An Estimator of Regulator Parameters in a Stochastic Setting

Shahrokh Valaee[1] *and Jean-Charles Grégoire*[2] *

[1] The Edward S. Rogers Sr. Department of Electrical and Computer Engineering,
University of Toronto, 10 King's College Road,
Toronto, Ontario, Canada, M5S 3G4.
Email: valaee@comm.utoronto.ca.

[2] INRS-EMT, Université du Québec,
Place Bonaventure, 800 rue de la Gauchetière Ouest, Suite 6900
Montréal, Québec, Canada, H5A 1C6.
Email: gregoire@inrs-emt.uquebec.ca

February 21, 2005

Abstract

This paper develops a new network provisioning and resource allocation scheme. We introduce the concept of the effective burstiness curve (EBC), which is defined as a percentile of the maximum burstiness curve. For a fixed service rate, EBC represents the size of a buffer for which the probability of buffer overflow is arbitrarily small. We show that EBC is a convex non-increasing function of the service rate. We also introduce the empirical effective burstiness curve (EEBC), an estimator of EBC, which can be obtained with a water-filling algorithm. For discrete queue size, EEBC can be evaluated with a recursive algorithm. The technique is applied to MPEG4 encoded video traces.

Keywords: burstiness curve, leaky-bucket regulator, fluid-flow traffic, quality-of-service, water-filling.

*The completion of this research was made possible in parts by Bell Canada's support through its Bell University Laboratories R & D program.

1 Introduction

There are two approaches to *quality-of-service* (QoS) provisioning in the literature of high-speed networking: the *stochastic* approach and the *deterministic* approach. In the stochastic approach for network provisioning, input traffic is usually modeled by a canonical probability distribution function. The performance is then quantified in terms of the mean or percentiles of the network behavior. The critical issue in stochastic approach is to find an appropriate probabilistic model for data. Most of the fairly rich literature in queueing theory is based on the Markov models for input traffic [1]. However, recent studies show that the classical Markov processes are inefficient models for Internet traffic [2, 3] and variable–bit–rate (VBR) encoded video traces [4, 5]. It has been observed that the autocorrelation function of these traffics decreases very slowly in time (long-range dependence). This is in sharp contrast to Markov models in which the autocorrelation function demonstrates an exponential decrease. Based on these observations, *self-similar* models with *fractional Brownian motion* processes have been proposed [6].

The deterministic approach avoids the hurdles of model mismatch by restraining the input traffic to certain boundaries [7, 8, 9, 10, 11, 12]. The bounds reflect the *worst-case* behavior of source. A source is called *greedy* if its traffic meets the upper bound [13]. It has been shown that a network with greedy sources attains its worst-case behavior. The deterministic approach usually quantifies the worst-case network performance in terms of maximum delay and maximum backlog as functions of the scheduling strategy and the parameters of a *regulator.* A regulator is a device located at network boundary and used to *shape* or *filter* the input traffic. A properly adjusted regulator anticipates occasions of congestion in the network and takes appropriate steps to avert congestion by shaping (delaying) and/or filtering (dropping) input traffic. In other words, a regulator should be capable of matching the needs of the input traffic to the status of network. An appropriately tuned regulator—supplied with a network feedback—decides, on the basis of the history of the source to which it is attached, to transmit, delay, and/or drop (mark) the traffic.

The deterministic approach suffers from a severe drawback: it produces very conservative solutions to network provisioning. Since it assumes that all sources are greedy simultaneously, it usually cannot benefit from *statistical multiplexing gain* which is an important element in aggregating stochastic sources. It has been demonstrated in the literature that the deterministic approach tends to under-utilize network resources [14, 15, 16].

This paper focuses on the development of a recursive approach to the selection of regulator parameters for a fluid-flow stochastic source. We use the *leaky bucket* regulator, which can be modelled by a single server queue with a constant service rate. The objective is to select appropriate leaky bucket parameters, i.e. token pool size and token replenishment rate, such as to, on the one hand, satisfy a certain level of contentment for the user—represented here as a small probability of loss—and on the other hand, be able to benefit from statistical multiplexing gain by aggregating several stochastic sources into a single flow. We do not, however, impose any canonical probability distribution function on the input traffic.

In this paper, the QoS metric is the probability of loss, represented by

$$\mathrm{P}(Q_\rho(t) > \sigma) \leq \epsilon \tag{1}$$

where $\epsilon < 1$ is a QoS index, which is usually very small, $Q_\rho(t)$ is the buffered workload in the single server queue with the constant service rate ρ, and σ is the corresponding buffer size. Note that for computing $Q_\rho(t)$ in (1), we assume that the buffer has infinite storage capacity. Therefore, the left-hand-side of (1) is the probability that the queue size exceeds σ.

Inequality (1) has been used by many researchers for performance analysis; see for instance [17, 18, 19, 20, 21, 22] and references therein. Lo Presti *et al* [17] separate the problem into a bufferless server and a storage system, and argue that the probability of loss is upper bounded by the summation of the probability of loss in each subproblem. Kesidis and Konstantapoulos [18] find the worst case upper bound on the probability of loss when the input traffic passes through a leaky bucket regulator. They assume that the queue size $Q_\rho(t)$ is a stationary process and then find the bound on queue occupancy and queueing delay. Chang *et al* [19] study the same problem under general traffic constraints. Both [18] and [19] assume that the node is a constant rate server. Vojnovic and Le Boudec [20] generalize the results of [18] and [19] to super-additive service curves. Kumaran and Mandjes [21] use the correlation of source traffic to derive approximate upper bounds for (1). Boorstyn *et al* [23] use the Central Limit Theorem and the Chernoff bound to obtain local and global effective envelopes as the upper bounds of a multiplexed traffic of independent flows. They assume that individual flows are regulated by leaky bucket regulators and then show that one can use (1) to improve the statistical multiplexing gain and to increase network utilization. Liebeherr *et al* [22] use the *statistical network calculus* to present a method for computing lower bounds of the service given to a single flow in a network in which service is provisioned to aggregates of flows. They show that a probabilistic service allocation for a single flow can be obtained from the service allocation of an aggregate by subtracting a probabilistic upper bound on the departures from all other flows. The upper bound is obtained by applying the Chernoff bound.

Inequality (1) has also been studied extensively under the condition $\sigma \gg 0$ where the theory of *large deviation* [24, 25] has been used to derive upper bounds on $\mathrm{P}(Q_\rho(t) > \sigma)$ [26, 27, 28]. It has been shown that if the input traffic is a stochastic process with exponentially decaying correlation function, for very large values of σ, one can approximate the upper bound of $\mathrm{P}(Q_\rho(t) > \sigma)$ by an exponentially decreasing function. However, for input traffic with long-range correlation the tail distribution of the queue size is sub-exponential [29, 27].

1.1 Our Approach

The references above share a common approach to the problem. They all provide an upper bound on the tail of loss probability. In other words, they formulate ϵ as a function of σ. Our approach in this paper is different. Here, for a fixed ϵ, we obtain the smallest vector (σ, ρ) that satisfies (1). We observe that (σ, ρ) is not unique, and therefore introduce the concept of *effective burstiness curve* (EBC), which we define as the size of the buffer, σ, in a single-server queue with constant service rate, ρ, with the probability of buffer overflow smaller than a positive constant ϵ. Indeed, a percentage of traffic will be lost (or marked as violating traffic) if EBC is used to select the buffer size or the service rate. The proposed approach illustrates one degree of freedom in choosing (σ, ρ) vector, which might be used by the network manager (or by the traffic regulator by consulting the network manager) to select σ or ρ; the other parameter is selected from EBC. We show that EBC is a percentile of the burstiness curve [30, 17, 31].

Our approach is the dual of the literature cited in the previous section, and could be more useful for network dimensioning. That is, if ϵ is known, then one can use our proposed technique to select an appropriate (σ, ρ) to satisfy ϵ. Indeed, we propose a trade-off curve between σ and ρ so that the desired ϵ is obtained. By keeping ϵ sufficiently small, the network can provide an acceptable level of QoS for the user. This is particularly important when the violating traffic is simply marked and submitted to network. Since statistical multiplexing improves bandwidth usage, there is a high probability that the violating traffic will safely pass through the network.

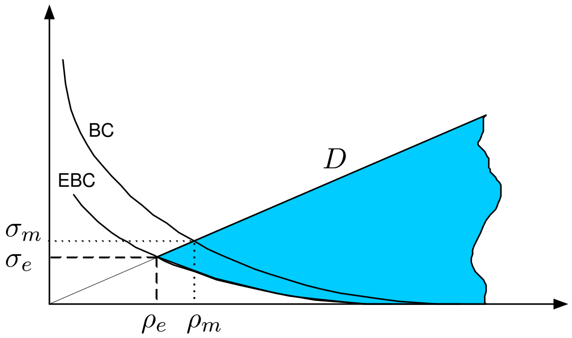

Figure 1: A typical effective burstiness curve (EBC) along with the corresponding burstiness curve (BC). The parameters of a properly-tuned leaky bucket are set at the intersection of EBC and the delay line.

Another major difference between our approach and the earlier ones is that we do not merely seek long-run solutions. In all works cited above, the QoS index has been studied under the assumption that time has been extended to infinity and that the system has converged to an equilibrium state. Therefore, $\mathrm{P}(Q_\rho(t) > \sigma)$ has been replaced by an upper envelope, which is usually obtained from the Chernoff bound. Although no canonical modelling of the probability distribution function of the input is required, the application of the Chernoff bound ignores the transitory behavior of traffic. In practice however sources are non-stationary and traffic generation time is limited. A more appropriate estimator should learn from the temporal behavior of source traffic. We devise such an estimator in this paper.

We will show that EBC is a convex, monotonically decreasing function of the service rate, and also that EBC of a multiplexed traffic is smaller than the summation of EBC of individual flows. Indeed, when independent traffic flows are aggregated, the aggregate shows statistical multiplexing gain. Therefore, if all sources are regulated by the suitable leaky buckets, the multiplexed traffic has a smaller EBC and the violating traffic of some users could possibly pass through the network without interference.

Fig. 1 shows a typical EBC (for a fixed ϵ). In this figure, "BC" denotes the burstiness curve as defined in [30, 17, 31], which is identical to EBC for $\epsilon = 0$. In [31], we used the intersection of BC and the delay line (to be discussed in Section 2) to set the traffic regulator. This point is shown as (σ_m, ρ_m) in Fig. 1. In this paper, we set the parameters of regulator at the intersection of EBC and the delay line—shown as (σ_e, ρ_e) in the figure. Therefore, considerable savings in bandwidth and buffer size are obtained.

Direct computation of EBC needs the probability distribution function of the queueing process which may not be available in practice. In this paper, we propose an estimator for EBC called the *empirical effective burstiness curve* (EEBC). EEBC is defined over a limited number of source traffic samples and is a consistent estimator of EBC. We will show that, for some traffic types—such as MPEG4-encoded traces—using EEBC will substantially reduce the required bandwidth and buffer size compared to when BC is used. Nonetheless, this saving is obtained by introducing an arbitrarily small traffic loss.

We will also show that for a discrete time input process EEBC can be obtained by a recursive *water-filling* procedure. Water-filling provides an algorithmic approach to computing EEBC. Water-filling indeed distributes the queue occupancy over time and gradually dilates the presence of sharp peaks in the queue size. The proposed water-filling approach produces a recursive scheme through

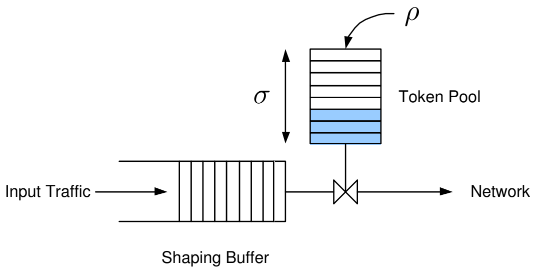

Figure 2: An illustration of the leaky bucket regulator with a shaping buffer.

which EEBC is updated by the arrival of each new sample. The process can therefore be used online. We further show that the water-filling algorithm can be used for a queue with a time-varying service rate.

