximum backlog beyond its previous limit. Then, from (36) and Proposition 5, we have

$$\sup_{0\le t\le T} Q_T(t;\rho_T^*) = \sup_{T'\le t\le T} Q_{T'T}(t;\rho_T^*). \tag{44}$$

Thus, it can be seen that the optimum point (σ_T^*, ρ_T^*) only depends on $(\sigma_{T'}^*, \rho_{T'}^*)$ and the maximum queue length in the interval $[T', T]$.

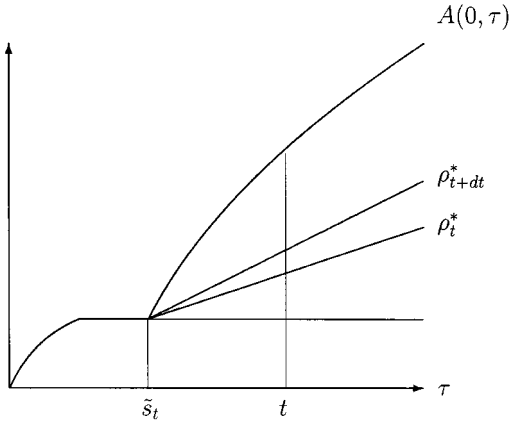

Fig. 3. Busy cycle containing t.

The two cases above correspond, respectively, to the points of *freeze* and *increase*. While in freeze, there exists $\delta T > 0$ such that $(\sigma^*_{T+\delta T}, \rho^*_{T+\delta T}) = (\sigma^*_T, \rho^*_T)$. In the state of increase, for any $\delta T > 0$, $\sigma^*_{T+\delta T} > \sigma^*_T$, and similarly $\rho^*_{T+\delta T} > \rho^*_T$.

Theorem 2: The dynamics of the optimum point (σ^*_t, ρ^*_t) can be represented by the following differential equations. If t is a point of freeze

$$\dot{\sigma}^*_t = D_M \dot{\rho}^*_t = 0. \tag{45}$$

If t is a point of increase

$$\dot{\sigma}^*_t = D_M \dot{\rho}^*_t = (t - \tilde{s}_t) + (a_t - \rho^*_t) \tag{46}$$

where $\tilde{s}_t$ is the beginning of the busy cycle containing t (see Fig. 3).

Proof: Let

$$T' = t \tag{47}$$

$$T = t + dt. \tag{48}$$

The two cases above correspond, respectively, to the following two instances.

Case I:

$$\sigma^*_{t+dt} = \sigma^*_t \tag{49}$$

$$\rho^*_{t+dt} = \rho^*_t. \tag{50}$$

Therefore, we have

$$\dot{\sigma}^*_t = \dot{\rho}^*_t = 0. \tag{51}$$

Case II:

$$\begin{aligned}\sigma^*_{t+dt} &= \sup_{t \le t' \le t+dt} Q_{t,\,t+dt}(t';\, \rho^*_{t+dt}) \\ &= \sup_{t \le t' \le t+dt} \sup_{0 \le s \le t'} [A(s, t') - (t' - s)\rho^*_{t+dt}].\end{aligned}$$

For dt very small, we have

$$\begin{aligned}\sigma^*_{t+dt} &= \sup_{0 \le s \le t+dt} [A(s, t+dt) - (t + dt - s)\rho^*_{t+dt}] \\ &= \sup_{0 \le s \le t+dt} [A(s, t) + a_t\, dt - (t - s)\rho^*_{t+dt} - \rho^*_{t+dt}\, dt] \\ &= \sup_{0 \le s \le t+dt} [A(s, t) - (t - s)\rho^*_{t+dt}] + (a_t - \rho^*_{t+dt})\, dt \\ &= Q_t(t;\, \rho^*_{t+dt}) + (a_t - \rho^*_{t+dt})\, dt.\end{aligned} \tag{52}$$

Note that this case only happens if $a_t > \rho^*_t$. This means that the input rate must be larger than the service rate; otherwise backlog will not grow. Since t is a point of increase for σ^*_t, the backlog at time t should be equal to σ^*_t, i.e.

$$\sigma^*_t = Q_t(t;\, \rho^*_t). \tag{53}$$

From (52) and (53), we get

$$\begin{aligned}\sigma^*_{t+dt} - \sigma^*_t &= Q_t(t;\, \rho^*_{t+dt}) - Q_t(t;\, \rho^*_t) + (a_t - \rho^*_{t+dt})\, dt \\ &= (t - \tilde{s}_t)(\rho^*_{t+dt} - \rho^*_t) + (a_t - \rho^*_{t+dt})\, dt.\end{aligned}$$

Now dividing the two sides by dt and letting $dt \to 0$, we have

$$\dot{\sigma}^*_t = (t - \tilde{s}_t)\dot{\rho}^*_t + (a_t - \rho^*_t) \tag{54}$$

and the proof is complete. □

In summary, the dynamics of the optimum point can be expressed using a finite state machine (FSM) with two states 0 and 1. While in State 0 (point of freeze), the optimum point (σ^*_t, ρ^*_t) is constant and does not change. In State 1 (point of increase), the dynamics of the optimum point is abided by the differential equation (54). It is straightforward to see that the solution to (54) is

$$\sigma^*_t = \rho^*_t D_M = \frac{D_M}{t - \tilde{s}_t + D_M} A(\tilde{s}_t, t). \tag{55}$$

VI. Reflection Mapping

The technique can also be formulated in terms of the concept of reflection mapping [13]. Reflection mapping has been used in [13] to formulate flow controllers and to establish their optimality.

