Correlated Token Bucket Shapers for Multiple Traffic Classes

Tsern-Huei Lee
Department of Communication Engineering
National Chiao Tung University
Hsinchu, Taiwan
ROC
e-mail:tlee@banyan.cm.nctu.edu.tw

Abstract

Traffic shaping is an important technique for quality of service guarantee in a network that provides integrated services. It is often used in an access node to regulate user traffic submitted to the network. Since the leaky bucket algorithm has been selected by international standard organizations as the traffic policing algorithm, an ideal shaper should change the temporal characteristics of user traffic to conform to the criterion. It is well known that token bucket is equivalent to leaky bucket and is often used to name shapers that shape traffic in accordance with the leaky bucket criterion. However, in standard documents, the leaky bucket algorithm was described for a single traffic class. There is no discussion for multiple traffic classes. One possible way is to use multiple independent shapers such that each traffic class is regulated by one shaper independent of other traffic classes. This approach is simple but may not be reasonable. In this paper, we propose to use multiple correlated token buckets to shape user traffic with multiple classes. In our proposed architecture, cheaper traffic classes can utilize the unused bandwidth subscribed for more expensive traffic classes. Besides, each traffic class is guaranteed a minimum bandwidth specified by its leaky bucket parameters.

1. Introduction

Quality of service (QoS) guarantee is an important function of a converged network that supports integrated data, voice and video services. In order to achieve QoS guarantee requested by different connections, a framework of QoS mechanisms including call admission control, traffic policing and service scheduling are necessary. Upon receiving a connection request, the call admission control function determines based on current network states, traffic parameters and the requested QoS level, whether or not to grant the request. Some examples of call admission control algorithms can be found in [1]-[3]. Traffic policing is a function that regulates user traffic to ensure that the submitted traffic conforms to its specified traffic parameters. The leaky bucket algorithm [4] has been adopted by some international standard organizations as the scheme to determine whether or not the traffic generated by a user is conforming to its specified parameters [5]-[7]. Finally, service scheduling is a function that decides the service order of user traffic so that different QoS levels can be achieved for different connections. References [8]-[10] are some well-known service scheduling algorithms.

In this paper, we consider the problem of traffic regulation in a packet network. For ease of description, we assume that all packets are of equal length as in asynchronous transfer mode (ATM) networks. The results can be easily generalized to variable-length packets. We further assume that the edge node of the network adopts the leaky bucket algorithm for traffic policing and all non-conforming packets are discarded.

To avoid submitting non-conforming packets to the network, a user can use a shaper to change the temporal characteristics of his/her traffic. With leaky bucket policing, such a shaper can be neatly described by token generation and consumption and

This work was supported by the Ministry of Education under the contract number 89-E-FA04-1-4.

thus is often referred to as a token bucket shaper. The leaky bucket policing algorithm for a single traffic class can be found in [5]. In this paper, we study the shaping function for a user who generates multiple traffic classes. A possible solution is to use multiple token bucket shapers so that each traffic class is regulated by a shaper independent of other classes. However, we think this is not fair to users. For example, consider the scenario that a user subscribes bandwidths w_1 and w_2 for two traffic classes, class 1 and class 2, respectively. Assume that class 1 has a more stringent QoS requirement than class 2. In other words, for the same amount of data, class 1 requires more network resources than class 2 and thus is more expensive to subscribe than class 2. From the user's point of view, he/she should be allowed to utilize the unused bandwidth subscribed for class 1 traffic to transmit class 2 data. Obviously, independent shapers do not possess the flexibility. In this paper, we propose to use correlated token bucket shapers to solve this problem.

The rest of this paper is organized as follows. In Section 2, we review the token bucket shaper in details. Our proposed correlated token bucket shapers are presented in Section 3. Section 4 contains concluding remarks.

2. The Token Bucket Shaper

Before describing the token bucket shaper, let us consider the leaky bucket policing algorithm. There are two parameters I (increment) and L (limit) in the leaky bucket algorithm, where I represents the work brought in by every packet and $I+L$ is the bucket size. Therefore, one can use the pair (I, L) to represent a leaky bucket with parameters I and L. The states maintained for each leaky bucket include *LCT* (last compliance time) and X (bucket occupancy). *LCT* is the arrival time of the last conforming packet and X is the updated bucket occupancy at *LCT*. The initial values of X and L are zero and negative infinity, respectively. An arriving packet is non-conforming if and only if the calculated new bucket occupancy is greater than L. *LCT* and X are updated only if a packet is conforming. The virtual scheduling algorithm is another way to describe the leaky bucket algorithm [5]. Figure 1 shows the virtual scheduling and the continuous-state leaky bucket algorithms provided in [5]. In the ATM Forum, a packet is called a cell and these traffic management algorithms are called generic cell rate algorithms (GCRA). Notice that an equivalent pair (σ, ρ) can be used to describe a leaky bucket, where σ and ρ denote, respectively, the bucket size and the leaky rate. If (I, L) and (σ, ρ) represent the same leaky bucket, then we have $\sigma = I+L$ and $\rho = 1/I$. In this paper, we will use both pairs interchangeably.