2 Problem Formulation

A regulator, located at the boundary of a network, can be considered as a device that anticipates congestion inside the network and takes proper actions—by transferring the congestion to the network boundary—to alleviate the problem altogether or lessen the destructive effects of congestion. In this paper, we focus on leaky bucket regulators. Here, we view a leaky bucket regulator as a token pool replenished by a constant rate token generator and an infinite overflow (shaping) buffer. Fig. 2 illustrates our formulation of the leaky bucket regulator that is represented by the pair (σ, ρ), with σ being the token pool size and ρ the token replenishment rate. A packet of data emitted by the source consumes a number of tokens (equal to the size of the packet) from the pool and enters the network immediately. Consumed tokens are replaced by the token generator. A packet arriving at an empty pool can be either marked as a nonconforming packet and submitted immediately to the network or delayed in the "shaping" buffer until an equal amount of tokens is generated. In the latter case, the size of the shaping buffer is assumed to be very large so that no packet is lost in the regulator. The size of the token pool, σ, also represents the maximum burst size of the traffic. In this paper, we assume that nonconforming traffic is marked and transmitted immediately. Nonconforming traffic will be prone to loss if congestion occurs inside the network. In this configuration of leaky bucket, the size of the shaping buffer is zero.

In [31], we devised an algorithm that could be used to select the (σ, ρ) parameters in a lossless traffic regulation paradigm. Our approach was based on estimating the worst-case burstiness of the projected traffic. The leaky bucket parameters were then selected in a parsimonious procedure; (σ, ρ) was selected on the burstiness curve of the source. Any point below the burstiness curve could have resulted in traffic loss.

Here, we take a different perspective. We allow a small part of traffic to be lost or marked as nonconforming traffic. This loss can be accepted if significant savings, in terms of bandwidth and/or buffer size, can be achieved. We devise such an algorithm here. Our problem—given a source whose statistics are unknown—is to set the parameters of the corresponding leaky bucket regulator.

We will require that two constraints on the backlog and the delay of traffic in a single-server queue be met. We first envision the network as a single-server node with a minimum guaranteed rate; this assumption is frequently used in networking literature [7, 13]. In our formulation, the leaky bucket regulator is selected so as to restrain the percentage of the marked traffic below a prescribed threshold. Indeed, we select (σ, ρ) so that

$$\mathrm{P}(Q_\rho(t) > \sigma) \leq \epsilon \tag{2}$$

where $Q_\rho(t)$ is the instantaneous queue size in a server with the service rate ρ.

We add a second constraint, in the form of an inequality

$$f(\sigma, \rho) \geq 0 \tag{3}$$

in which the function f is given. Examples of f are suggested by giving the design problem a particular network context. Consider, for instance, a network in which access to the nodal output buffers is mediated by a *generalized processor sharing* (GPS) scheduler or one of its non-preemptive variants [13]. A session is said to be *stable* in this connection if the corresponding GPS coefficient is at least as large as the token generation rate ρ in the associated leaky bucket [13]. In this case, the connection is considered 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, with 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

$$\sigma \leq \rho D_{M}, \tag{4}$$

where D_M is an upper bound on the maximal end–to–end delay deemed acceptable to the source. This corresponds to the inequality formulated in terms of f with $f(\sigma, \rho) = \rho D_M - \sigma$. The results reported in the ensuing sections are for this particular f.

Note that choosing f in the form of (4) is not a restrictive assumption. Indeed, the inequality constraints (2) and (4) taken together amount to selecting a (σ, ρ) pair inside a region denoted by the shaded area in Fig. 1. Both constraints are satisfied inside this region. The constraints simply indicate the boundary of the region. Since bandwidth is usually the scarce resource in network, we choose (σ, ρ) at the cross section of EBC (to be defined in Section 3) and the delay line $\sigma = \rho D_M$— the leftmost point in Fig. 1. Therefore, any function f, which intersects EBC can be selected as the constraint function. Assuming that the network will reserve resources for the projected flow based on the σ and ρ parameters of the regulator, we select the intersection of the two constraints (2) and (4) to guarantee that bandwidth is parsimoniously allocated to input traffic [31].

3 Effective Burstiness Curve

Consider a single server handling the traffic of a single user. The accumulated traffic of the user over an interval $[s, t)$ is represented by $A(s, t)$. The input traffic is assumed to be generated by a stationary and ergodic stochastic process that satisfies

$$\begin{aligned} \sup_t \sup_{\tau>0} \frac{A(t, t+\tau)}{\tau} &= \rho_M && (5) \\ \lim_{\tau\to\infty} \frac{A(t, t+\tau)}{\tau} &= \bar{\rho} \quad \text{uniformly in } t. && (6) \end{aligned}$$

ρ_M is the maximum rate of traffic and $\bar{\rho}$ is the average rate. The objective is to select appropriate service rate and buffer size so that a certain QoS is guaranteed for the user. The QoS is quantified in terms of traffic delay and loss.

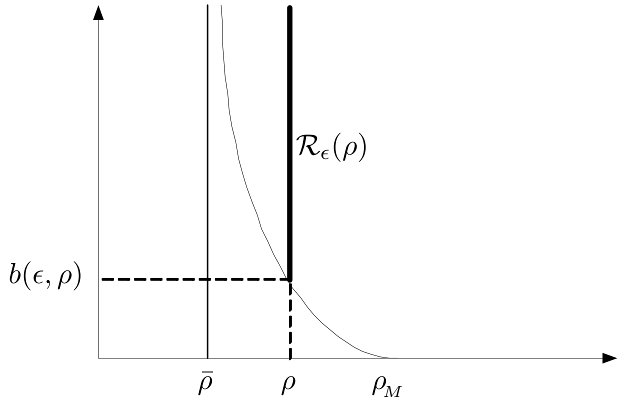

Figure 3: An example of the effective burstiness curve and $\mathcal{R}_\epsilon(\rho)$.

The backlog for a server with the constant rate ρ and infinite buffer size at time t can be written as

$$Q_\rho(t) = \sup_{s \le t}\{A(s,t) - (t-s)\rho\}. \tag{7}$$

In a stationary regime, one can represent the queue size at the origin by

$$Q_\rho(0) = \sup_{t \ge 0}\{A_{-t} - t\rho\} \tag{8}$$

where $A_{-t} \triangleq A(-t, 0)$.

Definition 1 *For any $0 \le \epsilon \le 1$ and for any fixed $\rho > \bar{\rho}$, let $\mathcal{R}_\epsilon(\rho) \triangleq \Big\{b \,|\, P\Big(Q_\rho(0) \ge b\Big) \le \epsilon\Big\}$. The effective burstiness curve (EBC) is defined as $b(\epsilon,\rho) \triangleq \inf \mathcal{R}_\epsilon(\rho)$.*

In Fig. 3, we show an example of EBC and $\mathcal{R}_\epsilon(\rho)$. Note that if we choose b_1 and b_2 subject to $b_1 < b_2$ and $b_1 \in \mathcal{R}_\epsilon(\rho)$, then $b_2 \in \mathcal{R}_\epsilon(\rho)$. Furthermore, if $\epsilon \le \epsilon'$, then $\mathcal{R}_\epsilon(\rho) \subseteq \mathcal{R}_{\epsilon'}(\rho)$. Indeed, $\mathcal{R}_\epsilon(\rho)$ is the set of all buffer sizes that have a buffer overflow probability not larger than ϵ when the server rate is ρ. Any buffer size selected in $\mathcal{R}_\epsilon(\rho)$ guarantees that the buffer overflow will remain below ϵ. At $\epsilon = 0$, we will get $b(0,\rho)$, the *burstiness curve* defined in [30, 31]—also called the *buffer-bandwidth trade-off curve* [17]. The following lemma states some of the properties of EBC.

Lemma 1 *The effective burstiness curve satisfies:*

(i) *For any $\epsilon \ge 0$, $b(\epsilon,\rho)$ is a non-increasing function of ρ;*

(ii) *For any $\rho > \bar{\rho}$, $b(\epsilon,\rho)$ is a non-increasing function of ϵ;*

(iii) *$b(\epsilon,\rho) = 0$ for $\rho \ge \rho_M$;*

(iv) *Let $Q_\rho(t)$ and $Q'_\rho(t)$ be the queue size of the single server queue serving two different input traffics. If $Q_\rho(t) \le Q'_\rho(t)$ for all t, then $b(\epsilon,\rho) \le b'(\epsilon,\rho)$ where $b(\epsilon,\rho)$ and $b'(\epsilon,\rho)$ are the EBC associated to $Q_\rho(t)$ and $Q'_\rho(t)$, respectively.*

Proof: The proof of parts (i) and (ii) is straightforward. Part (iii) follows from (ii) and the fact that $b(0,\rho) = 0$ for $\rho \ge \rho_M$ [31]. Part (iv) follows from the definition of the EBC. □

In the following theorem, we prove the convexity of EBC.

Theorem 1 *For fixed ϵ, the effective burstiness curve $b(\epsilon, \rho)$ is a convex function of ρ.*

Proof: See Appendix A.

The convexity of EBC is important. Consider two traffics $A_1(t)$ and $A_2(t)$ and let both have the same probability distribution function. Therefore, both traffics will have the same EBC represented by $b(\epsilon, \rho)$. Now let $A_1(t)$ be served by a single server with the rate ρ_1 and $A_2(t)$ be served with another server with the rate ρ_2. The convexity of EBC indicates that $b(\epsilon, \alpha\rho_1 + \beta\rho_2) \leq \alpha b(\epsilon, \rho_1) + \beta b(\epsilon, \rho_2)$ for $\alpha + \beta = 1$. That is , the aggregate traffic in a homogeneous network ($A_1(t)$ and $A_2(t)$ of the same type) needs a smaller buffer size per traffic. The following theorem shows that this property also holds in a network with non-homogenous traffic.

Theorem 2 *Let $Q^{(i)}_{\rho_i}(0) = \sup_{t\geq 0}\{A^{(i)}_{-t} - \rho_i t\}$, $\mathcal{R}^{(i)}_\epsilon(\rho_i) = \{b \mid P(Q^{(i)}_{\rho_i}(0) \geq b) \leq \epsilon\}$ and $b^{(i)}(\epsilon, \rho_i) = \inf \mathcal{R}^{(i)}_\epsilon(\rho_i)$ for $i = 1, \ldots, L$. Define $A_{-t} \triangleq \sum_{i=1}^{L} \alpha_i A^{(i)}_{-t}$, and $\rho \triangleq \sum_{i=1}^{L} \alpha_i \rho_i$, where $\alpha_i \geq 0$. Let also $Q_\rho(0) \triangleq \sup_{t\geq 0}\{A_{-t} - \rho t\}$, $\mathcal{R}_\epsilon(\rho) = \{b \mid P(Q_\rho(0) \geq b) \leq \epsilon\}$ and $b(\epsilon, \rho) = \inf \mathcal{R}_\epsilon(\rho)$. Then*

$$b(\epsilon, \rho) \leq \sum_{i=1}^{L} \alpha_i b^{(i)}(\epsilon, \rho_i). \tag{9}$$

Proof: See Appendix B.

Theorem 2 indicates that the total burstiness of the multiplexed traffic is smaller than the summation of the burstiness of individual users. Therefore, if we assume $\alpha_i = 1$ for all $i = 1, \ldots, L$, then we can conclude that statistical multiplexing can reduce the EBC. Similar results have been reported in [30] for the maximum burstiness curve.

In the following corollary, we strengthen the results of Theorem 2.

Corollary 1 *In Theorem 2, let $\rho_i = \rho$ for all $i = 1, \ldots, L$, and $\sum_{i=1}^{L} \alpha_i \leq 1$. Then*

$$b(\epsilon, \rho) \leq \sum_{i=1}^{L} \alpha_i b^{(i)}(\epsilon, \rho). \tag{10}$$

Proof: Replace ρ_i by ρ in the right-hand-side of (9) to get $b(\epsilon, \rho \sum_{i=1}^{L} \alpha_i) \leq \sum_{i=1}^{L} \alpha_i b^{(i)}(\epsilon, \rho)$. Now use Lemma 1-($i$) to get that $b(\epsilon, \rho) \leq b(\epsilon, \rho \sum_{i=1}^{L} \alpha_i)$ for $\sum_{i=1}^{L} \alpha_i \leq 1$. □

In Definition 1, the effective burstiness curve is defined assuming stationary traffic over an infinite interval. In practice, using infinite interval is unrealistic. In the sequel, we investigate the case of finite intervals and discrete time setting. We will use a fluid-flow approximation with $\rho_M = \infty$.

3.1 Empirical Effective Burstiness Curve

The queue size for a constant rate server in a discrete time setting satisfies Lindley's equation,

$$q_n = [q_{n-1} - \rho]^+ + r_n \tag{11}$$

where q_n is the backlog at time n, ρ is the service given over a time unit, r_n is the amount of traffic arriving at time n, and $[a]^+ \triangleq max\{a, 0\}$ for any real number a.