For a given process x_t, its reflection in the time varying interval $I_t = [\alpha_t, \beta_t]$ is defined as

$$q_t \triangleq x_t + \ell_t - u_t \tag{56}$$

where ℓ_t and u_t are chosen such that

- $\alpha_t \le q_t \le \beta_t$;
- $\ell_{0^-} = u_{0^-} = 0$ and ℓ_t (respectively, u_t) increases only when $q_t = \alpha_t$ (respectively, $q_t = \beta_t$).

ℓ_t and u_t are called the *lower boundary process* and the *upper boundary process*, respectively. Here, we use single boundary reflection mapping (either $\alpha_t = -\infty$ or $\beta_t = +\infty$.)

It can be shown that for a case in which only the lower boundary is present ($\beta_t = +\infty$), we have

$$\ell_t = \min\{-\inf_{0 \le s \le t}(x_s + \alpha_s), 0\} \tag{57}$$

$$q_t = x_t + \min\left\{-\inf_{0 \le s \le t}(x_s + \alpha_s), 0\right\}. \tag{58}$$

Similarly, if only the upper boundary is present ($\alpha_t = -\infty$)

$$u_t = \max\left\{\sup_{0 \le s \le t}(x_s - \beta_s), 0\right\} \tag{59}$$

$$q_t = x_t - \max\left\{\sup_{0 \le s \le t}(x_s - \beta_s), 0\right\}. \tag{60}$$

We use two properties of the reflection mapping, namely *causality* and *minimality* [13].

1) Causality:
   For any $s > 0$, the reflection of the process at time $s+t$ with $t > 0$ depends only on the value of the process from s^- onward, neglecting the entire past of the process before s^-.
2) Minimality:
   For given $I_t = [\alpha_t, \beta_t]$, the boundary process u_t (respectively, ℓ_t) is the smallest process that reflects x_t into I_t.

Theorem 3: The burstiness σ_T^* given in (55) is an upper boundary process for the reflection of the backlog process, downward at zero.

Proof: Define

$$x(t;\,\rho) = A(0,\,t) - \rho t. \tag{61}$$

Backlog at time $0 \le t \le T$ is

$$\begin{aligned} Q_T(t;\,\rho) &= \sup_{0\le s\le t}[x(t;\,\rho) - x(s;\,\rho)] \\ &= x(t;\,\rho) - \inf_{0\le s\le t} x(s;\,\rho). \end{aligned} \tag{62}$$

Using (58), we note that $Q_T(t;\,\rho)$ is the reflection of $x(t;\,\rho)$ upward at zero.

Now find the reflection of $Q_T(t;\,\rho_T^*)$ downward at zero. From the definition of reflection mapping, we have

$$Q_T(t;\,\rho_T^*) - u_t \le 0 \tag{63}$$

where u_t is the upper boundary process. Thus

$$Q_T(t;\,\rho_T^*) \le u_t. \tag{64}$$

From (59), we get

$$u_t = \sup_{0\le s\le t} Q_T(s;\,\rho_T^*). \tag{65}$$

Using the assumption that u_t is nondecreasing, for every T, we have

$$\sigma_T^* = \sup_{0\le t\le T} Q_T(t;\,\rho_T^*) = u_T. \tag{66}$$

□

Theorem 3 suggests that all properties of the reflection mapping—say, minimality and causality—can be extended to the concept of burstiness. Minimality of u_T indicates that σ_T^* is the smallest process that reflects $Q_T(t;\,\rho_T^*)$ below zero. This aligns with our objective, assigning the $(\sigma,\,\rho)$ parameters as parsimoniously as possible—the minimal upper-boundary process corresponds to a tight token pool size. Furthermore, the boundary process in reflection mapping is causal, meaning that for a given u_s, the upper boundary process u_t, for $t > s$, is independent of u_τ, $\tau < s$. We have implicitly used this property to derive an algorithm in Section VII.

An interesting influence of using the reflection mapping formulation is that the technique might be applied to nondifferentiable processes. This assumption incorporates traffics with instantaneous bursts in the proposed algorithm. To this end, we digitize the time to a fine grid and apply the technique on that grid. The optimal rate ρ_t^* should be chosen such as to handle the excess traffic accumulated during the last sampling interval. Hence, for traffics with sudden bursts, the differential equation (46) can be substituted with a difference equation. In the following section, we state the proposed on-line algorithm.

VII. The On-Line Algorithm

Let time be decomposed into nonoverlapping intervals of Δ seconds each. The optimum point $(\sigma_t^*,\,\rho_t^*)$ is computed at the end of each interval. The input traffic is introduced to a constant-rate server and the backlog is monitored and used to modify the value of the optimum point. The following steps summarize the algorithm.

