

Waiting Time and Cell Loss Probability Analysis for the Buffered Leaky Bucket

Douglas S. Holtsinger
Harry G. Perros

Center for Communications and Signal Processing
Department of Computer Science
North Carolina State University

TR-92/13
September 1992

WAITING TIME AND CELL LOSS PROBABILITY
ANALYSIS FOR THE BUFFERED LEAKY BUCKET¹

Douglas S. Holsinger†, Harry G. Perros‡

†Department of Electrical and Computer Engineering, and
Center for Communications and Signal Processing
North Carolina State University
Raleigh, NC 27695

‡Department of Computer Science, and
Center for Communications and Signal Processing
North Carolina State University
Raleigh, NC 27695

¹Supported in part by BellSouth, GTE Corporation, and NSF and DARPA under cooperative agreement NCR-8919038 with the Corporation for National Research Initiative.

Abstract

The use of preventive congestion control mechanisms has begun to receive increasing attention for applications in high-speed communication networks. The leaky bucket policing mechanism has been proposed for use as part of an overall preventive congestion control strategy in high-speed networks. We analyze approximately the waiting time distribution and the cell loss probability for the buffered leaky bucket policing mechanism, assuming that the source behaves as a two-state on/off arrival process with arbitrary distributions for the time spent in each state. Our results indicate that if the token pool size of the leaky bucket is sufficiently large, then the mean cell delay becomes negligible, and also it becomes roughly independent of the cell queue size and the average burst length. However, the percentile of the cell delay can vary substantially depending on the average burst length, becoming proportional to the product of the cell queue size and the token interarrival time even for fairly small cell loss probabilities (i.e. 10^{-5}).

1 Introduction

The design of congestion control mechanisms for high-speed networks has remained a difficult task because of the large propagation delays in these networks. These delays may give rise to long periods between the onset and detection of congestion conditions by

the appropriate network control elements [16]. Congestion conditions may be aggravated by the large bandwidth-delay product of the communication links between nodes, since these links may deliver a large number of cells to a congested node before the network control elements can throttle the offending source [2]. These problems may be avoided if connections are allocated their peak rate, but the resulting low bandwidth utilization may not be desirable since the network will support fewer connections. For these reasons, efforts have been focused on the development of preventive congestion control mechanisms for high-speed networks, rather than reactive controls [17].

The policing mechanism is an important component of a preventive congestion control strategy. The policing mechanism ensures that the traffic source does not submit excessive traffic into the network. A resource allocation algorithm operates in conjunction with the policing mechanism by allocating network resources to the connection [8]. Resources such as buffers and bandwidth may be allocated once during the setup of a connection, or allocated dynamically during the call.

In this paper, we analyze the buffered leaky bucket policing mechanism [15] assuming that the source behaves as two-state on/off arrival process with arbitrary distributions for the amount of time spent in each state. We derive the cell loss probability and the waiting time distribution of cells.

The outline of this paper is as follows. In section 2 we describe the leaky bucket policing

mechanism and provide an approximate analysis of the cell loss probability of the leaky bucket. Section 3 contains an approximate analysis of the waiting time distribution of cells, and in section 4, we present a comparison of the model with simulation, along with other numerical results. Finally in section 5, we present a summary of the results and our conclusions.

2 The leaky bucket policing mechanism

2.1 Description of the leaky bucket

Many policing mechanisms have been proposed and analyzed in the literature (for example, see [3, 10, 12]), but the leaky bucket policing mechanism appears to have received the most attention [1, 4, 5, 6, 7, 9, 11, 13, 14]. The leaky bucket policing mechanism is shown in figure 1. The leaky bucket is normally located at the user-network interface, where it polices the cell stream that is submitted into the network.

There are two queues; a cell queue of size C and a token pool of size K . Cells which arrive to the leaky bucket are required to consume a token from the token pool before proceeding into the network. If a cell arrives and there are tokens available in the token pool, then the cell departs immediately with a token. Otherwise, the cell waits in the cell queue if it is not full. During the time that the cell queue is full, arriving cells are

dropped.

Time is slotted, and a maximum of one cell may arrive per slot. Every N slots, a token is added to the token pool, which holds a maximum of K tokens. If the token pool is full upon a token arrival, then the token is dropped.

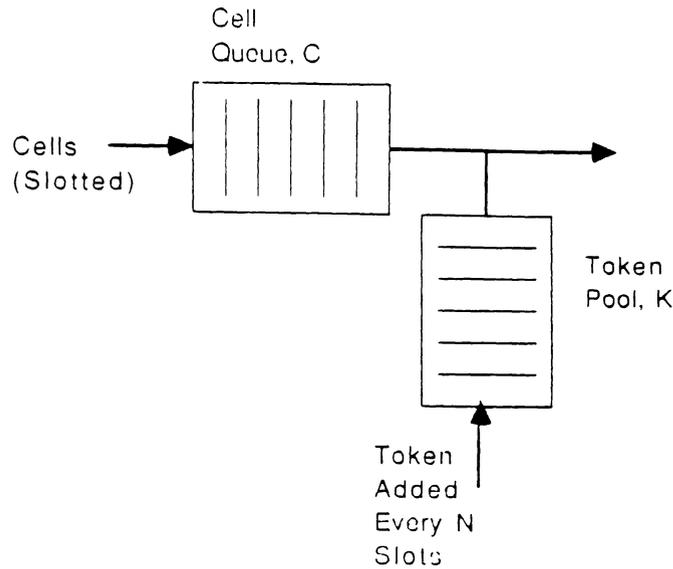


Figure 1: The Leaky Bucket

2.2 State probability distribution

The system we shall analyze is shown in figure 1. The traffic source is modeled as a two-state on/off arrival process, as shown in figure 2. The arrival process alternates between the burst and silence states. When the process is in the burst state, one cell is generated per slot. During the silence state, no cells are generated. A burst of cells is generated

after the arrival process moves from the silence state to the burst state. The times spent in the burst and silence states are assumed to be independent and arbitrarily distributed.

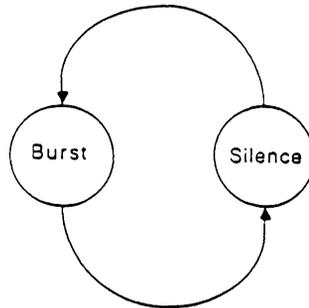


Figure 2: Two-state on/off Arrival Process

Our analysis of the leaky bucket is based upon a Markov chain embedded immediately before the beginning of each burst. The state of the system is represented by the number of slots remaining until the token pool is completely filled with tokens, assuming that no cells arrive. As shown in table 1, from the state of the system we can obtain the number of tokens in the token pool, the number of cells in the cell queue, and the number of slots remaining until the next token arrives. Figure 3 illustrates the timing of cell and token arrivals. Tokens arrive at time $(n)^+$, cells arrive at time $(n+1)^-$ and depart immediately if a token is available, and the system changes state at time $(n+1)$.

Each cell which consumes a token causes the state of the system to increase by N , since it takes N slots to generate a token. Each token arrival causes the state of the system

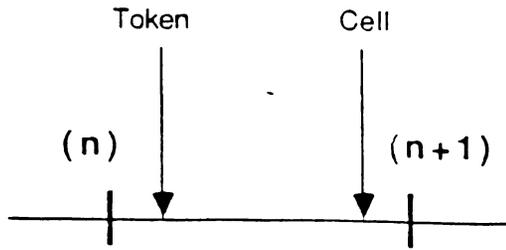


Figure 3: Timing of Token and Cell Arrivals

State	Residual Token		
	Tokens	Cells	Arrival Time
0	K	0	N
1, ..., N	K-1	0	1, ..., N
N+1, ..., 2N	K-2	0	1, ..., N
...			
(K-1)N+1, ..., KN	0	0	1, ..., N
KN+1, ..., (K+1)N	0	1	1, ..., N
...			
(K+C-1)N+1, ..., (K+C)N	0	C	1, ..., N

Table 1: Relationship between the states and the token/cell count

to decrease by N . Since a token arrives every N slots, the token generation process causes the state of the system to decrease by one during each slot. Thus, during the burst period when cells pass through the leaky bucket, the state of the system increases by $(N - 1)$ at each slot. During the silence period when no cells arrive, the state of the system decreases by one at each slot.

