Worst Average Queueing Delay of Multiple Leaky-Bucket-Regulated Streams and Jumping-Window Regulated Stream

Daniel C. Lee

Abstract: **This paper presents deterministic, worst-case analysis of a queueing system whose multiple homogeneous input streams are regulated by the associated leaky buckets and the queueing system that has a single stream regulated by the jumping-window. Queueing delay averaged over all items is used for performance measure, and the worst-case input traffic and the worst-case performance are identified for both queueing systems. For the former queueing system, the analysis explores different phase relations among leaky-bucket token generations. This paper observes how the phase differences among the leaky buckets affect the worst-case queueing performance. Then, this paper relates the worst-case performance of the former queueing system with that of the latter (the single stream case, as in the aggregate streams from many users, whose item arrivals are regulated by one jumping-window). It is shown that the worst-case performance of the latter is identical to that of the former in which all leaky buckets have the same phase and have particular leaky bucket parameters.**

Index Terms: **Leaky bucket, jumping window, queueing.**

I. INTRODUCTION

This paper presents deterministic, worst-case analysis of two queueing systems and relate their results. One is the multiplexer (single-server queue) wherein leaky-bucket-regulated [1] traffic streams merge. Fig. 1 illustrates this system. The other is a single-server queue that has only one input stream, in which the input traffic is regulated by the maximal amount of traffic allowed in each of the non-overlapping time-windows with a fixed size (jumping window [2] regulation). Fig. 2 illustrates this system. As a performance measure for comparing different traffic patterns passing the regulations, the delay averaged over all items is adopted.

Although there are different versions of the leaky bucket scheme, they all share the common basic idea of regulating the rate and burstiness of items' (packets') entry into the network. Each leaky bucket regulator in Fig. 1 is illustrated in Fig. 3. From a source, items (packets) arrive at the regulator's buffer and gets queued. A token must exist in the bucket when an item is admitted into the the system, and one token is consumed out of the bucket every time an item is admitted into the system. Tokens are generated in the bucket periodically with a specified rate r. The token bucket has a fixed size σ. If the token bucket is full at the time of token generation, the newly generated token is discarded. (In some version of the leaky bucket regulator, an item gets dropped if there is no token at the bucket at the time of the item's arrival. Such a regulation can be modelled by buffer size 0.) This scheme is specified by two parameters: The

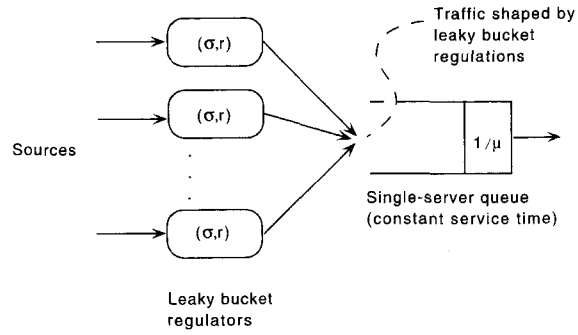


Fig. 1. Multiple leaky-bucket-regulated streams.

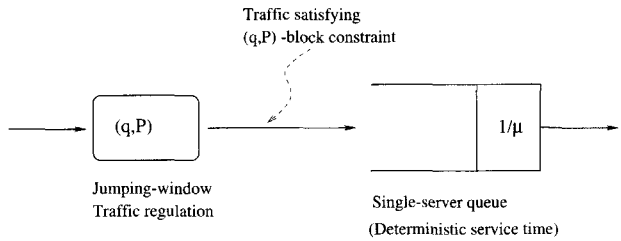


Fig. 2. Single stream regulated by jumping-window.

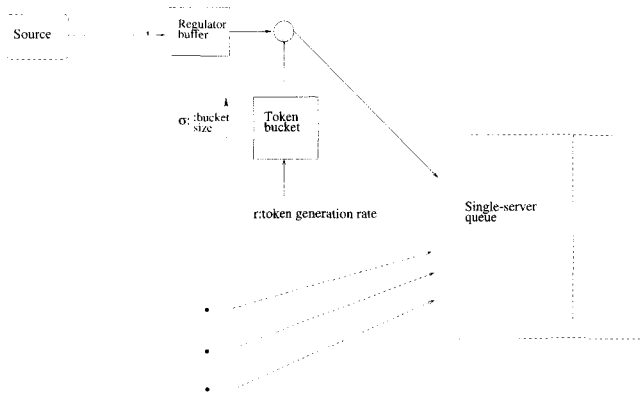


Fig. 3. Leaky bucket regulation.

Manuscript received April 30, 2003; approved for publication by Erik A. van Doorn, Division III Editor, September 4, 2003.

D. Lee is with Department of Electrical Engineering, University of Southern California, U.S.A., email: dclee@usc.edu.

token generation rate r and the bucket size σ. The token generation rate quantifies the allowed rate of admissions, and the bucket size quantifies the allowed burstiness of the traffic admitted. (The regulator's buffer size may affect the delay at the regulator. However, the present paper addresses the worst-case performance at the single-server queue in Fig. 1, and it turns out that regulator buffer size is inconsequential to the worst-case performance at the single-server queue. This will become clear in subsequent sections.)

The queueing system illustrated in Fig. 2 has an input traffic regulation (jumping window) that enforces an upper bound on the number of items admitted in each of the non-overlapping time intervals of a fixed length. We denote by P the length of the time intervals, and by q the upper bound on the amount of traffic in each time interval. Thus, in each of the intervals

$$[(i-1)P, iP] \qquad i = 1, 2, 3, \cdots,$$

up to q items are admitted. By adjusting the parameters q and P, this scheme can regulate the burstiness of the traffic as well as the rate. The long–term average rate of the traffic is kept less than or equal to q/P. For the same value of q/P, a larger P (thus a larger q also) indicates that more bursty traffic is allowed because the larger amount (up to q) of traffic can be concentrated on the same small length of time interval (within the pre-set interval of length P).

The basic difference of the jumping-window traffic regulation from the leaky bucket is that the traffic is regulated in non-overlapping, pre-set time intervals separately. When a new time interval starts, the traffic regulation forgets the history of the traffic entry that took place in the past intervals. (The mechanism is described in more details in Section V.) In the case of the leaky bucket, the credit count is incremented gradually. In the jumping-window traffic regulation discussed in the present paper, the credit count is restored to q at the end of each pre-set time interval of length P. The leaky bucket is widely known as the open-loop congestion regulation for the ATM (Asynchronous Transfer Mode) networks. The jumping window scheme was also considered as a per-connection traffic regulation for broadband packet networks [2]. This scheme was also implemented in a later version of ARPANET [3, p511]. For the IP differentiated [4] services, both schemes can be considered as a traffic conditioning agreement within the service level agreement between DS-domains for a particular traffic class.

Under the formulation of continuous-time queueing system (items can enter the queueing system at any point in the continuum of the time interval), this paper derives the worst–case traffic that passes each of the traffic regulations illustrated in Figs. 1 and 2. Then, the worst-case average queueing delay is quantified as a function of the regulation parameters (σ, r, q, P). A noteworthy result of this paper is that the worst-case traffic passing the leaky bucket regulations is not of the ON-OFF type. Moreover, in the system illustrated in Fig. 1, this paper pays a special attention to the phase differences of the leaky bucket operations regulating multiple streams. In particular, in section V this paper derives the worst-case performance for the case that the token generating times of all leaky bucket operations (for different streams) perfectly coincide and for the case that the token generation times of all the leaky buckets are perfectly interleaved. Then, it is further shown that the performance for the case of an arbitrary phase relation falls in between those two cases. In section V, the worst traffic passing through the jumping window regulation is derived. Then, this paper finds an interesting relation between the two queueing systems. Section V states that the worst aggregated traffic of the multiple leaky-bucket-regulated traffic is identical to the worst traffic regulated by the jumping window scheme as long as the leaky bucket parameters and the frame length are in a certain relation.