1) Initialize the algorithm to $t = 0$, $\rho_t^* = 0$, $\sigma_t^* = 0$, $Q(t) = 0$, $\tilde{s}_t = 0$.
2) Update the time index as
$$t = t + \Delta. \tag{67}$$
End the algorithm if $t \ge T$, else continue,
$$Q(t) = \max\{A(\tilde{s}_t,\,t) - (t - \tilde{s}_t)\rho_t^*,\,0\}. \tag{68}$$
If $Q(t) = 0 \Rightarrow \tilde{s}_t = t$.
3) If $Q(t) \le \sigma_t^*$, go to Step 2 (State 0); Else (State 1),
$$\rho_t^* = \frac{A(\tilde{s}_t,\,t)}{t - \tilde{s}_t + D_M}, \tag{69}$$
$$\sigma_t^* = \rho_t^* D_M. \tag{70}$$
Go to Step 2.

Note that the parameter D_M could be treated generically—this might suggest that the algorithm be applied to various delay lines and the whole burstiness curve be approximated by interconnecting the induced points.

The adaptation time T in Step 2 is a parameter of design. Note that the flow of traffic is not interrupted in this technique and T only acts as a means to end the algorithm and can be very large. A large T captures the burstiness of the traffic in a more reliable framework. For all that, the parameters of the regulator—if the burstiness is used to assign a regulator to the traffic—can be floating and might change as the time advances. In practice, however, one might be willing to freeze the regulator and mark or even stop the nonconforming traffic. Hence, a mechanism should be employed to select an appropriate T. Note that this might be fragile and inaccurate—a fairly regular source might happen to become bursty in the future. If the algorithm ends before such an impetuous behavior of the source is activated, the burstiness of the data will be unseen and uncaptured. To alleviate the problem one might come up with a rigorous rule of selecting T. For instance, the QoS that the user expects can be utilized as a prescription to cease the algorithm. The algorithm could be ended (T is reached) as soon as the QoS is satisfied for the user—the mechanism, however, might not be trivial.

An alternative would be to monitor the growth of the token pool size and/or token generation rate. If a regular behavior for the growth of these parameters could be found, T would be the minimal time at which the convergence of the growth is ratified. Take, for instance, a greedy source transmitting traffic with rate ρ_M starting at time zero. Then, the algorithm is always in the point of increase and we have

$$\rho_t^* = \frac{t\rho_M}{t + D_M}. \tag{71}$$

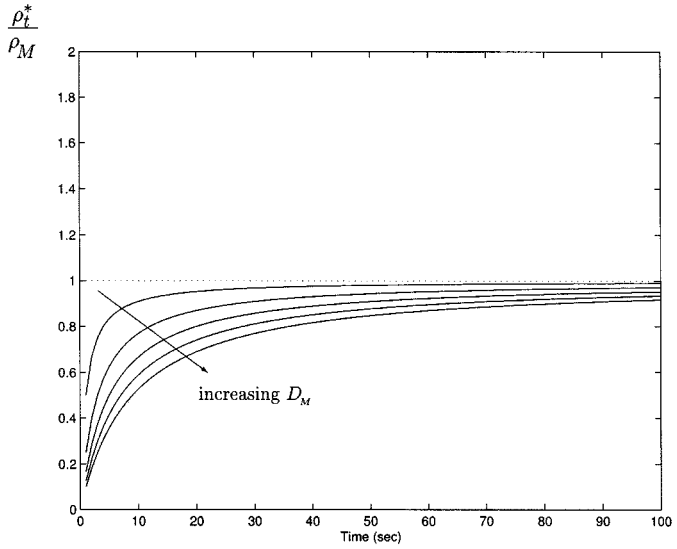

Fig. 4. Normalized convergence rate of the algorithm for a greedy source.

Fig. 4 illustrates the plot of ρ_t^*, normalized with respect to ρ_M, for various delay bounds D_M. Note a hyperbolic increase of the parameter. For this example, T can be selected as the time at which ρ_t^* is sufficiently close to ρ_M—again the closeness is a parameter of design.

For the traffic of Fig. 4, increasing D_M reduces the rate of convergence of the algorithm. This conclusion can be extended to other type of traffic, as well. Recall that the growth rate of the parameters is administered by (69). In [11], the supremum of (69) for $\hat{s}_t = 0$ has been defined as the *deterministic effective bandwidth* of the traffic accumulated during the interval $[0, t]$. Note that in a point of increase, the burstiness of the traffic changes. In fact, for a time t being a point of increase

$$\rho_t^* = \frac{A(\tilde{s}_t, t)}{t - \tilde{s}_t + D_M} = \sup_{0 \le s \le t} \rho_s^*. \tag{72}$$

Hence, ρ_t^* is the deterministic effective bandwidth of the traffic calculated in the interval $[0, t]$.

VIII. Applicability of the Algorithm

The convergence rate of the algorithm for a nongreedy source is upper bounded by (71). The rate of convergence depends on the stochastic behavior of source. Roughly speaking, a source illustrating a typical transmission characteristics—such as a semiperiodic video traffic—will behave nearly like a periodic source. Hence, for such a source, a hyperbolic growth rate for ρ_t^* (similarly σ_t^*) can be expected. This characteristic might be exploited to get a measure of adjustment of the regulator to the projected flow.

