Performance Characteristics of a Packet-Based Leaky-Bucket Algorithm for ATM Networks*

Toshihisa OZAWA
Department of Business Administration, Komazawa University
1-23-1 Komazawa, Setagaya-ku, Tokyo 154-8525, Japan
E-mail: toshi@komazawa-u.ac.jp

1999.10.19

Abstract

A packet-based leaky-bucket algorithm functions like the early packet discard (EPD), and accepts a newly arriving packet if the probability that all the cells of the packet are accepted is high. We derive some performance characteristics of the cell and packet arrival processes that are accepted by the leaky-bucket algorithm. From these analyses, a method to determine the values of the parameters of the leaky-bucket algorithm and certain relations between this leaky-bucket algorithm and the generic cell rate algorithm (GCRA) are obtained.

Keywords: ATM network, packet communication network, leaky bucket, queueing model, performance evaluation

1 Introduction

Asynchronous Transfer Mode (ATM) is one of the fundamental technologies used to construct computer communication networks. Almost all computer communications now use Internet Protocol (IP) as the network layer protocol, where user data is divided into some packets called IP datagrams. In computer communications achieved over ATM networks, such a packet is further divided into cells of 53 bytes in data length including the headers at a router or a host, which terminates ATM virtual circuits (VCs). The packet is transmitted by cell-by-cell transmission in the ATM networks. Some performance problems may arise here because of the disagreement between the transmission data unit of IP and that of ATM: the former is a packet and the latter is a cell. One solution to resolve these problems is to implement packet-based traffic control mechanisms into the ATM networks. A typical example of such a mechanism is the early packet discard (EPD) [1, 2]. When congestion occurs at an ATM switch, some cells may be dropped at the output buffer while other cells of the packets that contain the dropped cells will still be transmitted. These transmitted cells waste network resources because their packets cannot be assembled at the destination, and this may cause a decrease in packet throughput. In order to prevent this phenomenon from occurring, the EPD drops all the cells of a newly arriving packet when the queue length of the output buffer is greater than a given threshold.

In this paper, we deal with a packet-based leaky-bucket algorithm, which functions like the EPD, and study its characteristics from the viewpoint of performance assuming it to be used as the mechanism of usage parameter control (UPC) in ATM networks. In the next section, we introduce a generalized packet-based leaky-bucket algorithm with cell and token buffers, and provide a certain equivalency between the cell buffer and the token buffer. This equivalency allows us to use a simple algorithm with only a cell buffer in the rest of the paper. The processes of accepted cells and packets are analyzed in Sections 3 and 4. These results give a method to determine the value of the parameters of the leaky-bucket algorithm and certain relations between the packet-based leaky-bucket algorithm and the generic cell rate algorithm (GCRA) [3].

*This is a draft version of a paper that appeared in IEICE Transactions on Communications, E82-B, 1, pp. 305-308 (1999).

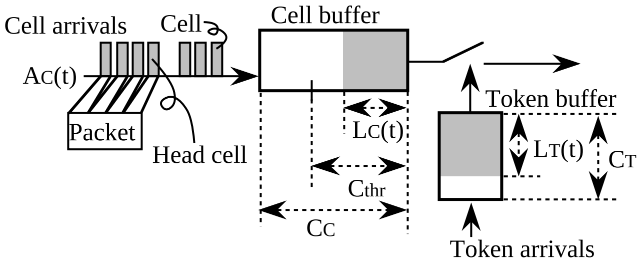

Figure 1: A packet-based leaky-bucket model.

2 Packet-based leaky-bucket algorithm

First we introduce a generalized packet-based leaky-bucket algorithm, which is represented as the queueing model with the cell and token buffers shown in Fig. 1, where tokens are generated according to a given process and a cell that can get a token departs from the cell buffer[1]. Notations are defined as follows:

- $A_C(t, s)$: The number of cells arriving during $(t, t + s]$.
- $L_C(t)$: The number of cells in the cell buffer at time t.
- C_C: The capacity of the cell buffer.
- C_{thr}: A given threshold for packet acceptance.
- $L_T(t)$: The number of tokens in the token buffer at time t.
- C_T: The capacity of the token buffer.

