Telecommunication Systems 21:1, 35–63, 2002
© 2002 Kluwer Academic Publishers. Manufactured in The Netherlands.

Buffered and Unbuffered Leaky Bucket Policing: Guaranteeing QoS, Design and Admission Control *

NATARAJAN GAUTAM ngautam@psu.edu
Harold and Inge Marcus Department of Industrial and Manufacturing Engineering,
The Pennsylvania State University, 310 Leonhard Building, University Park, PA 16802, USA

Abstract. Traffic shaping and smoothing using buffers or leaky buckets does not necessarily improve Quality of Service (QoS). In fact there is a trade-off between controlling user traffic and guaranteeing QoS to the users. We consider the first two stages (source node and border node before entering a network cloud) of an end-to-end QoS problem and assume that the QoS requirements across each of the two stages are given. We formulate and solve a mathematical programming problem to select optimal leaky bucket parameters that would enable high-speed telecommunication network providers to optimize traffic policing subject to guaranteeing a negotiated Quality of Service requirement across the first stage namely the source end. We address both the buffered and unbuffered leaky bucket cases where using fluid models we characterize the output process from the leaky buckets for general traffic sources. Using the optimal leaky bucket parameters and output characteristics (effective bandwidths in particular), we solve design and connection admission control problems given QoS requirements at the second stage, namely the border node.

Keywords: leaky bucket policing, quality of service, admission control, traffic regulation

1. Introduction

The proliferation of the Internet and its excessive congestion has led researchers working on emerging high-speed telecommunication networks to develop tools to police and control the traffic at the user or source end. These policing mechanisms need to not only ensure that the telecommunication network traffic generated by the sources are kept below a negotiated threshold but also ensure that the users receive a reasonable performance called Quality of Service (QoS) that they have been promised. QoS is typically in terms of packet loss rate, delay, delay–jitter and bandwidth, and must be guaranteed end-to-end (i.e. source to destination).

One policing mechanism is the leaky bucket (see [Cidon and Gopal, 12; Gu et al., 22; Gün et al., 23; Vamvakos and Anantharam, 34; Butto et al., 4; Callegati et al., 5; Holtsinger and Perros, 25; Sohraby and Sidi, 33; Wu and Mark, 35; Yin and Hluckyj, 36]). A leaky bucket is essentially a credit management mechanism that controls the traffic entering the network. A single or a series of leaky buckets can be used to optimally regulate the source traffic in communication networks (see [Anantharam and Konstantopoulos, 2]).

* This work was partially supported by NSF Grant No. NCR-9406823, and by the Center for Advanced Computing and Communication at Duke University.

There is a wide variety of network traffic, for example, data, audio, video, etc. These different types of traffic have varying QoS requirements that the network provider must guarantee. For example real-time traffic can tolerate some loss but not much delay and non-real-time traffic can tolerate some delay but not much loss. In this paper we study the QoS in terms of the loss and delay that the user traffic faces upon introducing a leaky bucket at the source of traffic. We consider the scenario (shown in figure 1) where sources policed by leaky buckets generate traffic which is aggregated and multiplexed at a border node before entering a network cloud. The network cloud is usually administered, owned and controlled by different organizations as compared to the source node and the border node. Therefore a meaningful way to provide end-to-end QoS is to budget it over the individual stages of the network. We assume in this paper that the QoS budget across the first two stages (namely, the source nodes and the border node) for the different traffic streams are specified.

Consider figure 2 where various system inputs, outputs and requirements are illustrated. At the first stage (associated with the source nodes), we formulate and solve a nonlinear programming problem to choose optimal leaky bucket parameters, given the source characteristics and the QoS requirements. At the second stage, where there

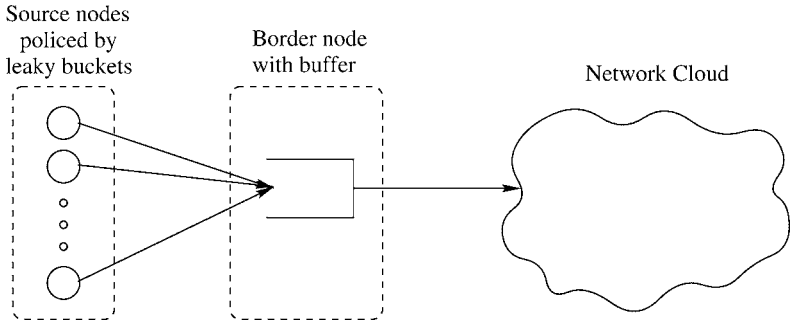

Figure 1. The problem setting.

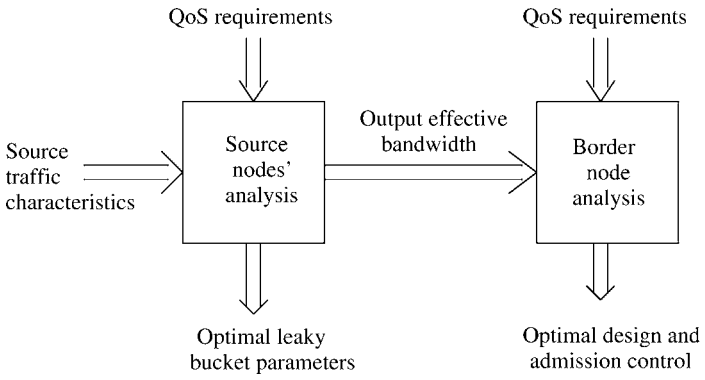

Figure 2. Two-stage system: inputs, outputs and requirements.

is a single buffer on the border node, given the QoS requirements, we solve optimal design and admission control problems using the effective bandwidth of the output from the leaky buckets at the source. Since using effective bandwidth analysis at a buffer to measure QoS performance is reasonably standard, the main research effort in this paper is in obtaining (but not illustrating the use of) the output effective bandwidth from the leaky buckets. For that same reason, we use a simplistic network structure of one buffer (as opposed to a network of buffers) to illustrate the use of the output effective bandwidth analysis with the understanding that well-known results can be used if the single node is replaced by a network. The concept of effective bandwidths is now well documented and accepted (see [Gibbens and Hunt, 20; Chang and Thomas, 6; Kesidis et al., 26; Elwalid and Mitra, 18; Choudhury et al., 11; Elwalid et al., 16; Kulkarni, 27; de Veciana et al., 13, 15]). In this paper we use stochastic fluid-flow models to describe the traffic flow, following the large literature using fluid-flow models for communication systems (see [Anick et al., 3; Elwalid and Mitra, 17], etc.). Chen and Yao [9, 10], Ott and Shanthikumar [31], Harrison [24], Chen and Mandelbaum [8], etc., demonstrate how to convert any discrete arrival system into a fluid-flow system and apply the fluid-flow model results.

In summary, we address two main issues in this paper: one is, given the characteristics of the input source, how to set the parameters of the leaky bucket, such that the negotiated QoS guarantees at the source node can be met. The other issue deals with the border node where we consider both the design of network parameters such as buffer sizes and link speeds as well as connection admission control which is to decide whether or not to accept an arriving connection for admission based on the currently admitted sources. The contribution of this research includes the optimal leaky bucket parameter problem formulation (including deriving expressions for the constraints), the algorithm for the optimization problem and the output effective bandwidth analysis for design and admission control at the border node so that negotiated QoS guarantees are met.

Broadly, there are two types of leaky buckets, the buffered and the unbuffered leaky buckets. In section 2 we describe the notation for both buffered and unbuffered leaky bucket models. In section 3 we consider the source node to formulate and solve a mathematical programming problem to optimally choose the leaky bucket parameters. In section 4 we consider the border node to solve channel capacity design and connection admission control problems using effective bandwidth of the output from leaky buckets derived in section 5. In section 6 we present the numerical results for various cases of buffered and unbuffered leaky buckets design and admission control problems. Finally, in section 7 we state the conclusions and future extensions of the work.

2. Leaky bucket preliminaries

In this section we first describe the working and the notation used for both buffered as well as unbuffered leaky buckets. “Leaky Bucket” is a control mechanism for admitting data into a network. It consists of a data buffer and a token pool as shown in figure 3.

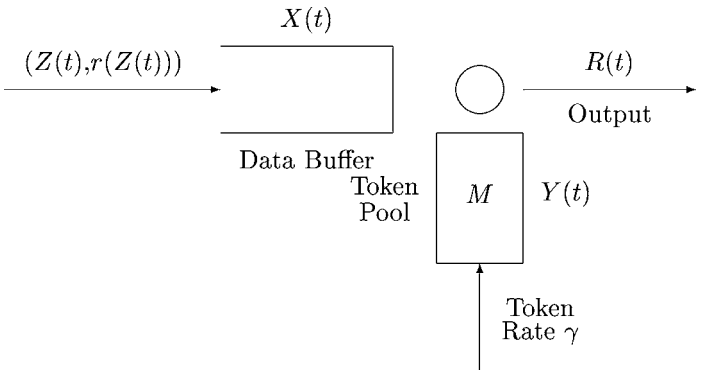

Figure 3. Single leaky bucket.

We use a fluid-flow leaky bucket model assuming that the data traffic and tokens can be modeled as fluids. Tokens are generated continuously at a fixed rate γ into the token pool of size M. The new tokens are discarded if the token pool is full. External data traffic enters the data buffer (of size B_D) from a source modulated by an environmental process $\{Z(t), t \geqslant 0\}$. Traffic is generated by this source at rate $r(Z(t))$ at time t.

If there are tokens in the token pool, the incoming fluid takes an equal amount of tokens and enters the network. If the token pool is empty then we have two alternative implementations:

- Buffered leaky bucket: the packets wait in the infinite capacity data buffer ($B_D = \infty$) for tokens to arrive.
- Unbuffered leaky bucket: there is no data buffer ($B_D = 0$) for the packets and any packet that does not find a token enters the network carrying a "violation" tag. Later such violation traffic can be dropped if congestion develops.

The leaky bucket is usually located at the user end. When the user (or source) generates traffic to a destination in the network, the leaky bucket acts as a credit management mechanism that controls the traffic entering the network. In practice, since the traffic flows through several different networks owned by different organizations, one way to provide QoS is to appropriately budget the required performance over the different network domains. In this paper we concentrate on guaranteeing QoS across the first two nodes, namely, the "source node" and the "border node" which are typically owned by the same organization.