We introduce the following approximations which help to simplify the analysis of the system. We assume that if a cell arrives to a full token pool (state 0), then the state of the system always increases by $(N - 1)$. The state of the system will remain unchanged if a cell arrives to a system with a full cell queue. As an approximation, we assume that if a cell arrives and the cell queue is full, (states $(K + C - 1)N + 1$ through $(K + C)N$), then the state of the system always becomes $(K + C)N$.

Figure 4 illustrates a possible evolution of the state of the system, assuming that $K + C = 3$ and $N = 4$. During the burst period, cells arrive one per slot, and during the silence period, no cells arrive. During a burst in which a cell is lost, if the last cell in a burst arrives at time $(m)^-$, then the next token will arrive at time $(m + N - 1)^+$ if no cells arrive before time $(m + N - 1)$. These approximations will help simplify the state evolution equation. Through simulation we have found that these approximations introduce a negligible amount of error in many instances.

We define the following random variables:

X_n = State of the system at the beginning of the n th burst period

\hat{B}_n = Increase in the state of the system during the n th burst period

B_n = Length of the n th burst period in slots

S_n = Length of the n th silence period in slots

The system evolves according to the following equation:

$$X_{n+1} = [\min(X_n + \hat{B}_n, (K + C)N) - S_n]^+ \quad (1)$$

with X_n taking on values from 0 to $(K + C)N$. Let $P_i(n)$ be the probability that $X_n = i$ at the beginning of the n th burst, and let $\bar{P}(n)$ be the associated state probability vector. Letting $q_{l,i}$ represent the transition probability from state l to i , and denoting Q as the matrix of transition probabilities $[q_{l,i}]$, we can write $\bar{P}(n + 1) = \bar{P}(n)Q$. If we let

$$P_i = \lim_{n \rightarrow \infty} P_i(n),$$

then we can solve a linear system of equations

$$\bar{P} = Q\bar{P} \quad (2)$$

by replacing one of the equations with the condition that $\sum_i P_i = 1$.

To derive the transition probabilities $q_{l,i}$, we note that the interval between embedded points consists of a burst period followed by a silence period. The time spent in each burst

period (and silence period) is independent and identically distributed, so the subscript n can be dropped from the random variables B_n , \hat{B}_n , and S_n . Now we define

$$\begin{aligned}\hat{w}_i &= P(\hat{B} = i) \\ &= P((N - 1)B = i) \\ w_i &= P(B = i) \\ s_i &= P(S = i).\end{aligned}$$

Now we can write

$$q_{l,i} = \bar{w}_{(K+C)N-1} s_{(K+C)N-i} + \sum_{j=1}^{(K+C)N-l} \hat{w}_j s_{l+j-i} \quad \begin{array}{l} l = 0, \dots, (K+C)N, \\ i = 1, \dots, (K+C)N \end{array} \quad (3)$$

$$q_{l,0} = \bar{w}_{(K+C)N-1} s_{(K+C)N} + \sum_{j=1}^{(K+C)N-l} \hat{w}_j \bar{s}_{l+j} \quad l = 0, \dots, (K+C)N \quad (4)$$

where

$$\begin{aligned}\bar{w}_i &= P(\hat{B} > i) \\ &= 1 - \sum_{j=1}^i \hat{w}_j \\ \bar{s}_i &= P(S > i) \\ &= 1 - \sum_{j=1}^i s_j\end{aligned}$$

(Note that the state $(K + C)N$ cannot be reached from any other state). The above solution produces a state space of size $((K + C)N + 1)$, which may be too large to solve efficiently for some configurations. Below we describe an alternative solution method

which relies on aggregating the states, and results in reducing the size of the state space to $O(K + C)$.

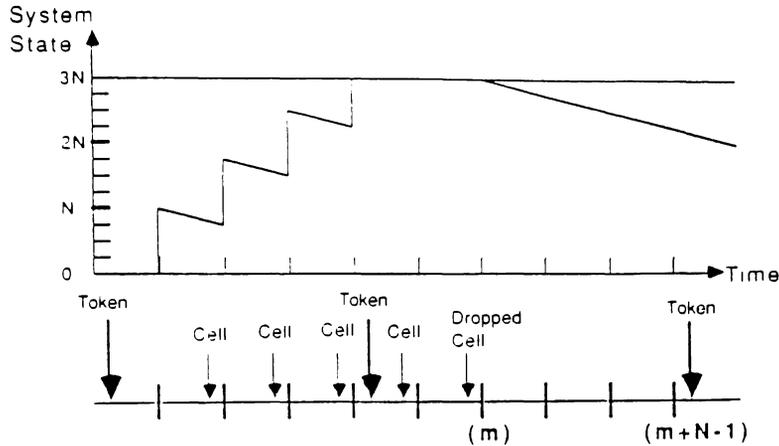


Figure 4: State evolution, $N = 4, K + C = 3$

2.3 Alternative solution method

The state of the system is represented by the number of slots remaining until the token queue is filled with tokens. During the burst period, the system moves from state i to state $(i + N - 1)$ at each slot. The alternative solution method consists of lumping the states together to construct an aggregate Markov Chain. The lumping of states is done so that during the burst period, the aggregated system moves from aggregate state i to

Original States	Aggregate State
$0, \dots, N - 2$	0
$N - 1, \dots, 2N - 3$	1
...	...
$i(N - 1), \dots, (i + 1)(N - 1) - 1$	i
...	...
$M(N - 1), \dots, (M + 1)(N - 1) - 1$	M
...	...
$Z(N - 1), \dots, (K + C)N$	Z

Table 2: Aggregation of States

aggregate state $(i + 1)$ at each slot.

The aggregate states and their corresponding original states are shown in table 2. As will be seen shortly, this particular method for aggregating the states allows the transition probabilities associated with the burst period to be written directly in terms of the distribution of the length of the burst period. Unlike the original states shown in table 1, the aggregate states do not correspond to a token/cell count.

The aggregate state Z is defined as the largest integer Z such that $Z(N - 1) \leq (K + C)N$, and the aggregate state M is defined as the largest integer M such that $M(N - 1) \leq KN$. Note that the aggregate state Z may contain less than $N - 1$ original states.

Let \widetilde{X} denote the random variable representing the state in the aggregated system, and let \bar{P}_i denote the probability that $(\widetilde{X} = i)$ at an embedded point. We define the following random variables in the aggregated system

\bar{B} = Increase in the state of the system during the burst period

\tilde{S} = Decrease in the state of the system during the silence period

and the associated probability distributions are

$$a_i = P(\bar{B} = i)$$

$$b_i = P(\tilde{S} = i).$$

During the burst period, the system moves from state i to state $(i + 1)$ at each slot, and so $P(\bar{B} = i)$ is simply the probability that the burst period lasts i slots, which we have defined previously as w_i . Hence $a_i = w_i$.

The aggregation step results in the loss of information of the residual token arrival time and the number of tokens in the token pool, since an aggregate state contains $(N - 1)$ original states and not N original states (see table 1). However, this information can be recovered from the aggregate states through a simple approximation. Given that the system is in aggregate state j , the probability that the system is actually in one of the original states $j(N - 1)$ through $(j + 1)(N - 1) - 1$ is assumed to be uniformly distributed among these $(N - 1)$ original states.

Using this approximation, we can write the transition probabilities corresponding to the silence period as:

$$\begin{aligned}
b_i &= P(\bar{S} = i) \\
&= \sum_{j=0}^{N-2} \left[\frac{j}{N-1} s_{(i-1)(N-1)+j} + \frac{N-1-j}{N-1} s_{i(N-1)+j} \right] \quad i = 0, 1, 2, \dots
\end{aligned} \tag{5}$$

where we recall that s_i is the probability that the silence period lasts i slots.