We assume that the first cell (or last cell) of each packet can be identified only by using cell level information. For example, this can be achieved by using the values of the payload types (PTs) of cells when ATM adaptation layer (AAL) type 5 is used [4]. A packet-based cell acceptance algorithm is defined as follows, like the EPD:

- Assume that the first cell of a packet arrives at t. If $L_C(t) < C_{thr}\,(C_{thr} > 0)$ or $L_T(t) > 0\,(C_{thr} = 0)$, then the first cell enters the cell buffer and the other cells of the packet also enter the cell buffer as long as it is not full at their arrival times.
- If $L_C(t) \geq C_{thr}$, then all the cells of the packet including the first cell are discarded (do not enter the cell buffer) even if the cell buffer is not full.

Hereafter we call a cell entering the cell buffer "an accepted cell" and a packet whose first cell is accepted "an accepted packet."

Letting $U(t)$ be defined by $U(t) = L_T(t) - L_C(t) + C_{thr}$, we can replace the condition "$L_C(t) < C_{thr}\,(C_{thr} > 0)$ or $L_T(t) > 0\,(C_{thr} = 0)$" of the packet-based cell acceptance algorithm with "$U(t) > 0$." Using the same sample path argument of Theorem 3.1 in [5], we obtain the following proposition.

Proposition 1 *Let $O_C(t)$ be the number of cells discarded or that overflowed during $(0, t]$. The sample paths of $O_C(t)$ remain unchanged if C_{thr} is changed to C'_{thr} provided that C_T is changed to $C'_T = C_T + (C_{thr} - C'_{thr})$, C_C is changed to $C'_C = C_C - (C_{thr} - C'_{thr})$, and $U(0)$ is unchanged.* □

From this proposition, we can assume that $C_T = 0$ without loss of generality for analyzing how user cells and packets are accepted by the packet-based leaky-bucket algorithm. Hence we assume this hereafter for simplicity, and abbreviate the packet-based leaky-bucket algorithm as the PLBA. We further assume the following:

[1]It is assumed that if a token is generated at the same time as the arrival of a cell then the departure of the cell that gets the token occurs before the arrival.

- Only one VC is connected with the cell buffer; the peak cell rate (PCR) of the VC is denoted by R_1.
- All the cells of each packet consecutively arrive at the cell buffer.
- Tokens are generated one by one at $1/R_0$ intervals only when there exist cells in the cell buffer, where R_0 is a guaranteed minimum cell rate.
- Let C_C be $C_C = C_{thr} + \lceil (1 - R_0/R_1)M_{max} \rceil$, where M_{max} is the maximum packet size (the number of cells) and $\lceil x \rceil$ is the smallest integer that is not less than x.

From these assumptions, all the cells of an accepted packet can successfully enter the cell buffer. Since this PLBA can be represented as a $G/D/1/C_C$ model, we define workload $W_C(t)$ as the residual time for all the cells in the cell buffer at time t to depart from it. In terms of $W_C(t)$, $L_C(t)$ is given as $L_C(t) = \lceil W_C(t)R_0 \rceil$.

In the ATM forum, this type of packet-based leaky-bucket algorithm with only a cell buffer is considered as a conformance definition of the guaranteed frame rate service (GFR), where the value of C_{the} is set to be M_{max} and that of C_C is set to be $2M_{max}$.

3 Process of accepted packets

What packet process is to be accepted by the PLBA and how the parameters of the PLBA are to be set are concerns for users. Our idea to explore them is to pay attention to a time scale T_S, which can be used for defining an empirical mean cell rate as $A_C(t, T_S)/T_S$.

Let t_j be equal to $j \cdot T_S$ and assume that $A_C(t_j, T_S) \leq \lfloor R_0 T_S \rfloor$ for every $j \geq 0$, where $\lfloor x \rfloor$ is the largest integer that is not greater than x. We here suppose that $C_{the} = \infty$ and $L_C(t) = W_C(t) = 0$ for all $t \leq 0$. Using the Lindley's formula, it can be seen that, for a given $W_C(t_j)$, the value of $W_C(t_{j+1})$ becomes the maximum when all $\lfloor R_0 T_S \rfloor$ cells arrive in $[t_{j+1} - \tau_0, t_{j+1}]$, where $\tau_0 = (\lfloor R_0 T_S \rfloor - 1)/R_1$. From this, we obtain

$$
\begin{aligned}
W_C(t_{j+1}) &\leq [W_C(t_j) - (T_S - \tau_0)]^+ + \omega_0 \\
&\leq [W_C(t_j) - \omega_0]^+ + \omega_0, \qquad (1)
\end{aligned}
$$

where $\omega_0 = \lfloor R_0 T_S \rfloor / R_0 - \tau_0$. By the mathematical induction, an upper bound of $W_C(t_j)$ is, therefore, given as $W_C(t_j) \leq \omega_0$ for every $j \geq 0$. Using a similar argument, for a given $W_C(t_j)$, an upper bound of the workload in $(t_j, t_{j+1}]$ is given as

$$
\sup_{t \in (t_j, t_{j+1}]} W_C(t) \leq W(t_j) + \omega_0. \qquad (2)
$$

An upper bound of $L_C(t)$ is, therefore, given as $L_C(t) \leq \lceil 2\omega_0 R_0 \rceil \leq L_{upp}$, where L_{upp} is given as

$$
L_{upp} = \lceil 2\{(1 - R_0/R_1)R_0 T_S + R_0/R_1\} \rceil. \qquad (3)
$$

Finally, we obtain the following result.