We now describe the notation used for the source nodes and the border node (refer to figure 4). Assume that there are N (a fixed positive integer) sources of traffic at a source node. Also assume that the ith ($i = 1, 2, \ldots, N$) input source is policed by a leaky bucket with parameters γ_i and M_i. We shall consider both the buffered and unbuffered leaky bucket cases. The output from the N leaky buckets is multiplexed onto the border node (with a single buffer of size B and constant output capacity c). Upon exiting this buffer, the multiplexed traffic enters a downstream node such as a router or switch owned by a different organization from where each source of traffic is

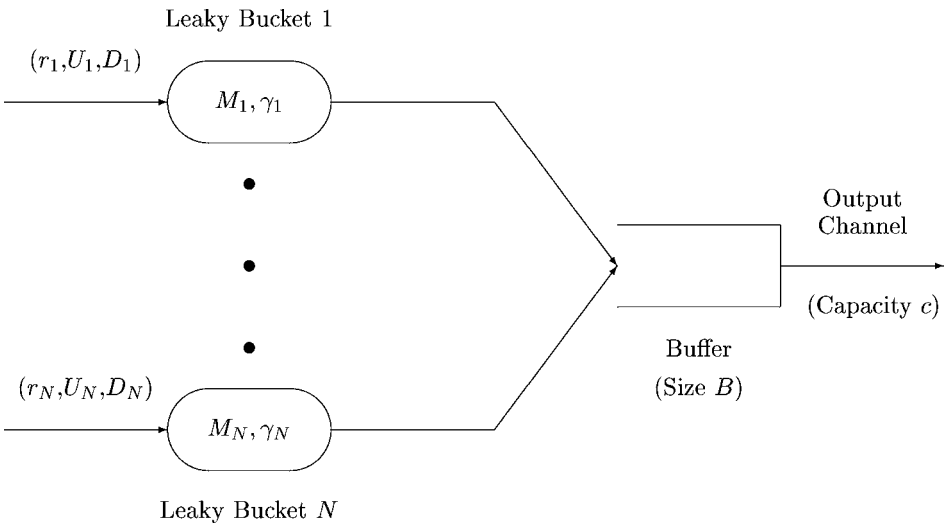

Figure 4. The leaky buckets node model.

appropriately routed to its destination. Note that in the unbuffered case all the packets (tagged or untagged) enter this buffer of size B. However, when the buffer gets full, the tagged packets are first dropped from the buffer before any of the untagged ones are affected.

3. Source node: optimal leaky bucket parameters

We now solve the problem of selecting optimal leaky bucket parameters, given the QoS requirements across the source nodes (see the left half of figures 2 and 4). First the setting is described in terms of the QoS requirements and source traffic characteristics (section 3.1). Then an optimization problem is formulated for both buffered and unbuffered leaky buckets (section 3.2). Finally, an algorithm is described to solve the optimization problem (section 3.3).

3.1. QoS requirements and source traffic characteristics

In this paper we consider selecting optimal leaky bucket parameters M_i and γ_i $(1 \leqslant i \leqslant N)$ subject to satisfying the following QoS constraints (arising from breaking down the end-to-end QoS guarantee requirements into individual node QoS requirements) at the source node:

- In case of the buffered leaky bucket, the QoS guarantee specifies that as long as the ith source adheres to its agreed upon characteristics, the fraction of the traffic that faces a delay of more than a fixed amount d_i^* is bounded above by ζ_i. (We refer to this as the waiting time constraint for the buffered leaky bucket.)

- In the case of the unbuffered leaky bucket, the QoS guarantee specifies that as long as the ith source adheres to its agreed upon characteristics, the fraction of the traffic that gets tagged as violation traffic is bounded above by ζ_i. (We refer to this as the tagging constraint for the unbuffered leaky bucket.)

Note that since there is no loss at the leaky buckets, we are not considering QoS loss requirement here.

Assume that the ith ($i = 1, 2, \ldots, N$) source is a general on-off source with on-times generally distributed with CDF $U_i(\cdot)$ (and mean τ_U^i) and off-times generally distributed with CDF $D_i(\cdot)$ (and mean τ_D^i). Note that the letters U and D are used to denote "up" and "down" as a surrogate for "on" and "off", respectively. When source i is on, traffic is generated at rate r_i also known as the peak rate, and, when the source is off, no traffic is generated. The mean input rate from the ith source is

$$m_i = \frac{r_i \tau_U^i}{\tau_D^i + \tau_U^i}. \tag{1}$$

The Laplace–Stieltjes Transforms (LSTs) of the on-times and off-times for the ith source are denoted by $\widetilde{U}_i(s)$ and $\widetilde{D}_i(s)$.

3.2. Formulation

To choose the optimal leaky bucket parameters subject to satisfying given QoS waiting time or tagging constraints across the source node, consider the following nonlinear programming (NLP) problem:

$$\text{Minimize} \sum_{i=1}^{N} \gamma_i$$

Subject to:

I. Waiting time or tagging constraint at the leaky bucket,

II. $m_i < \gamma_i \leqslant r_i, \quad \text{for } 1 \leqslant i \leqslant N,$

III. $\sum_{i=1}^{N} M_i \leqslant \overline{M}.$

We first explain the objective function. The parameter γ_i of the ith leaky bucket serves the following function: no matter how badly the source behaves, the data rate in arbitrarily long bursts that the source can send into the network is bounded above by γ_i. Thus if all sources were to simultaneously misbehave, the network will get traffic at a maximum rate of $\sum_{i=1}^{N} \gamma_i$. Hence it makes sense to minimize $\sum_{i=1}^{N} \gamma_i$ in order to ensure that this worst case situation is kept the best possible.

Constraint (I) arises out of the QoS guarantee stipulations in section 3.1. Explicit algebraic expressions for constraint (I) for the buffered and unbuffered leaky bucket cases are provided in sections 3.2.1 and 3.2.2, respectively. In constraint (II), $m_i < \gamma_i$

is needed for stability, and $\gamma_i \leqslant r_i$ is needed to keep the the leaky bucket operation nontrivial. The quantitative expression for constraint (I), to be derived later, is valid only in the range $m_i < \gamma_i \leqslant r_i$. In constraint (III), the parameter M_i can be thought of the largest instantaneous burst that the leaky bucket will allow from the ith source. Thus if $\overline{M}$ is the largest burst that the network can handle (for example, we may set $\overline{M} = B$ or $\overline{M} = B/2$, where B is the size of the buffer in the border node) then it makes sense to add constraint (III). It will be seen that this turns out to be a crucial constraint.

3.2.1. Buffered leaky buckets: constraint (I)
Consider the buffered leaky bucket system where the data buffer has infinite capacity. Recall that d_i^* is defined such that the expected fraction of traffic from the ith source that faces a delay greater than d_i^* must be less than or equal to ζ_i. The following theorem illustrates the waiting time constraint at the input buffer.

Theorem 1. The waiting time constraint at the buffered leaky bucket is satisfied if

$$C_i^* \frac{\gamma_i}{m_i} \mathrm{e}^{-\eta_i(M_i+\gamma_i d_i^*)} \leqslant \zeta_i, \tag{2}$$

where

$$C_i^* = \frac{\widetilde{U}_i(-\eta(r_i-\gamma_i))-1}{\eta_i(\tau_U^i+\tau_D^i)} \frac{r_i}{(r_i-\gamma_i)\gamma_i} \Bigg/ \min_{x\geqslant 0}\left\{\frac{\int_x^\infty \mathrm{e}^{\eta_i(r_i-\gamma_i)(y-x)}\,\mathrm{d}U_i(y)}{1-U_i(x)}\right\}, \tag{3}$$

η_i is the solution to

$$\widetilde{U}_i(-\eta_i(r_i-\gamma_i))\widetilde{D}_i(\eta_i\gamma_i) = 1.$$

Proof. See appendix A. □

Therefore we can use the inequality (2) above as constraint (I) in our formulation for buffered leaky buckets. In section 3.2.3 we illustrate the algebraic expressions for C_i^* and η_i for the special cases when the sources are exponential and erlang on-off sources. Otherwise C_i^* and η_i can be calculated using the results in [Gautam et al., 19].

3.2.2. Unbuffered leaky buckets: constraint (I)
Consider the unbuffered leaky bucket system where there are no input buffers or data buffers and packets enter the network with "violation" tags if there are no tokens available to them. The tagging constraint can be obtained by suitably modifying the waiting time constraint at the input buffer for the buffered leaky bucket as stated in the following theorem:

Theorem 2. The tagging constraint at the unbuffered leaky bucket is satisfied if

$$C_i^* \frac{\gamma_i}{m_i}\left(\frac{r_i-\gamma_i}{m_i}\right)\mathrm{e}^{-\eta_i M_i} \leqslant \zeta_i, \tag{4}$$

where C_i^* is as defined in equation (3), η_i is the solution to $\widetilde{U}_i(-\eta_i(r_i-\gamma_i))\widetilde{D}_i(\eta_i r_i) = 1$, and ζ_i is the expected fraction of traffic entering the network carrying a violation tag.

Proof. See appendix B. □

Therefore we can use inequality (4) as constraint (I) in our formulation for unbuffered leaky buckets.

3.2.3. Special cases

We now describe some closed-form results for special cases of the on-off input processes to the leaky buckets. The expressions will be used in the numerical examples in section 6. When source i (for $i = 1, \ldots, N$) is an exponential on-off source with on-time CDF $U_i(x) = 1 - \mathrm{e}^{-\alpha_i x}$, off-time CDF $D_i(x) = 1 - \mathrm{e}^{-\beta_i x}$ and peak rate r_i, we can show using equation (3) that (see [Gautam et al., 19])

$$C_i^* = \frac{m_i}{\gamma_i} \tag{5}$$

and

$$\eta_i = \frac{r_i(\gamma_i - m_i)\alpha_i}{(r_i - \gamma_i)(r_i - m_i)\gamma_i}, \tag{6}$$

where $m_i = r_i\beta_i/(\alpha_i + \beta_i)$.