This paper considers a queueing system with constant service time, so the queue analysis in this paper will be non-probabilistic in nature, unlike most practices in the queueing theory. The formalism of the constant service time has an advantage of isolating the the effect of the arrival stream's burstiness from the effect of the randomness caused by the service times. In addition, the model with constant service time may be directly applicable to modeling a network service that assures a certain bandwidth all the time (e.g., virtual leased line for the internet differentiated service).

The worst traffic in general is important because it serves to draw a limit in the range of possible network performance and thus guides the network controller to assign control parameters accordingly. It is often desired to simulate the network performance with the worst case input [5]; e.g., in order to have conservative estimate of the network performance for the case that the characteristics of the user population are changing [6]. The vision of the broadband packet-switching network guaranteeing the quality of service for each individual flow (or connection) has motivated much literature addressing the delay and queue length in the worst case [7], [8]. The performance metric employed for that purpose is for the most part the maximum delay among all packets or the maximum queue length in the time horizon. While addressing the worst-case scenario, the present paper considers the queueing delay averaged over all items as the performance metric. For theoretical purpose, the average queueing delay is good for providing fundamental queueing-theoretic intuition, and the parameters of leaky bucket and jumping window regulations are good indications of the intensity and the burstiness of the allowed input traffic classes. In addition, the average queueing delay can be a good performance metric even for the computer/communication network applications (especially, non-real-time applications) when the aggregated traffic consisting of many flows is regulated by the leaky bucket or jumping-window schemes. Such aggregation takes place in the provision of the differentiated service [4]. Indeed, the leaky bucket and its variations have been recommended as the traffic conditioning mechanisms for the internet differentiated service [4]. Thus, the leaky bucket parameters can be used as numerics to be negotiated between the network service provider and its subscribers in the service level agreement (SLA) [4]. The insight into the the relation between these parameters and the performance in the worst-case can contribute as a piece of guiding knowledge to the SLA negotiation.

II. RELATED WORKS

There are an enormous number of papers presenting studies of the leaky bucket regulation in various contexts. (To highlight

some, we note [7]–[30], and more references discussing fundamental issues relating the leaky bucket can be traced from the papers in [31] and the references provided by those papers. Issue [32] contains papers in the context of congestion control in high-speed packet switched networks.) Reference [33] analyzes the cell loss rate of the cell stream regulated by the jumping window. Moreover, references [2], [34]–[37] compare the performance of the leaky bucket and the jumping window. However, the performance metric employed by these references are different from the one discussed in the present paper, and the present paper focuses on the worst traffic.

For the open-loop control approach to the QoS (quality of service) guarantee, there have been studies to identify the worst-case traffic and to evaluate the associated QoS. QoS estimation under the worst case arrival pattern presents a new field in teletraffic theory [38]. The studies in this field can be categorized by the combination of two criteria; the performance measure with which to decide what is the worst and the set of traffic patterns among which to decide the worst one. Much of the existing literature uses the cell (packet) loss rate as the performance measure and considers the set of traffic passing through the leaky bucket regulation [1]. Reference [38] considers the cell loss ratio(CLR) as the performance measure of a multiplexer with deterministic bandwidth, wherein different traffic sources are multiplexed. Under the assumption that the cell arrival patterns are stationary and ergodic, upper bounds of the cell loss ratio ("conservative CLR estimation") are elaborately defined, and then the worst case cell arrival patterns for a tightly bounding conservative CLR estimation were identified from the set of patterns conforming the leaky bucket constraint. References [5], [39], [40] again consider CLR as a performance measure of the multiplexer, wherein input traffic streams are all from the leaky-bucket-based sources. Various cases were discussed; for example, in some cases the traffic sources input to the multiplexer are constrained to have identical patterns (homogeneous traffic), and in other cases without such constraint. In particular, much of the literature discussed whether the on-off process is the worst pattern. In some cases (e.g., homogeneous sources and unbuffered multiplexer), the on-off process was proven to be the worst [39]. However, [5] and [39] provide cases wherein the three-state source causes worse cell loss ratio than the on-off sources. Reference [41] employs loss rate, but for the application of multimedia communication, any traffic failing to meet a certain delay requirement is counted as loss. (This performance measure has been also used in [42] under a completely deterministic formalism.)

References [40], [43], [44] consider as a performance measure the queue length distribution of the multiplexer with an infinite buffer. Again, with this performance measure, the issue is discussed of whether the on-off pattern is the worst one passing through the leaky bucket. Reference [43] studies the traffic pattern consisting of periodic bursts of a maximum length under the cell delay variation constraint at the peak rate followed by a silence period. This is an on-off pattern, and the papers study the queue length distribution for the cases that multiple sources with such traffic patterns are multiplexed. Reference [44] provides through simulation a traffic pattern that results in the queue length's survival function (complementary cumulative distribution) worse than the on-off pattern. In fact, [40] considers both the queue length distribution and the cell loss rate as performance measures and compares the on-off pattern and the pattern presented in [44]. The simulation results indicate that the on-off pattern exhibits the worse cell loss rate yet better queue length distribution than the pattern presented in [44]. In [45], the performance measure in determining the "worst" is the variation of the inter-arrival times. With this measure, the worst traffic passing through leaky bucket regulations is evaluated.

Most literature mentioned above views the traffic as a stationary stochastic process. Even for the periodic traffic pattern, a random phase is used to make the traffic a stationary stochastic process. The present paper does not view the input traffic as a stochastic process but constructs the worst pattern through a deterministic optimization procedure. In fact, the present paper explores the effect of phase relation between the multiple streams instead of randomizing the phase. Also, the present paper considers the queueing delay average over all items as a performance criterion for traffic comparison, not the loss rate.

For users requiring deterministic QoS constraints, the studies relating to obtaining tight bounds on the worst delay of traffic regulated by leaky bucket have been elegantly presented in various contexts in different forms [7]–[10]. This approach can be also viewed as bounding performance of the worst traffic where the performance measure is the maximum delay experienced by traffic. The present paper considers the queueing delay averaged over all items, in contrast to the worst possible delay experienced by an item, as a performance criterion for traffic comparison. Results on the average queueing delay of the leaky-bucket-regulated traffic was presented as early as in [19][1]. More recent example of works on that performance measure can be found in [30].

III. PRELIMINARIES

Before discussing main results, we note some properties of the queueing system that we will use in our analysis. We want to relate the queueing delay averaged over all items with the queueing delay averaged within individual busy periods. We denote by w_i the i-th item's queueing delay in the queue. By "queueing delay", we refer to the time between the admission of an item till the beginning of the service of that item. In any admission schedule of a stable queueing system, the resulting sample path of the queue length will be a sequence of busy periods. We denote the number of admissions in the n-th busy period by ν_n. We denoted by R_n the sum of the queueing delays of these ν_n items. Then, the queueing delay per item averaged within the n-th busy period is R_n/ν_n. The following lemma relates this quantity with the queueing delay averaged over all items.

Lemma 1: For any input schedule, if the number of items served in individual busy periods is bounded (i.e. $\{\nu_n | n = 1, 2, \cdots\}$ is bounded), and service times of items are bounded, then we have

$$\limsup_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} w_m \leq \sup_k \frac{R_k}{\nu_k}.$$

[1]Some results in the present paper have been presented in *IEEE INFOCOM'94*.