On the other hand, a nonreal-time best-effort traffic can introduce an unexpected burstiness growth rate. The burstiness of such a source might stay low for a very long period of time, or change in a sudden rush. Due to instantaneous bursts, selection of T might not be trivial. The best-effort nature of a nonreal-time traffic, however, permits application of a buffer to smoothen the burstiness.

The techniques we have addressed in this paper may fall into two categories. In Section IV, we proposed an iterative procedure for the adjustment of a regulator based on a stored fragment of data. The adaptation period was assumed to be large enough to convey the burstiness behavior of the input traffic. Iterations were performed on the same observation window and repeated until a convergence was observed. Therefore, the technique is off-line and cannot be applied to real-time traffics. Nonetheless, a fixed adaptation period cannot characterize the burstiness of a whole trace. Yet, it is applicable to nonreal-time data and can be used to capture the burstiness of a delay-insensitive traffic. For nonreal-time traffic, an adjusted regulator can be employed as a shaper—the nonconforming traffic is delayed and introduced into the network in a more regular setting. Hence, the technique of Section IV is applied to adjust a regulator to data, and then the regulator is employed as a shaper with frozen parameters.

A second contribution of this paper is the on-line algorithm proposed in Section VII. As noted, the technique of Section IV will fail to catch the burstiness in the traffic outside the adaptation window. The second algorithm proposes a recursive method to capture the burstiness of the whole trace. The procedure starts with $T = \infty$ and will keep running adaptively to the progress of the source traffic. Thus, the recursive method can quickly adapt to any prospective burstiness in the forthcoming data. The characterized leaky-bucket parameters are still useful for network provisioning; the network can use this information to reserve required resources such as bandwidth and buffers for the introduced traffic.

The proposed technique in Section VII can be applied to both real-time and nonreal-time traffics. For a nonreal-time traffic, once the convergence of the algorithm could be conceived, the estimated leaky-bucket parameters are frozen. The nonconforming traffic is then delayed and injected into the network on a more regular basis. For real-time traffic, however, the adjustment of the leaky bucket cannot be halted and must continue with the burstiness behavior of the projected traffic. In this case, the algorithm will keep running and the results of the leaky-bucket adjustment are used for source classification and network resource provisioning. If the setting of the leaky-bucket regulator is frozen at a prescribed time, the nonconforming data in the ensuing traffic will be tagged and delivered to the network. The tagged traffic illustrates the burstiness of the source that has not been captured by an earlier adjustment of the regulator due to a finite adaptation period. It is then up to the network to either re-initiate the adaptation of the regulator and/or provide extra resources for the tagged traffic.

IX. Simulation Results

The objective is to design a leaky-bucket regulator for the traffic of a given source. The leaky bucket should be designed such as to minimize the token generation rate. The leaky-bucket parameters are determined independent of the state of the network. The simulation is performed on a real Motion JPEG encoded video trace taken from [14]. The trace is a full-length video of the movie Jurassic Park and represents the number of bytes contained in each encoded frame.

The trace of the movie for 600 s is depicted in Fig. 5. It is seen that the input rate is time varying—the maximum and minimum frame sizes are 21 099 and 1977, respectively. The burstiness

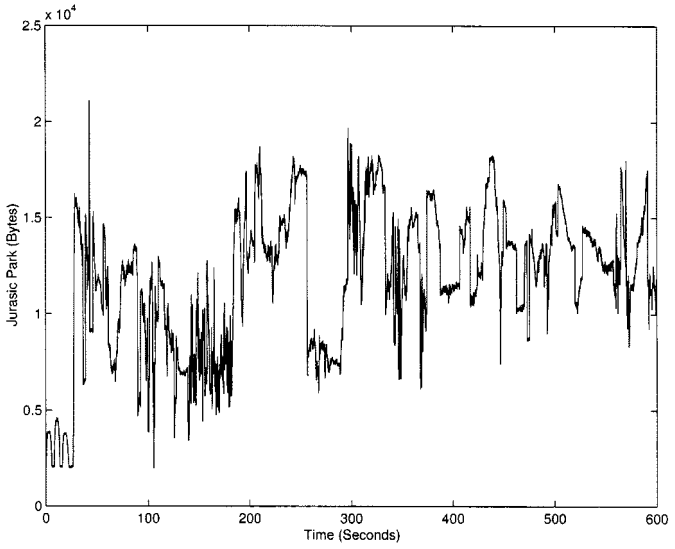

Fig. 5. Video trace for ten minutes of the movie Jurassic Park.