Let us consider the token bucket shaper now. Assume that the edge node in the network side uses a leaky bucket (I, L) to regulate some user's traffic. On the other side, the user adopts a token bucket shaper to change the temporal characteristics of his/her traffic so that every packet submitted to the network is conforming and no packet will be lost due to violation of the specified traffic parameters. For simplicity, we assume that L is an integral multiple of I. The derivation is similar if this assumption is relaxed. To achieve the goal of making all packets conforming, one can use a token pool which can store up to $L/I + 1$ tokens. Initially, the token pool is full. A packet can be submitted to the network only if the token pool is non-empty. Moreover, a token is consumed whenever a packet is granted its entrance to the network. New tokens are refilled to the token pool with a rate of one token per I units of time. It is well known that leaky bucket and token bucket are equivalent and thus with such a token bucket shaper every packet will be conforming to the traffic parameters. Figure 2 illustrates such a token bucket shaper.

3. Correlated Token Bucket Shapers

The token bucket shaper described in the last section is applicable to a single class of traffic. In this section, we consider the case when a user generates multiple classes of traffic. Assume that there are n traffic classes and thus we need n token bucket shapers. As mentioned before, a simple way for the network to regulate n classes of traffic is to use one leaky bucket for each traffic class independent of other classes. However, such a solution is not fair to network users. It is more reasonable if a user is allowed to utilize the unused bandwidth subscribed for a more expensive class to transmit data of a cheaper class. For example, consider the case when $n = 2$ and assume that class 1 is more expensive than class 2 per unit bandwidth. Let (σ_1, ρ_1) and (σ_2, ρ_2) be the leaky buckets specified for class 1 and class 2 traffic, respectively. A reasonable policing criterion is to use (σ_1, ρ_1) to regulate class 1 traffic and use $(\sigma_1+\sigma_2, \rho_1+\rho_2)$ to regulate the aggregated traffic of both classes. Generalization of the idea to more than two classes is straightforward. For n traffic classes, there are n leaky buckets and the k^{th} bucket $(\sum_{i=1}^{k}\sigma_i, \sum_{i=1}^{k}\rho_i)$ is used to regulate the aggregated traffic of class 1, class 2, …, and class k. However, under such criterion, the bandwidth subscribed for a more expensive traffic class could be totally used by cheaper classes if the user does not manage in appropriately. The purpose of the paper is to propose a shaper architecture, called the correlated token bucket shapers, so that, in addition to allowing bandwidth borrowing, every class has a minimum guaranteed bandwidth that conforms to its leaky bucket parameters.

In our proposed architecture, for each traffic class i, there is a token pool, called token pool i, that is used to guaranteed the minimum bandwidth for class i traffic. Also, there is an associated queue, called queue i, to store class i packets when token pool i is empty. The token generator that generates tokens for class i traffic is called token generator i and the tokens it generates are called class i tokens. In our proposed correlated token bucket shapers, a class i token could be placed in token pool j which satisfies $j \geq i$. For simplicity, we assume that the capacity of the link connecting the user and the network is infinite. As a consequence, queue i is always empty if token pool i is non-empty and vise versa.

Our proposed correlated token bucket shapers consist of n token generators, n token pools, and a token placement module that decides which token pool a new token should be placed in. The token generator i generates class i tokens with rate ρ_i. Assume that a new class i token is generated. The token is placed in token pool i if it is not full. In case that token pool i is full, it is placed in token pool j, where j is the smallest number greater than i such that queue i is non-empty (which implies token pool i is empty under the assumption of infinite link capacity). If all queues are empty, then place the token in token pool j, where j is the smallest number greater than i such that token pool j is not full. If token pool j is full for all $j \geq i$, then the token is lost. Figure 3 illustrates the proposed architecture for $n = 3$.