Also, consider the case when source i ($i = 1, \ldots, N$) is an on-off source with $Erlang(N_U, \alpha_i)$ on-time distribution, $Erlang(N_D, \beta_i)$ off-time distribution and peak rate r_i. The Erlang distribution convention used here is such that an $Erlang(k, l)$ random variable has mean k/l and variance k/l^2. Note that the mean on and off times τ_U and τ_D respectively are $\tau_U = N_U/\alpha_i$ and $\tau_D = N_D/\beta_i$. Also, the LSTs of the on and off times are $\widetilde{U}_i(s) = (\alpha_i/(\alpha_i + s))^{N_U}$ and $\widetilde{D}_i(s) = (\beta_i/(\beta_i + s))^{N_D}$, respectively. In [Gautam et al., 19] it is shown that (can also be derived from equation (3))

$$C_i^* = \frac{(\alpha_i/(\alpha_i - \eta_i(r_i - \gamma_i)))^{N_U} - 1}{\tau_U + \tau_D} \frac{r_i}{\gamma_i(r_i - \gamma_i)\eta_i\{\alpha_i/(\alpha_i - \eta_i(r_i - \gamma_i))\}} \tag{7}$$

and η_i is the solution to

$$\left(\frac{\beta_i}{\beta_i + \eta_i\gamma_i}\right)^{N_D}\left(\frac{\alpha_i}{\alpha_i - \eta_i(r_i - \gamma_i)}\right)^{N_U} = 1. \tag{8}$$

3.3. Analysis: solving the NLP

Before solving the NLP in section 3.2, we combine the buffered and unbuffered leaky bucket cases into a single case and state a common solution methodology. We restate the nonlinear optimization problem for the buffered and unbuffered leaky bucket, and, discuss a solution procedure. Note that the LHS of constraint (II)

$$m_i < \gamma_i$$

would be automatically satisfied as $\gamma_i = m_i$ will imply that $M_i = \infty$ which is not possible using our algorithm that always searches within the feasible region. Hence we can drop the LHS of constraint (II) with the understanding that while solving, γ_i will never approach m_i as M_i will quickly approach ∞ in that case. Using theorems 1 and 2, we restate the NLP as:

$$\min\left\{\sum_{i=1}^{N}\gamma_i\right\},$$

subject to the constraints,

$$H(\gamma_i)\mathrm{e}^{-\eta_i M_i} \leqslant \zeta_i, \quad \text{for } i = 1, 2, \ldots, N, \tag{9}$$

$$M_1 + M_2 + \cdots + M_N \leqslant B, \tag{10}$$

$$\gamma_i \leqslant r_i, \quad \text{for } i = 1, 2, \ldots, N, \tag{11}$$

where

$$H(\gamma_i) = \begin{cases} C_i^* \dfrac{\gamma_i}{m_i}\mathrm{e}^{-\eta_i\gamma_i d_i^*} & \text{if buffered leaky bucket} \\ C_i^* \dfrac{\gamma_i}{m_i}\left(\dfrac{r_i - \gamma_i}{m_i}\right) & \text{if unbuffered leaky bucket} \end{cases}$$

with (see equation (3))

$$C_i^* = \frac{\widetilde{U}_i(-\eta(r_i - \gamma_i)) - 1}{\eta_i(\tau_U^i + \tau_D^i)} \frac{r_i}{(r_i - \gamma_i)\gamma_i} \Bigg/ \min_{x \geqslant 0}\left\{\frac{\int_x^\infty \mathrm{e}^{\eta_i(r_i-\gamma_i)(y-x)}\,\mathrm{d}U_i(y)}{1 - U_i(x)}\right\}, \tag{12}$$

and η_i the solution to $\widetilde{U}_i(-\eta_i(r_i - \gamma_i))\widetilde{D}_i(\eta_i\gamma_i) = 1$.

We assume that we do not have the trivial case $\gamma_i = r_i$ and $M_i = 0$ for all $i = 1, \ldots, N$ as the only feasible solution to the optimization problem since that would mean that there is no use for the leaky buckets. Also, if there is an optimal solution such that the constraint $M_1 + M_2 + \cdots + M_N \leqslant B$ is not binding, an alternate optimal solution can be found by arbitrarily increasing a particular M_j value (since $\eta_j > 0$ the waiting time or tagging constraint will continue to be satisfied) such that $M_1 + M_2 + \cdots + M_N = B$. Therefore we solve the optimization problem using the constraint $M_1 + M_2 + \cdots + M_N = B$.

Using a standard nonlinear programming technique, namely, the Karush–Kuhn–Tucker conditions (or KKT conditions which can be obtained in any standard text covering nonlinear programming) we derive the following algorithm to solve the optimization problem. Greif and Golub [21] describe the KKT conditions in detail and illustrate solution techniques.

Algorithm.

1. Choose an arbitrary $w > 0$
2. Solve for γ_i (for all $1 \leqslant i \leqslant N$) in
$$w\frac{\partial H(\gamma_i)}{\partial \gamma_i} + \frac{\partial \eta_i}{\partial \gamma_i} w H(\gamma_i) \frac{1}{\eta_i} \log\left[\frac{\zeta_i}{H(\gamma_i)}\right] = -H(\gamma_i)\eta_i$$
3. Set $\gamma_i = \min\{\gamma_i, r_i\}$
4. Repeat steps 2 and 3 by appropriately modifying w until
$$\sum_{i:\gamma_i < r_i} -\frac{1}{\eta_i} \log\left[\frac{\zeta_i}{H(\gamma_i)}\right] = B$$
5. Set the optimal leaky bucket parameters as
$$\gamma_i^* = \gamma_i$$
$$M_i^* = \begin{cases} 0 & \text{if } \gamma_i = r_i \\ -\dfrac{1}{\eta_i} \log\left[\dfrac{\zeta_i}{H(\gamma_i)}\right] & \text{otherwise} \end{cases}$$

Note that under special cases when the sources are exponential or erlang on-off sources, a suitable binary search can be performed to identify a w that solves the equation in step 4 of the above algorithm.

4. Border node: design and admission control

Now we use the optimal leaky bucket parameters for design and admission control schemes at the border node (see the right halves of figures 2 and 4). The design problem is to determine the optimal value of the output channel capacity c (see figure 4). For the connection admission control problem, a decision needs to be made whether or not to accept a request for connection into the network.

For both the design as well as the admission control problems at the border node, we consider the following QoS requirements for each source i ($1 \leqslant i \leqslant N$):

- In the case of the buffered leaky bucket, the fraction of the traffic from the ith source that is discarded by the buffer (of size B) at the border node due to overflows is bounded above by ε_i. Also, the traffic of class i flowing out of the border node should not face a delay larger than κ_i at the border node.
- In the case of the unbuffered leaky bucket, the fraction of non-violation traffic from source i that is discarded by the buffer (of size B) at the border node due to overflows is bounded above by ε_i. Also, the non-violation traffic of class i flowing out of the border node should not face a delay larger than κ_i at the border node.

To study the QoS requirements for design and admission control at the border node, we use the well documented and accepted effective bandwidth methodology. An overview of the effective bandwidth methodology is first presented. Then design and admission control issues will be addressed.

4.1. Effective bandwidths: an overview

The concept of effective bandwidths is used in the analysis of design and admission control problems. We recapitulate some of the recent results on effective bandwidths from [Chang and Thomas, 6; Chang and Zajik, 7; de Veciana et al., 13, 15; Kesidis et al., 26; Kulkarni, 27].

Consider a single buffer fluid model driven by a random environmental process $\{Z(t), t \geqslant 0\}$ (see figure 5). When the environment is in state $Z(t)$, the fluid enters the buffer at rate $r(Z(t))$. Let $B(t)$ be the amount of fluid in the buffer at time t. The buffer has infinite capacity and is serviced by a channel of constant output rate c. Let $A(t)$ be the total amount of fluid input from the source to the buffer in time t. Thus

$$A(t) = \int_0^t r\big(Z(u)\big)\,\mathrm{d}u. \tag{13}$$

We define effective bandwidth of the source, $eb(v)$, for $v > 0$, as

$$eb(v) = \lim_{t\to\infty} \frac{1}{vt} \log E\big\{\exp\big(vA(t)\big)\big\}. \tag{14}$$

The stochastic behavior of the traffic source is captured by the effective bandwidth in an asymptotic sense. It is known that $eb(v)$ is an increasing function of v. Also as $v \to 0$, $eb(v) \to E[r(Z(\infty)]$ (the mean traffic generation rate) and as $v \to \infty$, $eb(v) \to \sup_t\{r(Z(t)\}$ (the peak traffic generation rate). If K independent sources with effective bandwidths $eb_1(v)$, $eb_2(v), \ldots, eb_K(v)$ are multiplexed, the resultant traffic's effective bandwidth is $\sum_{k=1}^{K} eb_k(v)$. From large deviations theory, for large x,

$$\lim_{t\to\infty} P\big(B(t) > x\big) \approx \mathrm{e}^{-\eta x}, \tag{15}$$

where η is the solution to

$$eb(\eta) = c.$$

It is not easy to calculate the effective bandwidth using equation (14). However, when the environmental processes can be modeled as Continuous time Markov Chains

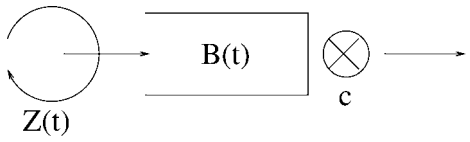

Figure 5. Single buffer fluid model.

(CTMCs), Semi-Markov Processes (including general on-off sources), Markov Regenerative Processes (MRGP) or regenerative processes, etc., we can compute their effective bandwidths using the results shown in [Elwalid and Mitra, 18; Kesidis et al., 26; Kulkarni, 27], etc. Some of them (useful for this paper) are described below using the notation $e(M)$ for the largest real eigenvalue of a square matrix M:

- Suppose $\{Z(t),\ t \geqslant 0\}$ is a CTMC with generator matrix Q and diagonal matrix R such that $R_{ii} = r(i)$. Then the effective bandwidth of this CTMC source is $eb(v) = (1/v)e(Q + vR)$.
- Suppose $\{Z(t),\ t \geqslant 0\}$ is an alternating on-off process such that the LSTs of the on and off times are $\widetilde{U}(\cdot)$ and $\widetilde{D}(\cdot)$, respectively. Also, traffic is generated at rates r and 0 when the source is in the on and off state, respectively. Then define $\Lambda(u, v) = \widetilde{U}(uv - rv)\widetilde{D}(uv)$ so that $u^*(v) = \inf\{u > 0:\ e(\Lambda(u, v)) < \infty\}$. Then the effective bandwidth of this on-off source is a unique solution to $\Lambda(eb(v), v) = 1$ whenever a solution exists such that $u^* < eb(v) < r$. When a solution does not exist, $eb(v) = u^*$.

The input to the border node with a single buffer is the multiplexed traffic from N leaky buckets. In order to do the effective bandwidth analysis we need to calculate the effective bandwidth of the output from the leaky bucket. The output traffic from a leaky bucket cannot be modeled as CTMCs, Semi-Markov Processes, or MRGPs for which effective bandwidths can be computed. Therefore a detailed analysis of calculating the effective bandwidth of the output from buffered and unbuffered leaky buckets are described in sections 5.1 and 5.2, respectively.

4.2. Design problem: determining the channel capacity

Suppose we want to design the channel capacity c to handle a given set of sources policed by buffered or unbuffered leaky buckets. Let $\varepsilon = \min_i\ \varepsilon_i$ (this means all sources will face a loss-probability of at most ε). If $eb_i(\delta)$ is the effective bandwidth of the traffic entering the border node from leaky bucket i (see figure 4), then it is known that (see [Chang and Thomas, 6; Kesidis et al., 26]) the QoS loss criterion is satisfied if

$$\sum_{i=1}^{N} eb_i(\delta) < c$$

where $\delta = -\log(\varepsilon)/B$ and B is the buffer size. This result is valid in the asymptotic region

$$B \to \infty, \qquad \varepsilon \to 0 \quad \text{so that} \quad -\frac{\log(\varepsilon)}{B} \to \delta \in (0, \infty),$$

otherwise the results are approximate but usually conservative. Also, the maximum delay constraint can be satisfied if $c > B/\kappa_i$ for all i. Therefore the main research question (that will be addressed in section 5) is how to obtain the effective bandwidth of the output from a leaky bucket.