Proof: Denote by $N(m)$ the number of completed busy periods up to the admission time of the mth item. Define $L(m) = \sum_{n=1}^{N(m)} \nu_n$. We then have

$$\begin{aligned}
&\limsup_{M\to\infty} \frac{\sum_{m=1}^{M} w_m}{M} \\
&= \limsup_{M\to\infty} \left[\frac{\sum_{n=1}^{N(M)} R_n}{M} + \frac{\sum_{m=L(M)+1}^{M} w_m}{M} \right] \\
&\le \limsup_{M\to\infty} \frac{\sum_{n=1}^{N(M)} R_n}{\sum_{n=1}^{N(M)} \nu_n} + \limsup_{M\to\infty} \frac{\sum_{m=L(M)+1}^{M} w_m}{M} \\
&= \limsup_{M\to\infty} \frac{\sum_{n=1}^{N(M)} R_n}{\sum_{n=1}^{N(M)} \nu_n} \\
&= \limsup_{N\to\infty} \frac{\sum_{n=1}^{N} R_n}{\sum_{n=1}^{N} \nu_n} \\
&= \limsup_{N\to\infty} \frac{\sum_{n=1}^{N} \nu_n \frac{R_n}{\nu_n}}{\sum_{n=1}^{N} \nu_n} \\
&\le \limsup_{N\to\infty} \frac{\sum_{n=1}^{N} \nu_n \sup_k \frac{R_k}{\nu_k}}{\sum_{n=1}^{N} \nu_n} = \sup_k \frac{R_k}{\nu_k}.
\end{aligned}$$

□

This lemma enables us to focus on only one busy period in order to find the worst arrival schedule to a queueing system. Note that the queueing systems discussed in this paper satisfy the assumptions of this lemma for any allowable set of arrivals, as long as the parameters of traffic regulations guarantee the arrival rate less than the service rate.

We now note elementary properties of a single busy period of the queueing system with constant service time. We denote by $\alpha(t)$ the number of admissions (arrivals) until t time unit has elapsed since the beginning of the busy period. Note that function $\alpha(t)$ fully specifies the admission schedule within the busy period. We refer to this function as "cumulative admission function". Denote by $1/\mu$ the service time. Then, within the busy period, $\alpha(t) - \lfloor \mu t \rfloor$ is the number of items in the queueing system. For a busy period that serves a fixed number of items, say ν, the total queueing delay is

$$\int_0^{\nu/\mu} \{\alpha(t) - \lfloor \mu t \rfloor\}\, dt - \frac{\nu}{\mu},$$

and thus the following lemma is obvious.

Lemma 2: For schedules, $\alpha(t), \tilde{\alpha}(t)$, admitting the same number of items, ν in a single busy period, if $\alpha(t) \ge \tilde{\alpha}(t)$, $0 \le \forall t \le \nu/\mu$, then schedule $\alpha(t)$ results in total queueing delay no less than $\tilde{\alpha}(t)$.

IV. WORST AGGREGATED TRAFFIC PASSING HOMOGENEOUS LEAKY BUCKETS

In this section, we derive the worst-case performance of the queueing system into which parallel leaky-bucket-constrained item (packet) streams are fed (Fig. 1). We denote by S the number of traffic streams in Fig. 1. The performance measure is the queueing delay averaged over all items without regard to their sources. We pay keen attention to the phase relation between the leaky bucket regulations.

A. Identical Leaky-Bucket Phase

Suppose that all S leaky bucket operations generate tokens at the same time; therefore, S tokens are generated simultaneously every $1/r$ time units. Consider the admission schedule generated by the following algorithm:

Algorithm 1:

1. Wait until all sources have a full token bucket.
2. Immediately prior to the next token generation, each source admits σ items; immediately after this token generation, each source admits another item.
3. Each source admits one item at each of the next J token generation epochs; go to 1.

For this admission pattern, the total queueing delay per busy period is

$$\sum_{i=0}^{S\sigma+S-1} i\frac{1}{\mu} + \sum_{j=1}^{J}\sum_{l=0}^{S-1} \left\{ S\sigma\frac{1}{\mu} + j(\frac{S}{\mu} - \frac{1}{r}) + l\frac{1}{\mu} \right\},$$

$$\text{for such } J \text{ as } S\sigma\frac{1}{\mu} + J(\frac{S}{\mu} - \frac{1}{r}) \ge 0.$$

The number of admitted items in this busy period is $S\sigma+S+JS$, so the average queueing delay per item in this busy period is

$$\begin{aligned}
&h(J,\sigma,r,S) \\
&\equiv \frac{1}{S\sigma+S+JS} \times \\
&\quad \left[\sum_{i=0}^{S\sigma+S-1} i\frac{1}{\mu} + \sum_{j=1}^{J}\sum_{l=0}^{S-1} \left\{ S\sigma\frac{1}{\mu} + j(\frac{S}{\mu} - \frac{1}{r}) + l\frac{1}{\mu} \right\} \right] \\
&= J\frac{1}{2}(\frac{S}{\mu} - \frac{1}{r}) + \frac{S\sigma+S-1}{2\mu} + \frac{\sigma}{2r} - \frac{\sigma}{2r}\left(\frac{\sigma+1}{\sigma+1+J}\right) \\
&= J\frac{1}{2}(\frac{S}{\mu} - \frac{1}{r}) + \frac{S\sigma+S-1}{2\mu} + \frac{\sigma}{2r}\frac{J}{\sigma+1+J}. \qquad (1)
\end{aligned}$$

Among the input patterns generated by Algorithm 1, let us consider which parameter J yields the maximal average queueing delay per item in a busy period. By searching the value of J that makes $\partial h/\partial J$ vanish with the domain of h extended to the set of nonnegative real values of J and by noting unimodality of h for nonnegative J, we can derive the integer maximum:

$$J^* = \begin{cases} 0 & \text{if } Sr/\mu \le 1/(\sigma+1) \\ J_l & \text{if } Sr/\mu > 1/(\sigma+1) \text{ and} \\ & \quad h(J_l,\sigma,r,S) \ge h(J_h,\sigma,r,S) \\ J_h & \text{if } Sr/\mu > 1/(\sigma+1) \text{ and} \\ & \quad h(J_l,\sigma,r,S) < h(J_h,\sigma,r,S), \end{cases} \qquad (2)$$

where

$$J_l \equiv -(\sigma+1) + \left\lfloor \sqrt{\frac{\sigma(\sigma+1)}{1-Sr/\mu}} \right\rfloor, \qquad (3)$$

$$J_h \equiv -(\sigma+1) + \left\lceil \sqrt{\frac{\sigma(\sigma+1)}{1-Sr/\mu}} \right\rceil. \qquad (4)$$

Theorem 1: For S sources with leaky buckets identical in phase, each with rate r and bucket size σ, the average queueing delay per item has the following upper bound:

$$\limsup_{M\to\infty} \frac{1}{M}\sum_{m=1}^{M} w_m \le h(J^*, \sigma, r, S).$$

This upper bound is attained by Algorithm 1 with parameter J^* of (2).