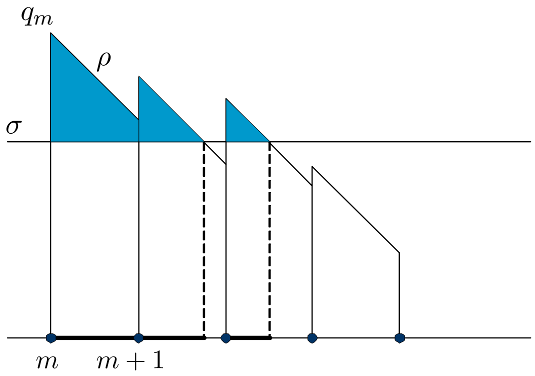

Figure 4: Queue size as a function of time for a queue with discrete inputs. The period of time for which $Q_\rho(t) > \sigma$ has been shown in heavy lines.

For a given token pool size σ, one is able to define

$$\mu_n(\sigma) = \frac{1}{n}\sum_{i=0}^{n-1} 1\Big\{q_i > \sigma\Big\} \min\Big\{\frac{q_i - \sigma}{\rho}, 1\Big\} \tag{12}$$

where $1\{.\}$ is the *indicator* function, that is

$$1\{\mathcal{A}\} = \begin{cases} 1 & \text{if the predicate } \mathcal{A} \text{ is true,} \\ 0 & \text{if the predicate } \mathcal{A} \text{ is false.} \end{cases} \tag{13}$$

Note that $\mu_n(\sigma)$ is also a function of ρ—we have dropped this parameter for brevity. Further, if q_n is the sample of the queue size $Q_\rho(t)$ at the nth time instant, $\mu_n(\sigma)$ will be the proportion of time that the queue size stays above the threshold, σ. See Fig. 4 for an illustration. The queue size at time t is given by

$$Q_\rho(t) = q_m - (t-m)\rho, \qquad \text{for } m < t \leq m+1. \tag{14}$$

Therefore, $\mu_n(\sigma)$ is indeed the period of time over which $Q_\rho(t) > \sigma$, normalized over the whole window of observation.

We can also represent $\mu_n(\sigma)$ as

$$\begin{aligned} \mu_n(\sigma) &= \frac{1}{n}\sum_{i=0}^{n-1}\int_i^{i+1} 1\{q_i - (t-i)\rho > \sigma\}\,dt \\ &= \frac{1}{n}\int_0^n 1\{Q_\rho(t) > \sigma\}\,dt. \end{aligned} \tag{15}$$

$\mu_n(\sigma)$ can thus be interpreted as the temporal average of the event $\{Q_\rho(t) > \sigma\}$. We also know that

$$\mathrm{P}\Big(Q_\rho(t) > \sigma\Big) = \mathrm{E}1\{Q_\rho(t) > \sigma\}, \tag{16}$$

with E1 denoting the expected value. For an ergodic process $Q_\rho(t)$ we can prove the following lemma.

Lemma 2 *In a stationary regime,*

$$\lim_{n\to\infty} \mu_n(\sigma) = P(Q_\rho > \sigma) \tag{17}$$

where $Q_\rho \triangleq \lim_{t\to\infty} Q_\rho(t)$ *is the stationary queue size.*

Note also that since queue $Q_\rho(t)$ is stable, $\{Q_\rho(t) > \sigma\}$ will be a persistent recurrent event. Therefore, Lemma 2 can also be proved using the "renewal" theory [32].

Definition 2 *The empirical effective burstiness curve (EEBC) for observations over the interval* $[0, n]$ *is defined as a function* $b_n(\epsilon, \rho)$ *that satisfies*

$$\mu_n\Big(b_n(\epsilon,\rho)\Big) = \epsilon. \tag{18}$$

Therefore, $b_n(\epsilon,\rho) = \mu_n^{-1}(\epsilon)$.

Using (15), we have

$$\frac{1}{n}\int_0^n 1\{Q_\rho(t) > b_n(\epsilon,\rho)\}dt = \epsilon. \tag{19}$$

Thus, for each n, EEBC indicates a threshold at which the total duration of the time interval where the event $\{Q_\rho(t) > b_n(\epsilon,\rho)\}$ occurs is $n\epsilon$. Note that at a given n, EEBC cannot be defined for all $0 \le \epsilon \le 1$. Indeed, the maximum ϵ is given by $\bar{\epsilon}_{n,\rho} = \mu_n(0)$. In the sequel, we show that EEBC is a non-increasing convex function of ρ.

Theorem 3 *EEBC has the following properties:*

(*i*) $b_n(\epsilon,\rho)$ *is a non-increasing function of* ρ*;*

(*ii*) $b_n(\epsilon,\rho)$ *is a non-increasing function of* ϵ*;*

(*iii*) *Let* q_n *and* q'_n *be the queue size of the single server queue serving two different input traffics. If* $q_n \le q'_n$ *for all* n*, then* $b_n(\epsilon,\rho) \le b'_n(\epsilon,\rho)$ *where* $b_n(\epsilon,\rho)$ *and* $b'_n(\epsilon,\rho)$ *are EEBCs associated to* q_n *and* q'_n*, respectively;*

(*iv*) $b_n(\epsilon,\rho)$ *is a convex function of* ρ*.*

Proof: See Appendix C.

In the following section, we will propose an algorithmic approach for computing EEBC. We will then use it to prove that $b_n(\epsilon,\rho)$ converges to $b(\epsilon,\rho)$ when $n \to \infty$.

4 Water-filling

In this section, we show that EEBC can be obtained with a water-filling procedure. Water-filling is particularly important since, as we will show later, it is performed "on-the-fly". That is, when a new packet arrives, EEBC is adjusted using the water-filling algorithm. We assume that time is slotted with each time instant represented by an integer $n \ge 1$.

Let $n = 1$. From the definition of EEBC, we should have $\mu_1(b_1(\epsilon,\rho)) = \epsilon$. Fig. 5(a) shows an example with the corresponding EEBC for a given ϵ. EEBC should be selected so that the fraction of time that the queue length is larger than $b_1(\epsilon,\rho)$ be equal to ϵ. Note that for any $q_0 - \rho \le \sigma \le q_0$,

$$\begin{aligned}\mu_1(\sigma) &= \int_0^1 1\{Q_\rho(t) > \sigma\}\,dt \\ &= \frac{q_0 - \sigma}{\rho}.\end{aligned} \tag{20}$$

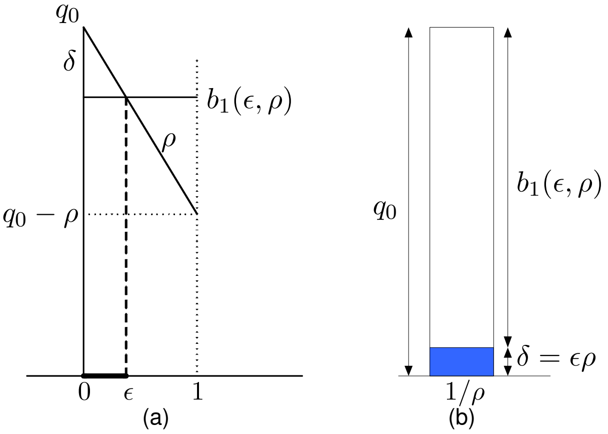

Figure 5: The queue size and EEBC for $n = 1$.

If $\sigma = q_0 - \epsilon\rho$, then $\mu_1(q_0 - \epsilon\rho) = \epsilon$. Therefore, $b_1(\epsilon, \rho) = q_0 - \delta$ where $\delta \triangleq \epsilon\rho$ indicates the vertical distance between the maximum queue size, q_0, and EEBC. Note that $b_1(\epsilon, \rho)$ is an estimate of EBC at the end of the first time slot.

We now show that $b_1(\epsilon, \rho)$ can be found by water-filling. Let us represent the first time instant by a container with the height q_0, and the width $1/\rho$. Let this container hold ϵ units of liquid. Then, EEBC is the empty portion of the container; see Fig. 5(b).

We continue water-filling representation by letting $n = 2$. For $n = 2$, we have two queue samples denoted by q_0 and q_1. Note first that any percentile of the queue size is independent of the order at which queue sizes q_0, and q_1 arrive. One might visualize the queue size by considering the quadrangles with the maximum height $q_i, i = 0, 1$ and the decreasing slope of the upper line ρ. The quadrangles can be arranged in any arbitrary order. Therefore, without loss of generality, we assume $q_0 \geq q_1$. From the definition of EEBC, the fraction of time that the queue size is larger than $b_2(\epsilon, \rho)$ is ϵ. Let us visualize a horizontal line located at q_0 and allow this line to move gradually downward and measure the length of the interval for which the queue size is larger than the horizontal line. We represent the interval for which the queue size is larger than this horizontal line by I_2. The length of I_2 starts at zero and gradually increases. When the total length of I_2 is equal to 2ϵ, we stop and measure the height of the horizontal line, which is equal to $b_2(\epsilon, \rho)$. There are two cases that we will discuss separately: (i) $q_1 \leq q_0 - 2\delta$; and (ii) $q_1 > q_0 - 2\delta$.

In case (i), q_1 is much smaller than q_0, hence the length of I_2 becomes 2ϵ before the horizontal line arrives at q_1. This case has been illustrated in Fig. 6(a) for $\epsilon = 0.25$. In this figure, the event $1\{Q_\rho(t) > b_2(\epsilon, \rho)\}$ has been illustrated in heavy lines. For $\epsilon = 0.25$ and $n = 2$, the total temporal extent of the heavy line is 0.5 units. Since the total length of I_2 is equal to 2ϵ, we have $b_2(\epsilon, \rho) = q_0 - 2\delta$, where $\delta = \epsilon\rho$.

EEBC in case (i) can also be obtained by water-filling. Each time instant is modeled by a container with the height q_i and the width $1/\rho$ holding ϵ units of water. If the two containers can exchange liquid, the water will pour into the larger container with the height q_0. The height of the empty portion of the container, $q_0 - 2\delta$, is EEBC.

Now consider the second case as illustrated in Fig. 6(b). From the definition of EEBC, the total length of time for which the queue length is larger than $b_2(\epsilon, \rho)$ is 2ϵ. Similar to case (i), we move the horizontal line, starting at q_0, downward until the length of the total interval for which the

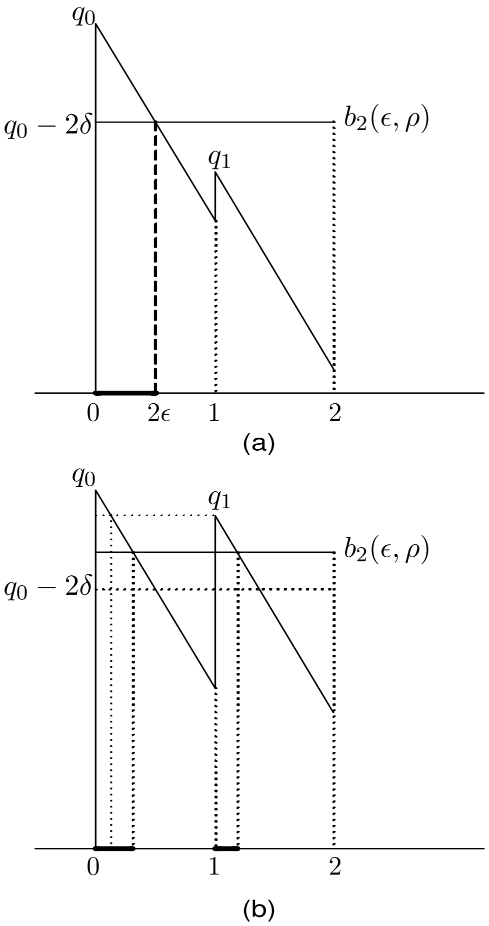

Figure 6: The queue size and EEBC with $\epsilon = 0.25$ for $n = 2$: (a) $q_0 \geq q_1 + 2\delta$ and (b) $q_0 < q_1 + 2\delta$.

queue size is larger than the line is 2ϵ. When the horizontal line arrives at q_1, the total length of I_2 is equal to $\frac{q_0 - q_1}{\rho}$. The horizontal line should move another $\frac{2\delta - (q_0 - q_1)}{2}$ units downward to complete the journey. At that point, the total length of I_2 is $\frac{q_0 - q_1}{\rho} + 2\frac{2\delta - (q_0 - q_1)}{2\rho} = 2\frac{\delta}{\rho} = 2\epsilon$. Therefore, $b_2(\epsilon, \rho) = q_1 - \frac{2\delta - (q_0 - q_1)}{2} = \frac{q_0 + q_1}{2} - \delta$. Note in Figure 6(b) that $b_2(\epsilon, \rho)$ is at the midpoint of q_1 and $q_0 - 2\delta$.