In summary, to address the design problem, we first solve the optimization problem using the algorithm in section 3.3 for these given sources and a given buffer size B, and obtain the optimal values of γ_i and M_i for $1 \leqslant i \leqslant N$. With the optimal parameters γ_i and M_i, we use the results of sections 5.1 and 5.2 to obtain the effective bandwidth $eb_i(v)$ of the output from each leaky bucket. Then the minimum capacity needed to satisfy the loss probability constraint is hence given by

$$c^* = \max\left\{\frac{B}{\min_i(\kappa_i)}, \sum_{i=1}^{N} eb_i(\delta)\right\}.$$

4.3. Connection admission control problem

Suppose the capacity c and the buffer size B are given. Say, we have N sources requesting service. We use the optimal parameters γ_i and M_i that have been computed by the algorithm in section 3.3. Then, from the analysis in sections 5.1 and 5.2 we compute $eb_i(\delta)$, where δ is explained in section 4.2. Let

$$c^* = \sum_{i=1}^{N} eb_i(\delta).$$

If $c^* < c$ and $c > B/\min_i(\kappa_i)$ then we can admit all the sources, otherwise some will have to be denied access. Furthermore, if we have already admitted N sources, and a new $(N+1)$st source arrives, we use the new optimal parameters M_i and γ_i $(1 \leqslant i \leqslant N+1)$, and compute (using the output effective bandwidth analysis in section 5)

$$c^* = \sum_{i=1}^{N+1} eb_i(\delta).$$

If $c^* < c$ and $c > B/\min_i(\kappa_i)$, we can admit the new source and reset the leaky bucket parameters of all the existing sources.

5. Output effective bandwidth analysis

The output from a leaky bucket acts as an input to a downstream network node such as the border node. Hence, in this section we characterize the output from the leaky bucket and calculate the effective bandwidth of the output traffic. The analysis for the buffered and unbuffered cases are a little different and need to be treated separately.

5.1. Buffered leaky bucket: output effective bandwidth analysis

Refer to figure 3 for the notation used here. We consider a more general stochastic process $Z(t)$ to derive the output effective bandwidth. The on-off source used in the earlier sections is a special case of this $Z(t)$ process. Let $X(t)$ be the amount of traffic in the data buffer at time t. Let $Y(t)$ be the amount of tokens in the token pool at time t

($Y(t) \leqslant M$). Note that fluid starts accumulating in the data buffer ($X(t) > 0$) only when the token pool is empty ($Y(t) = 0$). As long as tokens are available ($Y(t) > 0$), fluid does not wait at the data buffer ($X(t) = 0$). Therefore $X(t)Y(t) = 0$, for all t. Clearly, when the token pool is not empty ($Y(t) > 0$), the output from the leaky bucket is at rate $r(Z(t))$ at time t and when the token pool is empty, the output from the leaky bucket is at rate γ. Hence the output rate from the leaky bucket at time t, $R(t)$, is given by

$$R(t) = \begin{cases} \gamma & \text{if } Y(t) = 0, \\ r\big(Z(t)\big) & \text{if } Y(t) > 0. \end{cases} \tag{16}$$

Define a process $\{W(t), t \geqslant 0\}$ (see [Anantharam and Konstantopoulos, 1]) as

$$W(t) = X(t) + M - Y(t). \tag{17}$$

We will see later that this process is critical in obtaining the effective bandwidths as well as in the appendices A and B to derive the expressions for constraint (I) in the optimization problem. In fact, we will observe that if we replace the leaky bucket with an infinite-size buffer with output capacity γ, the buffer content process of that buffer will be identical to the $W(t)$ process. To determine the effective bandwidth of the output from the leaky bucket, we first characterize the $\{W(t),\ t \geqslant 0\}$ process. The dynamics of the $X(t)$ and the $Y(t)$ processes are given by

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \begin{cases} r\big(Z(t)\big) - \gamma & \text{if } X(t) > 0, \\ 0 & \text{if } X(t) = 0, \end{cases} \tag{18}$$

$$\frac{\mathrm{d}Y(t)}{\mathrm{d}t} = \begin{cases} \gamma - r\big(Z(t)\big) & \text{if } 0 < Y(t) < M, \\ -\big\{r\big(Z(t)\big) - \gamma\big\}^+ & \text{if } Y(t) = M, \\ 0 & \text{if } Y(t) = 0, \end{cases} \tag{19}$$

where for any quantity a, $\{a\}^+ = \max(a, 0)$.

From equation (17) we get,

$$\begin{aligned} W(t) > M &\quad\Rightarrow\quad X(t) > 0 \text{ and } Y(t) = 0, \\ 0 < W(t) \leqslant M &\quad\Rightarrow\quad X(t) = 0 \text{ and } 0 < Y(t) < M, \\ W(t) = 0 &\quad\Rightarrow\quad X(t) = 0 \text{ and } Y(t) = M. \end{aligned}$$

In fact, $R(t)$ can be written as

$$R(t) = \begin{cases} \gamma & \text{if } W(t) \geqslant M, \\ r\big(Z(t)\big) & \text{if } W(t) < M. \end{cases}$$

Also,

$$\begin{aligned} \frac{\mathrm{d}W(t)}{\mathrm{d}t} = \frac{\mathrm{d}X(t)}{\mathrm{d}t} - \frac{\mathrm{d}Y(t)}{\mathrm{d}t} &= \begin{cases} r\big(Z(t)\big) - \gamma & \text{if } X(t) > 0 \text{ and } Y(t) = 0, \\ r\big(Z(t)\big) - \gamma & \text{if } X(t) = 0 \text{ and } 0 < Y(t) < M, \\ \big\{r\big(Z(t)\big) - \gamma\big\}^+ & \text{if } X(t) = 0 \text{ and } Y(t) = M \end{cases} \\ &= \begin{cases} r\big(Z(t)\big) - \gamma & \text{if } W(t) > 0, \\ \big\{r\big(Z(t)\big) - \gamma\big\}^+ & \text{if } W(t) = 0. \end{cases} \end{aligned} \tag{20}$$

Thus the dynamics of the $W(t)$ process are identical to those of the buffer-content process of an infinite-sized buffer with output capacity γ and input rate $r(Z(t))$ at time t. Therefore to obtain the properties of the $W(t)$ process, for example, its probability distribution, all one needs to do is look up the vast literature on the buffer-content process (see [Kulkarni, 28] and the references therein) of an infinite sized buffer with output capacity γ and input rate $r(Z(t))$. We exploit the structure of the $\{W(t),\ t \geqslant 0\}$ process in the analysis that follow.

Sample paths of $Z(t)$, $X(t)$, $Y(t)$, and $W(t)$ are shown in figure 6. Define the first passage time V (see figure 6) as

$$V = \inf\{t > 0:\ W(t) = 0 \mid W(0) = 0, W(0+) > 0\}, \tag{21}$$

where the term "0+" denotes the time instant immediately after $t = 0$. Note that V is the

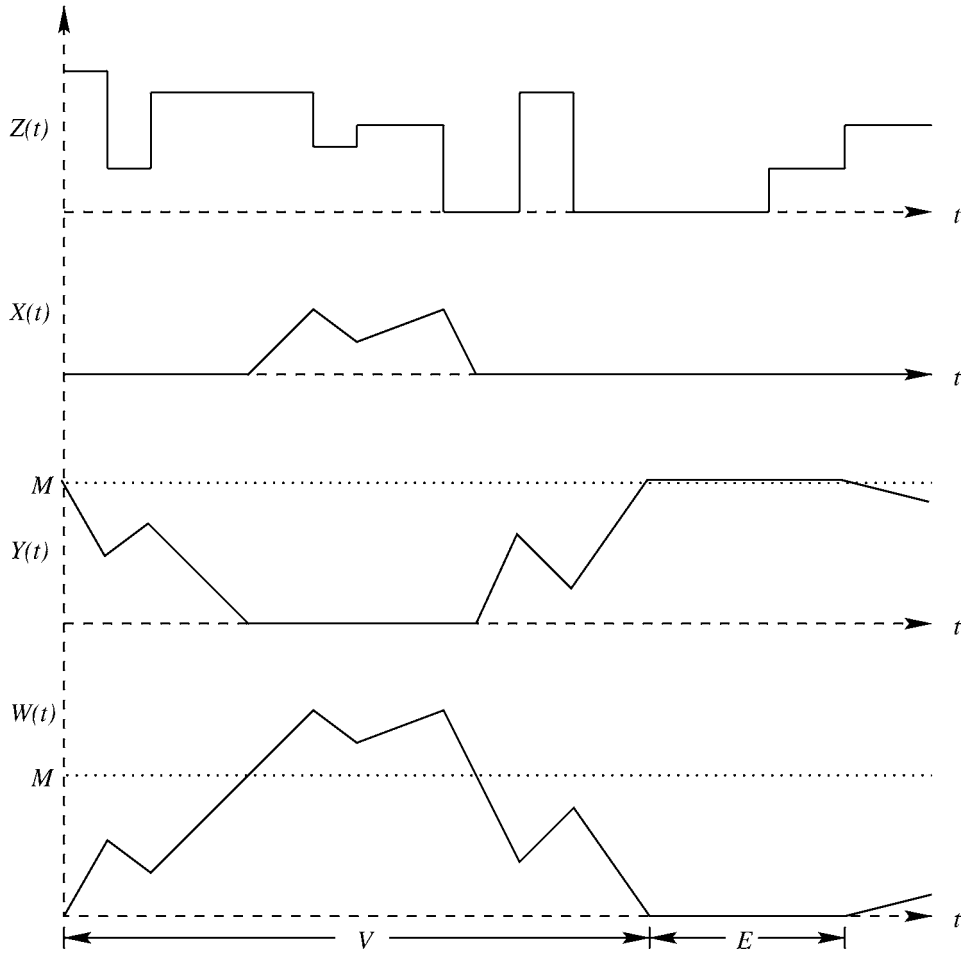

Figure 6. $Z(t)$, $X(t)$, $Y(t)$, and $W(t)$ for buffered leaky buckets.

time duration between when $W(\cdot)$ becomes nonzero until it for the first time becomes zero. It is also the time duration when the token buffer is not full (i.e. $Y(\cdot) < M$).

Let $\Theta(V)$ be the total amount of traffic output from the leaky bucket in time V. During the time interval $(0, V)$, $W(t) > 0$ and token pool is non-full. Hence the tokens enter the token pool at rate γ during the time interval $(0, V)$. Since the token pool is full at times 0 and V, the total number of tokens removed from the pool over $(0, V)$ must be the same as the total number of tokens that entered the pool over $(0, V)$ which is γV since γ is the rate of token generation and V is the time. Hence we get

$$\Theta(V) = \gamma V. \tag{22}$$

Define $A(t)$ as the total fluid arrival into the leaky bucket from the source in time t. Also, let $O(t)$ be the total fluid output from the leaky bucket in time t. Using the result in equation (22), the following theorem states the effective bandwidth of the output of the leaky bucket when $\{Z(t),\ t \geqslant 0\}$ is a semi-Markov process (SMP). (Note that de Veciana [14] derives the effective bandwidth of the output of the leaky bucket for a discrete traffic model. The following theorem is the equivalent result for a fluid traffic model. Also, the proof uses a different approach as that of [de Veciana, 14].)