Proof: Consider the schedule of admitting a fixed number of items, say ν in a busy period to maximize the total queueing delay. Ignoring the distinction between different sources, we can describe the state of token generation by (x, q), where x is the total number of tokens in all the buckets, and q is the remaining time until the next token generation (in bulk). Note that $0 \le x \le S\sigma$, $0 < q \le 1/r$, and S tokens are generated at each token generation epoch. Consider the schedule that admits items according to Algorithm 1 (with sufficiently large J) until all ν items are admitted. (Note that the last bulk admission may not have size S, depending on the number ν.) Denote by $\alpha_c(t)$ the cumulative admission function for this schedule; that is, the number of admissions until time t has elapsed since the beginning of the busy period. Then,

$$\alpha_c(t) = \min(S\sigma + iS, \nu), \forall t \in [\frac{(i-1)}{r}, \frac{i}{r}), i = 1, 2, \cdots. \quad (5)$$

Denote by $\tilde{\alpha}_c(t)$ the cumulative admission function for an arbitrary schedule, which starts the busy period at an arbitrary state of token generation (x, q) under this coinciding token generation pattern. Note that in each interval $[(i-1)/r, i/r), i = 1, 2, \cdots$, only S tokens are generated. Therefore, we have

$$\tilde{\alpha}_c(t) \le \min(x + iS, \nu) \le \min(S\sigma + iS, \nu) = \alpha_c(t), \quad \forall t \in [\frac{(i-1)}{r}, \frac{i}{r}), \ i = 1, 2, \cdots,$$

with the beginning of the busy period as the frame of reference $t = 0$. Therefore, from Lemma 2, $\alpha_c(t)$ maximizes the total queueing delay, and thus the average queueing delay, for a fixed number of items. We claim that the number of admissions, ν^* that maximizes the average queueing delay per item in a busy period satisfies

$$\nu^* = S\sigma + S + JS \quad \text{for some integer } J. \quad (6)$$

This is proved in Appendix I. Therefore, the maximal average queueing delay per item in a busy period is $h(J, \sigma, r, S)$ for some J. Hence, the maximal average queueing delay per item in a busy period is $\max_J h(J, \sigma, r, S) = h(J^*, \sigma, r, S)$. From Lemma 1, for any admission schedule,

$$\limsup_{M\to\infty} \frac{1}{M}\sum_{m=1}^{M} w_m \le h(J^*, \sigma, r, S),$$

and this bound is attained by Algorithm 1. □

B. Perfectly Interleaved Phase

Tokens for each source are generated periodically at every $1/r$ time unit. In this subsection we consider the case that the token generation times of S sources are perfectly interleaved, so that the time between the token generations of different sources is exactly $1/(Sr)$. In this case, the tokens for S leaky buckets are generated cyclically. The queueing delay averaged over all items depends only on the overall admission schedule of items without regard to which admission is for which source. The overall admission schedule is only constrained by the total number of tokens available in the buckets and the token generation schedule without regard to which tokens for which sources. In this case of perfectly interleaved token generations, tokens are generated periodically at period $1/(Sr)$, and the total number of tokens in all the buckets can be up to $S\sigma$. In fact, the mechanics of tokens generation is exactly identical to the case of the single source with rate Sr and bucket size $S\sigma$ if the distinction between tokens for different sources is ignored. Therefore, the worst case average queueing delay is identical to the case of a single source. The expression is $\max_J h(J, S\sigma, Sr, 1) \equiv h(J_p, S\sigma, Sr, 1)$. Note that the worst-case average queueing delay is constructed by the busy period starting with burst of size $S\sigma + 1$ followed by J_p arrivals $1/(Sr)$ apart from each other in time.

C. Arbitrary Phase

Theorem 2: The average queueing delay in the worst case under an arbitrary leaky-bucket phase relation lies between the worst-case bounds for the 'perfectly interleaved phase' and the 'identical leaky-bucket phase' That is,

$$h(J_p, S\sigma, Sr, 1) \le \liminf_{M\to\infty} \frac{1}{M}\sum_{m=1}^{M} w_m \le \limsup_{M\to\infty} \frac{1}{M}\sum_{m=1}^{M} w_m \le h(J^*, \sigma, r, S).$$

Proof: Suppose that we try to make the total queueing delay of a busy period as large as possible with a fixed number of admissions, ν. Denote by $\tilde{\alpha}(t)$ the cumulative admission function for an arbitrary admission schedule of the ν items for an arbitrary leaky-bucket phase relation, which starts the busy period at an arbitrary state of the token generation (x, q). (Variable x is the total number of tokens in buckets, and q is the time until the next token generation. We refer to the beginning of the busy period as $t = 0$.) We then have constraint

$$\tilde{\alpha}(t) \le \min(x + iS, \nu), \quad \forall t \in [\frac{(i-1)}{r}, \frac{i}{r}), \quad i = 1, 2, \cdots,$$

because in each time interval $[\,(i-1)/r\,,\ i/r\,)$ exactly one token is generated for each source. Thus, we have

$$\tilde{\alpha}(t) \le \min(x + iS, \nu) \le \min(S\sigma + iS, \nu) = \alpha_c(t), \quad \forall t \in [\frac{(i-1)}{r}, \frac{i}{r}), \quad i = 1, 2, \cdots,$$

(The last equality was mentioned in (5).) Therefore, for any fixed number of admissions, $\alpha_c(t)$ results in the largest total queueing delay in one busy period (Lemma 2). Therefore, using Lemma 1, we prove

$$\limsup_{M\to\infty} \frac{1}{M}\sum_{m=1}^{M} w_m \le h(J^*, \sigma, r, S).$$

Now we need to compare the perfectly interleaved phase with general phase relations. Consider again maximizing the total delay of a single busy period with a fixed number of items, ν. For the perfectly interleaved phase, let $\alpha_p(t)$ be the maximal arrival schedule described at the end of Section IV-B. Take an instance of an arbitrary phase relation among leaky buckets, which we denote as PH_a. Without loss of generality, let us say that a token for source 0 is generated at time 0, and that the following token generation for source $1, 2, \cdots, S-1$ occurs at time $0 \le g_1 \le g_2 \le \cdots \le g_{S-1} < 1/r$, respectively. Each source generates a token with period $1/r$, so source k generates tokens at

$$m\frac{1}{r} + g_k \qquad m = 0, 1, 2, \cdots.$$

Without loss of generality, we can say that source k of the 'perfectly–interleaved–phase' system generates a single token at times

$$m\frac{1}{r} + \frac{k}{Sr} \qquad m = 0, 1, 2, \cdots.$$

Define $\xi_k \equiv g_k - k/(Sr)$. Suppose $\xi_k \le 0, \forall k \in \{0, 1, \cdots, S-1\}$. Without loss of generality we can start the busy period for the 'perfectly interleaved phase' at time 0 and use the maximal schedule $\alpha_p(t)$. For the case PH_a we can abstain from admitting an item long enough before time 0 so that the token bucket is full immediately before time 0. That way, we can start a busy period with the maximum bulk admission of $S\sigma + 1$ items (or more if multiple leaky buckets generate tokens at time 0) and admit an item at each subsequent token generation. We denote by $\bar{\alpha}(t)$ the number of admissions in $[0, t]$ (cumulative admission function) according to this schedule for PH_a. Then, because $\xi_k \le 0, \forall k$, the ith admission of $\bar{\alpha}(t)$ takes place no later than the ith admission of schedule $\alpha_p(t)$ for $i = 1, 2, \cdots, \nu$. Equivalently, we have $\bar{\alpha}(t) \ge \alpha_p(t)$, $\forall t$. Suppose $\xi_k > 0$ for some k. Take the largest ξ_k and define

$$\xi_{k^*} \ge \xi_k, \qquad \text{for all } k.$$

Consider starting a busy period for the instance PH_a at time g_{k^*} with the maximum bulk admission, in which the bulk size is at least $S\sigma + 1$. Again, the subsequent admissions take place at token generation times without skipping one. Denote by $\bar{\alpha}(t)$ the number of admissions according to this schedule until time t has elapsed since the beginning of the busy period. We compare this schedule with the maximal schedule for the 'perfectly interleaved phase', $\alpha_p(t)$, which also start the busy period with the bulk admission of $S\sigma + 1$ items. For the purpose of comparison, we start the busy period of the perfectly interleaved case at time $k^*/(Sr)$ (token generation time of source k^*). Then, we compare the perfectly interleaved phase and PH_a regarding the time gap from the beginning the busy period to the token generation of source k. The difference between the two cases in time gap is

$$(g_k - g_{k^*}) - (\frac{k}{Sr} - \frac{k^*}{Sr}) = \xi_k - \xi_{k^*} \le 0, \qquad \text{for each } k.$$