The state transition probabilities $\bar{q}_{l,i}$ for the aggregated system take the same form as equations (3) and (4), but with $(K+C)N$ replaced by Z , \hat{w}_i replaced by a_i , and s_i replaced by b_i . We have

$$\bar{q}_{l,i} = \bar{a}_{Z-l} \bar{b}_{Z-i} + \sum_{j=1}^{Z-l} a_j \bar{b}_{l+j-i} \quad l = 0, \dots, Z \quad i = 1, \dots, Z \tag{6}$$

$$\bar{q}_{l,0} = \bar{a}_{Z-l} \bar{b}_Z + \sum_{j=1}^{Z-l} a_j \bar{b}_{l+j} \quad l = 0, \dots, Z \tag{7}$$

where we define

$$\begin{aligned}
\bar{a}_i &= P(\bar{B} > i) \\
&= 1 - \sum_{j=1}^i a_j \\
\bar{b}_i &= P(\bar{S} > i) \\
&= 1 - \sum_{j=1}^i b_j
\end{aligned}$$

While the probabilities a_i and b_i cannot be obtained explicitly, it is only necessary to compute a finite number of these probabilities.

As before, we denote \tilde{Q} as the the matrix of transition probabilities $[\tilde{q}_{l,i}]$, and \tilde{P} as the state probability vector, and we solve a linear system of equations $\tilde{P} = \tilde{Q}\tilde{P}$ by replacing one of the equations with the condition that $\sum_i \tilde{P}_i = 1$.

2.4 Cell loss probability

Let P_{loss} denote the cell loss probability, defined as the fraction of cells which are dropped by the leaky bucket. The cell loss probability is determined assuming that the state probabilities are computed using the alternative solution method.

Let the state of the system at time slot m be \tilde{X}_m (see figure 3). A cell arriving at instant $(m+1)^-$ is dropped if $\tilde{X}_m + 1 > Z$, where Z is the largest aggregate state number. All cells which arrive in the same burst after the first lost cell are subject to rejection, and the state of the system remains at Z until the last cell in a burst arrives. Tokens will continue to arrive during this period, and will depart with a cell every N slots. Hence, the proportion of cells which are dropped during this period is approximately

$$\frac{N-1}{N}$$

Suppose that the state of the system at the beginning of the n th burst is $\tilde{X}_n = k$. Given that a randomly selected cell arrives in a burst of length i in which at least one cell is dropped in the burst (i.e. $i+k > Z$), the probability that the cell is dropped can be

written as

$$\frac{N-1}{N} \frac{i+k-Z}{i}. \quad (8)$$

The probability that a randomly selected arriving cell arrives during a burst of length i , is written as

$$\frac{iP(\text{burst length} = i)}{\sum_i iP(\text{burst length} = i)} \quad (9)$$

or

$$\frac{ia_i}{E[B]} \quad (10)$$

where $E[B]$ is the expected length of a burst period in slots.

Let $P_{\text{loss}}(k)$ represent the probability that a randomly selected cell is lost in a burst, given that the system state is k at the beginning of a burst period. Using equations (8) and (10), we write,

$$\begin{aligned} P_{\text{loss}}(k) &= \sum_{i=Z-k+1}^{\infty} \frac{ia_i}{E[B]} \frac{N-1}{N} \frac{i+k-Z}{i} \\ &= \frac{N-1}{N} \sum_{i=Z-k+1}^{\infty} \frac{(i+k-Z)a_i}{E[B]} \end{aligned}$$

The cell loss probability can now be written as

$$P_{\text{loss}} = \sum_{k=0}^Z \bar{P}_k P_{\text{loss}}(k)$$

which can be expressed as

$$D_{\text{loss}} = \frac{N-1}{V} \left[1 - \frac{Z - E[\tilde{X}] - \sum_{k=0}^Z \bar{P}_k \sum_{i=1}^{Z-k} (Z-k-i)a_i}{E[B]} \right] \quad (11)$$

where $E[\tilde{X}] = \sum_k k \cdot \bar{P}_k$, and $E[B]$ is the expected length of a burst period.

Equation (11) has a simple interpretation. The term $(Z - E[\tilde{X}])$ is the expected number of cells which the leaky bucket can accept during a single burst period without rejecting a cell. The summation is the expected number of additional cells which the leaky bucket could have accepted at the end of a burst period. Thus, the numerator is the expected number of cells which arrive before the first rejected cell in a burst, which divided by the denominator gives the probability that a cell arrives before the first rejected cell in a burst. The term $\frac{N-1}{V}$ gives the probability that the leaky bucket rejects a cell which does not arrive before the first rejected cell in a single burst period.

Equation (11) indicates that the cell loss probability does not depend on the individual sizes of the token pool (K) and the cell queue (C), but on their sum, in agreement with the results of Berger [4].

3 Waiting time distribution

The waiting time distribution of cells is determined from the aggregate state probabilities calculated in section 2.3 and the cell loss probability calculated in section 2.4. We assume that the source is modeled as a two-state on/off arrival process as shown in figure 2.

In order to simplify the analysis of the waiting time distribution, we require that all aggregate states contain $(N - 1)$ original states, including the largest aggregate state Z .

Referring to table 2, this requires

$$(K + C)N - (N - 1)Z = N - 2,$$

or

$$Z = \frac{(K + C)N - N + 2}{N - 1},$$

where Z is the largest aggregate state, and it assumes integer values. To simplify the determination of the probability that the waiting time of a cell is zero, we require that

$$KN - M(N - 1) = N - 2$$

or

$$M = \frac{KN - N + 2}{N - 1}.$$

where M assumes integer values.

The information about the number of tokens in the token pool and the residual token interarrival time is needed to calculate the waiting time distribution of cells. This information is lost when the states are aggregated during the solution of the state probabilities. We recover this information by using the same approximation given in section 2.3. Given that the system is in aggregate state j , the probability that the system is actually in one of the original states $j(N - 1)$ through $(j + 1)(N - 1) - 1$ is uniformly distributed among these $(N - 1)$ states. We use this an approximation for the original state probability distribution, which is in turn used for calculating the waiting time distribution.

Let \hat{P}_l represent the probability that the system is in state l , using the previous approximation. Then we can write

$$\hat{P}_l = \bar{P}_{\lfloor l/(N-1) \rfloor} \quad l = 0, \dots, (K + C)N \quad (12)$$

where \bar{P}_l is the aggregate state probability. In the following analysis, we calculate the probability that a randomly selected cell which is accepted has a waiting time of j slots, which is denoted as $P\{W = j\}$.

3.1 Zero waiting time

A randomly selected cell which arrives at time $(n)^-$ will have zero waiting time if no cells are contained in the cell queue when the system changes state at time (n) and there is at least one token available. Referring to table 1, this event can only occur if the state

of the system is less than or equal to KN at time (n).

The random selection of a cell implies that the position of the cell is uniformly distributed in the burst in which it is contained. In order for the cell to have zero waiting time, its position in the burst x , and the state of system at the beginning of a burst l must satisfy

$$l + (N - 1)x \leq KN.$$

Since x can only take on values greater than zero, it follows that x and l satisfy

$$0 \leq l \leq KN - N + 1 \tag{13}$$

and

$$x \leq \lfloor \frac{KN - l}{N - 1} \rfloor \tag{14}$$

Let U represent the random variable for the length of the burst period in which a randomly selected cell is contained, and let $P\{U = i\}$ denote the probability that a randomly selected arriving cell is contained in a burst of length i . The probability $P\{U = i\}$ is given by equation (10), repeated here,

$$P\{U = i\} = \frac{ia_i}{E[B]}$$

where $E[B]$ is the expected length of a burst period and a_i is the probability that the burst period lasts i slots.