Theorem 3. Let $\{Z(t),\ t \geqslant 0\}$ be an SMP on a finite state space $\mathcal{S}$. Let $O(t)$ be the total output from the leaky bucket over $[0, t]$. The effective bandwidth of the output of the leaky bucket

$$eb_O(v) = \lim_{t\to\infty} \frac{1}{vt} \log E\big\{\exp\big(vO(t)\big)\big\}$$

is given in terms of the effective bandwidth of the input, $eb_A(v)$, as

$$eb_O(v) = \begin{cases} eb_A(v) & \text{if } 0 \leqslant v \leqslant v^*, \\ \dfrac{v^*}{v} eb_A(v^*) - \gamma \dfrac{v^*}{v} + \gamma & \text{if } v > v^*, \end{cases} \tag{23}$$

where v^* is obtained by solving

$$\frac{\mathrm{d}}{\mathrm{d}v^*}\big[v^* eb_A(v^*)\big] = 0$$

and

$$eb_A(v) = \lim_{t\to\infty} \frac{1}{vt} \log E\left\{\exp\left(v \int_0^t r\big(Z(t)\big)\,\mathrm{d}t\right)\right\}.$$

Proof. Define the set $\mathcal{G}$ (comprising of all states of the SMP with traffic generation rates larger than γ) as follows

$$\mathcal{G} = \big\{i\colon\ r(i) > \gamma,\ i \in \mathcal{S}\big\}.$$

The output rate from the leaky bucket, $R(t)$, at time t in equation (16) can be rewritten as

$$R(t) = \begin{cases} \gamma & \text{if } W(t) \geqslant M, \\ r(i) & \text{if } 0 < W(t) < M,\ i \in \mathcal{S} \text{ and } Z(t) = i. \end{cases} \tag{24}$$

We showed earlier that the dynamics of the $W(t)$ process are identical to those of the buffer-content process of an infinite-sized buffer with output capacity γ and input rate $r(Z(t))$ at time t. However on a sample-path basis the output from the leaky bucket is not identical to that of a single buffer with output capacity γ and input rate $r(Z(t))$ at time t. But we now show by observing the output process at appropriate Markov regenerative epochs that the asymptotic properties such as the effective bandwidth would be the same. We first consider the actual $W(t)$ process and later the fictitious $W(t)$ process which is the fluid contents of a buffer with output capacity γ and input rate $r(Z(t))$ at time t.

- *The actual $W(t)$ process in the leaky bucket context*: Consider the bivariate stochastic process $\{(W(t), Z(t)),\ t \geqslant 0\}$ that modulates the output from a leaky bucket according to equation (24). Now, suppose $W(0) = 0$ and $Z(0) = i$, for $i \in \mathcal{G}$. Then, $\{(W(t), Z(t)),\ t \geqslant 0\}$ is a Markov regenerative process (MRGP) that Markov-regenerates whenever it reaches the state $(0, j)$, $j \in \mathcal{G}$. The length of the Markov-regenerative cycle is seen to be $S_1 = V + E$, where V is as in equation (21) and E is the duration when $W(t) = 0$ and $Z(t) = i$, $i \in \mathcal{S} - \mathcal{G}$ (see figure 6). Now, from equation (22), the total output during V is γV, while the total output during E is, say, $F(E)$. Hence the total output during the first Markov-regenerative cycle is

$$F_1 = \gamma V + F(E).$$

- *The fictitious $W(t)$ process which is the total fluid in a buffer with same input as the leaky bucket and output capacity γ*: Consider a source modulated by the $\{Z(t),\ t \geqslant 0\}$ process inputs fluid at rate $r(Z(t))$ at time t into the buffer. For this model, we noticed that the buffer-content process is identical to $\{W(t),\ t \geqslant 0\}$. Now, suppose $W(0) = 0$ and $Z(0) = i$, for $i \in \mathcal{G}$. Then, $\{(W(t), Z(t)),\ t \geqslant 0\}$ for this model is also a Markov regenerative process that Markov-regenerates whenever it reaches the state $(0, j)$, $j \in \mathcal{G}$. The length of the Markov-regenerative cycle is seen to be $S_1 = V + E$, where V is as in equation (21) and E is the duration when $W(t) = 0$ and $Z(t) = i$, $i \in \mathcal{S} - \mathcal{G}$. Whenever the buffer is non-empty, the output rate is γ, hence the total output during V is γV. Also, the total output during E is $F(E)$. Therefore the total output during the first Markov-regenerative cycle is

$$F_1 = \gamma V + F(E).$$

Since the same MRGP models the two cases, the output from the "fictitious" single buffer model and the output from the actual leaky bucket model, it follows that in the two cases the effective bandwidths are identical. The results of the effective bandwidths of the output from a buffer are derived in [Chang and Zajik, 7; Kulkarni and Gautam, 29]. Hence we have the effective bandwidth of the output from the leaky bucket (which

is identical to that of the output from the buffer with capacity γ) in terms of the effective bandwidth of the input as given in the theorem. □

Therefore, given the effective bandwidth of the input traffic to the leaky bucket, it is easy to obtain the effective bandwidth of the output traffic from the leaky bucket by simply replacing the leaky bucket by a single infinite capacity buffer with capacity γ and measuring the output effective bandwidth of this infinite capacity buffer in terms of its input. As mentioned in section 4.1, when the environmental processes of the input traffic can be modeled as CTMCs, Semi-Markov Processes, MRGPs, or regenerative processes, etc., we can compute their effective bandwidths using the results shown in [Elwalid and Mitra, 18; Kesidis et al., 26; Kulkarni, 27], etc.

Note that the effective bandwidth of the leaky bucket is identical to that of the output from a buffer (with output rate γ), and is independent of M! This means that M does not play any role as far as reducing the effective bandwidth, but acts strictly as a policing device that prevents arbitrarily large peak-rate bursts from entering the network.

We would like to comment upon a curious discontinuous behavior at this point. Although the effective bandwidth of the output is related to that of the input as stated in theorem 3 for all $M < \infty$, we have $eb_O(v) = eb_A(v)$ if $M = \infty$. This is because the $\{(W(t), Z(t)),\ t \geqslant 0\}$ process is transient if $M = \infty$, thus making the above analysis inapplicable. However, in that case (i.e. $M = \infty$), the leaky bucket is transparent and $O(t) = A(t)$ for all $t \geqslant 0$ assuming that the token buffer is full at time 0, thus making the two effective bandwidths identical. In practice using $M = \infty$ is never a good idea, and hence this discontinuity will not bother us in our analysis.

5.2. Unbuffered leaky bucket: output effective bandwidth analysis

For the unbuffered leaky bucket, we only consider the case when the environmental process governing the fluid input from a source, $\{Z(t),\ t \geqslant 0\}$, is a 2-state on-off process ($Z(t) = 0$ or 1, which implies whether the source is off or on respectively at time t). Therefore we say that the fluid input is from a general on-off source with on time distribution $U(\cdot)$ (with mean τ_U) and off time distribution $D(\cdot)$ (with mean τ_D). When the source is on it generates traffic at rate r and at rate 0 when off. Therefore $r(Z(t)) = rZ(t)$.

In this unbuffered leaky bucket case, a packet that arrives at the leaky bucket is sent into the network with a "violation" tag if no tokens are available at the time of its arrival. We will concentrate on the untagged packets as the tagged ones would be dropped in the event of a congestion. Let $Y(t)$ and $W(t)$ be as defined in section 5.1. Note that $X(t) = 0$ for all t in this unbuffered leaky bucket case. A sample path of $W(t)$ is shown in figure 7. Since there is no data buffer, $W(t) = M - Y(t)$ and $W(t)$ ranges from 0 to M. Note that $W(t)$ process is identical to a buffer content process of a fluid queue with on-off input, constant output with rate γ, and, a finite buffer of size M.

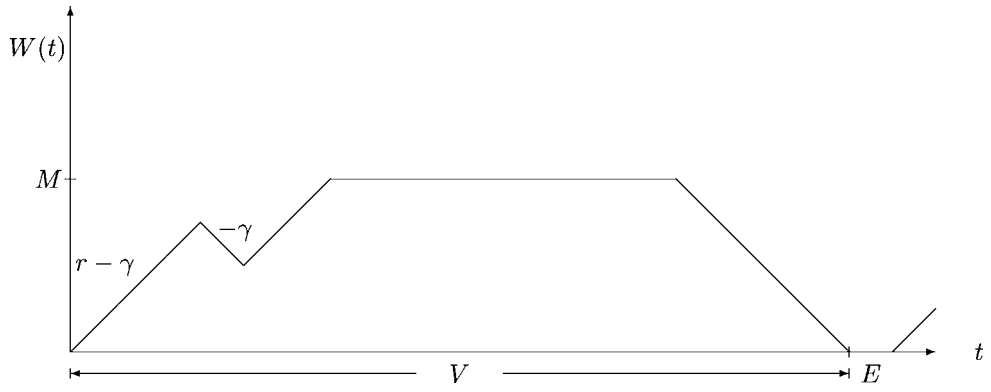

Figure 7. $W(t)$ process for unbuffered leaky bucket.

To obtain the effective bandwidth of the output process, we follow the procedure used in section 5.1. The output rate from the leaky bucket is $R(t)$ at time t and is given by

$$R(t) = \begin{cases} \gamma & \text{if } W(t) = M, \\ r & \text{if } W(t) < M \text{ and } Z(t) = 1, \\ 0 & \text{if } W(t) < M \text{ and } Z(t) = 0. \end{cases} \tag{25}$$

Let V be as in equation (21). Then equation (22) remains valid in the unbuffered case. Hence the effective bandwidth of the output process from the unbuffered leaky bucket is equivalent to that of of the output process from a single finite buffer (of size M) with general on-off source input and output capacity γ. However, the effective bandwidth of the output cannot be easily written in terms of that of the input due to the fluid loss (as a result of untagged traffic) at the input buffer.