The water-filling procedure can again be used to find EEBC. Here, we assume two containers with heights q_0 and q_1, and the width $1/\rho$. Each container holds ϵ units of liquid as illustrated in Fig. 7(a). In Fig. 7, we have extended the height of the containers by δ to include the cases at which $q_i = 0$. In such cases, the capacity of the container is zero but δ units of liquid should still be generated at each time instant. Note that this extra height is not used in computing EEBC. As shown in Fig. 7(b), water-filling gives $b_2(\epsilon, \rho) = \frac{q_0 + q_1}{2} - \delta$, which is the same value obtained by the direct method.

We continue this process by letting $n = 3$, where such as before we assume $q_0 \geq q_1 \geq q_2$. Here, we consider three cases: (*i*) q_0 is substantially larger than q_1 and q_2; (*ii*) q_1 is close to q_0 but is much larger than q_2; and (*iii*) q_0, q_1, and q_2 are close to each other. In the sequel, we will study the three cases in details.

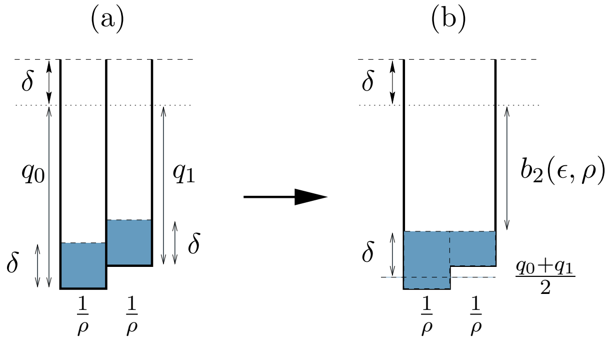

Figure 7: The water-filling procedure for $n = 2$ with $q_1 > q_0 - 2\delta$.

To find EEBC we visualize a horizontal line located at q_0 and allow this line to move gradually downward and measure the length of the interval for which the queue size is larger than the horizontal line. Let us represent the interval for which the queue size is larger than this horizontal line by I_3. The length of I_3 starts at zero and gradually increases. When the total length of I_3 is equal to 3ϵ, the horizontal line coincides with $b_3(\epsilon, \rho)$.

In case (i), since q_0 is much larger than q_1 and q_2, the length of I_3 will become 3ϵ before the horizontal line arrives at q_1. Therefore, we will have $b_3(\epsilon, \rho) = q_0 - 3\delta$ where $\delta = \epsilon\rho$. Note that here we have implicitly assumed that $3\epsilon < 1$. Using the waterfilling paradigm, we can find the solution by pouring liquid into the containers formed by q_0, q_1, and q_2. Fig. 8(a) illustrates the waterfilling solution for this case.

For case (ii), we note that I_3 will be decomposed into two parts located in the neighborhoods of q_0 and q_1. When the total length of I_3 is equal to 3ϵ, the horizontal line coincides with $b_3(\epsilon, \rho)$. This interval first grows in the first time slot until the horizontal line arrives at q_1 and then splits into two parts and grows with the same rate both in the vicinity of q_0 and q_1 until the total length becomes 3ϵ. When the horizontal line touches q_1, the total length of I_3 is $\frac{q_0-q_1}{\rho}$. The horizontal line should move another $\frac{3\delta-(q_0-q_1)}{2}$ units downward to complete the journey. At this point, the total length of I_3 is $\frac{q_0-q_1}{\rho} + 2\frac{3\delta-(q_0-q_1)}{2\rho} = 3\frac{\delta}{\rho} = 3\epsilon$. Therefore, $b_3(\epsilon, \rho) = q_1 - \frac{3\delta-(q_0-q_1)}{2} = \frac{q_0+q_1}{2} - \frac{3}{2}\delta$.

This case can also be solved with the waterfilling algorithm. Here, we distribute 3ϵ units of liquid over two time instants and arrive at the solution illustrated in Fig. 8(b). Note that in this case the total volume of the liquid in the containers is $(q_0 - b_3(\epsilon, \rho))/\rho + (q_1 - b_3(\epsilon, \rho))/\rho = 3\epsilon$, which results at $b_3(\epsilon, \rho) = \frac{q_0+q_1}{2} - \frac{3}{2}\delta$.

We now consider the third case. Similar to case (ii), we consider a horizontal line, first located at q_0, and gradually move it down until the length of I_3 becomes 3ϵ. In case (iii), we will consider a situation at which $b_3(\epsilon, \rho)$ is smaller than both q_1 and q_2. Therefore, we expect that I_3 will have three segments located in all three time slots. Since we have assumed that q_1 is larger than q_2, the horizontal line will first arrive at q_1. When the horizontal line arrives at q_1, the length of I_3 will be $\frac{q_0-q_1}{\rho}$. The length of I_3 then grows in two segments located in the first and the second time slots. When the horizontal line arrives at q_2 the total length of I_3 is $\frac{q_0-q_1}{\rho} + 2\frac{q_1-q_2}{\rho}$. At this point the horizontal line has already moved $q_0 - q_2$ units downward. The horizontal line should move extra $\frac{3\delta-(q_0-q_2)-(q_1-q_2)}{3}$ units downward. At this point the total length of I_3 will be $\frac{q_0-q_1}{\rho} + 2\frac{q_1-q_2}{\rho} + 3\frac{3\delta-(q_0-q_2)-(q_1-q_2)}{3\rho} = 3\frac{\delta}{\rho} = 3\epsilon$. EEBC can then be found from $b_3(\epsilon, \rho) = q_2 - \frac{3\delta-(q_0-q_2)-(q_1-q_2)}{3} = \frac{q_0+q_2+q_3}{3} - \delta$. This solution can also be found by waterfilling as illustrated in

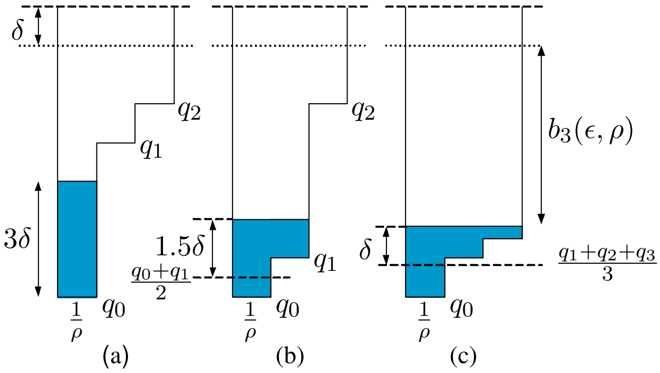

Figure 8: The water-filling process for $n = 3$ and $q_0 \geq q_1 \geq q_2$:
(a) $q_2 \leq q_1 \leq \dfrac{q_0 + q_1 - 3\delta}{2}$; (b) $q_2 \leq \dfrac{q_0 + q_1 - 3\delta}{2} \leq q_1$; (c) $\dfrac{q_0 + q_1 - 3\delta}{2} \leq q_2 \leq q_1$.

Fig. 8(c) since $(q_0+b_3(\epsilon,\rho))/\rho+(q_1+b_3(\epsilon,\rho))/\rho+(q_2+b_3(\epsilon,\rho))/\rho = 3\epsilon$. Hence $b_3(\epsilon,\rho) = \frac{q_0+q_2+q_3}{3} - \delta$.

This procedure can be repeated for other values of n. Here, we devise a waterfilling algorithm to obtain EEBC at a fixed ρ. Let the queue size over the interval $[i, i+1)$ be the decreasing function $Q_\rho(t) = q_i - (t-i)\rho$. Each time instant is represented by a container holding ϵ units of liquid. The ith container has the height $q_i + \delta$ and the width $1/\rho$. The whole liquid is distributed over all containers with the waterfilling process. In the water-filling algorithm, the height of the empty portion of the container at any time instant n less δ represents EEBC, $b_n(\epsilon,\rho)$. To guarantee that a zero queue size will also affect EEBC (i.e. hold ϵ units of water), the height of each container has been assumed to be at least $\delta = \epsilon\rho$.

The waterfilling algorithm gives us a powerful tool to find an appropriate buffer size to achieve the bound (1) in a single server queue. Besides providing a computationally simple algorithm (as we will show in Section 4.2) waterfilling allows us to solve the problem for cases at which the service rate is not constant. To observe this property, assume that the service rate is constant in each time slot but varies from one time slot to next. The waterfilling algorithm can then be formed as follows. Each time slot is represented by a container with the height $q_i + \epsilon\rho_i$ and the width $1/\rho_i$, holding ϵ units of liquid. If all containers can exchange water, the level of the liquid in the system will be EEBC.

4.1 Discrete Queues

Let $0 = a_0 < a_1 < \ldots < a_M$ represent the ordered set of quantization levels. We seek an iterative procedure to track EEBC in real-time. EEBC at time $n-1$ is represented by b_{n-1} (the parameters ϵ and ρ have been dropped for brevity.) We derive a recursive formula and use it to update EEBC upon the arrival of a new packet.

The states are decomposed into two sets of "wet" and "dry" states. By a wet state at time instant n, we mean a state m for which $b_n < a_m$—all wet states contain liquid in the water-filling paradigm. A dry state at time instant n is a one that is not wet. For a given b_n with $a_m < b_n \leq a_{m+1}$ all states a_k, $k \geq m+1$ are wet and all states a_k, $k \leq m$ are dry.

The total number of time instants at which the queue size has been at state m over the interval

$[0, n]$ is represented by

$$\ell_{m,n} \triangleq \sum_{i=1}^{n} 1\{q_i = a_m\}, \qquad \text{for } m = 0, \ldots, M. \tag{21}$$

Note that $\sum_{m=0}^{M} \ell_{m,n} = n$. The aggregated temporal extent of the states over the interval $[0, n]$ is represented by

$$\ell_n(m) \triangleq \sum_{k=m}^{M} \ell_{k,n} \quad \text{for } m = 0, \ldots, M. \tag{22}$$

Note that for each n, we have $\ell_n(0) = n$ and $\ell_n(m) \leq \ell_n(k)$ for $m \geq k$. It is also possible to show that in a stationary regime,

$$\lim_{n\to\infty} \frac{\ell_n(m)}{\ell_n(0)} = \mathrm{P}\Big(Q_\rho(0) \geq a_m\Big). \tag{23}$$

Let, for $m = 1, \ldots, M$,

$$v_{m,n} \triangleq (a_m - a_{m-1})\ell_n(m) \tag{24}$$

represent the amount of liquid required to fill-up the whole volume stretched between the states a_m and a_{m-1} at time instant n.

Let b_{n-1} be known at time $n-1$. Upon the arrival of a new packet at time instant n, we proceed to find a new EEBC, b_n. We distinguish two cases: (i) the new queue size is a dry state, (ii) the new queue size is a wet state. We discuss these two cases separately.

- **Case (i):**

  Assume $a_m < b_{n-1} < a_{m+1}$. Therefore, a_{m+1} is wet and a_m is dry. Let the new queue size be $q_n = a_k$, where a_k is a dry state, that is, $k \leq m$. The container in the new time instant carries δ units of liquid that should be distributed over all states in a water-filling procedure. First, we update $\ell_n(j)$ as

  $$\ell_n(j) = \begin{cases} \ell_{n-1}(j) + 1 & \text{if } j \leq k, \\ \ell_{n-1}(j) & \text{if } j > k. \end{cases} \tag{25}$$

  Define the amount of liquid, which is required to move b_{n-1} to a_m, by

  $$\check{v}_{b,n} \triangleq (b_{n-1} - a_m)\ell_n(m+1). \tag{26}$$

  Let also $V_{m,n}^{(j)}$, $j = 1, \ldots, m$ denote the aggregate of the amount of liquid required to fill-up the volume extended between states a_{m-j} and a_m. Then

  $$V_{m,n}^{(j)} \triangleq \sum_{i=0}^{j-1} v_{m-i,n}. \tag{27}$$

  Assume $V_{m,n}^{(0)} = 0$ and note that $0 < V_{m,n}^{(1)} < \ldots < V_{m,n}^{(m)} < V_{m,n}^{(m)} + n\delta$. The new EEBC is found by comparing $\delta - \check{v}_{b,n}$ to these values:

  $$b_n = \begin{cases} b_{n-1} - \dfrac{\delta}{\ell_n(m+1)} & \text{if } \delta < \check{v}_{b,n}, \\ a_{m-j+1} - \dfrac{\delta - \check{v}_{b,n} - V_{m,n}^{(j-1)}}{\ell_n(m-j+1)} & \text{if } V_{m,n}^{(j-1)} < \delta - \check{v}_{b,n} \leq V_{m,n}^{(j)},\ j \neq 0. \end{cases} \tag{28}$$

- **Case** (*ii*)

  Again assume $a_m < b_{n-1} < a_{m+1}$ and let q_n correspond to a wet state. Each container at any time interval carries δ units of liquid. Therefore, the level of liquid in the container should be compared to $q_n - \delta$. If $q_n - \delta < b_{n-1}$, the excessive liquid in the new container should be divided over all pre-existing wet states. An approach similar to the one presented in case (i), with δ in (28) replaced by $\delta - q_n + b_{n-1}$, can be used to distribute the liquid over all corresponding states. If, on the other hand, $q_n - \delta > b_{n-1}$, the new container can sink part of the liquid in the system and hence increase EEBC. An example of this case has been illustrated in Fig. 9.