Notice that an "overflow" token is immediately usable by selecting a non-empty queue, if there is one. With the proposed correlated token bucket shapers, all packets submitted to the network are conforming to the leaky bucket policing criterion discussed above. Moreover, although the policing criterion allows, our architecture prohibits a cheaper class from using the tokens in the token pools of more expensive classes. In other words, a token bucket is dedicated to each traffic class so that class i traffic is guaranteed a minimum bandwidth that satisfies (σ_i, ρ_i) specification for all i. Another point that is worth mentioning is that although a class j packet may consume a class i ($i < j$) token, it is not "upgraded" to class i so that out of order delivery can be avoided.

4. Concluding Remarks

Let us consider implementation of the proposed correlated token bucket shapers. For usual access nodes, there should be only a few traffic classes. Therefore, dedicating one timer for each token generator should be a cost effective implementation. As for traffic policing, one possible solution is to simulate the behavior of the correlated token bucket shapers. However, since the number of users monitored by an edge node of the network could be large, we suggest to modify the continuous-state leaky bucket algorithm for traffic policing. For a user with n traffic classes, the edge node maintains n leaky buckets and the k^{th} leaky bucket is used to regulate the aggregated traffic of class 1, class 2, …, and class k. When a class i packet arrives, the edge node checks leaky buckets i, i+1, …, and n. The packet is conforming only if it does not violate any of the checked leaky buckets. The last compliance time and bucket occupancy of all the checked leaky buckets are updated if a packet is conforming.

In our analysis, we assumed that the capacity of the link between user and network is infinite. The result is similar when the link capacity is finite as in all real networks. For finite link capacity, the token placement module can be modified so that, when token pool i is full, a new class i token is placed in token pool j, where j is the smallest number greater than i such that token pool j is not full and the number of tokens in token pool j is less than the number of packets in queue j. If no such j exists, then the token is placed in token pool k, where k is the smallest number greater than i such that token pool k is not full. If token pool j is full for all $j > i$, then the token is lost. The modification, however, does not affect the design of traffic policing. Other variations of the token placement module can be further investigated.

Another assumption made in this paper is that all packets are of equal length. This assumption can be relaxed if tokens and packets are calculated in some data unit, say bytes. The granularity affects precision of traffic regulation and implementation complexity.

We believe that users should be allowed to transmit packets of cheaper classes with the unused bandwidth subscribed for more expensive classes. Therefore, the correlated token bucket shapers presented in this paper should be useful. The tradeoff is increased complexity in traffic policing. Instead of checking one leaky bucket, the edge node of the network has to check up to n leaky buckets when a packet arrives. Since the value of n is expected to be small for every user, the increased complexity should be acceptable.

References

[1] R. Gibbens, F. Kelly, and P. Key, "A decision-theoretic approach to call admission control in ATM networks," IEEE J. Sel. Areas Commun., vol. 13, pp. 1101-1114, Aug. 1995.

[2] T. H. Lee, K. C. Lai, and S. T. Duann, "Design of real-time call admission controller for ATM networks," *IEEE/ACM Trans. Networking,* vol. 4, no. 5, pp. 758-765, 1996.

[3] D. Nandita, J. Kuri, and H. S. Jamadagni, "Optimal call admission control in generalized processor sharing (GPS) schedulers," *IEEE Infocom*, April 2001.

[4] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 3, pp. 325-334, April 1991.

[5] Traffic management specification, version 4.0, ATM Forum/95-0013R8, Oct. 1995.

[6] J. Heinanen and R. Guerin, "A single rate three color marker," ietf, rfc 2697, Sept. 1999.

[7] T. Finland and R. Guerin, "A two rate three color marker," ietf, rfc 2698, Sept. 1999.

[8] L. Zhang, "Virtual clock: A new traffic control algorithm for packet switching networks," *ACM Tran. Computer Systems*, May 1991.

[9] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," IEEE/ACM Trans. Networking, vol. 1, no. 3, pp. 344-357, June1993.

[10] J. C. R. Bennett and H. Zhang, "WF^{2}Q: Worst-case fair weighted fair queueing," *IEEE Infocom*, March 1996.

[11] T. H. Lee, Policing and shaping devices for multiple traffic classes, Patent pending.