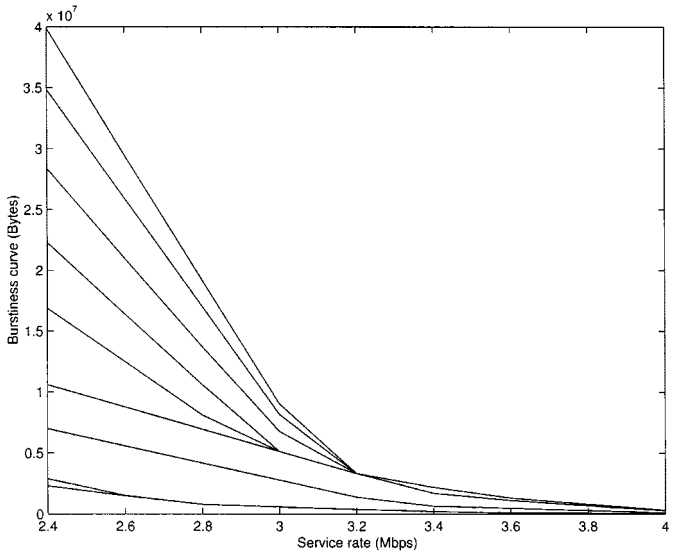

Fig. 6. Burstiness curve for the movie Jurassic Park. The burstiness curve was computed at the end of each minute of observed data.

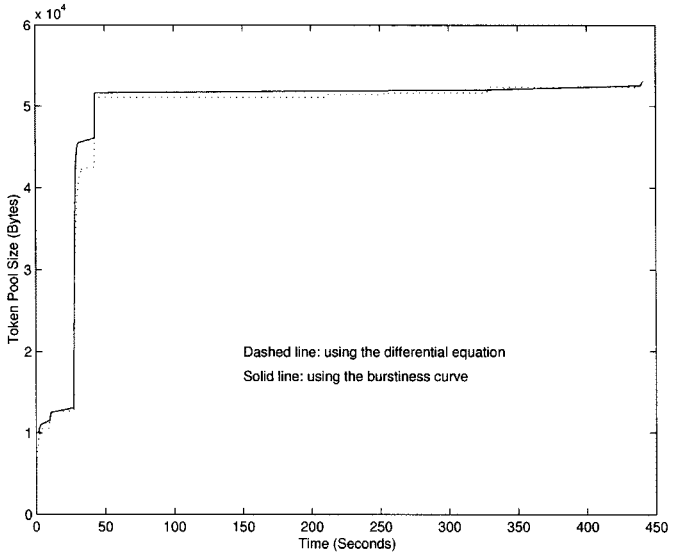

Fig. 7. Dynamics of the optimum point (burstiness) using the two proposed techniques for the same trace of the movie Jurassic Park.

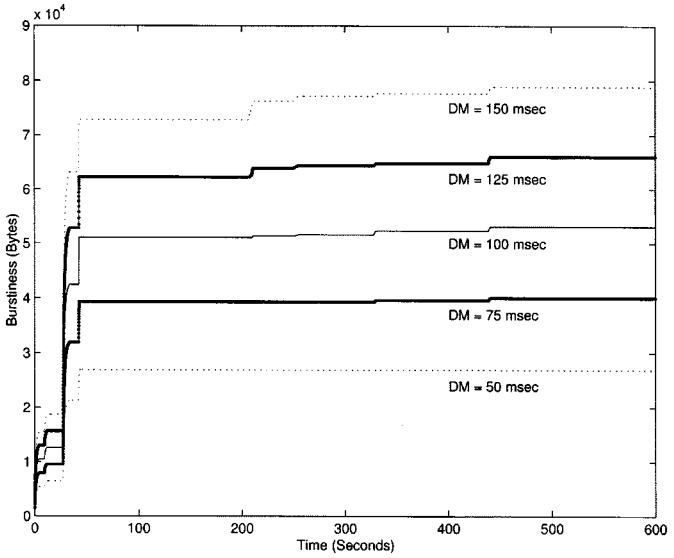

Fig. 8. Evolution of the burstiness of the movie Jurassic Park as function of the maximum delay.

curve for this trace is illustrated in Fig. 6. Here, the time is decomposed into nonoverlapping snapshots of one minute each. The burstiness curve is calculated at the end of each snapshot.

We compute the parameters of a leaky bucket using the two techniques discussed in this paper. In the first technique, we find the crossing point of the burstiness curve with the line $\sigma = \rho D_M$ with $D_M = 100$ ms. Here, the burstiness curve is calculated on a grid of rates ranging from 0.8 to 5.6 Mb/s with the step size of 40 kb/s. For the second technique, we use the algorithm proposed in Section VII. The results have been shown in Fig. 7. It is seen that the two algorithms perform closely. The difference is due to using a coarse grid for rate; for a finer grid, the dashed line approaches the solid line. Note that the algorithm of Section VII has a smaller computational cost and can also be applied on line.

In Fig. 8, the evolution of the burstiness of the traffic for the same trace of Jurassic Park has been illustrated for various delay bounds. Note that the burstiness increases with D_M. This is in contrast to the behavior of the optimum rate—the optimum service rate decreases with D_M. This behavior is in agreement with the results obtained in Fig. 4 for a greedy source.