Let $P\{\text{position} = \mathbf{x} \mid U = i\}$ represent the probability that the cell is in position \mathbf{x} of the burst, given that the cell is contained in a burst of length i . This probability can be written as

$$P\{\text{position} = \mathbf{x} \mid U = i\} = \frac{1}{i}. \quad (15)$$

Using equation (10), (12), and equations (13) through (15), the probability that a randomly selected cell experiences zero waiting time is written as

$$\begin{aligned} P\{W = 0\} &= P\{\text{Cell experiences no waiting} \mid \text{cell accepted}\} \\ &= \frac{\sum_{l=0}^{KN-N+1} \hat{P}_l \sum_{i=1}^{\infty} P\{U = i\} \sum_{\mathbf{z}=1}^{\min(i, \lfloor \frac{KN-l}{N-1} \rfloor)} P\{\text{position} = \mathbf{x} \mid U = i\}}{P\{\text{cell accepted}\}} \\ &= \frac{\sum_{l=0}^{KN-N+1} \hat{P}_l \sum_{i=1}^{\infty} i a_i \sum_{\mathbf{z}=1}^{\min(i, \lfloor \frac{KN-l}{N-1} \rfloor)} 1/i}{E[B] P\{\text{cell accepted}\}} \end{aligned}$$

which after some manipulation leads to

$$P\{W = 0\} = \frac{\sum_{l=0}^{KN-N+1} \hat{P}_l \left[\sum_{i=0}^{\lfloor \frac{KN-l}{N-1} \rfloor - 1} i a_i + \lfloor \frac{KN-l}{N-1} \rfloor \left(1 - \sum_{i=0}^{\lfloor \frac{KN-l}{N-1} \rfloor - 1} a_i \right) \right]}{E[B](1 - P_{\text{loss}})} \quad (16)$$

3.2 Non-zero waiting time

A randomly selected cell which arrives at time $(n)^-$ will have a waiting time of j slots ($j > 0$) if it departs at the token arrival instant $(n + j - 1)^+$. Referring to table 1, this

event occurs if the state of the system becomes equal to $(KN + j)$ immediately after the cell arrival at time (n) . Assuming that no previous cell in the burst had been dropped, we have

$$l + (N - 1)x = KN + j, \quad (17)$$

or

$$x = \frac{KN + j - l}{N - 1}, \quad (18)$$

where x is the position of the cell in the burst, and l is the state of the system at the beginning of a burst. The position x assumes integer values. If a cell had been dropped in the burst before the randomly selected cell arrived, then the randomly selected cell would be placed in the last cell queue position (state $(K + C)N$), and would experience a waiting time of CN . Hence equation (17) restricts the value of j to

$$1 \leq j < CN \quad (19)$$

We will subsequently examine the case where the cell experiences a waiting time of CN . Since the position x of the randomly selected cell is related to the size of the burst i by

$$1 \leq x \leq i, \quad (20)$$

we have from equation (17),

$$0 \leq l \leq KN + j - N + 1 \quad (21)$$

Substituting for x from equation (18) into equation (20), and using the fact that x and i assume integer values, we have

$$i \geq \lceil \frac{KN + j - l}{N - 1} \rceil. \quad (22)$$

Using equations (10), (??), (15), and equations (17) through (22), the probability that the random selected cell experiences a waiting time of j slots is given as

$$\begin{aligned} P\{W = j\} &= P\{\text{Cell experiences waiting time of } j \text{ slots} \mid \text{cell accepted}\} \\ &= \frac{\sum_{l=0}^{KN+j-N+1} \hat{P}_l \sum_{i=\lceil \frac{KN+j-l}{N-1} \rceil}^{\infty} P\{U = i\} \sum_{\{x: 1 \leq x \leq i, 1+(N-1)x=KN+j\}} P\{\text{position} = x \mid U = i\}}{P\{\text{cell accepted}\}} \\ &= \frac{\sum_{l=0}^{KN+j-N+1} \hat{P}_l \sum_{i=\lceil \frac{KN+j-l}{N-1} \rceil}^{\infty} a_i \text{int}(\frac{KN+j-l}{N-1})}{E[B](1 - P_{\text{loss}})} \quad 1 \leq j < CN \end{aligned}$$

where $\text{int}(x)$ is 1 if x is an integer, and is 0 otherwise. The above equation reduces to

$$P\{W = j\} = \frac{\sum_{l=0}^{KN+j-N+1} \text{int}(\frac{KN+j-l}{N-1}) \hat{P}_l \left[1 - \sum_{i=1}^{\lceil \frac{KN+j-l}{N-1} \rceil - 1} a_i \right]}{E[B](1 - P_{\text{loss}})} \quad 1 \leq j < CN \quad (23)$$

The probability that the cell experiences a waiting time of CN is written using equations (16) and (23) as

$$P\{W = CN\} = 1 - \sum_{j=0}^{CN-1} P\{W = j\} \quad (24)$$

The mean waiting time of cells is

$$E[W] = \sum_{i=0}^{CN} i P\{W = i\} \quad (25)$$

4 Numerical results

4.1 Comparison with simulation

Table 3 shows the particular traffic source which is used for our numerical studies. The Markov modulated arrival process is a two-state process which alternates between the active and inactive states. During the active state, cell arrivals occur one per slot, and during the inactive state, no cells arrive. The process spends a geometrically distributed amount of time in each state. The Markov modulated arrival process has been used in previous analyses of the leaky bucket policing mechanism as a suitable model for representing bursty arrival processes [6, 12].

Figures 5 and 6 show the mean waiting time of cells, for a token interarrival time of $N = 2$ and $N = 8$ respectively, assuming that the source behaves as a Markov modulated arrival process as described in table 3. The mean arrival rate ρ represents the fraction of slots which contain a cell arrival. The solid line indicates the results for the model, and the dotted line shows the results from a simulation of the leaky bucket. Each plot shows the mean waiting time of cells as a function of the leaky bucket size ($K + C$) for two different sizes of the cell queue. Overall, the simulation shows good agreement with the model, with better agreement for larger token pool sizes and larger token interarrival times.

Figures 7 and 8 illustrate the cell loss probability, for a token interarrival time of $N = 2$

Average Burst Length	10
Average Silence Length	90
Mean Arrival Rate	0.1
Distribution of on/off periods	Geometric

Table 3: Arrival process used in numerical results

and $N = 8$, respectively, assuming that the source behaves as a Markov modulated arrival process as described in table 3. Generally speaking, the model tends to overestimate the cell loss probability (solid line), with better agreement for larger token interarrival times.

4.2 Waiting time, fixed token pool size

For the remaining numerical results, we use the Markov modulated arrival process as described in table 3, while altering the size of the token pool (K) and the cell queue (C), except as otherwise noted.