(Note periodic nature of each source's token generation.) Therefore, the elapsed time from the beginning of the busy period to each admission in schedule $\bar{\alpha}(t)$ is no more than the one in schedule $\alpha_p(t)$. Thus, we have $\bar{\alpha}(t) \ge \alpha_p(t)$, and $\bar{\alpha}(t)$ results in the larger average queueing delay within a busy period (from Lemma 2). Therefore, using Lemma 1, we prove that there exists a schedule in PH_a that yields no less average queueing delay than the maximal schedule for the 'perfectly interleaved phase'. Thus,

$$h(J_p, S\sigma, Sr, 1) \le \liminf_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} w_m.$$

□

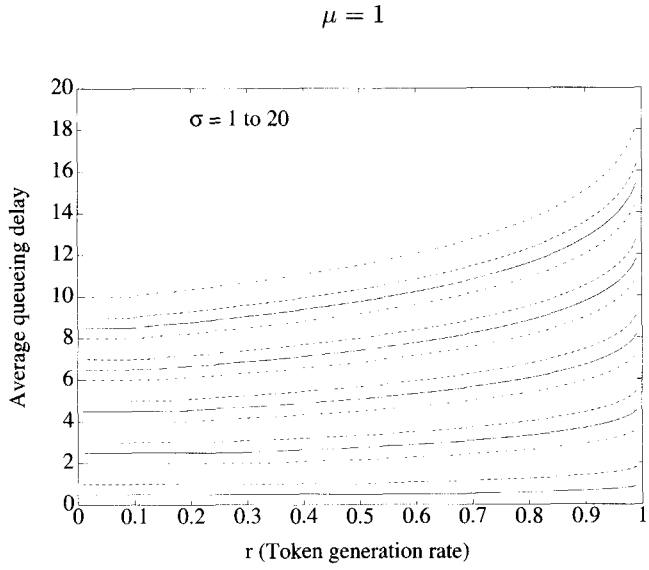


Fig. 4. Average delay in the worst case vs. token generation rate.

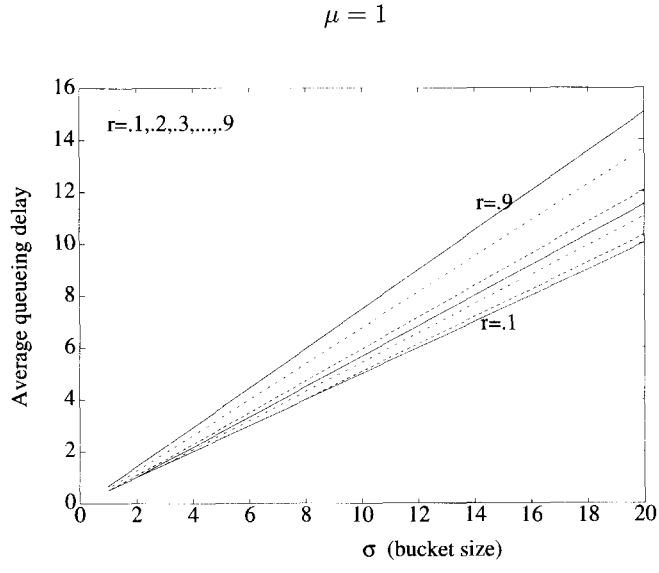


Fig. 5. Average delay in the worst case vs. bucket size.

D. On the Worst-Case Performance

For simple intuition we first consider the case of single source ($S = 1$) with bucket size σ and rate r. Theorem 1 and Algorithm 1 indicate that the worst packet admission pattern is not ON–

OFF type. The worst traffic repeats the following three patterns: Bulky admission with bulk size $\sigma + 1$ at a token generation, admission at each of the next J^* token generations, and then no admission at the next σ token generations. Note that this pattern is not of on–off type. Figs. 4 and 5 show for the case $S = 1$ the relationship between the worst average delay per customer, $h(J^*, \sigma, r, 1)$ and the leaky bucket parameters. Fig. 5 indicates that the relation between the queueing delay and σ is very close to a linear relation. Let us compute the asymptotic slope. For sufficiently large σ, we have $r > \mu/(1+\sigma)$, so the ratio of the queueing delay to σ is

$$\frac{h(J^*, \sigma, r, 1)}{\sigma} = \frac{1}{2}\frac{1}{\mu} - \frac{J^*}{2\sigma}\left(\frac{1}{r} - \frac{1}{\mu}\right) + \frac{1}{2r}\frac{J^*}{\sigma + 1 + J^*}.$$

Also, from (3) and (4),

$$\lim_{\sigma\to\infty}\frac{J_l}{\sigma} = \lim_{\sigma\to\infty}\frac{J_h}{\sigma} = \lim_{\sigma\to\infty}\frac{J^*}{\sigma} = -1 + \frac{1}{\sqrt{1 - r/\mu}}. \quad (7)$$

Finally, we have

$$\lim_{\sigma\to\infty}\frac{h(J^*, \sigma, r, 1)}{\sigma} = \frac{1 - \sqrt{1 - r/\mu}}{r/\mu}\frac{1}{\mu}. \quad (8)$$

Therefore, the queueing delay is approximated by the following expression:

$$h(J^*, \sigma, r, 1) \simeq \frac{1 - \sqrt{1 - r/\mu}}{r/\mu}\frac{1}{\mu}\sigma.$$

This equation indicates the effect of r, σ, and their interaction effect on the worst-case average delay. The percentage error of this approximation for $\mu = 1$,

$$\frac{h(J^*, \sigma, r, 1) - \frac{1-\sqrt{1-r}}{r}\sigma}{h(J^*, \sigma, r, 1)},$$

is plotted in Fig. 6. For the case of multiple sources with identical leaky bucket parameters, the case of identical phase results in longer worst-case average delay than the perfectly interleaved phase. The worst-case performances of all other phase relations fall in between (Theorem 2).