However, when the sources are exponential on-off sources (i.e. on and off times are exponentially distributed) with mean on-time $1/\alpha$ and mean off-time $1/\beta$, the effective bandwidth of the output from the buffer can be calculated using the following LSTs in [Narayanan and Kulkarni, 30] as follows:

$$\begin{aligned} \widetilde{V}(w) &= E\{\mathrm{e}^{-wV}\} \\ &= \Big[(\beta + w + \gamma s_1)\mathrm{e}^{(s_0-s_1)M}\big(w(w+\beta+\gamma s_0+\alpha)+\alpha\gamma s_0\big) \\ &\quad - (\beta + w + \gamma s_0)\big(w(w+\beta+\gamma s_1+\alpha)+\gamma\alpha s_1\big)\Big] \\ &\quad \times \Big[\beta\big(\mathrm{e}^{(s_0-s_1)M}\big(w^2+w\beta+w\gamma s_0+\alpha w+\alpha\gamma s_0\big) \\ &\quad + \big(-w^2-w\beta-w\gamma s_1-\alpha w-\gamma\alpha s_1\big)\big)\Big]^{-1}, \end{aligned}$$

where

$$s_0 = \frac{-\hat{b} - \sqrt{\hat{b}^2 + 4w(w+\alpha+\beta)\gamma(r-\gamma)}}{2\gamma(r-\gamma)},$$

$$s_1 = \frac{-\hat{b} + \sqrt{\hat{b}^2 + 4w(w+\alpha+\beta)\gamma(r-\gamma)}}{2\gamma(r-\gamma)},$$

and

$$\hat{b} = (r-2\gamma)w + (r-\gamma)\beta - \gamma\alpha.$$

Also,

$$\widetilde{E}(w) = E\{e^{-wE}\} = \frac{\beta}{\beta+w},$$

where E is the duration when $W(t) = 0$ and $Z(t) = 0$, (see figure 7) which corresponds to a full token buffer and the new tokens overflowing with the traffic source being off. Using the effective bandwidth for on-off source analysis in section 4.1, one can derive u^* as

$$u^* = \frac{\gamma\beta - \alpha\gamma - r\beta + 2\sqrt{\alpha\beta\gamma(r-\gamma)}}{rv} + \gamma.$$

The effective bandwidth of the output of the unbuffered leaky bucket, $eb_O(v)$, is a unique solution to $\widetilde{V}(veb_O(v) - \gamma v)\widetilde{E}(veb_O(v)) = 1$, whenever a solution exists such that $u^* < eb_O(v) < r$. When a solution does not exist, $eb_O(v)$ is given by $eb_O(v) = u^*$.

When $M = 0$, it can be shown that the effective bandwidth of the output, $eb_O(v)$, reduces to the effective bandwidth of an on-off source with exp(α) on-times, exp(β) off-times, and, on-time traffic generation rate γ. This is as expected. However, we get the same discontinuous behavior as $M \to \infty$ as in the buffered case, and it arises for the same reason explained in the buffered case.

Closed-form algebraic expressions for $eb_O(v)$ are intractable even when the sources are exponential on-off sources. Therefore for general on-off sources, we develop an approximation method given below. When the traffic carrying the "violation" tag is an extremely small fraction of the output traffic from the leaky bucket (a fraction of the order of 10^{-4} is typical), then as an approximation, the effective bandwidth of the untagged packets, $eb_O(v)$, is considered to be equal to the effective bandwidth of the input ($eb_A(v)$) to the leaky bucket. Note that since $O(t)$ is stochastically less than $A(t)$ for all t, $eb_O(v) \leqslant eb_A(v)$. Thereby this approximation is indeed a conservative one and hence can be appropriately used in our optimization models.

6. Results

In this section we present some numerical examples to illustrate the optimal leaky bucket parameters at the source nodes, the effective bandwidth of traffic between the source node and the border node, and, design and admission control problems at the border node for various cases of buffered and unbuffered leaky buckets. The title of each example corresponds to the type of problem in the border node, the type of leaky bucket, and, the type of input traffic source.

6.1. *Capacity design, buffered leaky buckets, Erlang on-off sources case*

Consider buffered leaky buckets where the sources of traffic belong to four different classes such that there are three class-1 sources, two class-2 sources, four class-3 sources and three class-4 sources. All the sources are Erlang on-off sources with parameters N_U, α, N_D, β, and r, as defined in section 3.2.3 and with QoS parameters ζ and d^* at the source node as defined in section 3.1, and κ and ε at the border node as defined in section 4. The numerical values of the parameters are summarized in table 1. The buffer size of the border node is $B = 10$. Solving the optimization problem using the algorithm in section 3.3 subject to satisfying the QoS constraints (besides other constraints) at the source, we obtain the optimal leaky bucket parameters γ^* and M^* for the buffered leaky bucket for each of the classes of traffic (see table 1).

Then using the output effective bandwidth analysis (detailed in section 5.1), we obtain $eb_i(\delta)$ (for $i = 1, 2, 3, 4$), where $\delta = -\log\{\min_i \varepsilon_i\}/B$. Therefore from the analysis in section 4.2, the optimal channel capacity at the border node can be designed as $c = 3eb_1(\delta) + 2eb_2(\delta) + 4eb_3(\delta) + 3eb_4(\delta)$. For the numerical values in table 1, the optimal channel capacity is 10.141.

6.2. *Design and control table, buffered leaky buckets, Erlang on-off sources case*

Consider the case when there are two types of sources, say, real-time sources and non-real-time sources that generate traffic policed by buffered leaky buckets. For example, there are k_1 iid type 1 Erlang on-off sources with $N_U^1 = 5$, $\alpha_1 = 2$, $N_D^1 = 2$, $\beta_1 = 5$, $r_1 = 2.0$, $\zeta_1 = 10^{-5}$, $d_1^* = 0$, $\varepsilon_1 = 0.0001$, $\kappa_1 = 2$ and k_2 iid type 2 Erlang on-off sources with $N_U^2 = 3$, $\alpha_2 = 3$, $N_D^2 = 2$, $\beta_2 = 5$, $r_2 = 1.2$, $\zeta_2 = 0.003$, $d_2^* = 0$, $\varepsilon_2 = 10^{-7}$, $\kappa_2 = 20$. The border node buffer has capacity $B = 10$. For a given (k_1, k_2), we can obtain the optimal γ_1^*, γ_2^*, M_1^* and M_2^* by solving the nonlinear optimization problem. Using the effective bandwidth of the output of a type i source $eb_i(\delta)$, the loss probability constraint is satisfied if

$$k_1 eb_1(\delta) + k_2 eb_2(\delta) < c,$$

and the delay constraint is satisfied if $B/c < \min(\kappa_1, \kappa_2)$. If both are satisfied we say that the pair (k_1, k_2) is feasible.

Table 2 gives the values of γ_1^*, γ_2^*, M_1^*, M_2^*, $eb_1(\delta)$, and $eb_2(\delta)$ for the pairs $\{(k_1, k_2)\colon\ 1 \leqslant k_1 \leqslant 5,\ 1 \leqslant k_2 \leqslant 5\}$. The legend at the bottom of the table describes the

Table 1
Source and QoS parameters and optimal leaky bucket parameters.

Class	N_U	α	N_D	β	r	ζ	d^*	ε	κ	γ^*	M^*
1	4	4	3	5	1.2	0.001	0.1	0.0000001	2	0.990656	0.443116
2	3	2	6	3	1.4	0.0001	0.0	0.00001	1	1.400000	0.000000
3	2	4	4	5	2.1	0.00035	0.2	0.0000001	3	1.324647	1.641966
4	5	6	2	2	1.3	0.000007	0.5	0.000001	10	0.975620	0.712276

Table 2
Design and admission control table.

k_1 / k_2	1		2		3		4		5	
1	1.808	0.903	1.8386	0.9240	1.8634	0.9343	1.8800	0.9446	1.8938	0.9548
	5.772	4.236	3.6002	2.7791	2.5251	2.3484	1.9965	2.0170	1.6334	1.7558
	1.785	0.893	1.7928	0.9026	1.7930	0.9054	1.7930	0.9070	1.7930	0.9074
2	1.833	0.917	1.8607	0.9343	1.8800	0.9446	1.8938	0.9548	1.9076	0.9583
	3.905	3.153	2.6257	2.3484	1.9965	2.0171	1.6334	1.7557	1.3233	1.6801
	1.792	0.900	1.7930	0.9054	1.7930	0.9070	1.7930	0.9074	1.7930	0.9074
3	1.858	0.934	1.8800	0.9446	1.8965	0.9549	1.9076	0.9617	1.9186	0.9686
	2.730	2.348	1.9965	2.0171	1.5675	1.7558	1.3233	1.6095	1.1059	1.4813
	1.793	0.905	1.7930	0.9070	1.7930	0.9074	1.7930	0.9074	1.7930	0.9074
4	1.880	0.944	1.8993	0.9548	1.9131	0.9617	1.9241	0.9686	1.9324	0.9720
	1.996	2.017	1.5037	1.7558	1.2115	1.6095	1.0059	1.4813	0.8654	1.4225
	1.793	0.907	1.7930	0.9074	1.7930	0.9074	1.7930	0.9074	1.7930	0.9074
5	1.902	0.955	1.9159	0.9651	1.9269	0.9720	1.9379	0.9754	1.9434	0.9789
	1.442	1.756	1.1580	1.5435	0.9579	1.4225	0.7775	1.3671	0.6938	1.3149
	1.793	0.907	1.7930	0.9074	1.7930	0.9074	1.7930	0.9074	1.7930	0.9074

Legend:

γ_1^*	γ_2^*
M_1^*	M_2^*
$eb_1(\delta)$	$eb_2(\delta)$

values corresponding to each (3×2) cell. Notice that γ_i^* values change very little across the table as compared to the M_i^* values. One reason is that there is a constraint on the sum of M_i^* values but the γ_i^* values are by themselves independent of other γ_j^* values. Also observe that the effective bandwidth values are almost identical across all the cells. The explanation for that is the output effective bandwidth is usually equal to the input effective bandwidth (unless $v > v^*$, see section 5.1) which remains the same in all the cases.

This table 2 can be used for both the design problem as well as the admission control problem as follows. For example, suppose we want to be able to handle 3 sources of type 1 and 4 of type 2 at the border node. Then for the pair $(3, 4)$ we see that the sum of the output effective bandwidths is $3 \times 1.7930 + 4 \times 0.9074 = 9.0086$. Also $B/\min(\kappa_1, \kappa_2) = 5$. Hence we must choose $c > 9.0086$ in order to handle this traffic. On the other hand, suppose $c = 12.2$ is given. Then the pair $(4, 5)$ is feasible if we use the optimal parameters from table 2, however the pair $(5, 5)$ is infeasible. Thus the call admission can be done using such a table. For implementing these call admission policies, a look-ip table can be maintained with values similar to table 2 and referred to whenever a decision needs to be made.

6.3. Admissible region, buffered leaky buckets, Erlang on-off sources case

Consider k_1 identical sources belonging to class 1 and k_2 identical sources belonging to class 2 traffic. All feasible pairs (k_1, k_2) are shown in the region R (including the boundary) of figure 8 for the case of buffered leaky buckets with input from two classes (real-time and non-real-time) of Erlang on-off sources with $N_U^1 = 7$, $\alpha_1 = 1$, $N_D^1 = 6$, $\beta_1 = 0.4$, $r_1 = 1.2$, $\zeta_1 = 10^{-8}$, $d_1^* = 0$, $\varepsilon_1 = 0.0001$, $\kappa_1 = 40$ $N_U^2 = 8$, $\alpha_2 = 2.4$, $N_D^2 = 5$, $\beta_2 = 0.4$, $r_2 = 2.0$, $\zeta_2 = 10^{-5}$, $d_2^* = 0$, $\varepsilon_2 = 10^{-7}$, $\kappa_2 = 100$. The border node buffer has capacity $B = 200$ and the border node output channel capacity is 8.1.