The token generation rate γ can be expressed in terms of the mean cell arrival rate ρ , and a dimensioning factor D as

$$\gamma = \frac{1}{N} = D \rho \quad 0 < \rho \leq 1$$

where the dimensioning factor can take on real values such that the token interarrival time N is an integer. When $D = 1$, the mean cell arrival rate equals the token generation

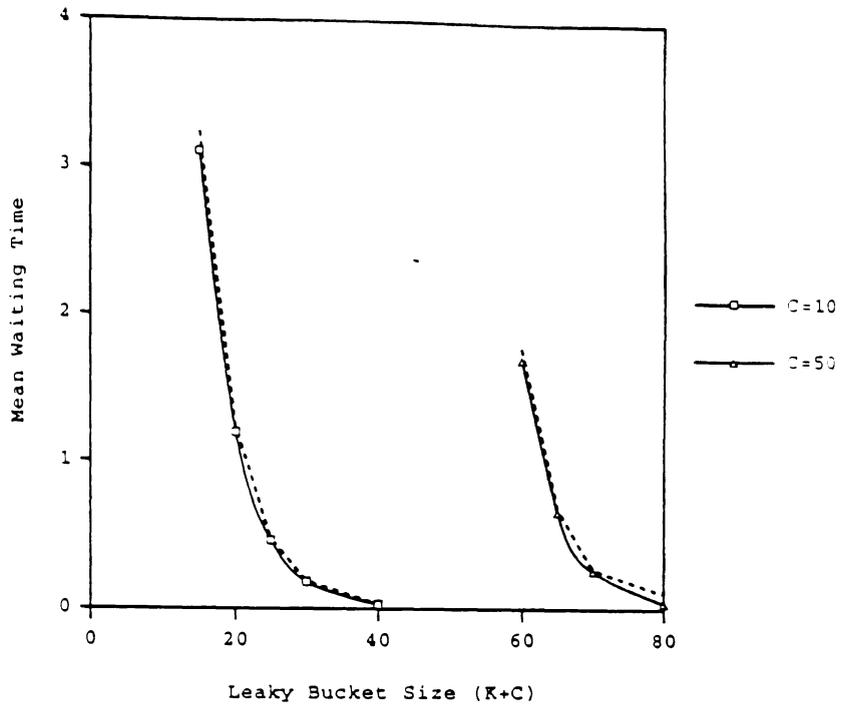


Figure 5: Mean Waiting Time, $N = 2$

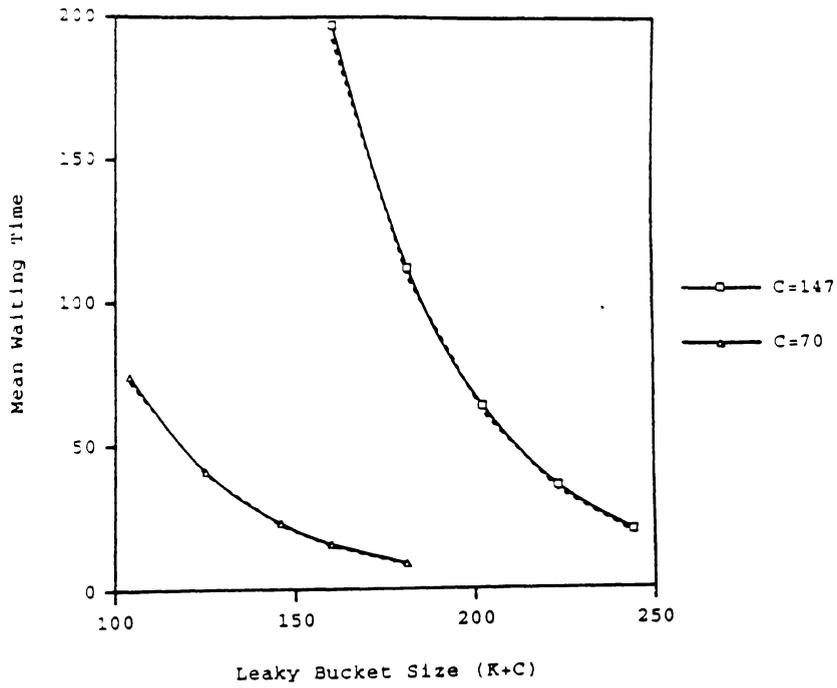


Figure 6: Mean Waiting Time, $N = 8$

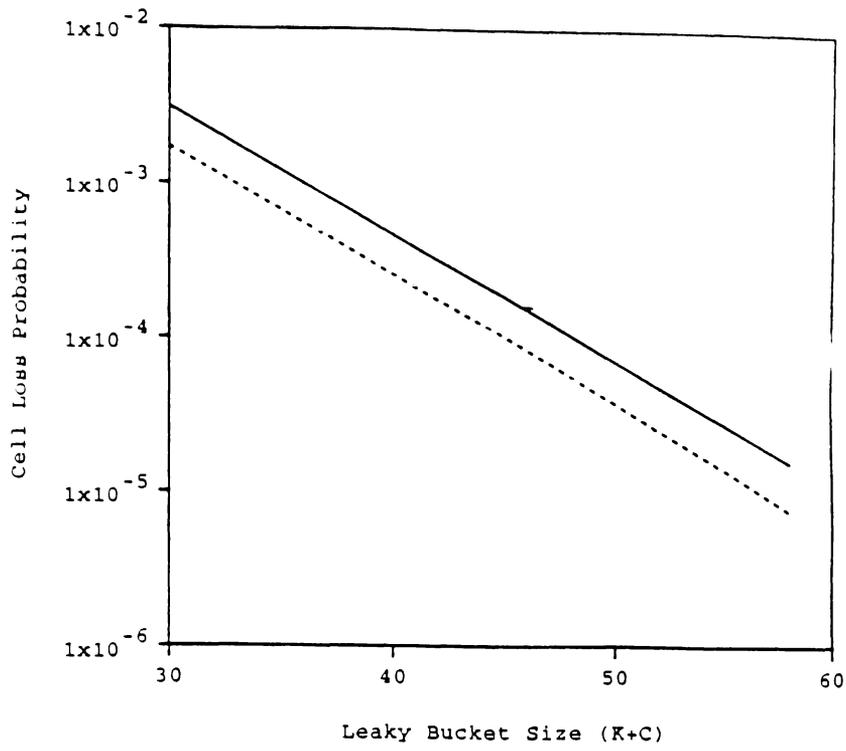


Figure 7: Cell Loss Probability, $N = 2$

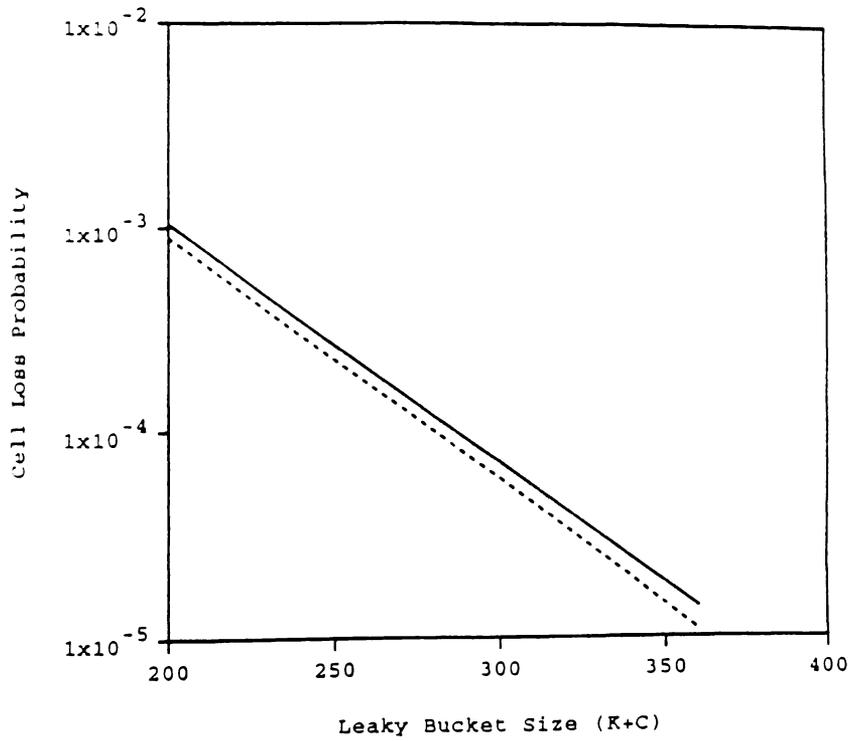


Figure 8: Cell Loss Probability, $N = 8$

rate, and the mean departure rate is policed very tightly. When D increases to $\frac{1}{\rho}$, a token arrives every slot and the leaky bucket passes all arriving cells without dropping or delaying cells. For our purposes, we use dimensioning factors of $D = 5$ and $\Gamma = 1.25$ corresponding to a token interarrival time of $N = 2$ and $N = 8$, and a mean arrival rate of $\rho = 0.1$.