Now we compare the case of single traffic source having leaky bucket parameters $(S\sigma, Sr)$ and multiple $(S > 1)$ sources each having leaky bucket parameters (σ, r). Theorem 2 indicates that multiple sources exhibit the delay performance worse than or equal to a single source for a given amount of total bucket capacity $(S\sigma)$ and token generation rate (Sr). However, the asymptotic growth of the maximal average queueing delay as a function of the total bucket size $S\sigma$ is the same for both cases, as seen by the following equations derived from (1), (3), (4), (7), and (8).

$$\lim_{\sigma\to\infty}\frac{h(J^*, \sigma, r, S)}{S\sigma} = \frac{1 - \sqrt{1 - Sr/\mu}}{Sr/\mu}\frac{1}{\mu} = \lim_{\sigma\to\infty}\frac{h(J_p, S\sigma, Sr, 1)}{S\sigma}.$$

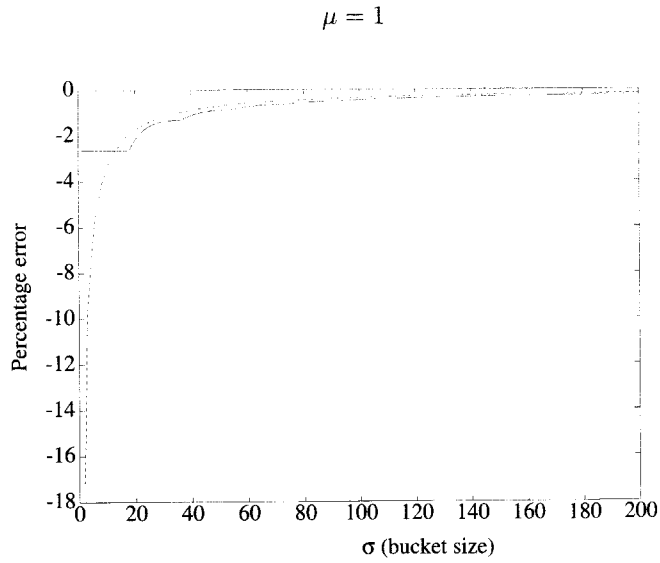


Fig. 6. Percentage error.

V. JUMPING-WINDOW REGULATION

In this section, we analyze the arrival pattern constrained by the jumping–window scheme that results in the largest average queueing delay. Up to q items are admitted in each of the intervals

$$[(i-1)P, iP] \qquad i = 1, 2, 3, \cdots.$$

We refer to this constraint as "(q, P) – *block constraint*." We refer to a sequence of admission times as an input schedule. In this section, we examine the worst input schedule under this regulation, in terms of the average queue length or queueing delay. In studying the worst-case performance of this (q, P) – *block constraint*, we make the following interesting observation:

Theorem 3: The set of input schedules allowed by the (q, P)-block constraint is identical to the set of aggregated input schedules of q streams each of which is regulated by the leaky bucket regulation with parameters $\sigma = 1$, $r = 1/P$ and whose leaky bucket regulations are identical in phase.

Proof: We compare a (q, P) – *block constraint* with the leaky bucket regulation of multiple sources that have the leaky buckets operating in phase. Consider a system with q streams where each stream is regulated by a leaky bucket with token generation rate $r = 1/P$ and bucket size $\sigma = 1$, as illustrated in Fig. 7. Consider the case that token generations of all q leaky buckets are synchronized (identical phase). We assume that for each stream, tokens are generated at times

$$t = 0, P, 2P, 3P, \cdots,$$

and that token buckets are full immediately prior to $t = 0$. Immediately after each of these token generation times, each of q token buckets contains exactly one token, so the total number of tokens is q. Therefore, in each interval $(\ nP\ ,\ nP + P\ ]$, no more than q items are admitted into the queue in Fig. 7. Therefore, any allowable traffic into the queue in Fig. 7 satisfies the (q, P) – *block constraint*. Also, we can construct

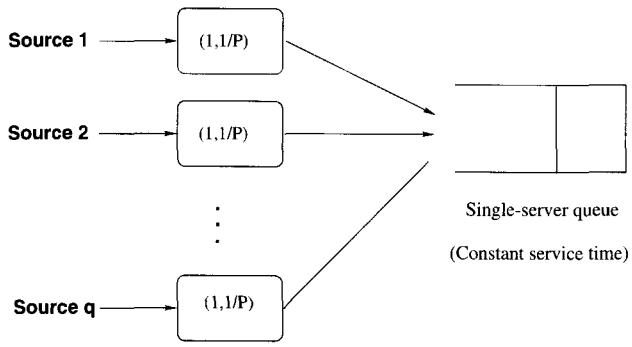


Fig. 7. Queue with constant service time and with multiple input streams constrained by the leaky bucket operations.

any traffic satisfying the (q, P) − *block constraint* as the legitimate traffic entering the queue in Fig. 7. Suppose that an input schedule satisfying the (q, P) − *block constraint* admits x_n items in time interval $(nP, nP + P]$ for $n = 0, 1, \cdots$. Due to (q, P) − *block constraint*, we have $x_n \leq q, n = 0, 1, 2, \cdots$. Therefore, we can select x_n sources in Fig. 7 to admit an item in each interval at the time of item admission under (q, P) − *block constraint*. This is possible because all q buckets have a token at the beginning of each time interval. We have established that the set of admission schedules satisfying the (q, P) − *block constraint* is identical to the set of admission schedules that can be generated by multiple streams under leaky bucket regulations having an identical token generation phase and parameters $\sigma = 1, r = 1/P$. □

There are two interesting implications of Theorem 3. First, in order to find the worst-case traffic passing the single (q, P) − *block constraint*, we can refer to the results regarding the worst-case aggregated traffic the streams regulated by multiple leaky buckets operating in identical phase, which have been presented in Section IV-A. Therefore, the traffic that maximizes the average queueing delay for the system described in Fig. 7 is also maximal for the case of (q, P) − *block constraint*. The second implication is about aggregating smooth streams with identical characteristics into a class, as in the differentiated service class of smooth flows. There is a particular equivalency in terms of the worst-case performance between regulating individual streams of smooth traffic and regulating the aggregate of those streams. Individual stream regulated with leaky bucket size $\sigma = 1$ can be considered smooth. As illustrated by the worst-case traffic in previous sections, the aggregate of those streams can create bursts, especially if all the leaky buckets are somehow synchronized in their token generation (identical phase). As far as the worst-case performance is concerned, regulating q streams individually, each stream to be smooth with leaky bucket parameters $\sigma = 1$ and a rate $1/P$, and regulating their aggregate with (q, P) − *block constraint* have the same effect.

VI. SUMMARY AND DISCUSSIONS

This paper presented the worst-case aggregate traffic arrival patterns for the queueing system whose multiple homogeneous input streams are constrained by the associated leaky buckets and the queueing system that has a single stream constrained by (q, P) − *block constraint*. Queueing delay averaged over all items is used for performance measure under the assumption of the constant service time. In the former queueing system, the worst-case traffic passing through the leaky bucket regulators is not of ON-OFF type. Regarding the phase relations among leaky-bucket token generations, as far as the worst traffic is concerned, the in-phase case yields the largest average queueing delay, and the perfectly spaced phase relation yields the smallest average. It was also shown that the worst-case arrival pattern and the corresponding average for the (q, P) − *block constraied* input are identical to those for the q-input-stream queueing system whose input streams are regulated by q leaky buckets operating in phase with $\sigma = 1$ and $r = 1/P$.

In Section IV, this paper considered a queueing system fed by multiple input streams with an identical leaky bucket regulation, (σ, r). Such a queueing system models the configuration in which a particular bandwidth is allocated for serving a particular class of streams; that is, the class of streams that have the same traffic characteristics. Multiple streams with an identical leaky bucket regulation lend themselves to tractable derivation of the worst aggregate traffic, as seen in this paper.

An interesting generalization for future study is to consider the traffic aggregation of the streams regulated by different leaky bucket parameters, (σ_0, r_0), (σ_1, r_1), (σ_2, r_2), $\cdots$, (σ_{S-1}, r_{S-1}). Identifying the worst aggregate traffic passing through these regulations is left for future study. The full-scale analysis seems challenging. However, we can start from a partial generalization in which sources have the same token generation rate; namely, the case (σ_0, r), (σ_1, r), (σ_2, r), $\cdots$, (σ_{S-1}, r). In this case, from Lemmas 1 and 2 again, it is expected that the following algorithm, with optimized variable $\hat{J}$, constructs the worst-average aggregate traffic for the case of the identical leaky-bucket phase.