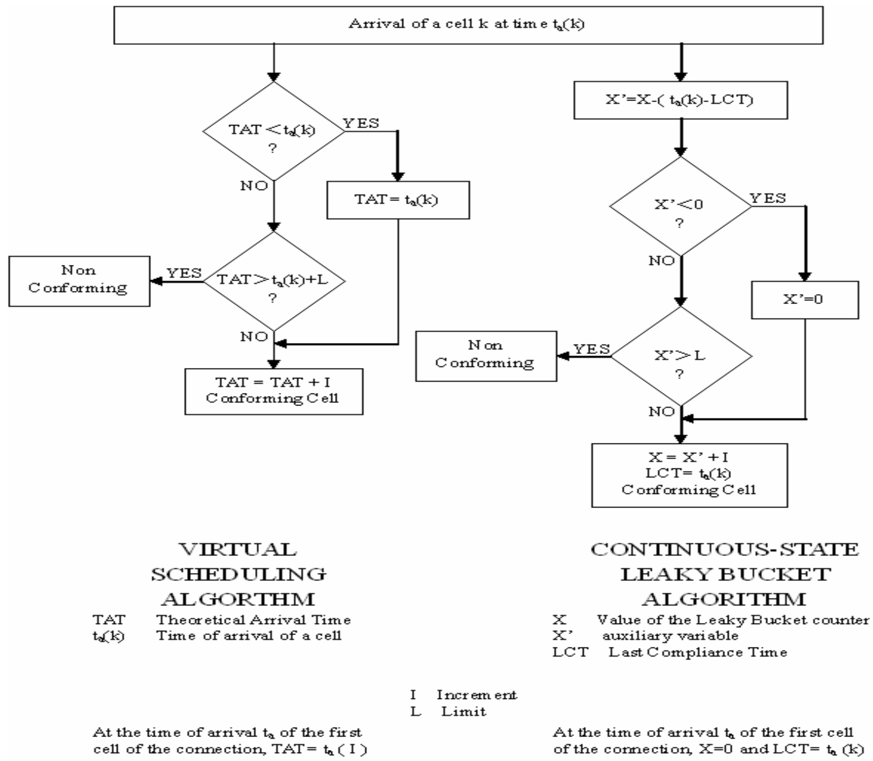

Figure 1. Equivalent versions of the Generic Cell Rate Algorithm.

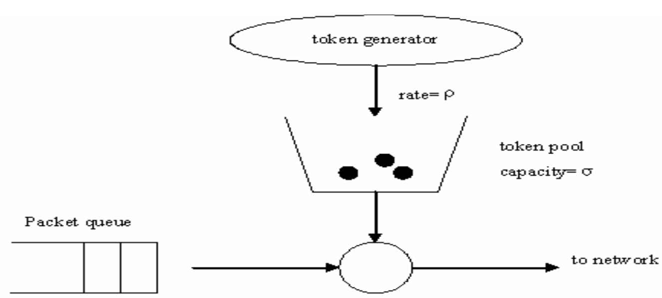

Figure 2. A token bucket shaper.

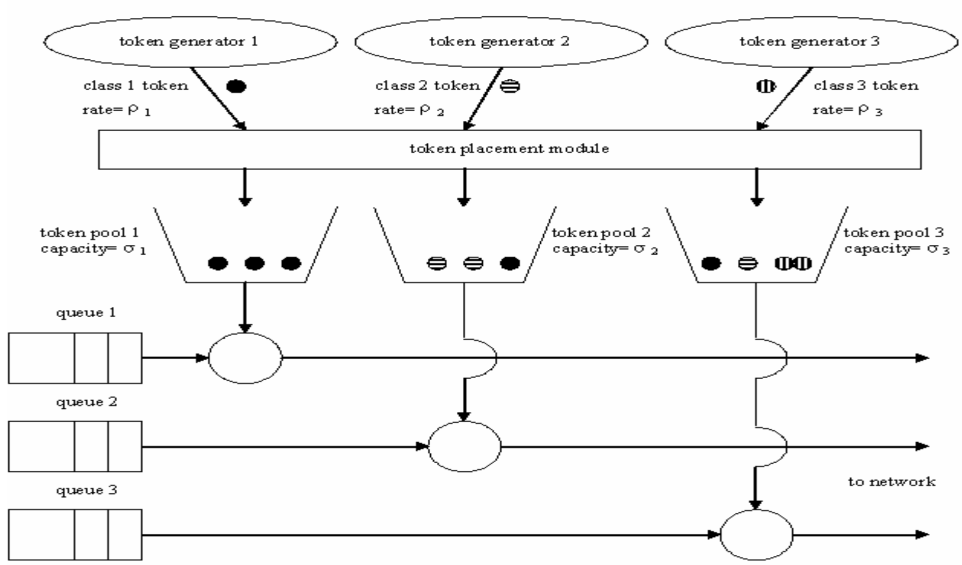

Figure 3. A correlated token bucket shapers for $n = 3$.