Figure 9 shows the mean waiting time of cells for a token interarrival time of $N = 8$. Referring to figure 8 it can be seen that increasing the leaky bucket size decreases the cell loss probability, and so increasing the leaky bucket size has the effect of increasing the number of accepted cells, and hence increasing the waiting time. However, as seen in figure 9, increasing the cell queue size beyond a certain point, does not increase the waiting time, suggesting that the token pool size plays a more significant role than the cell queue size in determining the waiting time as the cell loss probability decreases.

Figure 10 shows that the maximum value of the 99.9% delay time can be an order of magnitude larger than the corresponding mean waiting time. The maximum value of the 99.9% delay time can be closely approximated as $C \cdot N$ for leaky bucket sizes of less than 200, since here the cell loss probability is large enough such that the cell queue is occupied much of the time. As the leaky bucket size increases, the maximum value of the 99.9% delay time tends to level off.

Figures 11 and 12 show the mean waiting time of cells and the maximum value of the

99.9% delay time of cells for a token interarrival time of $N = 2$. The same observations apply here, except the mean waiting time and the maximum delay time are much smaller. As seen in figure 12, the maximum delay time slopes away from the function $C \cdot N$ much more sharply as the leaky bucket size increases than when the token interarrival time is $N = 8$, as in figure 10.

Our results contrast sharply with the results given by Sohraby and Sidi [14], in which they suggest that the token interarrival time, and not the token pool size, is the dominant factor for determining the waiting time characteristics. They used a leaky bucket with an infinite cell queue size, and they used relatively small token pool sizes ($K = 20$).

Our results show that both the token interarrival time and the token pool size play a dominant role in determining the mean waiting time, and that a sufficiently large token pool size can allow for a negligible mean waiting time. Beyond a certain point, increasing the cell queue size does not affect the mean waiting time significantly, although it can have a profound effect on the maximum value of the 99.9% delay time.