Algorithm 2:

1. Wait until all sources have a full token bucket; that is, σ_0, $\sigma_1, \sigma_2, \cdots, \sigma_{S-1}$.
2. Immediately prior to the next token generation, each source i admits σ_i items, $i = 0, 1, 2, \cdots, S - 1$; immediately after this token generation, each source admits another item.
3. Each source admits one item at each of the next $\hat{J}$ token generation epochs; go to 1.

It is further expected that the maximal-average-delay traffic can be constructed in a similar way for the case of the perfectly interleaved phase. Furthermore, it is expected that Theorems 1 and 2 will be easily extended to the case of different bucket sizes with an identical token generation rate.

The next level of generalization is for the case of S streams with arbitrary leaky bucket parameters (σ_i, r_i), $i = 0, 1, 2, \cdots, S - 1$. In order to identify the worst aggregate traffic passing through this set of regulations, we can still use Lemmas 1 and 2 of this paper. As done in this paper, the key step in identifying the worst traffic would be to construct a busy pe-

riod that yields the maximum possible delay averaged over one busy period. Let us describe the state of the buckets by (x_i, ϵ_i), $i = 0, 1, 2, \cdots, S-1$, where x_i denotes the number of tokens in Source i's bucket and ϵ_i is the time until its next token generation. One variable to determine in order to find the maximizer busy period is the state of the leaky buckets at the beginning of the busy period. From experiences gained from Section IV, we can expect that such a maximizer busy period will start when all buckets are full; namely, when $x_i = \sigma_i, \forall i$. However, how to choose $\epsilon_i, i = 0, 1, \cdots, S-1$ and the number of items admitted in the busy period still need to be determined in order to maximize the average delay per busy period. Such optimization is left for future work. Another concern to address is whether the maximizer busy period can be repeated. It is expected that the maximizer busy period can be repeated as long as token generation rates have a relation:

$$\frac{n_0}{r_0} = \frac{n_1}{r_1} = \cdots \frac{n_{S-1}}{r_{S-1}},$$

for some set of integers $n_0, n_1, \cdots, n_{S-1}$.

Unlike the case of homogeneous streams discussed in this paper, in the case of different bucket sizes for different streams, we cannot expect that the aggregate traffic yielding the maximal average queueing delay will also have the maximal throughput. For example, let us consider a simple case of two sources with $\sigma_0 < \sigma_1$. Let us further assume that both sources have an identical token generation rate r and identical phase. According to the conjectured maximal-average-delay scenario presented in Algorithm 2, after a busy period, source 0 will lose a token (or tokens) until the next busy period starts. Thus, we expect that in the aggregate traffic of inhomogeneous streams yielding the worst average delay may not yield the worst average queue length because it is not expected to exert the maximal allowed throughput.

APPENDIX

I. Proof of (6)

Because ν^* maximizes the average queueing delay per item in a busy period,

$$\frac{\sum_{i=1}^{\nu^*-1} w_i}{\nu^* - 1} \leq w_{\nu^*}. \tag{9}$$

Suppose $\nu^* = S\sigma + S + (J-1)S + l$, $1 \leq l \leq S-1$. Then, we can admit another item at the same time as the ν^*-th item because the number of tokens generated at the time of the ν^*-th admission is less than the size of bulk admission at that time, and $w_{\nu^*+1} = w_{\nu^*} + (1/\mu)$. We now compare the queueing delay the (ν^*+1)-st item would have, w_{ν^*+1}, and the queueing delay averaged up to ν^*-th item, $(\sum_{i=1}^{\nu^*} w_i)/\nu^*$. We have

$$\begin{aligned} w_{\nu^*+1} - \frac{\sum_{i=1}^{\nu^*} w_i}{\nu^*} &= w_{\nu^*} + \frac{1}{\mu} - \frac{\sum_{i=1}^{\nu^*} w_i}{\nu^*} \\ &= \frac{(\nu^*-1)w_{\nu^*} - \sum_{i=1}^{\nu^*-1} w_i}{\nu^*} + \frac{1}{\mu} \\ &= \frac{\nu^*-1}{\nu^*}\left[w_{\nu^*} - \frac{\sum_{i=1}^{\nu^*-1} w_i}{\nu^*-1}\right] + \frac{1}{\mu} \\ &> 0 \quad \text{using inequality (9).} \end{aligned}$$

This implies that

$$\frac{\sum_{i=1}^{\nu^*} w_i}{\nu^*} < \frac{\sum_{i=1}^{\nu^*+1} w_i}{\nu^*+1}.$$

This contradicts the maximality of ν^*. Therefore, $\nu^* = S\sigma + S + JS$ for some integer J. □

REFERENCES

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

[2] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 325–334, Apr. 1991.

[3] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 2nd ed., 1992.

[4] S. Blake *et al.*, *An architecture for differentiated services*. Internet Society, Dec. 1998. RFC 2475.

[5] B. Erimli, J. Murphy, and J. Murphy, "On worst case traffic in ATM networks," in *Twelfth UK IEE Teletraffic Symp.*, (Windsor, United Kingdom), Mar. 1995.

[6] M. Bonatti and A. A. Gaivoronski, "Worst case analysis of ATM sources with application to access engineering of broadband multiservice networks," in *The Fundamental Role of Teletraffic in the Evolution of Telecommunication Networks* (J. Labetoulle and J. W. Roberts, eds.), pp. 559–569, Amsterdam: Elsevier Science, 1994.

[7] R. L. Cruz, "Calculus for network delay - part I: Network elements in isolation," *IEEE Trans. Infom. Theory*, vol. 37, pp. 114–131, Jan. 1991.

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

[9] R. L. Cruz, "Calculus for network delay - part II: Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, pp. 132–141, Jan. 1991.

[10] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated service networks: The multiple node case," *IEEE/ACM Trans. Networking*, vol. 2, pp. 137–150, Apr. 1994.

[11] H. Ahmadi, R. Guerin, and K. Sohraby, "Analysis of leaky bucket access control mechanism with batch arrival process," in *Proc. GLOBECOM'90*, Dec. 1990, pp. 344–349.

[12] A. W. Berger and W. Whitt, "The brownian approximation for rate–control throttles and the G/G/1/C queue," *Discrete Event Dynamic Systems: Theory and Applications*, pp. 7–60, 1992.

[13] A. W. Berger and W. Whitt, "The pros and cons of a job buffer in a token-bank rate-control throttle," *IEEE Trans. Commun.*, vol. 42, pp. 857–861, Feb./Mar./Apr. 1994.

[14] R. Boorstyn *et al.*, "Effective envelopes: Statistical bounds on multiplexed traffic in packet networks," in *Proc. IEEE INFOCOM 2000*, 2000, pp. 1223–1232.

[15] G. de Veciana, "Leaky buckets and optimal self-tuning rate control," in *Proc. GLOBECOM'94*, 1994, pp. 1207–1211.

[16] A. I. Elwalid and D. Mitra, "Traffic shaping at a network node: Theory, optimum design, admission control," in *Proc. IEEE INFOCOM'97*, 1997, pp. 444–454.

[17] M. G. Hluchyj and N. Yin, "A second-order leaky bucket algorithm to guarantee QoS in ATM networks," in *Proc. IEEE GLOBECOM'96*, 1996, pp. 1090–1096.