  Let $q_n = a_k$ for some $m+1 \leq k \leq M$ and assume $q_n - \delta > b_{n-1}$. In equilibrium, the amount of liquid poured into the new container is $q_n - \delta - b_n$. This liquid should be supplied by other containers. Define the volume of liquid reserved between b_{n-1} and a_{m+1} by

  $$\hat{v}_{b,n-1} \triangleq (a_{m+1} - b_{n-1})\ell_{n-1}(m+1). \tag{29}$$

  Such as (27), let $V^{(j)}_{k,n-1}$, $j = 1, \ldots, k-m-2$ denote the aggregated liquid between states a_{k-j} and a_k at time $n-1$. The total liquid to be distributed over all states $j = m+1, \ldots, k$ is equal to (see Fig. 9)

  $$W_n \triangleq \delta + \hat{v}_{b,n-1} + V^{(k-m-2)}_{k,n-1}. \tag{30}$$

  Upon the arrival of a new packet at time instant n, and assuming $q_n = a_k$, the aggregated temporal extents $\ell_n(j)$, $j = 0, \ldots, M$ will be updated as in (25). Now use (27) to find $V^{(j)}_{k,n}$, $j = 1, \ldots, k-m-2$. Again assume $V^{(0)}_{k,n} = 0$. A technique similar to the one presented in case (i) is used to distribute the liquid. Consider $0 < V^{(1)}_{k,n} < V^{(2)}_{k,n} < \ldots < V^{(k-m-2)}_{k,n}$. The total amount of liquid W_n is now compared to these values. If W_n belongs to a certain interval, all states with an index larger than the one corresponding to the given interval will be wet at time instant n and the remaining states will become dry. The new EEBC, b_n, is then given by

  $$b_n = \begin{cases} a_k - \dfrac{W_n}{\ell_n(k)} & \text{if } W_n \leq V^{(1)}_{k,n}, \\ a_{k-j} - \dfrac{W_n - V^{(j-1)}_{k,n}}{\ell_n(k-j)} & \text{if } V^{(j)}_{k,n} < W_n \leq V^{(j+1)}_{k,n}, \quad \text{for } 1 \leq j \leq k-m-3. \end{cases} \tag{31}$$

The total liquid used in the water-filling procedure in the interval $[0, n]$ is denoted by $\Delta_n = n\delta$. From the definition of δ, we have

$$\frac{\Delta_n}{n} = \epsilon\rho \tag{32}$$

which is independent from n, the length of the observation interval. From the construction of the water-filling procedure, we conclude that

$$\lim_{t\to\infty} \mathrm{P}\Big(Q_\rho(t) > b_n\Big) = \frac{\Delta_n}{n\rho} \tag{33}$$

where b_n corresponds to the level of liquid in the water-filling process with the total amount of liquid Δ_n. Therefore, the amount of liquid used will indicate the probability of buffer overflow. We will use this observation to prove the following theorem.

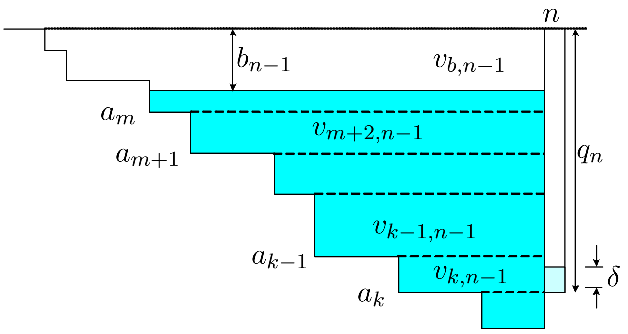

Figure 9: The water-filling procedure for a case at which $q_n - \delta < b_{n-1}$.

Theorem 4 *EEBC, $b_n(\epsilon, \rho)$, converges to EBC, $b(\epsilon, \rho)$, as $n \to \infty$.*

Proof: See Appendix D.

4.2 Computational Complexity

In this subsection, we study the computational complexity of the water-filling algorithm. We compute the computational complexity of the algorithm in the worst case—compute the maximum number of parameters at each time instant. The water-filling algorithm should store the values of $\ell_n(j)$, $j = 0, \ldots, M$. Upon the arrival of a new packet, the queue size, q_n, is calculated using Lindley's equation. As before, we assume that $a_m < b_{n-1} < a_{m+1}$ and $q_n = a_k$. If $q_n = a_k$, then all $\ell_n(j)$, $j = 0, \ldots, k$ will be increased, as suggested in (25). This will require, at most, M additions. We will also need to calculate $v_{j,n}$, $j = 1, \ldots, M$, $\check{v}_{b,n}$, and $\hat{v}_{b,n-1}$, given by (24), (26), and (29), respectively. At most, this step will use $(M + 2)$ multiplications and $(M + 2)$ additions. Note that the number of additions can be reduced to 2 if we store each $a_m - a_{m-1}$ and use them in (25). Furthermore, if we assume that $a_m - a_{m-1} = 1$, then the maximum number of multiplications will be reduced to 2. The next step is to compute $V_{m,n}^{(j)}$. At the worst case, this step may need M additions. Next, we have to compute $\delta - \check{v}_{b,n}$ or $W_n = \delta + \hat{v}_{b,n-1} + V_{k,n-1}^{(k-m-2)}$, depending on whether q_n is a dry or a wet state. This step needs 1 or 2 additions. In the final stage, we should compare $\delta - \check{v}_{b,n}$ or W_n to at most M levels. This can be done in $\log M$ steps. Finally, EEBC in (28) and (31) need at most 3 additions and one division. Therefore, the water-filling algorithm in total will need at most $3M + 7$ additions, $M + 3$ multiplications and $\log M$ comparisons. Note that since M is usually small, the computational complexity of the water-filling algorithm is limited.

5 Numerical Results

In this section, we find EEBC for MPEG4 encoded video traces. We use 20 minute traces of 9 MPEG4 encoded movies downloaded from [33]. The 9 movies are: Jurassic Park I, Silence Of The Lambs, Star Wars IV, Mr. Bean, Star Trek – First Contact, From Dusk Till Dawn, The Firm, Starship Troopers, and Die Hard III. Fig. 10 illustrates the trace of Star Wars IV as a function of time. The maximum packet size in this trace is 9370 bytes and is located at the 154th frame (6.16 seconds after the start of the trace). We decompose each trace into 4 non-overlapping segments each containing 5 minutes of the original trace and create 36 traces of 5 minute length each. In the sequel, we will use these traces to study our techniques.

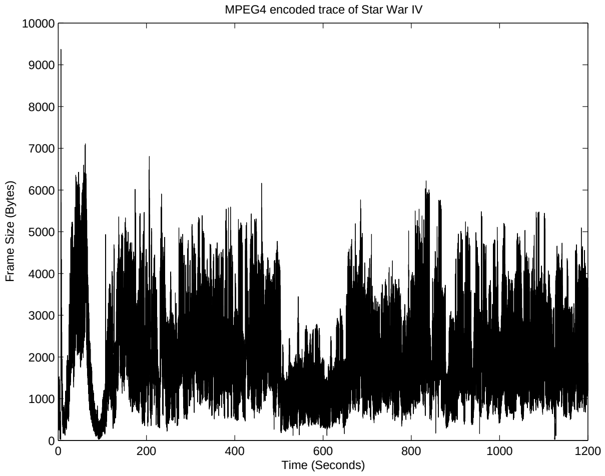

Figure 10: The MPEG4 encoded trace of the movie Star Wars IV.

The EEBC of all 36 traces for $\epsilon = 0.01$ are illustrated in Fig. 11. As expected from Theorem 3, EEBC is a convex, monotonically decreasing function of the service rate ρ. Note also that EEBC is trace dependent. Therefore, it is not possible to define a unique EEBC that can serve as the burstiness curve of a class of traces. Indeed, as illustrated in Fig. 11, the variance of EEBC of traces can be large. Fig. 12 shows the average EEBC for $\epsilon = 0, 0.01, 0.02, 0.03, 0.04, 0.05$. Note that EEBC for different values of ϵ is a convex decreasing function of ρ.

In Fig. 13, we show EEBC as a function of ϵ for five different values of the service rate in a semi-logarithmic scale. As noted, EEBC decreases with increasing ϵ. The decrease is more pronounced for smaller values of the service rate and moderate values of ϵ.

EEBCs of the movie Star Wars IV are shown in Fig. 14 in a semi-logarithmic plot along with the delay lines corresponding to 20 msec and 50 msec delay limits; the curves correspond to $\epsilon = 0, 0.01, \ldots, 0.05$. As illustrated, there exists a fairly large difference between the empirical maximum burstiness curve and the EEBC associated to $\epsilon = 0.01$. In this example, for $\epsilon = 0$ and $D = 20$ msec, the allocated bandwidth should be 400 KByte/sec. If $\epsilon = 0.01$ is used, the required bandwidth will be 180 KByte/sec. Therefore, using $\epsilon = 0.01$ instead of $\epsilon = 0$ will save approximately 55% of the required bandwidth. Note that indeed with the service rate of 180 KByte/second for the trace, at most 1% of data will be lost or placed in the shaping buffer and delayed beyond 20 msec. This phenomenon is due to the burstiness of video stream.

Fig. 15 illustrates the percentage of bandwidth that can be saved if the empirical maximum burstiness curve is replaced by EEBC. The curves have been found by locating the intersection of the EEBC and the delay lines. The percentage depends on the parameter ϵ. In this figure, for the delay of 50 msec, about 50% saving on bandwidth can be acquired if EEBC is used.

In order to show that the existence of the large distance between the burstiness curves for $\epsilon = 0$ and $\epsilon = 0.01$ is not restricted only to the video trace under investigation, we have investigated the EEBC of an MPEG4 encoded trace of the movie Jurassic Park for 1000 seconds. The results have been illustrated in Fig. 16. Note the fairly large separation between the burstiness curves for $\epsilon = 0$ and $\epsilon = 0.01$ in this example.

Fig. 17 illustrates EEBC of a convex combination of the traces for $\epsilon = 0.01$. The upper curve in the figure shows the average of EEBCs of all traces. The other curve is EEBC of the multiplexed

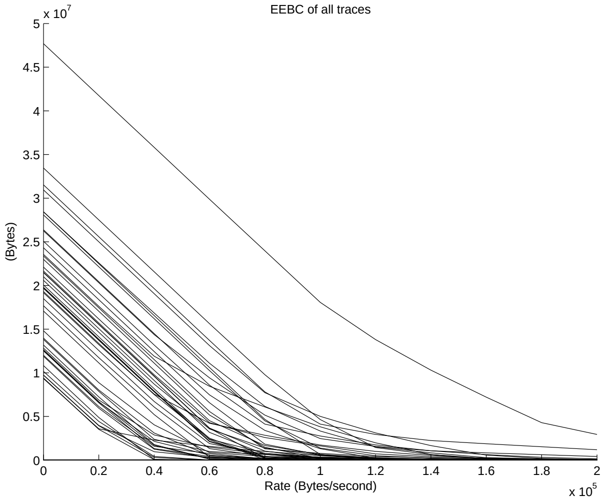

Figure 11: The EEBC of 36 traces of 5 minutes each taken from the 9 MPEG4 movies.