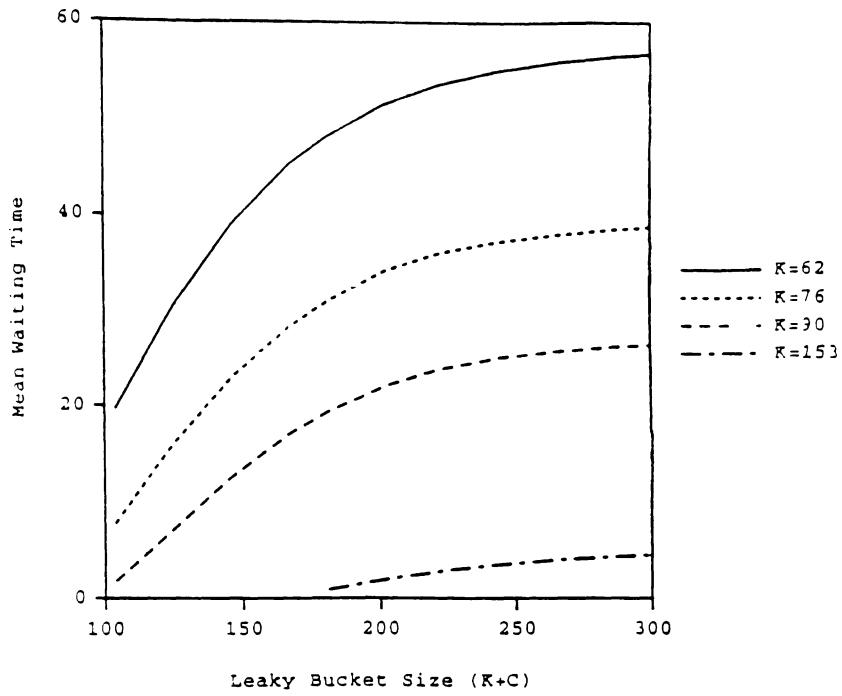


Figure 9: Mean Waiting Time, $N = 8$

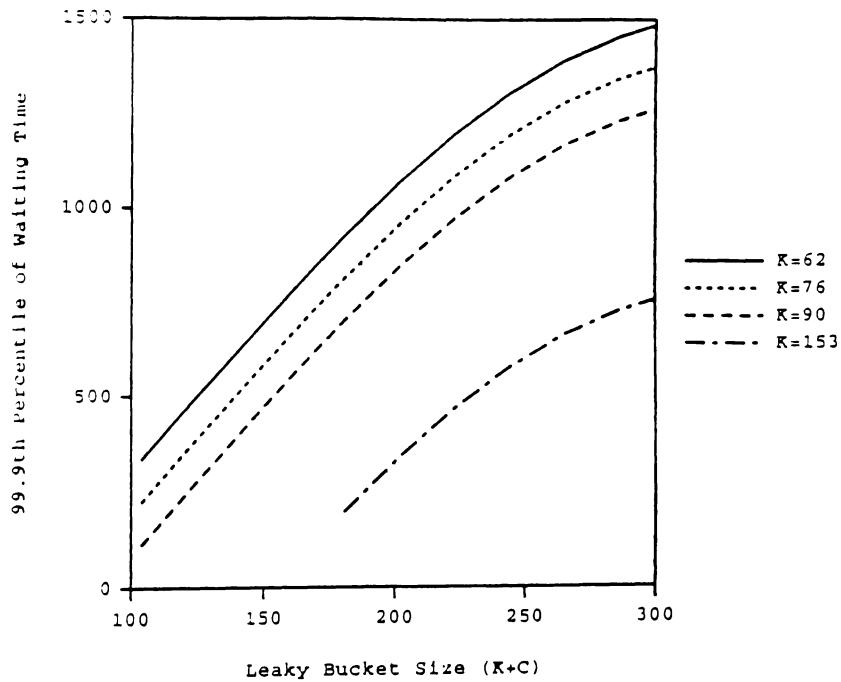


Figure 10: Maximum value of 99.9% delay time, $N = 8$

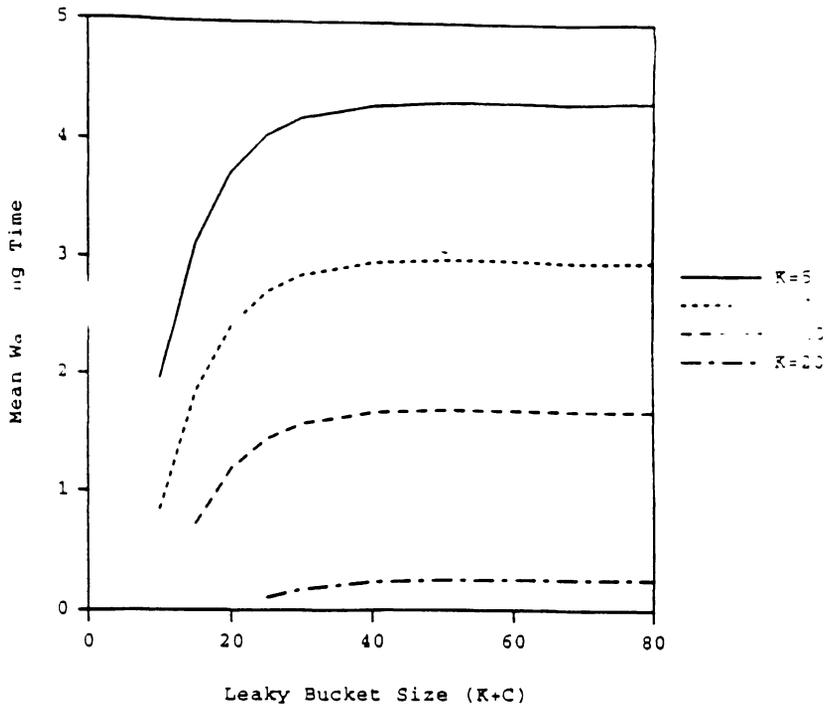


Figure 11: Mean Waiting Time, $N = 2$

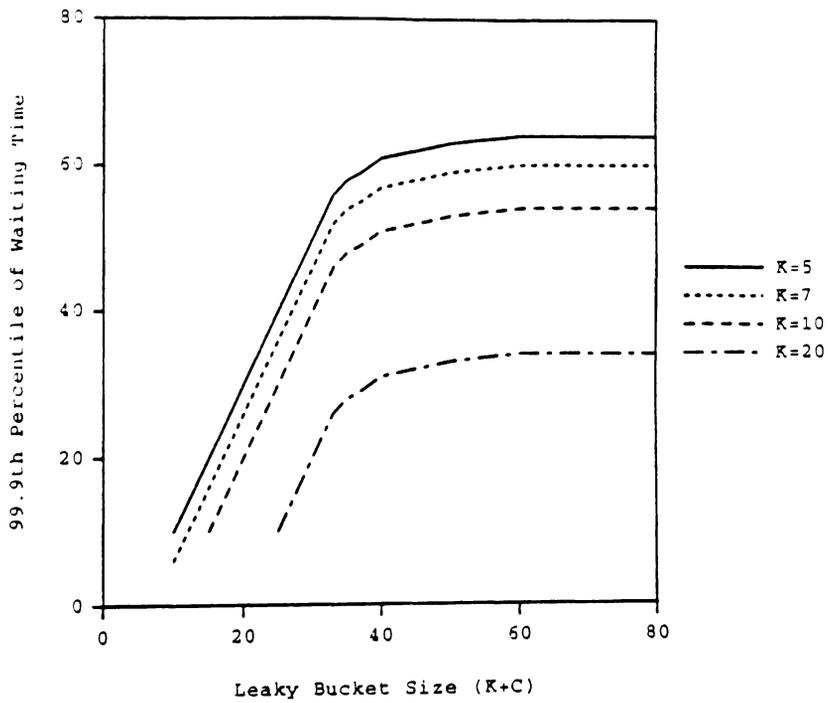


Figure 12: Maximum value of 99.9% delay time, $N = 2$

4.3 Waiting time, fixed leaky bucket size

Figure 13 shows the mean waiting time of cells for a leaky bucket size of $(K + C) = 300$ and $N = 8$, using a Markov modulated arrival process with arrival rate $\rho = 0.1$, and an average burst length $E[B]$ varying from 5 to 20. Assuming that the burst and silence periods are geometrically distributed, the traffic source can be completely characterized by the average burst length $E[B]$ and the mean arrival rate ρ .

As the token pool size increases, the mean waiting time becomes less dependent on the average burst length. However for this particular example, when the token pool size is smaller than 100, the average burst length has a substantial impact on the waiting time. Figure 14 shows similar results when the token interarrival time is set to $N = 2$ and the leaky bucket size is 60.

Figures 15 and 16 illustrate the maximum value of the 99.9% delay time for leaky buckets of size $K + C = 300$ and 60, respectively. For an average burst length of $E[B] = 20$, the delay time is roughly equal to $C \cdot N$. For smaller average burst lengths, the function $C \cdot N$ would tend to greatly overestimate the delay time.

Finally, shown in table 4 are the cell loss probabilities for each of the four traffic sources used in figures 13 through 16. From this table, it can be seen that the traffic sources which have a larger average burst length produce greater loss probabilities, and consequently

larger values of the maximum delay time, since the cell queue is more congested with cells. The mean waiting time is not as affected by the average burst length if the token pool size is sufficiently large (i.e. $K > 200$).