To study the effect of T (the adaptation period) in the performance of the algorithm, we simulate a scenario in which the traffic is regulated by a leaky bucket tailored to the burstiness of a traffic for 60, 300, and 600 s of the same trace. The maximum delay is $D_M = 150$ ms. The leaky bucket is used as a shaper—the excess traffic is queued and smoothened with the rate ρ_T^*. Fig. 9 represents the cumulative excess traffic for different values of the adaptation period for a trace of 2000 s of the same movie. Note that for small T, the burstiness of the source is not captured thoroughly—a notable amount of data is backlogged in the leaky bucket. The volume of the backlogged data decreases with increasing T. Fig. 10 illustrates the delay experienced by the traffic in the leaky bucket. Note that a large T con-

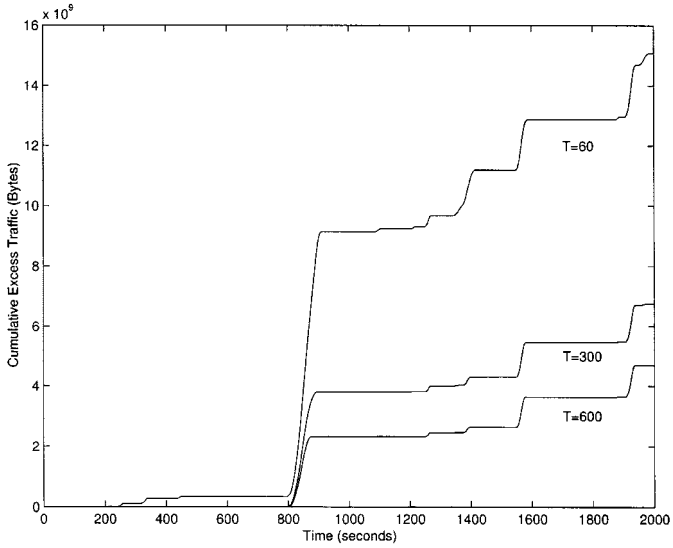

Fig. 9. Cumulative excess traffic to be regulated by an adapted leaky bucket. The adaptation periods are $T = 60$, $T = 300$, and $T = 600$ s.

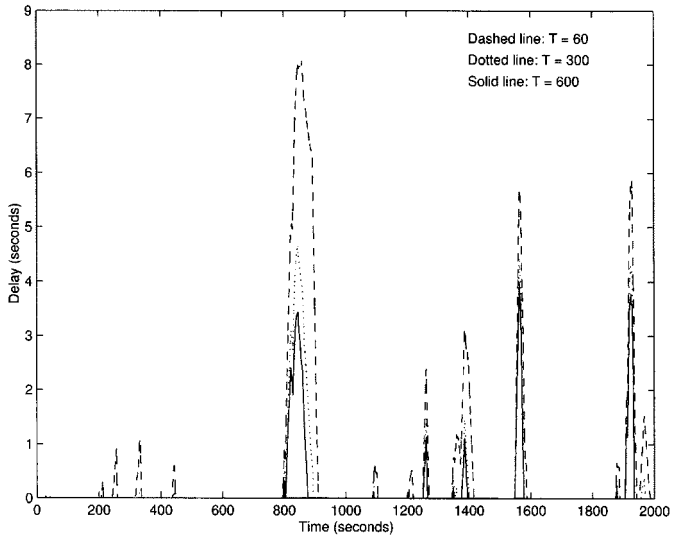

Fig. 10. Delay suffered by the excess traffic in the leaky bucket. The adaptation periods are $T = 60$, $T = 300$, and $T = 600$ s.

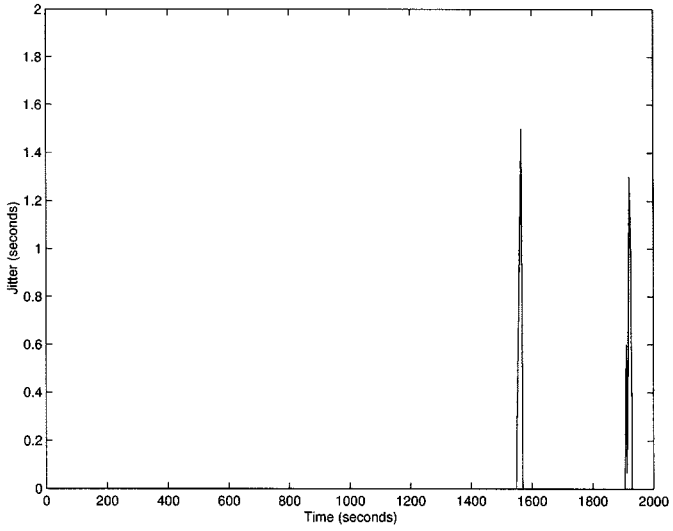

Fig. 11. Jitter suffered by the excess traffic in the leaky bucket. The adaptation period is $T = 1000$ s.