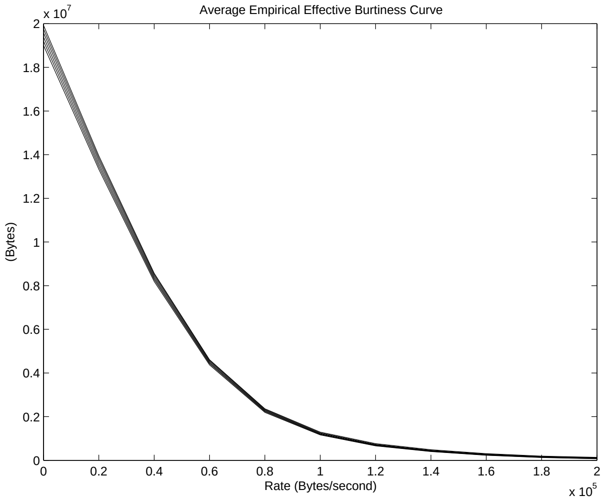

Figure 12: The average EEBC of all traces for $\epsilon = 0, 0.01, 0.02, 0.03, 0.04, 0.05$.

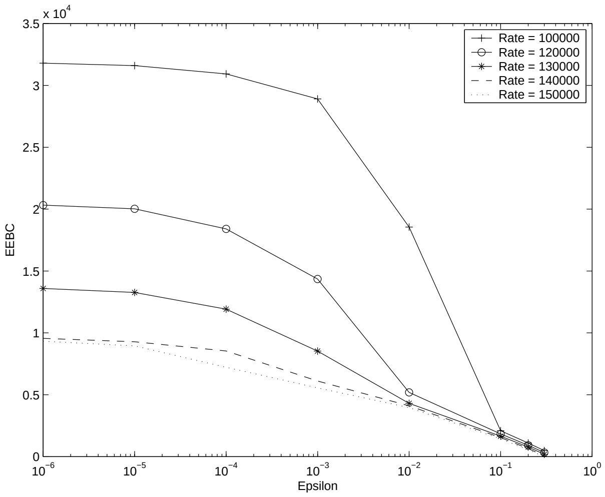

Figure 13: The EEBC as a function of ϵ for five different rate values.

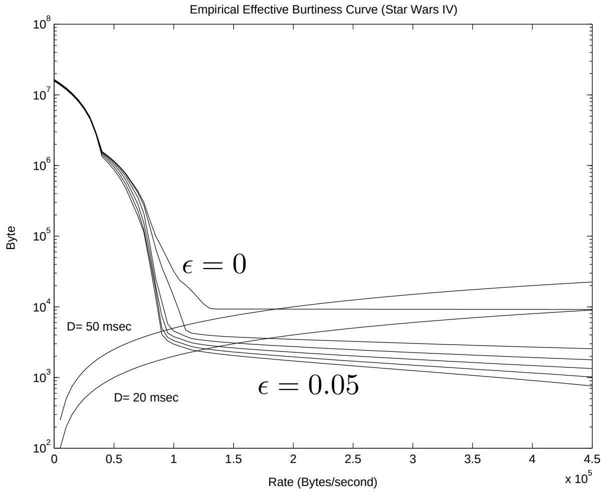

Figure 14: Intersection of EEBCs of the movie Star Wars IV for $\epsilon = 0, 0.01, \ldots, 0.05$ and the delay lines of 20 msec and 50 msec.

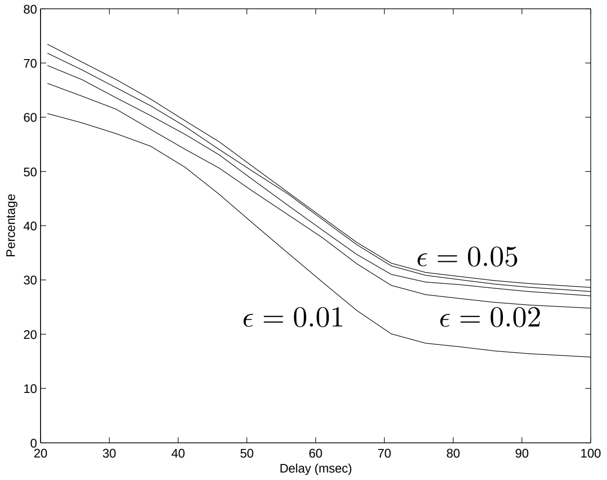

Figure 15: The percentage of bandwidth saved as a function of the maximum delay.

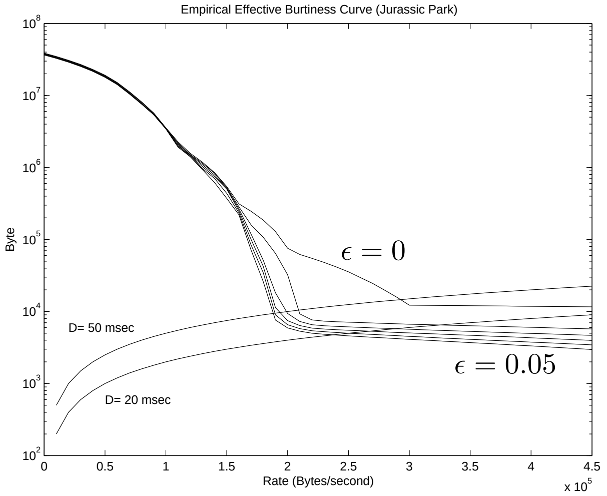

Figure 16: Intersection of the EEBC of the movie Jurassic Park for $\epsilon = 0, 0.01, \ldots, 0.2$ and the delay lines of 20 msec and 50 msec.

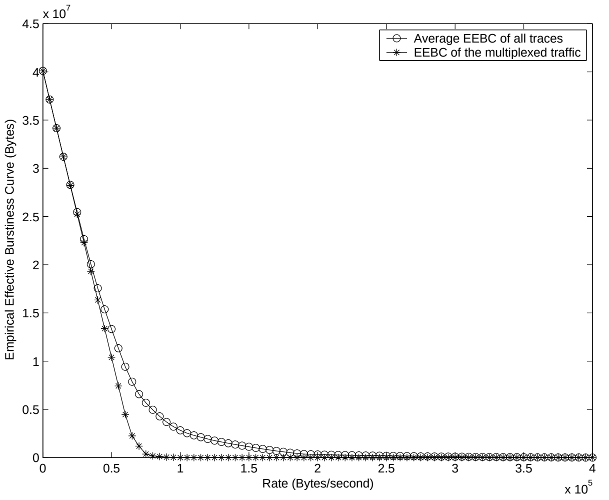

Figure 17: The convex combination of EEBCs of individual traces and EEBCs of the multiplexed trace.

trace, which is obtained by averaging all MPEG4 traces. Note that for very small service rates, the two curves coincide. Indeed, for $\rho = 0$, there will not be any service and the queue size will be equal to the addition of the traffic of all traces. For very high bandwidths, the system is over-provisioned and the two curves also tend to coincide. However, the curve of the multiplexed traffic decays much faster than the curve of the convex combination of EEBC of the traces. For intermediate bandwidths, the multiplexed traffic needs a smaller bandwidth.

Fig. 18 illustrates the percentage of buffer size that can be saved when the traces are multiplexed. Note that for very large bandwidth the relative saving is almost 100%, that is the required buffer size of the multiplexed traffic is much smaller than the convex combination of the buffer sizes of the individual MPEG4 traces. Therefore, for the same QoS, the multiplexed traffic requires much smaller buffer size per trace than the original traces stored in separate buffers.

In Fig. 19 and Fig. 20, we study the statistical multiplexing gain of EEBC. For any fixed buffer size ranging from 100 KBytes to 6 MBytes, we find the service rate required to maintain the loss bellow $\epsilon = 0.01$. First, we assume that each trace passes through a separate queue. For a fixed buffer size we find the service rates so that all traces have the loss rate $\epsilon = 0.01$. We add the service rates of all individual queues to get the total aggregated service rate. This value is shown by the upper curve in Fig. 19. In the second example, we create a single queue with a buffer size equal to the addition of all all buffers sizes of individual queues. We then pass the aggregated traffic, obtained by adding all traces together, through this queue and adjust the service rate to get the traffic loss $\epsilon = 0.01$. The corresponding service rate has been shown with the lower curve in Fig. 19. As expected, the service rate of the aggregated traffic is much smaller than the sum of the service rate of individual traces.

Fig. 20 illustrates the ratio of the sum of the service rates of individual traces and the service rate of the aggregated traffic. Note that the aggregated traffic uses about 25-50 times less bandwidth than the individual traces.

Fig. 21 illustrates the temporal behavior of EEBC for 10 minutes of the MPEG4 trace of the movie Star Wars IV for $\rho = 100, 150, 200$ KBytes/sec. Since the queue size is smaller for higher service rates, the empirical burstiness curve converges faster.

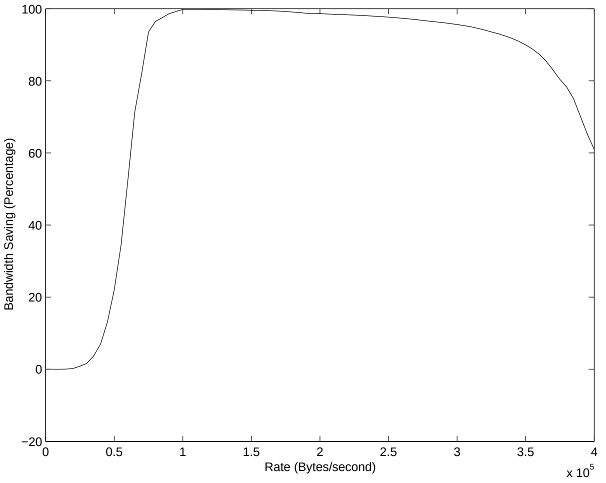

Figure 18: The percentage of burstiness (buffer size) that can be saved if the traces are aggregated.

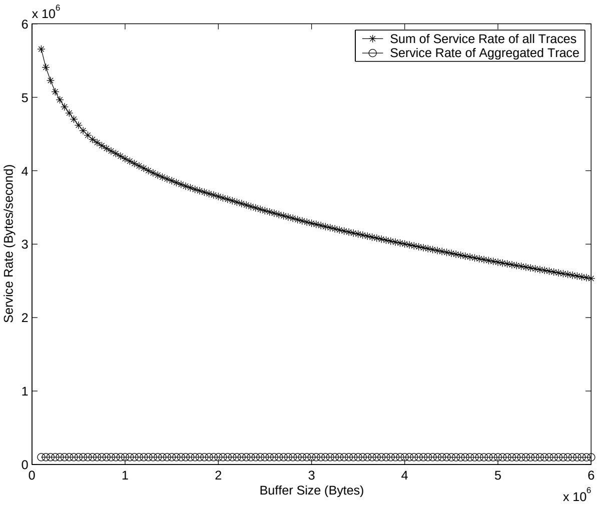

Figure 19: The service rate versus buffer size for individual traces and the aggregated trace.

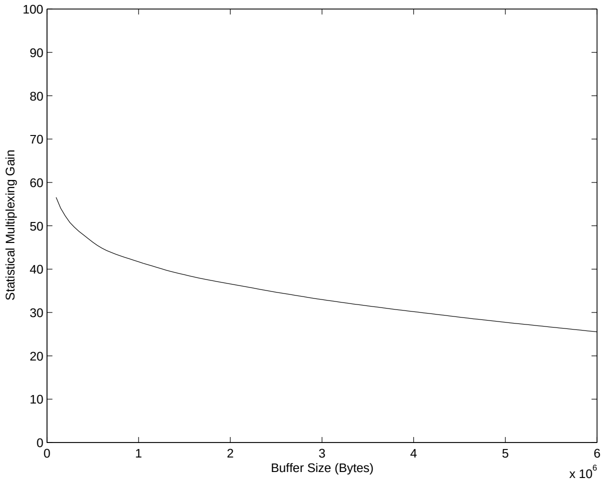

Figure 20: The statistical multiplexing gain as a function of the buffer size.

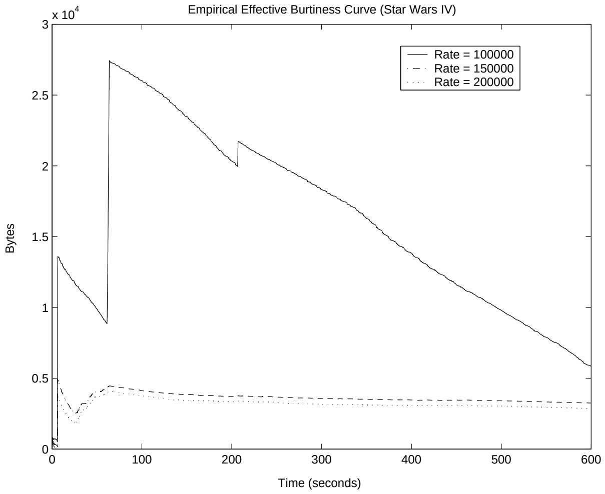

Figure 21: The temporal behavior of EEBC for three different values of the service rate.