[18] J. S. M. Ho, H. Uzunalioğlu, and I. Akyildiz, "Cooperating leaky bucket for average rate enforcement of vbr video traffic in ATM networks," in *Proc. IEEE INFOCOM'95*, 1995, pp. 1248–1255.

[19] D. C. Lee, "Effects of leaky bucket parameters on the average queueing delay: Worst case analysis," in *Proc. IEEE INFOCOM'94*, 1994, pp. 482–489.

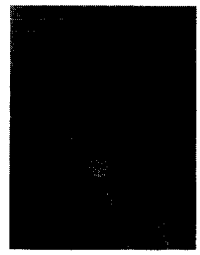
[20] D. S. Holtsinger and H. G. Perros, "Performance of the buffered leaky bucket policing mechanism," in *Proc. TriCom'92: High–Speed Communication Networks*, 1992, pp. 47–69.

[21] I. Khan and V. O. K. Li, "A traffic control mechanism for ATM networks," in *Proc. IEEE GLOBECOM'93*, (Houston, TX), 1993, pp. 1122–1126.

[22] R. Krishnan and J. A. Silvester, "The effect of variance reduction on the performance of the leaky bucket," in *Proc. ICC'95*, (Seattle), 1995, pp. 1974 –1980.

[23] J. F. Kurose, "On computing per–session performance bounds in high–speed multi–hop computer networks," in *Proc. ACM SIGMETRICS/IFIP Performance'92 Conf.*, (Newport, RI), June 1992, pp. 128–139.

[24] S.-K. Kweon and K. G. Shin, "Traffic-controlled rate monotonic priority scheduling of ATM cells," in *Proc. IEEE INFOCOM'96*, (San Francisco, CA), Mar. 1996, pp. 655–662.
[25] F. Lo Presti *et al.*, "Source time scale and optimal buffer/bandwidth tradeoff for heterogeneous regulated traffic in a network node," *IEEE/ACM Trans. Networking*, vol. 7, pp. 490–501, Aug. 1999.
[26] S. Low and P. P. Varaiya, "Burst reducing servers in ATM networks," *Queueing Systems*, vol. 20, pp. 61–84, Sept. 1995.
[27] J. A. S. Monteiro, M. Gerla, and L. Fratta, "Leaky bucket analysis for ATM networks," in *Proc. SBT/IEEE Int. Telecommun. Symp.*, 1990, pp. 498–502.
[28] C.-F. Su and G. de Veclana, "On the overflow probability of deterministically constrained traffic," in *Proc. ICC'97*, (Montreal), 1997, pp. 1704–1708.
[29] M. K. Wong and P. P. Varaiya, "A deterministic fluid model for cell loss in ATM networks," in *Proc. IEEE INFOCOM'93*, 1993, pp. 395–400.
[30] F. M. Guillemin *et al.*, "Extremal traffic and bounds for the mean delay of multiplexed traffic streams," in *Proc. IEEE INFOCOM 2002*, 2002.
[31] R. G. Gallager *et al.*, *IEEE J. Select. Areas Commun.*, vol. 13, Aug. 1995.
[32] K. Sohraby *et al.*, *IEEE J. Select. Areas Commun.*, vol. 9, Sept. 1991.
[33] A. Atkinson, "A traffic control scheme for virtual paths in an asynchronous transfer mode network," in *Proc. GLOBECOM'94*, 1994, pp. 1768–1773.
[34] K. C. Budka, "Stochastic monotonicity and convexity properties of rate-based flow control mechanisms," *IEEE Trans. Automatic Control*, vol. 39, pp. 544–548, Mar. 1994.
[35] P. Castelli, A. Forcina, and A. Tonietti, "Dimensioning criteria for policing functions in ATM netwokrs," in *Proc. IEEE INFOCOM'92*, 1992.
[36] K. Shimokoshi, "Resource management for multimedia broadband access networks," in *Proc. GLOBECOM'95*, 1995, pp. 728–733.
[37] S. Shioda and H. Saito, "Satisfying QOS standard with combined strategy for CAC and UPC," in *Proc. ICC'95*, (Seattle), 1995, pp. 965–969.
[38] T. Tsuchya and H. Saito, "The worst case cell arrival patterns in ATM networks," *IEICE Trans. Commun.*, vol. 81–B, pp. 996–1003, May 1998.
[39] B. T. Doshi, "Deterministic rule based traffic descriptors for broadband ISDN: Worst case behavior and connection acceptance control," in *The Fundamental Role of Teletraffic in the Evolution of Telecommunication Networks* (J. Labetoulle and J. W. Roberts, eds.), Amsterdam: Elsevier Science, 1994, pp. 591–600.
[40] T. Worster, "Modelling deterministic queues: The leaky bucket as an arrival process," in *The Fundamental Role of Teletraffic in the Evolution of Telecommunication Networks* (J. Labetoulle and J. W. Roberts, eds.), Amsterdam: Elsevier Science, 1994, pp. 581–590.
[41] M. Reisslein, K. Ross, and S. Rajagopal, "Guaranteeing statistical qos to regulated traffic: The single node case," in *Proc. IEEE INFOCOM'99*, (New York), Mar. 1999, pp. 1061 –1072.
[42] D. C. Lee, "Worst-case fraction of CBR teletraffic unpunctual due to statistical multiplexing," *IEEE/ACM Trans. Networking*, vol. 4, pp. 98–105, Feb. 1996.
[43] J. M. Barceló, J. García-Vidal, and O. Casals, "Worst-case traffic in a tree network of ATM multiplexers," *IEEE/ACM Trans. Networking*, vol. 8, pp. 507–516, Aug. 2000.
[44] N. Yamanaka, Y. Sato, and K. Sato, "Performance limitation of leaky bucket algorithm for usage parameter control and bandwidth allocation methods," *IEICE Trans. Commun.*, vol. E75–B, pp. 82–86, Feb. 1992.
[45] A. Skliros, "Characterizing the worst traffic profile passing through an ATM-UNI," in *First UK Workshop on Performance Modeling and Evaluation of ATM Networks*, IFIP, 1993, pp. 74–84.



Daniel C. Lee received his Ph.D. (1992) and S.M. (1987) both from the Massachusetts Institute of Technology in Electrical Engineering and Computer Science, B.S. (1985) in Electrical Engineering with Honors, and B.S. (1985) in Mathematics from the University of Maryland. His Ph.D. thesis published in 1992 at MIT and subsequent publications address various aspects of network modeling, performance analysis, control, and management. From 1993 to 1998, Dr. Lee was devoted to network systems engineering in the U.S. Naval Research Laboratory (NRL), Washington, DC. At the Center for Computational Science in NRL, Dr. Lee participated in the development of the object-oriented protocol software framework, CASiNO (Component Architecture for Simulating Network Objects), and network signaling simulators, SEAN (Signaling Entity for ATM Networks) and PRouST (PNNI Routing Simulation Toolkit). At the Naval Space Center in NRL, Dr. Lee developed a proxy agent that allows the interoperation of commercial standard network management system and the network management modules of the U.S. government's information collection and dissemination system, ICEbox. In addition to such development activities, Dr. Lee dedicated most of his time to the network modeling and simulation for improving the reliability, availability, maintainability, operability, and security of the information collection and dissemination systems. In 1998, Dr. Lee joined the Electrical Engineering Department of the University of Southern California and has been continuing the network research. His current interests include quality of service of wireless-, sensor-, WDM-networks, the next-generation internet, and multi-media transport. Dr. Lee's honors include: 1995 Alan Berman Research Publication Award from NRL, 1995 Navy's Outstanding Performance Award in NRL, and 1989 Teaching Award from MIT