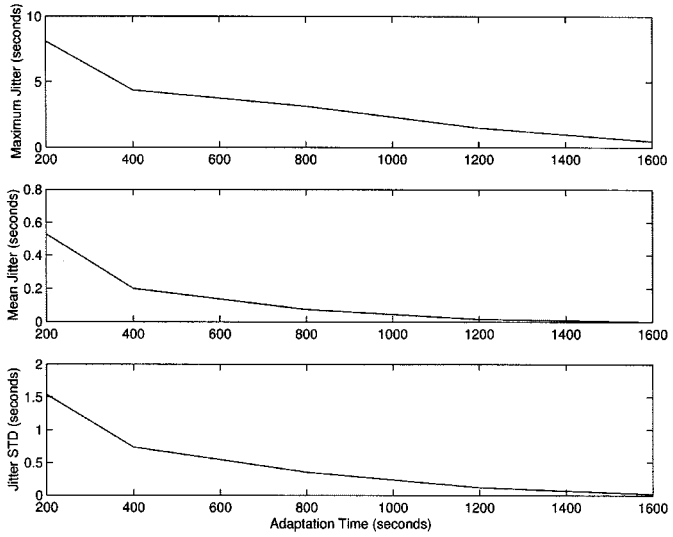

Fig. 12. Maximum, mean, and standard deviation for the jitter suffered by the nonconforming traffic in the leaky bucket as a function of the adaptation period.

stitutes a small delay. Inside the network, the delay is bounded by D_M provided that at least a rate of ρ_T^* is guaranteed for the traffic.

One can also use Figs. 9 and 10 to examine the performance of a network composed of a single node serving the traffic regulated by the designed leaky bucket. Assume a leaky bucket operating in a filtering mode (nonconforming traffic is tagged and passed to the network). Figs. 9 and 10 can then be interpreted as the accumulated queue size and delay for the tagged traffic in the node. Note again that to obtain an appropriate estimate of the traffic burstiness, one should use a fairly large T.

To study the QoS of a video trace regulated by the proposed leaky-bucket estimator, we compute the jitter for the nonconforming traffic in a regulator, acting as a traffic shaper and tuned to the input traffic for a prescribed adaptation period. The result for an adaptation period of $T = 1000$ s has been illustrated in Fig. 11. Here, we assume that the on-line algorithm ends in $T = 1000$ s and the traffic is smoothened by the regulator with frozen parameters. This is in contrast to the proposed policy in Section VIII (keep the algorithm running with $T = \infty$ for real-time traffics) in order to study the behavior of a real-time video traffic regulated by a leaky-bucket shaper. The jitter has been calculated on the last byte of each delayed frame. Fig. 12 illustrates the maximum, the mean and the standard deviation of the jitter versus the adaptation period. Note that the jitter is reduced by increasing the adaptation period. An appropriate jitter-removal buffer can be utilized in the receiver to thwart the induced jitter in the regulator. For instance, consulting Fig. 12, for an adaptation period of 20 min (1200 s), a jitter-removal buffer that stores up to two seconds of traffic, can be used to avert the whole effect of jitter and produce a jitter-free trace.

Finally, we use the leaky-bucket regulator as a filter (nonconforming traffic is discarded). The same trace of 2000 s was used

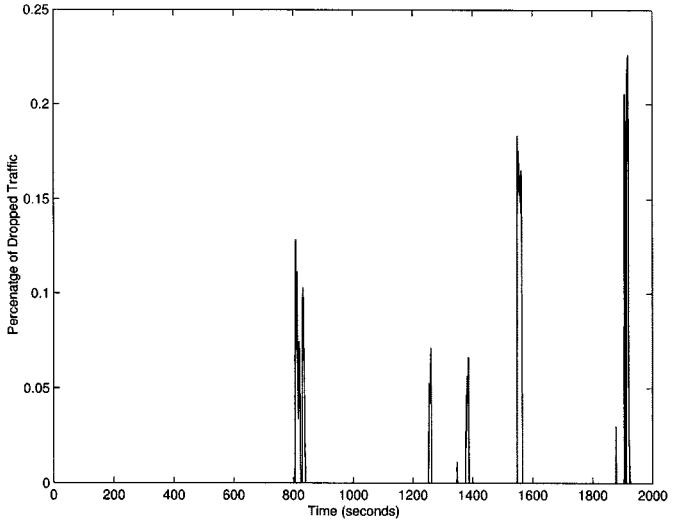

Fig. 13. Ratio of the dropped data to the total input traffic for an adaptation period of $T = 800$ s.

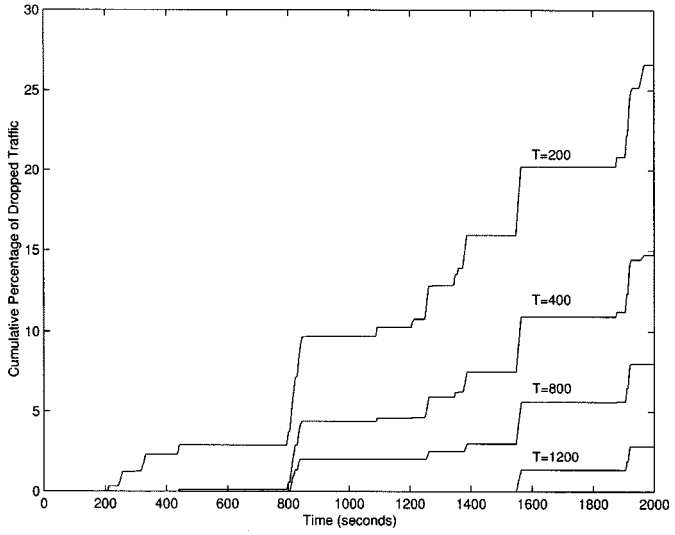

Fig. 14. Cumulative ratio of the dropped data to the total input traffic for the adaptation periods $T = 200, 400, 800,$ and 1200 s.

and the adaptation period was assumed to be 200, 400, 800, and 1200 s. The output traffic was monitored and the ratio of the dropped data to the total input traffic was computed. Fig. 13 illustrates the ratio of the discarded traffic to total input traffic for $T = 800$ s. The cumulative sum of the percentage of the discarded traffic is depicted in Fig. 14. A notable improvement is observed by increasing the adaptation period.