6.4. Capacity design, unbuffered leaky buckets, exponential on-off sources case

Consider unbuffered leaky buckets (notice that section 6.1 considers buffered) where the sources of traffic belong to three different classes such that there are three class-1 sources, two class-2 sources, and, four class-3 sources. All the sources are exponential on-off sources with parameters α, β and r as defined in section 3.2.3 with QoS requirements ζ at the source as defined in section 3.1 and κ and ε at the border node as defined in section 4. The numerical values of the parameters are summarized in table 3. The buffer size of the border node $B = 17$.

Solving the optimization problem using the algorithm in section 3.3, we obtain the optimal leaky bucket parameters γ^* and M^* for the unbuffered leaky bucket for each of the classes of traffic (see table 3). Then using the output effective bandwidth

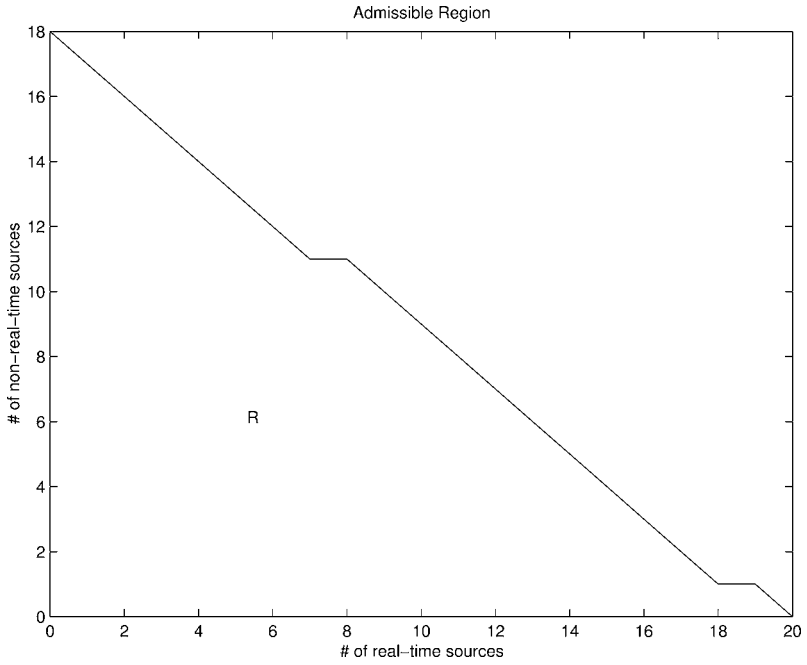

Figure 8. The 2-class buffered leaky bucket node admissible region.

Table 3
Source and QoS parameters and optimal leaky bucket parameters.

Class	α	β	r	ζ	ε	κ	γ^*	M^*
1	1.0	0.4	1.22	0.03	0.000001	40	0.8148	1.8801
2	0.5	0.2	1.83	0.08	0.0001	10	1.3921	2.3994
3	1.5	0.5	1.11	0.001	0.00000001	4	0.8061	1.6494

analysis illustrated in section 5.2, we obtain $eb_i(\delta)$ (for $i = 1, 2, 3$). Using the analysis in section 5.2, the effective bandwidth of the untagged traffic from the unbuffered leaky buckets for the three classes of traffic can be calculated as $eb_1(\delta) = 0.5987$, $eb_2(\delta) = 1.2251$ and $eb_3(\delta) = 0.4098$ for the numerical values in table 3. Therefore from the analysis in section 4.2, the optimal channel capacity can be designed as $c = 3eb_1(\delta) + 2eb_2(\delta) + 4eb_3(\delta)$. For the numerical values in table 3, the optimal channel capacity at the border node is 5.8855.

Now instead of using the initial analysis in section 5.2 to calculate the effective bandwidth of the untagged traffic from the unbuffered leaky buckets, if a conservative approach is taken and the effective bandwidth of the untagged traffic is approximated as that of the traffic input from the source into the unbuffered leaky buckets, then the effective bandwidths are $eb_1(\delta) = 0.6005$, $eb_2(\delta) = 1.3729$ and $eb_3(\delta) = 0.4141$. This approximation is explained towards the end of section 5.2. The resulting optimal channel capacity at the border node is 6.2037. Therefore in most cases it is not necessary to go through the tedious calculations mentioned in section 5.2, instead a conservative approach can be taken to obtain the effective bandwidths and hence the channel capacity of the border node.

6.5. Admission control, unbuffered leaky buckets, Erlang on-off sources case

Consider k_1 identical sources belonging to class 1 and k_2 identical sources belonging to class 2 traffic. All feasible pairs (k_1, k_2) are shown in the region R (including the boundary) of figure 9 for the case of unbuffered leaky buckets with input from two classes (real-time and non-real-time) of iid Erlang on-off sources with $N_U^1 = 8$, $\alpha_1 = 2.4$, $N_D^1 = 5$, $\beta_1 = 0.4$, $r_1 = 2.0$, $\zeta_1 = 0.001$, $\varepsilon_1 = 0.0001$, $\kappa_1 = 40$, $N_U^2 = 7$, $\alpha_2 = 1$, $N_D^2 = 6$, $\beta_2 = 0.4$, $r_2 = 1.2$, $\zeta_2 = 0.01$, $\varepsilon_2 = 0.000001$, $\kappa_2 = 100$. The border node buffer has capacity $B = 200$ and the output channel capacity is 8.1. This admissible region can be mapped onto a look-up table and used for admission control.

7. Conclusions and extensions

In this paper we considered the scenario of multiple sources that belong to multiple classes which generate traffic that is policed by leaky buckets. Given the QoS constraints in terms of loss and delay at the first two stages of the traffic path (namely, the source node and a border node, both of which are typically owned by the same organization) for

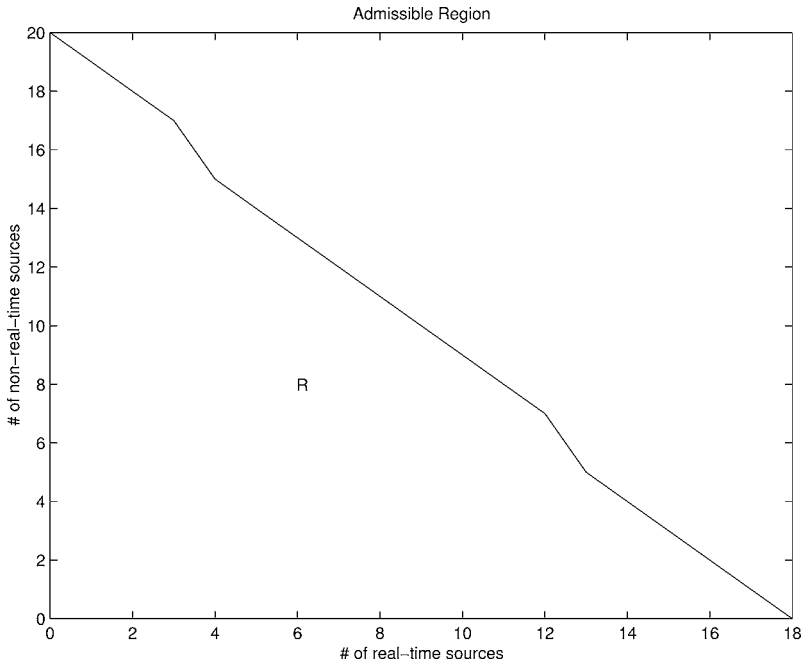

Figure 9. The 2-class unbuffered leaky bucket node admissible region.

all the classes of traffic, we formulated and solved a nonlinear programming problem to optimally select the leaky bucket parameters using the source input characteristics.

We considered two leaky bucket implementations: buffered and unbuffered. One of the constraints of the nonlinear optimization problem was the waiting time constraint for the buffered leaky bucket case and tagging constraint for the unbuffered leaky buckets case. We derived algebraic expressions for those constraints. Then we developed an algorithm using the Karush–Kuhn–Tucker (KKT) conditions to solve for the optimal leaky bucket parameters at the source node.

In order to analyze the traffic into the border node, we derived expressions for the effective bandwidth of the output from a leaky bucket when the input source is modulated by (i) a semi-Markov process for the buffered leaky bucket case, and, (ii) an exponential on-off source for the unbuffered leaky bucket case. Then we used the output effective-bandwidth to solve the QoS problem for overflow probability and delay at the border node buffer. We used the optimal leaky bucket parameters and output effective bandwidths to address network design and admission control problems at the border node. In terms of implementation, the numerical solutions can be executed off-line to compute the design and admission control schemes. These can be stored and used via table-look-up to implement on-line design decisions and admission control. The computations do not need to be executed at every decision, but only when the input parameters change.

An important extension that we propose to work on in the future is to solve a network-wide global optimization problem with leaky bucket sources at different nodes of a private network. The constraints are end-to-end QoS measures that have to be

guaranteed to the users. We will also consider dual leaky buckets and multiple tandem leaky buckets in this extension.

Acknowledgements

The author would like to thank the associate editor and all the anonymous reviewers for their comments and suggestions that led to considerable improvements in the content and presentation of this paper.

Appendix A. Proof of theorem 1

The proof follows the arguments in [Gün et al., 23] for the exponential on-off source case using the steady state analysis of the $\{(W(t), Z(t)),\ t \geqslant 0\}$ process defined in the proof of theorem 3. Consider the $\{W_i(t), t \geqslant 0\}$ process illustrated in figure 10. From section 5.1, we know that the $\{W_i(t), t \geqslant 0\}$ process behaves like the buffer-content process of a single infinite-buffer fluid-model with input from a general on-off source with distributions $U_i(\cdot)$ and $D_i(\cdot)$. Traffic is generated at rate r_i when the source is on, and at rate 0 when it is off. The output channel capacity is a constant γ_i. As $t \to \infty$, let $W_i(t) \to W_i$. Using the SMP bounds technique for general on-off sources in [Gautam et al., 19; Palmowski and Rolski, 32], we can derive

$$P(W_i > w) \leqslant C_i^* \mathrm{e}^{-\eta_i w},$$

where

$$C_i^* = \frac{\widetilde{U}_i(-\eta(r_i - \gamma_i)) - 1}{\eta_i(\tau_U^i + \tau_D^i)} \frac{r_i}{(r_i - \gamma_i)\gamma_i} \Bigg/ \min_{x \geqslant 0} \left\{ \frac{\int_x^\infty \mathrm{e}^{\eta_i(r_i - \gamma_i)(y-x)}\, \mathrm{d}U_i(y)}{1 - U_i(x)} \right\},$$

and η_i is the solution to $\widetilde{U}_i(-\eta_i(r_i - \gamma_i))\widetilde{D}_i(\eta_i\gamma_i) = 1$. Define Z_i as the steady state random variable representing the environmental process $\{Z_i(t),\ t \geqslant 0\}$, i.e., as $t \to \infty$,

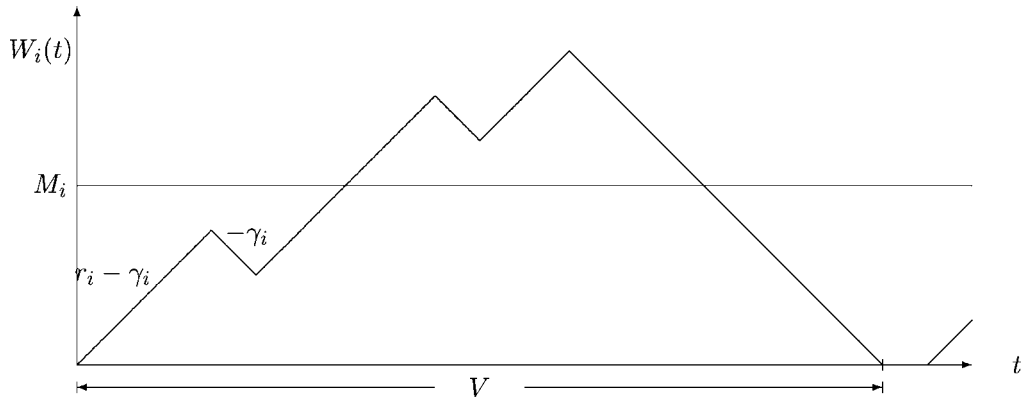

Figure 10. The $\{W_i(t),\ t \geqslant 0\}$ process.