6 Conclusion and Discussion

In this paper, we have developed a new traffic regulation scheme. The proposed technique uses the effective burstiness curve (EBC) which has been defined as a percentile of the queue size for a constant rate server. We have shown that the EBC is a convex non-increasing function of the service rate. It has also been shown that the EBC of a multiplexed traffic is smaller than the sum of the corresponding curves of individual flows. Therefore, multiplexing can reduce the EBC.

We have devised a "water-filling" algorithm to obtain the empirical effective burstiness curve (EEBC), which provides a sequence of convex decreasing curves converging to the true EBC. It has been shown that for discrete queues, EEBC can be obtained with a recursive algorithm. The proposed approach has been applied to MPEG4 encoded video traces. It has been shown that, for the given video traces, considerable savings in terms of required bandwidth and allocated buffer size can be obtained if the maximum burstiness curve is replaced by EBC.

The technique proposed in this paper can be applied to traffic streams in an on-line procedure. In an on-line estimation of EEBC, one can employ a number of auxiliary parallel constant-rate servers. The queue size in each server is used to determine EEBC at the given service rate. The total burstiness curve is then estimated by interpolating these points.

The traffic characterization studied in this paper is very useful in applications in which a prerecorded video trace is transmitted over a network with guaranteed bandwidth. One instance of such an application is video-on-demand in which a prerecorded video trace is transmitted at the request of a user. In such a case, if EEBC of the trace is available, it can be used to allocate an appropriate amount of bandwidth and buffer size to the transmitted stream in the network. The intersection of EEBC and the delay line $D_M \rho$ indicates a candidate point for resource allocation. If the bandwidth of the intersection point is available, it will be allocated, otherwise an alternative point—with presumably larger guaranteed delay—is selected. The burstiness at the selected point represents the maximum size of the buffer that should be reserved for the traffic, given that the minimum guaranteed bandwidth is indicated by the service rate of the selected point. The maximum burstiness curve at the selected service rate indicates the maximum buffer requirement both in the regulator and inside the network.

A The proof of Theorem 1

For simplicity of notation let $b_1 = b(\epsilon, \rho_1)$ and $b_2 = b(\epsilon, \rho_2)$ where

$$\mathrm{P}(Q_{\rho_1}(0) \geq b_1) \leq \epsilon \tag{34}$$

$$\mathrm{P}(Q_{\rho_2}(0) \geq b_2) \leq \epsilon \tag{35}$$

and define $\rho = \alpha\rho_1 + \beta\rho_2$ for $0 \leq \alpha \leq 1$, and $\beta = 1 - \alpha$. Since supremum is a convex function, we have, for all t,

$$\begin{aligned} Q_\rho(t) &= \sup_{s \leq t}\{A(s,t) - \rho(t-s)\} \\ &\leq \alpha \sup_{s\leq t}\{A(s,t) - \rho_1(t-s)\} + \beta \sup_{s \leq t}\{A(s,t) - \rho_2(t-s)\} \\ &= \alpha Q_{\rho_1}(t) + \beta Q_{\rho_2}(t). \end{aligned} \tag{36}$$

Define $\mathcal{R}_{\alpha,\epsilon}(\rho_1,\rho_2) \triangleq \left\{ b \,|\, \mathrm{P}\Big(\alpha Q_{\rho_1}(0) + \beta Q_{\rho_2}(0) \geq b\Big) \leq \epsilon \right\}$. Use part (iv) and (36) to conclude that $b(\epsilon, \rho) \leq \inf \mathcal{R}_{\alpha,\epsilon}(\rho_1,\rho_2)$. To establish convexity, we will show that $\alpha b_1 + \beta b_2 \in \mathcal{R}_{\alpha,\epsilon}(\rho_1,\rho_2)$ and hence $b(\epsilon,\rho) \le \alpha b_1 + \beta b_2$.

Assuming $\rho_1 < \rho_2$, there exists $\hat{t}$ such that

$$\begin{aligned} Q_{\rho_2}(0) &= A_{-\hat{t}} - \rho_2\hat{t}, & (37)\\ Q_{\rho_1}(0) &= Q_{\rho_1}(-\hat{t}) + A_{-\hat{t}} - \rho_1\hat{t}. & (38)\end{aligned}$$

From $\mathrm{P}\Big(Q_{\rho_1}(0) \ge b_1\Big) \le \epsilon$ and (38), we have

$$\mathrm{P}\Big(A_{-\hat{t}} - \rho_2\hat{t} \ge b_1 + (\rho_1 - \rho_2)\hat{t} - Q_{\rho_1}(-\hat{t})\Big) \le \epsilon. \tag{39}$$

Also from $\mathrm{P}\Big(Q_{\rho_2}(0) \ge b_2\Big) \le \epsilon$ and (37),

$$\mathrm{P}\Big(A_{-\hat{t}} - \rho_2\hat{t} \ge b_2\Big) \le \epsilon. \tag{40}$$

Since b_2 is the smallest burstiness curve satisfying (40), we conclude

$$b_1 + \rho_1\hat{t} - Q_{\rho_1}(-\hat{t}) \ge b_2 + \rho_2\hat{t}. \tag{41}$$

Now use (37) and (38) to get

$$\mathrm{P}\Big(\alpha Q_{\rho_1}(0) + \beta Q_{\rho_2}(0) \ge \alpha b_1 + \beta b_2\Big) = \mathrm{P}\Big(A_{-\hat{t}} \ge \alpha(b_1 + \rho_1\hat{t} - Q_{\rho_1}(-\hat{t})) + \beta(b_2 + \rho_2\hat{t})\Big) \tag{42}$$

Using (41) and (40), we have

$$\mathrm{P}\Big(\alpha Q_{\rho_1}(0) + \beta Q_{\rho_2}(0) \ge \alpha b_1 + \beta b_2\Big) \le \mathrm{P}\Big(A_{-\hat{t}} \ge b_2 + \rho_2\hat{t}\Big) \le \epsilon. \tag{43}$$

Therefore, $\alpha b_1 + \beta b_2 \in \mathcal{R}_{\alpha,\epsilon}(\rho_1,\rho_2)$ and the proof is complete.

B Proof of Theorem 2

To prove Theorem 2, we will need the following lemma.

Lemma 3 *Assume positive causal functions $f_i(t)$, $i = 1,\ldots,L$ that satisfy $\int_0^\infty f_i(t)dt = 1$. Choose b_i, $i = 1,\ldots,L$ so that $\int_{b_i}^\infty f_i(t)dt \le \epsilon_i$ for some $\epsilon_i > 0$. Then*

$$\int_{\sum_{i=1}^L b_i}^\infty f_1(t) * \ldots * f_L(t)\, dt \le \min_{1\le i\le L} \epsilon_i \tag{44}$$

where $*$ is the convolution operator.

Proof: Let $\hat{f}(t) \triangleq f_2(t) * \ldots * f_L(t)$. The left-hand-side of (44) can be written as

$$\begin{aligned} \int_{\sum_{i=1}^L b_i}^\infty f_1(t) * \hat{f}(t)\, dt &= \int_{\sum_{i=1}^L b_i}^\infty dt \int_0^\infty f_1(t-s)\hat{f}(s)\, ds \\ &= \int_0^\infty \hat{f}(s)\, ds \int_{s+\sum_{i=1}^L b_i}^\infty f_1(r)\, dr. & (45)\end{aligned}$$

Now use $b_1 < s + \sum_{i=1}^{L} b_i$ for all $s \geq 0$ and the fact that $\displaystyle\int_0^\infty f_1(t)dt = 1$ and $f_1(t) \geq 0$ to get

$$\int_{s+\sum_{i=1}^{L} b_i}^{\infty} f_1(r)\, dr \leq \epsilon_1. \tag{46}$$

Therefore, (45) can be written as

$$\int_{\sum_{i=1}^{L} b_i}^{\infty} f_1(t) * \hat{f}(t)\, dt \leq \epsilon_1 \int_0^\infty \hat{f}(s)\, ds. \tag{47}$$

Note that $\displaystyle\int_0^\infty \hat{f}(s)\, ds = 1$ since $\hat{f}(s)$ can be interpreted as the probability density function of $L-1$ independent random variables. Therefore

$$\int_{\sum_{i=1}^{L} b_i}^{\infty} f_1(t) * \hat{f}(t)\, dt \leq \epsilon_1. \tag{48}$$

Since $f_1(t)$ has been randomly selected, the left-hand-side of (48) is indeed smaller than $\min_{1\leq i \leq L} \epsilon_i$. □

Proof of Theorem 2: Similar to (36), using the convexity of supremum, we can show that

$$Q_\rho(t) \leq \sum_{i=1}^{L} \alpha_i Q_\rho^{(i)}(t). \tag{49}$$

Define $\mathcal{R}_{\underline{\alpha},\epsilon}(\rho) \triangleq \{b \,|\, \mathrm{P}(\sum_{i=1}^{L} \alpha_i Q_\rho^{(i)}(0) \geq b) \leq \epsilon\}$. Use Lemma 1-($iv$) and (49) to get $b(\epsilon,\rho) \leq \inf \mathcal{R}_{\underline{\alpha},\epsilon}(\rho)$. Next, we show that $\sum_{i=1}^{L} \alpha_i b^{(i)}(\epsilon,\rho) \in \mathcal{R}_{\underline{\alpha},\epsilon}(\rho)$.

Let the probability density function of $\alpha_i Q_\rho^{(i)}(0)$ be represented by $f_i(t)$. Therefore,

$$\int_{\alpha_i b^{(i)}(\epsilon,\rho)}^{\infty} f_i(t)\, dt = \mathrm{P}\Big(\alpha_i Q_\rho^{(i)}(0) \geq \alpha_i b^{(i)}(\epsilon,\rho)\Big) \leq \epsilon. \tag{50}$$

Since $Q_\rho^{(i)}(0)$, $i = 1, \ldots, L$ are independent random variables, we have

$$\mathrm{P}\Big(\sum_{i=1}^{L} \alpha_i Q_\rho^{(i)}(0) \geq \sum_{i=1}^{L} \alpha_i b^{(i)}(\epsilon,\rho)\Big) = \int_{\sum_{i=1}^{L} \alpha_i b^{(i)}(\epsilon,\rho)}^{\infty} f_1(t) * \ldots * f_L(t)\, dt. \tag{51}$$

Now use (50) and Lemma 3 to get

$$\mathrm{P}\Big(\sum_{i=1}^{L} \alpha_i Q_\rho^{(i)}(0) \geq \sum_{i=1}^{L} \alpha_i b^{(i)}(\epsilon,\rho)\Big) \leq \epsilon. \tag{52}$$

This completes the proof.

C Proof of Theorem 3

Parts (i) − (iii) are obvious by construction. We prove part (iv) here. The objective is to show that $b_n(\epsilon,\rho) \leq \alpha b_n(\epsilon,\rho_1) + \beta b_n(\epsilon,\rho_2)$ where $\rho = \alpha\rho_1 + \beta\rho_2$, $\alpha \geq 0$, $\beta \geq 0$, and $\alpha + \beta = 1$. For simplicity of notation, denote $b_1 = b_n(\epsilon,\rho_1)$ and $b_2 = b_n(\epsilon,\rho_2)$.

Let q_i indicate the queue size at the ith time instant for a server with the service rate ρ. The queue size can be represented by

$$q_i = \sup_{\ell \le i}\{A(\ell,i) - (i-\ell)\rho\}. \tag{53}$$

The subadditive property of the supremum gives

$$\begin{aligned} q_i &\le \alpha \sup_{\ell \le i}\{A(\ell,i) - (i-\ell)\rho_1\} + \beta \sup_{\ell \le i}\{A(\ell,i) - (i-\ell)\rho_2\} \\ &= \alpha q_{1,i} + \beta q_{2,i} \end{aligned} \tag{54}$$

where $q_{1,i}$ and $q_{2,i}$ represent the queue size at time instant i for the service rates ρ_1 and ρ_2, respectively.

Without loss of generality, we assume that $\rho_1 < \rho_2$. With this assumption the queue sizes $q_{1,i}$ and $q_{2,i}$ can be represented by

$$q_{2,i} = A(\hat{\ell},i) - (i-\hat{\ell})\rho_2 \tag{55}$$

$$q_{1,i} = q_{1,\hat{\ell}} + A(\hat{\ell},i) - (i-\hat{\ell})\rho_1 \tag{56}$$

where $\hat{\ell}$ is the time instant at which the supremum of (53) is achieved.