X. Summary and Conclusion

In this paper, we introduced a technique to adapt the parameters of a leaky bucket to the source traffic at call setup. The objective was to adjust the leaky bucket so as to avoid traffic discard at the network boundary, regardless of the possibility of congestion in the network. This can be used as an approach to the assessment of burstiness of a given stream of data. The leaky-bucket parameter assignment should have imposed a constraint on the traffic such that the maximum delay incurred was bounded below a prescribed threshold.

We identified a feasible region in the (σ, ρ) parameter space which satisfied both constraints. We further assumed that the bandwidth was expensive, so that the objective was to select a point in the feasible region with the smallest bandwidth. This assumption reduced the problem to intersecting the so-called burstiness curve with the maximum delay constraint. We used the term optimum to refer to the intersection point.

An iterative procedure was introduced which converged to the optimum point without the necessity of forming the whole burstiness curve. This could drastically reduce the computational cost. It was proved that the proposed procedure was convergent to the true optimum point.

The dynamics of the optimum point as a function of time was also studied. It was shown that the evolution of the optimum point could be managed by a differential equation. The solution to the differential equation was formulated in terms of a two-state machine. In State 0, the optimum point froze and in State 1, it increased.

The proposed algorithm was also formulated via reflection mapping. This extended the properties of the reflection mapping, such as minimality and causality, to the concept of burstiness and justified our parsimony in selecting the leaky-bucket parameters.

Finally, we backed up our algorithm by applying it to a real video trace. The simulation study showed that the algorithm successfully begetted the intrinsic burstiness nature of the data and found an appropriate leaky bucket in an on-line procedure.

Acknowledgment

The author would like to thank Prof. M. A. Kaplan of McGill University for his many contributions during the course of this research, and believes that, in fact, Prof. Kaplan is a co-author of this work.

References

[1] 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.
[2] ——, "A calculus for network delay—Part II: Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, pp. 132–141, Jan. 1991.
[3] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services network: The single node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 334–357, June 1993.
[4] ——, "A generalized processor sharing approach to flow control in integrated services network: The multiple node case," *IEEE/ACM Trans. Networking*, vol. 2, pp. 137–150, Apr. 1994.
[5] J. S. Turner, "New directions in communications (or which way to the information age?)," *IEEE Commun. Mag.*, vol. 24, pp. 8–15, Oct. 1986.
[6] S. Low and P. Varaiya, "A simple theory of traffic resource allocation in ATM," in *Proc. IEEE Globecom*, 1991, pp. 1633–1637.
[7] E. P. Rathgeb, "Modeling and performance comparison of policing mechanics for ATM networks," *IEEE J. Selec. Areas Commun.*, vol. 9, pp. 325–334, Apr. 1991.
[8] M. Butto, E. Cavallero, and A. Tonietti, "Effectiveness of the leaky-bucket policing mechanisms in ATM networks," *IEEE J. Selec. Areas Commun.*, vol. 9, pp. 335–342, Apr. 1991.
[9] 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.

[10] S. Chong and S. Q. Li, "(σ, ρ)-characterization-based connection control for guaranteed services in high-speed networks," in *Proc. IEEE INFOCOMM*, 1995, pp. 835–844.
[11] J. Y. Le Boudec, "Application of network calculus to guaranteed service networks," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1087–1096, May 1998.
[12] C. S. Chang, "Stability, queue length and delay of deterministic and stochastic queuing networks," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 913–931, May 1994.
[13] T. Konstantopoulos and V. Anantharam, "Optimal flow control schemes that regulate the burstiness of data," *IEEE/ACM Trans. Networking*, vol. 3, pp. 423–432, Aug. 1995.
[14] Motion JPEG video traces, W.-C. Feng. (2000). [Online]. Available: http://www.cis.ohio-state.edu/~wuchi

Shahrokh Valaee (S'88–M'00) was born in Tabriz, Iran. He received the B.Sc. and M.Sc. degrees from Tehran University, Tehran, Iran, and the Ph.D. degree from McGill University, Montreal, Canada, all in electrical engineering.

From 1994 to 1995, he was a Research Associate at INRS Telecom, University of Quebec, Montreal. Since 1996, he has been an Assistant Professor in the Department of Electrical Engineering, Tarbiat Modares University, Tehran, and also an Adjunct Professor in the Department of Electrical Engineering, Sharif University of Technology, Tehran. His current research in high-speed networking focuses on quality-of-service (QoS) guarantees, multiprotocol label switching (MPLS), and traffic modeling and flow classification.