Proposition 2 *Let the value of C_{thr} be greater than or equal to L_{upp}. For any packet arrival process and any packet sizes, if $A_C(j \cdot T_S, T_S)$ is less than or equal to $\lfloor R_0 T_S \rfloor$ for every $j \geq 0$, all packets are accepted by the PLBA.* □

The time intervals used in Proposition 2 are the jumping-window type. Replace them with sliding-window-type intervals, i.e., $A_C(t, T_S) \leq \lfloor R_0 T_S \rfloor$ for all $t \geq 0$. Let t_j be equal to $t - (k - j)T_S$, where $k = \lceil t/T_S \rceil$. Using the same argument as that used for the jumping-window-type intervals, an upper bound of $L_C(t)$ is given as $L_C(t) = L_C(t_k) \leq \lceil \omega_0 R_0 \rceil$. Hence we obtain the following:

Corollary 1 *In Proposition 2, if $A_C(j \cdot T_S, T_S)$ is replaced with $A_C(t, T_S)$ and t moves every non-negative real number, then L_{upp} can be replaced with $L'_{upp} \equiv \lceil (1 - R_0/R_1)R_0 T_S + R_0/R_1 \rceil$.* □

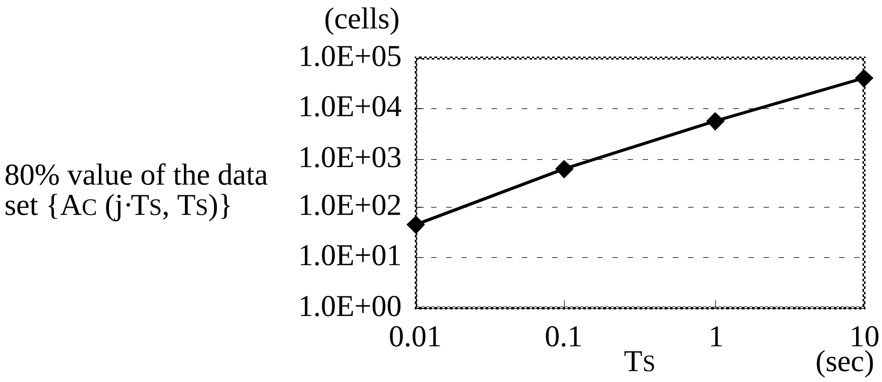

Figure 2: Relation between T_S and the 80% value of $\{A_C(j \cdot T_S,\ T_S)\}$.

Generally speaking, it is difficult for users to control packet output processes in detail, especially in cases where data flows are handled by Transmission Control Protocol (TCP). Proposition 2 asserts that such a control is not necessary in shorter time scales than T_S. It also gives a hint to determine the values of R_0, R_1, and C_{thr}; the value of T_S can be given as a time scale relevant to traffic control or traffic measurement. If the value of R_0T_S is obtained from real traffic data or some given knowledge, R_0 can be determined. As a result, the relation between C_{thr} and R_1 is obtained from the equation $C_{thr} = L_{upp}$, where L_{upp} is given by Eq. (3), and the value of R_1 is usually given in advance.

Here we show an example. Figure 2 represents the relation between T_S and the 80% value [2] of the data set $\{A_C(j \cdot T_S, T_S)\}$, which was obtained from real packet traffic data used in [6] by dividing packets into cells and adding cell headers[3]. Let the value of T_S be set at 0.1 sec (cf. an empirical mean round trip time) and that of R_1 be set at 23.6 Kcell/s ($\approx$ 10 Mbps). From Fig. 2, the value of R_0T_S is given as 571 cells, that of R_0 is given as 5.71 Kcell/s ($\approx$ 2.42 Mbps), and that of C_{thr} is given as 867 cells.