$Z_i(t) \to Z_i$. Note that $Z_i(t) = 0$ or $Z_i(t) = 1$ represents the source being in the off or on state respectively at time t. Using the results for alternating renewal processes, we can show that

$$P(Z_i = 1) = E(Z_i) = \frac{m_i}{r_i}.$$

The fraction of time the source is in the on state whenever W_i is greater than a positive number w is γ_i/r_i. This is because the ratio of on-times to off-times whenever $W_i > w$ is $\gamma_i/(r_i - \gamma_i)$. Therefore

$$P(Z_i = 1 \mid W_i > w) = \frac{\gamma_i}{r_i}.$$

Furthermore, the conditional steady-state distribution of the $\{W(t), t \geqslant 0\}$ process is given by

$$\begin{aligned} P\{W_i > w \mid Z_i = 1\} &= P(Z_i = 1 \mid W_i > w)\frac{P(W_i > w)}{P(Z_i = 1)} \\ &\leqslant C_i^* \frac{\gamma_i}{m_i} \mathrm{e}^{-\eta_i w}. \end{aligned}$$

Similar to Z_i, define X_i as follows: as $t \to \infty$, $X_i(t) \to X_i$. The waiting time constraint is equivalent to

$$P\{X_i > \gamma_i d_i^* \mid Z_i = 1\} \leqslant \zeta_i,$$

since a delay of d_i^* is equivalent to $\gamma_i d_i^*$ amount of fluid in the data buffer. Clearly,

$$P\{X_i > \gamma_i d_i^* \mid Z_i = 1\} = P\{W_i > M_i + \gamma_i d_i^* \mid Z_i = 1\}.$$

Hence the waiting time constraint is satisfied if

$$C_i^* \frac{\gamma_i}{m_i} \mathrm{e}^{-\eta_i (M_i + \gamma_i d_i^*)} \leqslant \zeta_i.$$

Appendix B. Proof of theorem 2

As $t \to \infty$, let $W_i(t) \to W_i$. For this proof we compare the W_i random variables of the buffered against those of the unbuffered leaky bucket cases. Instead of using a new set of notation, we represent it as a conditional probability. Using a sample path argument we can easily show that $P(W_i = M_i \mid \text{unbuffered leaky bucket } i)$ is always less than or equal to $P(W_i > M_i \mid \text{ buffered leaky bucket } i)$. Also for an unbuffered leaky bucket, whenever $W_i = M_i$, a fraction $(r_i - \gamma_i)/m_i$ of traffic enters the network with a violation tag.

The tagging constraint is equivalent to

$$\frac{r_i - \gamma_i}{m_i} P(W_i = M_i \mid Z_i = 1, \text{ unbuffered leaky bucket } i) \leqslant \zeta_i$$

and is satisfied if

$$\frac{r_i - \gamma_i}{m_i} P(W_i > M_i \mid Z_i = 1, \text{ buffered leaky bucket } i) \leqslant \zeta_i.$$

Hence using the proof of theorem 1 in appendix A, the tagging constraint is satisfied if

$$C_i^* \frac{\gamma_i}{m_i} \left(\frac{r_i - \gamma_i}{m_i} \right) e^{-\eta_i M_i} \leqslant \zeta_i.$$

References

[1] V. Anantharam and T. Konstantopoulos, Optimality and interchangeability of leaky buckets, in: *32nd Allerton Conference*, Monticello, IL, 1994, pp. 235–244.

[2] V. Anantharam and T. Konstantopoulos, A methodology for the design of optimal traffic shapers in communication networks, IEEE Transactions on Automatic Control 44(3) (1999) 583–586.

[3] D. Anick, D. Mitra and M.M. Sondhi, Stochastic theory of a data handling system with multiple sources, Bell System Technical Journal 61 (1982) 1871–1894.

[4] M. Butto, E. Cavallero and A. Tonietti, Effectiveness of the leaky bucket policing mechanism in ATM networks, IEEE Journal on Selected Areas in Communications 9 (1991) 335–342.

[5] F. Callegati, G. Corazza and C. Raffaelli, On the dimensioning of the leaky bucket policing mechanism for multiplexer congestion avoidance, in: *IEEE International Conf. on Information Engineering*, Vol. 2, 1993, pp. 617–621.

[6] C.S. Chang and J.A. Thomas, Effective bandwidth in high-speed digital networks, IEEE Journal on Selected Areas in Communications 13(6) (1995) 1091–1100.

[7] C.S. Chang and T. Zajik, Effective bandwidths of departure processes from queues with time varying capacities, in: *INFOCOM'95*, pp. 1001–1009.

[8] H. Chen and A. Mandelbaum, Discrete flow networks: Bottleneck analysis and fluid approximations, Mathematics of Operations Reserach 16(2) (1991) 408–446.

[9] H. Chen and D. D. Yao, A fluid model for systems with random disruptions, Operations Research 40 (Suppl. 2) (1992) S239–S247.

[10] H. Chen and D.D. Yao, Dynamic scheduling of a multiclass fluid network, Operations Research 41(6) (1993) 1104–1115.

[11] G.L. Choudhury, D.M. Lucantoni and W. Whitt, On the effectiveness of effective bandwidths for admission control in ATM networks, in: *Proceedings of ITC-14* (Elsevier Science, Amsterdam, 1994) pp. 411–420.

[12] I. Cidon and I.S. Gopal, Paris: An approach to integrated high-speed private networks, International Journal of Digital and Analog Cabled Systems 1(2) (1998).

[13] G. de Veciana, C. Courcoubetis and J. Walrand, Decoupling bandwidths for networks: A decomposition approach to resource management, in: *INFOCOM'94*, 1994, pp. 466–473.

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

[15] G. de Veciana, G. Kesidis and J. Walrand, Resource management in wide-area ATM networks using effective bandwidths, IEEE Journal on Selected Areas in Communications 13(6) (1995) 1081–1090.

[16] A.I. Elwalid, D. Heyman, T.V. Lakshman, D. Mitra and A. Weiss, Fundamental bounds and approximations for ATM multiplexers with applications to video teleconferencing, IEEE Journal on Selected Areas in Communications 13(6) (1995) 1004–1016.

[17] A.I. Elwalid and D. Mitra, Analysis and design of rate-based congestion control of high speed networks, part I: Stochastic fluid models, access regulation, Queueing Systems 9 (1991) 29–64.

[18] A.I. Elwalid and D. Mitra, Effective bandwidth of general Markovian traffic sources and admission control of high-speed networks, IEEE/ACM Transactions on Networking 1(3) (June 1993) 329–343.
[19] N. Gautam, V.G. Kulkarni, Z. Palmowski and T. Rolski, Bounds for fluid models driven by semi-Markov inputs, Probability in the Engineering and Informational Sciences 13 (1999) 429–475.
[20] R.J. Gibbens and P.J. Hunt, Effective bandwidths for the multi-type UAS channel, Queueing Systems 9 (1991) 17–28.
[21] C. Greif and G. Golub, Techniques for solving general KKT systems, Technical Report, SCCM, Stanford (2000).
[22] X. Gu, K. Sohraby and D.R. Vaman, *Control and Performance in Packet, Circuit, and ATM Networks* (Kluwer Academic, Boston, 1995).
[23] L. Gün, V.G. Kulkarni and A. Narayanan, Bandwidth allocation and access control in high-speed networks, Annals of Operations Research 49 (1994) 161–183.
[24] J.M. Harrison, *Brownian Motion and Stochastic Flow Systems* (Wiley, New York, 1985).
[25] D. Holtsinger and H. Perros, Performance analysis of leaky bucket policing mechanisms, in: *Proc. of Tricomm '92*, Raleigh, NC, 1992.
[26] G. Kesidis, J. Walrand and C.S. Chang, Effective bandwidths for multiclass Markov fluids and other ATM sources, IEEE/ACM Transactions on Networking 1(4) (1993) 424–428.
[27] V.G. Kulkarni, Effective bandwidths for Markov regenerative sources, Queueing Systems 24 (1997).
[28] V.G. Kulkarni, Fluid models for single buffer systems, in: *Frontiers in Queueing*, Probability Stochastics Series (CRC, Boca Raton, FL, 1997), pp. 321–338.
[29] V.G. Kulkarni and N. Gautam, Admission control of multi-class traffic with service priorities in high-speed networks, Queueing Systems 27 (1997) 79–97.
[30] A. Narayanan and V.G. Kulkarni, First passage times in fluid models with an application to two-priority fluid systems, in: *Proc. of the IEEE Internat. Computer Performance and Dependability Symposium*, 1996.
[31] T.J. Ott and J.G. Shanthikumar, Discrete storage processes and their poisson flow and fluid flow approximations, Queueing Systems 24 (1997) 101–136.
[32] Z. Palmowski and T. Rolski, The superposition of alternating on-off flows and a fluid model, Report No. 82, Mathematical Institute, Wrocław University (June 1996).
[33] K. Sohraby and M. Sidi, On the performance of bursty and modulated sources subject to leaky bucket rate-based access control schemes, IEEE Transactions on Communications 42(2–4) (1994).
[34] S. Vamvakos and V. Anantharam, On the departure process of a leaky bucket system with long-range dependent input traffic, Queueing Systems 28(1–3) (1998).
[35] G. Wu and J.W. Mark, Discrete time analysis of leaky bucket congestion control, in: *Proc. of ICC '92*, 1992.
[36] N. Yin and M.G. Hluckyj, Analysis of the leaky bucket algorithm for on-off data sources, Journal of High Speed Networks 2(1) 81–98.