The metric $\mu_n(\sigma)$ can also be written as

$$\mu_n(\sigma) = \frac{1}{n}\sum_{i=0}^{n-1} \min\Big\{\max\Big\{\frac{q_i - \sigma}{\rho}, 0\Big\}, 1\Big\}. \tag{57}$$

Define the truncating function $\mathcal{T}_\alpha^\beta(.)$ as

$$\mathcal{T}_\alpha^\beta(t) = \begin{cases} \alpha & \text{if } t \le \alpha \\ t & \text{if } \alpha < t < \beta \\ \beta & \text{if } \beta \le t \end{cases} \tag{58}$$

Using this function, (57) will be represented as

$$\mu_n(\sigma) = \frac{1}{n}\sum_{i=0}^{n-1} \mathcal{T}_0^1(\frac{q_i - \sigma}{\rho}). \tag{59}$$

From the definition of the EEBC, we have

$$\frac{1}{n}\sum_{i=0}^{n-1} \mathcal{T}_0^1\Big(\frac{q_{1,i} - b_1}{\rho_1}\Big) = \epsilon \tag{60}$$

$$\frac{1}{n}\sum_{i=0}^{n-1} \mathcal{T}_0^1\Big(\frac{q_{2,i} - b_2}{\rho_2}\Big) = \epsilon \tag{61}$$

Using (56) and (55), we get

$$\frac{1}{n}\sum_{i=0}^{n-1} \mathcal{T}_0^1\Big(\frac{A(\hat{\ell},i) - (i-\hat{\ell})\rho_1 - b_1 + q_{1,\hat{\ell}}}{\rho_1}\Big) = \epsilon \tag{62}$$

$$\frac{1}{n}\sum_{i=0}^{n-1} \mathcal{T}_0^1\Big(\frac{A(\hat{\ell},i) - (i-\hat{\ell})\rho_2 - b_2}{\rho_2}\Big) = \epsilon \tag{63}$$

Applying $\rho_1 \leq \rho_2$ in (62) and (63) gives

$$(i-\hat{\ell})\rho_2 + b_2 \leq (i-\hat{\ell})\rho_1 + b_1 - q_{1,\hat{\ell}}. \tag{64}$$

We now apply $\alpha b_1 + \beta b_2$ in the definition of the EEBC and use (54) to get

$$\frac{1}{n}\sum_{i=0}^{n-1}\mathcal{T}_0^1\Big(\frac{q_i-(\alpha b_1+\beta b_2)}{\rho}\Big) \leq \frac{1}{n}\sum_{i=0}^{n-1}\mathcal{T}_0^1\Big(\frac{\alpha q_{1,i}+\beta q_{2,i}-(\alpha b_1+\beta b_2)}{\rho}\Big). \tag{65}$$

Substitute (55) and (56) in (65) and get

$$\begin{aligned}\frac{1}{n}\sum_{i=0}^{n-1}\mathcal{T}_0^1\Big(\frac{q_i-(\alpha b_1+\beta b_2)}{\rho}\Big) &\leq \frac{1}{n}\sum_{i=0}^{n-1}\mathcal{T}_0^1\left(\frac{A(\hat{\ell},i)-\alpha\Big((i-\hat{\ell})\rho_1+b_1-q_{1,\hat{\ell}}\Big)-\beta\Big((i-\hat{\ell})\rho_2+b_2\Big)}{\rho}\right)\\ &\leq \frac{1}{n}\sum_{i=0}^{n-1}\mathcal{T}_0^1\Big(\frac{A(\hat{\ell},i)-(i-\hat{\ell})\rho_2-b_2}{\rho}\Big) \qquad (66)\end{aligned}$$

where we have used (64). Note that ρ is a convex combination of ρ_1 and ρ_2 and therefore $\rho \leq \rho_2$. Use this inequality in (66) to get

$$\frac{1}{n}\sum_{i=0}^{n-1}\mathcal{T}_0^1\Big(\frac{q_i-(\alpha b_1+\beta b_2)}{\rho}\Big) \leq \epsilon. \tag{67}$$

Since the left hand side of (67) is smaller than ϵ it is possible to find an EEBC $b_n(\epsilon,\rho) \leq \alpha b_1+\beta b_2$ such that

$$\frac{1}{n}\sum_{i=0}^{n}\mathcal{T}_0^1(\frac{q_i-b(\epsilon,\rho)}{\rho}) = \epsilon. \tag{68}$$

And the proof is complete.

D Proof of Theorem 4

Note first that $\lim_{n\to\infty}\ell_n(m)=\infty$ for all $m=0,\ldots,M$. This is due to the fact that in a stable queue all states are recurrent. We investigate two cases:

Case (i): q_n is a dry state. Note that

$$\lim_{n\to\infty}V_{b,n} = \lim_{n\to\infty}\Big(b_{n-1}-a_m\Big)\ell_n(m+1)=\infty. \tag{69}$$

Therefore, for very large n, we have $\delta \in I_0$ in (28) and

$$b_n = b_{n-1} - \frac{\delta}{\ell_n(m+1)}. \tag{70}$$

This indicates that $\|b_n - b_{n-1}\| \to 0$ for $n \to \infty$. Hence, $\{b_n\}$ is a Cauchy sequence and is convergent.

Case (ii): q_n is a wet state. Here, W_{n-1} grows much faster than $V_{q,n}$ and thus for very large n we have $W_{n-1} \in I_{k-m-1}$ in (31). Hence

$$b_n = a_{m+1} - \frac{V_{q,n-1}+\sum_{i=1}^{k-m-2}V_{k-i,n-1}+V_{b,n-1}-V_{q,n}-\sum_{i=1}^{k-m-2}V_{k-i,n}}{\ell_n(m+1)} \tag{71}$$

We use approximations, $V_{q,n-1} \approx V_{q,n}$, $V_{k-i,n-1} \approx V_{k-i,n}$ and $\ell_{n-1}(m+1) \approx \ell_n(m+1)$, to get $b_n \approx b_{n-1}$.

From the above discussion we conclude that the sequence $\{b_n\}$ is convergent and assign $\lim_{n\to\infty} b_n = b_\infty$. Note also that from (33) we have

$$\mathrm{P}(Q_\rho(0) > b_\infty) = \epsilon. \tag{72}$$

The objective is to show that $b_\infty = b(\epsilon, \rho)$. We prove this by contradiction. First, let $b_\infty < b(\epsilon,\rho)$. This is in contradiction to the definition of the EBC and therefore is not valid. Second, let $b_\infty > b(\epsilon,\rho)$. A water-filling procedure resulting in $b(\epsilon,\rho)$ will require $\Delta_n + \Big(b_\infty - b(\epsilon,\rho)\Big)\ell_\infty(m+1)$ units of fluid where we have assumed $a_m < b(\epsilon,\rho) < b_\infty < a_{m+1}$. Therefore, using (33), one gets

$$\lim_{n\to\infty} \frac{1}{n\rho}\Big[\Delta_n + \Big(b_\infty - b(\epsilon,\rho)\Big)\ell_\infty(m+1)\Big] = \epsilon + \Big(\frac{b_\infty - b(\epsilon,\rho)}{\rho}\Big)\mathrm{P}\Big(Q_\rho(0) > a_m + 1\Big) = \epsilon'. \tag{73}$$

Since both $b_\infty - b(\epsilon,\rho) > 0$ and $\mathrm{P}\Big(Q_\rho(0) > a_m + 1\Big) > 0$, we have that $\epsilon' > \epsilon$. This is also in contradiction to the definition of $b(\epsilon,\rho)$.

References

[1] L. Kleinrock, *Queueing Systems*, vol. I. John Wiley & Sons, 1975.

[2] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic," in *ACM SIGCOMM*, pp. 183–193, 1993.

[3] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, June 1995.

[4] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," *ACM SIGCOMM*, September 1994.

[5] B. K. Ryu and A. Elwalid, "The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities," *ACM SIGCOMM Computer Communications Review*, vol. 26, pp. 3–14, October 1996.

[6] I. Norros, "Studies on a model for connectionless traffic, based on fractional Brownian motion," *Conference on Applied Probability in Engineering, Computer and Communication Sciences*, June 1993.

[7] 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.

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

[9] T. Konstantopoulos and V. Anantharam, "Optimal flow control schemes that regulate the burstiness of data," *IEEE/ACM Trans. Networking*, vol. 3, pp. 423–432, August 1995.

[10] S. Rajagopal, M. Reisslein, and K. W. Ross, "Packet multiplexers with adversarial regulated traffic," in *Proceeding IEEE INFOCOM*, vol. 1, pp. 347–355, 1998.

[11] C. S. Chang, *Performance Guarantees in Communication Networks.* Springer Verlag, 2000.

[12] J. Y. Le Boudec and P. Thiran, *Network Calculus.* Springer Verlag, 2001. Also available at: http://ica1www.epfl.ch/PS_files/NetCal.htm.

[13] 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.

[14] H. Zhang and E. W. Knightly, "Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models," in *ACM Sigmetrics*, pp. 211–220, 1994.

[15] E. W. Knightly, D. E. Wrege, J. Liebeherr, and H. Zhang, "Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic," in *the ACM SIGMETRICS/PERFORMANCE*, pp. 98–107, 1995.

[16] M. Reisslein, K. W. Ross, and S. Rajagopal, "A framework for guaranteeing statistical QoS," *IEEE/ACM Trans. Networking*, vol. 10, pp. 27–42, Feb. 2002.

[17] F. Lo Presti, Z. L. Zhang, J. Kurose, and D. Towsley, "Source time scale and optimal buffer/bandwidth trade-off for regulated traffic in an ATM node," *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 490–501, 1999.

[18] G. Kesidis and T. Konstantopoulos, "Extremal shape-controlled traffic patterns in high-speed networks," *IEEE Transactions on Communications*, vol. 48, pp. 813–819, May 2000.

[19] C. S. Chang, Y. M. Chiu, and W. T. Song, "On the performance of multiplexing independent regulated inputs," *Proceeding of ACM Sigmetrics*, pp. 184–193, May 2001.

[20] M. Vojnovic and J. Y. Le Boudec, "Bounds for independent regulated inputs multiplexed in a service curve network element," *IEEE Trans. on Comm.*, vol. 51, pp. 735–740, May 2003.

[21] K. Kumaran and M. Mandjes, "Multiplexing regulated traffic streams: Design and performance," in *Proceeding IEEE INFOCOM*, vol. 1, pp. 527 –536, March 2001.

[22] J. Liebeherr, S. Patek, and A. Burchard, "Statistical per-flow service bounds in a network with aggregate provisioning," in *Proceeding IEEE INFOCOM*, pp. 1680–1690, 2003.

[23] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn, "Effective envelopes: statistical bounds on multiplexed traffic in packet networks," in *Proceeding IEEE INFOCOM*, pp. 1223–1232, 2000.

[24] N. O'Connell, "Large deviation with applications to telecommunications," Nov. 1999. Lecture Notes for a course presented at Uppsala University.

[25] A. Dembo and O. Zeitouni, *Large Deviation Techniques and Applications*, vol. 38. Springer, Application of Mathematics, 2 ed., 1998.

[26] P. W. Glynn and W. Whitt, “Logarithmic asymptotics for steady-state tail probabilities in a single-server queue,” *Studies in Applied Probability*, pp. 131–156, 1994.

[27] N. G. Duffield and N. O’Connell, “Large deviation and overflow probabilities for the general single-server queue with applications,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 118, pp. 363–374, 1995.

[28] A. Elwalid, D. Heyman, and T. V. Lakshman, “Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing,” *IEEE J., Select., Areas Commun.*, vol. 13, pp. 1004–1016, August 1995.

[29] I. Norros, “A storage model with self-similar input,” *Queueing Systems*, vol. 16, pp. 387–396, 1994.

[30] S. Low and P. Varaiya, “A simple theory of traffic resource allocation in ATM,” in *Proceeding IEEE Globecom*, pp. 1633–1637, 1991.

[31] S. Valaee, “A recursive estimator of worst-case burstiness,” *IEEE/ACM Trans. Networking*, vol. 9, pp. 211–222, April 2001.

[32] W. Feller, *An Introduction to Probability Theory and its Applications*, vol. I. John Wiley & Sons, 1968.

[33] http://www-tkn.ee.tu berlin.de/research/trace/trace.html