4 Process of accepted cells

What cell arrival process to pass through the PLBA is a concern for network designers. To explore this, we compare the GCRA [3] with the PLBA in terms of the upper bound of accepted cell flows. Here we use an ordinary leaky bucket algorithm with only a cell buffer as the GCRA, and assume that the capacity of the cell buffer is equal to C_C. Of course, this GCRA accepts arriving cells whenever the cell buffer is not full.

Let $D_C(t,s)$ and $D_C^{GCRA}(t,s)$ denote the number of cells accepted by the PLBA during $(t, t+s]$ and that of cells accepted by the GCRA during the same interval, respectively. Let $V_C(t,s)$ denote the cumulative decrease in workload of the PLBA during $(t, t+s]$. The relation between $W_C(t)$ and $W_C(t+s)$ is given as

$$W_C(t+s) \quad = \quad W_C(t) + D_C(t,s)/R_0 - V_C(t,s). \tag{4}$$

Since $W_C(t+s) \leq C_C/R_0$, $W_C(t) \geq 0$, and $V_C(t,s) \leq s$, we obtain the following restriction for $D_C(t,s)$.

Proposition 3

$$D_C(t,s) \quad \leq \quad sR_0 + C_C \quad \textit{for all } \ t, s \geq 0. \tag{5}$$

□

$D_C^{GCRA}(t,s)$ has the same restriction [7][4] as $D_C(t,s)$, i.e., $D_C^{GCRA}(t,s) \leq sR_0 + C_C$ for all $t, s \geq 0$.

[2] This value is given as $\inf\{x \,|\, \frac{1}{N}\sum_{j=1}^{N} 1(A_C(j \cdot T_S, T_S) \leq x) > 0.8\}$, where N is the number of intervals and $1(\cdot)$ is an indicator function.

[3] Here we assume the case where AAL type 5 is used.

[4] In the GCRA that we considered, a cell that finds no other cells in the cell buffer at its arrival also wait time of $1/R_0$ for its departure. Hence $sR_0 + C_C$ can be used as the upper bound of $D_C^{GCRA}(t,s)$, instead of $\lceil sR_0 + C_C \rceil$.

Remind that $D_C(0,t)$ and $D_C^{GCRA}(0,t)$ are the stochastic counting process that represent the process of accepted cells for the PLBA and that for the GCRA, respectively. Let Ω_C and Ω_C^{GCRA} denote the set of sample paths of $D_C(0,t)$ and that of sample paths of $D_C^{GCRA}(0,t)$. We here suppose that $\omega \in \Omega_C$. If ω is used as a cell arrival process for the GCRA, all the cells in ω are accepted by the GCRA, i.e., $\omega \in \Omega_C^{GCRA}$. This means that $\Omega_C \subset \Omega_C^{GCRA}$. Hence call acceptance controls (CACs) that are based on the worst case performance of cell arrival processes conforming to the GCRA (e.g. [8]) can be applied to cell arrival processes conforming to the PLBA.

5 Conclusions

We introduced a packet-based leaky-bucket algorithm for ATM-based packet communication networks, and characterized accepted cell and packet processes when this leaky-bucket algorithm was used as the mechanism of the UPC in the network. Our results also presented a method to determine the value of the parameters of the leaky-bucket algorithm.

References

[1] A. Romanow and S. Floyd, "Dynamics of TCP traffic over ATM networks," IEEE Journal on Selected Areas in Communications, vol. 13, no. 4, pp.633–641, May 1995.

[2] K. Kihara and T. Ozawa, "A selective cell discard scheme using packet-size information," Trans. of the IEICE, vol. J80-B-I, no. 10, pp.670–678, October 1997.

[3] ITU-T Draft Recommendation I.371, "Traffic congestion control in B-ISDN," 1993.

[4] ITU-T Recommendation I.363, "B-ISDN ATM Adaptation Layer (AAL) specification," March 1993.

[5] A. W. Berger and W. Whitt, "The impact of a job buffer in a token-bank rate-control throttle," Stochastic Models, 1992.

[6] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version), IEEE/ACM Trans. on Networking, vol. 2, no. 1, pp.1–15, February 1994. (Data is available at *http://ita.ee.lbl.gov/index.html*.)

[7] A. L. Neidhardt, F. Huebner, and A. Erramilli, "Shaping and policing of fractal traffic," IEICE Trans. Commun., vol. E81-B, no. 5, pp.858–868, May 1998.

[8] S. Shioda and H. Saito, "Connection admission control guaranteeing negotiated cell loss ratio of cell streams passing through usage parameter control," IEICE Trans. Commun., vol. E80-B, no. 3, pp.399–411, March 1997.