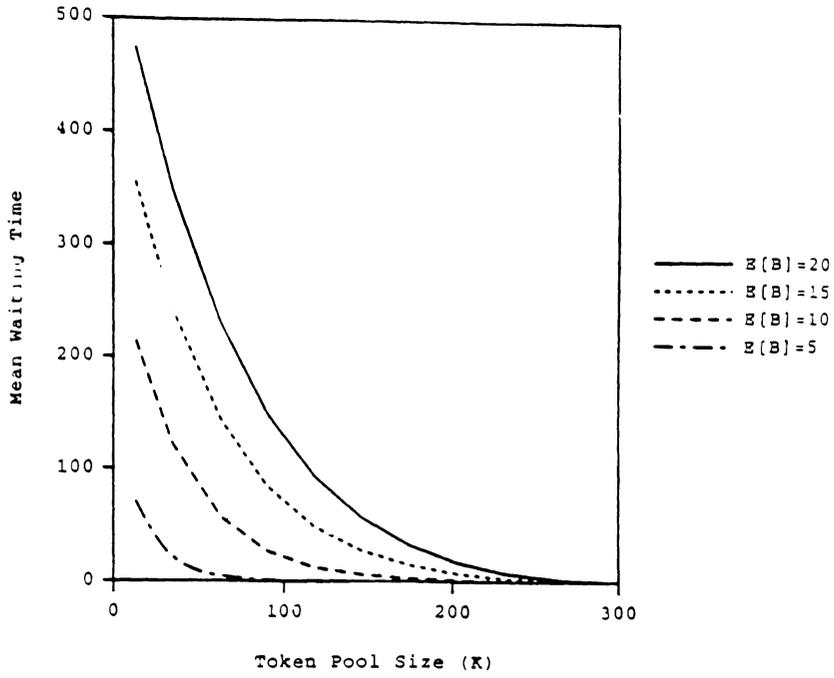


Figure 13: Mean Waiting Time, $\rho = 0.1$, $N = 8$, $K + C = 300$

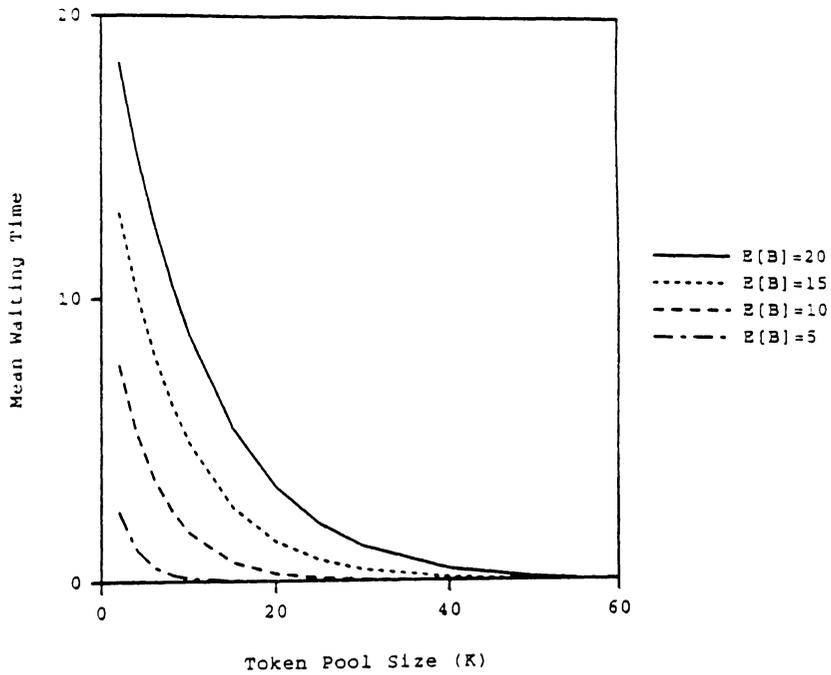


Figure 14: Mean Waiting Time, $\rho = 0.1$, $N = 2$, $K + C = 60$

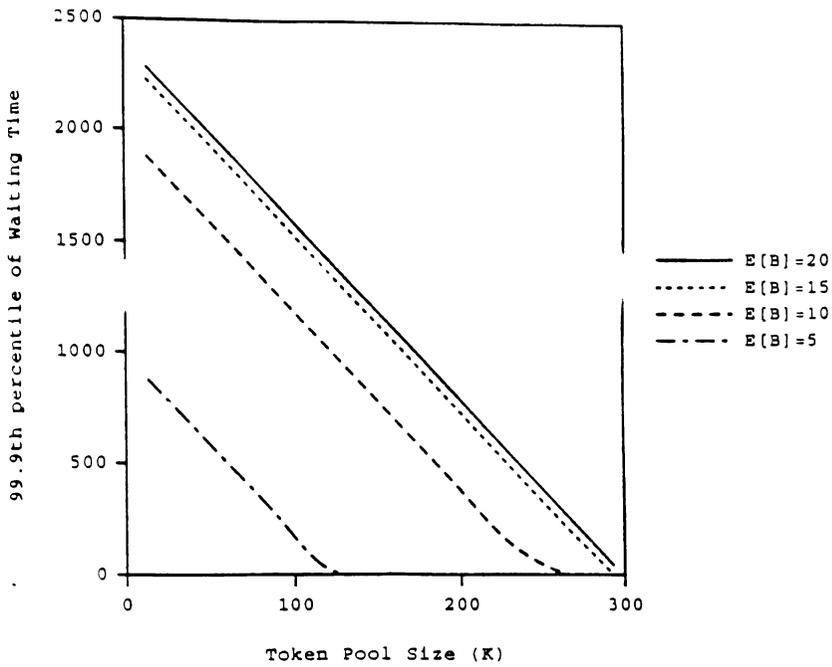


Figure 15: Maximum value of 99.9% delay time, $\rho = 0.1$, $N = 8$, $K + C = 300$

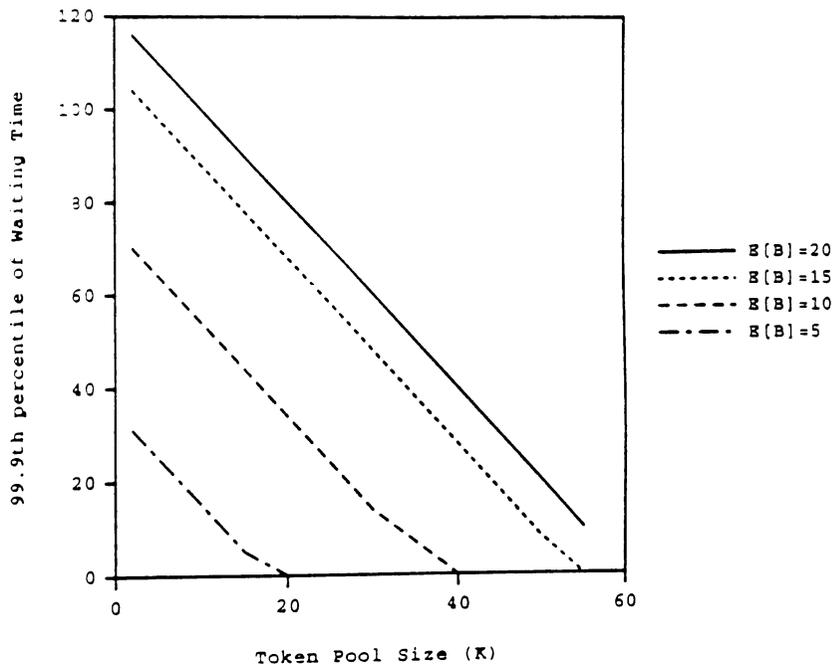


Figure 16: Maximum value of 99.9% delay time, $\rho = 0.1$, $N = 2$, $K + C = 60$

Average Burst Length, $E[B]$	Cell Loss Probability	
	$N = 8, K + C = 300$	$N = 2, K + C = 60$
5	7.54×10^{-9}	1.54×10^{-11}
10	6.25×10^{-5}	5.49×10^{-6}
15	1.01×10^{-3}	2.75×10^{-4}
20	3.97×10^{-3}	1.84×10^{-3}

Table 4: Cell Loss Probability for Markov modulated arrival sources

5 Summary and conclusions

We have provided an approximate analysis of the leaky bucket policing mechanism, using an on/off traffic source with arbitrary distributions for the length of the burst and silence periods. The model shows good agreement with a simulation for predicting the cell loss probability and the mean waiting time.

Our results indicate that the token pool size and the token generation rate play a significant role in determining the mean waiting time. The size of the cell queue and the average burst length do not significantly affect the mean waiting time if the token queue is sufficiently large.

However, even for relatively small loss probabilities (10^{-5}), the maximum value of the 99.9% delay time can be proportional to the product of the cell queue size and the token

interarrival time. In some instances, the percentile of the cell delay can be more than an order of magnitude larger than the mean waiting time. At substantially smaller loss probabilities (10^{-9}), the percentile delay introduced by the cell queue can be held negligible if the token pool size is sufficiently large.

References

- [1] H. Ahmadi, R. Guérin, and K. Sohraby. Analysis of leaky bucket access control mechanism with batch arrival process. In *IEEE Globecom*, pages 344–349, 1990.
- [2] J.J. Bae and T. Suda. Survey of traffic control schemes and protocols in ATM networks. *Proceedings of the IEEE*, 79(2):170–189, February 1991.
- [3] K. Bala, I. Cidon, and K. Sohraby. Congestion control for high speed packet switched networks. In *IEEE Infocom*, pages 520–526, 1990.
- [4] A.W. Berger. Performance analysis of a rate-control throttle where tokens and jobs queue. *IEEE JSAC*, 9(2):165–170, February 1991.
- [5] A.W. Berger, A.E. Eckberg, T. Hou, and D.M. Lucantoni. Performance characterizations of traffic monitoring, and associated control, mechanisms for broadband “packet” networks. In *IEEE Globecom*, pages 350–354, 1990.

- [6] M. Butto', E. Cavallero, and A. Tonietti. Effectiveness of the leaky bucket policing mechanism in ATM networks. *IEEE JSAC*, 9(3):335–342, April 1991.
- [7] A.E. Eckberg, D.T. Luan, and D.M. Lucantonio. Bandwidth management: A congestion control strategy for broadband packet networks – characterizing the throughput-burstiness filter. In *Int. Teletraffic Cong. Specialist Sem.*, Adelaide, Australia, September 1989.
- [8] A.E. Eckberg, D.T. Luan, and D.M. Lucantonio. Meeting the challenge: congestion and flow control strategies for broadband information transport. In *IEEE Globecom*, pages 1769–1773, 1989.
- [9] D.S. Holsinger and H.G. Perros. Performance analysis of leaky bucket policing mechanisms. Technical report, Department of Computer Science, North Carolina State University, 1991.
- [10] J.A.S. Monteiro, M. Gerla, and L. Fratta. Input rate control for ATM networks. In *ITC-13*, pages 117–122, 1991.
- [11] M. Murata, Y. Ohba, and H. Miyahara. Analysis of flow enforcement algorithm for bursty traffic in ATM networks. In *IEEE Infocom*, pages 2453–2462, 1992.
- [12] E.P. Rathgeb. Modeling and performance comparison of policing mechanisms for ATM networks. *IEEE JSAC*, 9(3):325–334, April 1991.

- 13] M. Sidi, W. Liu, I. Cidon, and I. Gopal. Congestion control through input rate regulation. In *IEEE Globecom*, pages 49.2.1–49.2.5, 1989.
- 14] K. Sohrabv and M. Sidi. On the performance of bursty and correlated sources subject to leaky bucket rate-based access control schemes. In *IEEE Globecom*, pages 426–434, 1990.
- 15] J.S. Turner. New directions in communications (or which way to the information age?). *IEEE Communications Magazine*, 24(10):8–15, October 1986.
- 16] G.M. Woodruff and Rungroj Kositpaiboon. Multimedia traffic management principles for guaranteed ATM network performance. *IEEE JSAC*, 8(3):437–446, April 1990.
- 17] G.M. Woodruff, R.G.H. Rogers, and P.S. Richards. A congestion control framework for high-speed integrated packetized transport. In *IEEE Globecom*, pages 7.1.1–7.1.5, 1988.