The Token-Bank Leaky Bucket Mechanism for Group Connections in ATM Networks*

Sheng-Lin Wu and Wen-Shyen E. Chen†
Institute of Computer Science
National Chung-Hsing University
Taichung, Taiwan ROC
email: echen@cs.nchu.edu.tw

Abstract

A well accepted policing mechanism for ATM networks is the leaky bucket mechanism. The original leaky bucket has been shown to be effective for traffic with a constant bit rate, but reacts poorly to bursty traffic. Although there are some other policing mechanisms in the literature that provide improvements over the original leaky bucket, most of them enforce the negotiated parameters for individual source traffic and do not fully utilize the statistical multiplexing of multiple connections. As a result, the bandwidth utilization is low. Even though some source policing mechanisms have been proposed for group connections, they either fail to explore statistical multiplexing fully or do not provide a scheme to protect well-behaved sources from malicious ones in the group.

In this paper, we propose a new source policing mechanism, called the Token-Bank Leaky Bucket, for group connections in ATM networks. The mechanism explores the statistical multiplexing of multiple connections in the group and allows the unused bandwidth to be effectively shared by the connections that need it. In addition, it sets a limit of excessive data cells a source can send in a cycle in order to protect well-behaved sources against malicious sources. It is shown by simulations to have a lower violation probability and better bandwidth utilization even for small group connections when compared to the original leaky bucket with a similar configuration.

Keywords: Leaky Bucket, ATM Networks, B-ISDN, policing, Token-Bank Leaky Bucket

1 Introduction

Today, various real-time multimedia data, including voice, image, video, are widely used in computer applications. These multimedia data can be transmitted between computers in time only when the network can provide large bandwidth. The traditional networks that provide less than hundred megabits do not seem enough to serve today's applications adequately. ATM (Asynchronous Transfer Mode) [1] has been selected to be the transport technique for Broadband Integrated Services Digital Network (B-ISDN). It is available at various speeds from hundreds of Megabits to Gigabits. It is connection-oriented and provides assurance of Quality of Service (QoS), such as cell loss probability and delay, for the connections. To set up a connection, a source specifies its traffic parameters and negotiates with the underlying ATM network. The traffic parameters consist of three primitives: Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) and Maximum Burst Size (MBS). The PCR specifies the upper bound on the traffic that can be submitted on the connection; the SCR describes the upper bound on the conforming average rate of the connection; and the MBS determines the maximum number of cells that can be submitted back-to-back with the PCR.

During connection setup, a Call Admission Control (CAC) procedure is invoked by the network to verify if the traffic parameters of the new connection can be accepted without degrading the services provided to the existing connections. Once the new connection is accepted and established, the source should follow the negotiated traffic parameters. However, it is possible that a malicious source sends data traffic more than the negotiated traffic parameters, disturbing the services of other connections. Therefore, one of the fundamental network control issues is how to effectively police sources. An ideal policing mechanism should mark or drop data cells generated in excess of the negotiated value and should result in a low violation probability for well-behaved sources.

Several policing mechanisms have been proposed.

*This research was supported in part by the National Science Council, the Republic of China, under contract number NSC85-2221-E-005-002.

†Corresponding author

One of the well-known policing mechanisms is the leaky bucket mechanism as depicted in Fig. 1 [2, 3, 4, 6]. The leaky bucket consists of a token buffer with a predetermined maximum buffer size and the tokens are generated with a specific token generating rate. A data cell can pass through the leaky bucket only if it can get a token from token buffer. The original leaky bucket has been shown to be effective for traffic with a constant bit rate, but reacts poorly to bursty traffic [3, 4].

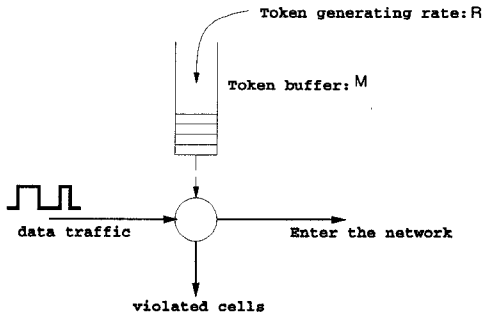

Figure 1: The Leaky Bucket mechanism

There are some other policing mechanisms introduced in the literature to improve the original leaky bucket mechanism. The multi-level leaky bucket mechanism described in [5] uses multiple leaky rates to mimic the source behavior to reduce the violation probability. The (L,M,T) mechanism in [6] monitors the average rate from a few previous bursts to control the minimum distance that is allowed between two adjacent bursts.

A stringent policing mechanism enforces the negotiated parameters for individual source traffic. However, such a restriction degrades the statistical multiplexing of multiple connections. The statistical multiplexing on group connections can enhance the utilization of the bandwidth, especially in the presence of bursty traffic. For example, some policing mechanisms are intended for group connections instead of a single connection [7]. These group-based policing mechanisms introduce some strategies, such as dynamic bandwidth allocation, to improve the statistical multiplexing. To take advantage of statistical multiplexing of group connections, the cooperating leaky bucket mechanism [7] works exactly the same as the original leaky bucket mechanism, except when there exists one or more connections which are silent, it distributes the unused leaky rates to other buckets. However, some connections might obtain extra leaky rate even when they do not need it. In addition, the dynamic bandwidth allocation used in this mechanism requires an accurate traffic prediction for bandwidth distribution. To accurately estimate the traffic usage before actual transmission is difficult. Consequently, over-estimation is often used to ensure the QoS of connections. This in turn reduces the utilization of the bandwidth.

In this paper, we propose a new policing mechanism for group connections, called Token-Bank Leaky Bucket (TBLB). It consists of a token buffer of size one and a counter for each connection in the group, and a shared token pool for the connection group as a whole. Each connection can store ("credit") tokens that exceed its buffer capacity to the shared token pool (the Bank). If a connection needs more tokens to send data cells, it can request ("debit") tokens from the shared token pool, if available. A priority rule is used to determine which data cells can get the tokens. In addition, the TBLB mechanism has a "credit limit" for each connection to prevent malicious sources from debiting tokens well beyond their negotiated limits. Because the connections are able to send extra data cells by debiting tokens from the shared token pool even when their individual token buffers depletes, the overall utilization of bandwidth can be improved, and violation probability reduced.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to the original leaky bucket mechanism. The details of the TBLB mechanism are described in Section 3. Simulation results of the TBLB mechanism are presented in Section 4. The conclusion remarks are given in Section 5.

2 The Leaky Bucket Mechanism

The original leaky bucket mechanism [2, 3, 4, 6] is the most widely accepted mechanism for the source policing [7] in the ATM network. As depicted in Fig. 1, the original leaky bucket mechanism has a token buffer with capacity of M tokens. The tokens will be generated with a specific rate R, determined when the connection is established. To enter the network, a data cell have to get one token from the token buffer. If the token buffer becomes empty, the arriving data cell will be deemed as a *violation* to the traffic contract. Depending on the policing mechanism used, the violated data may be discarded or enter the network with a low Cell Loss Priority (CLP=1) [2].

The leaky bucket polices the traffic with the token generating rate R and the size of the token buffer M. The parameter R describes the SCR of the connection. Since each conforming data cell obtains a token from the token buffer, the total number of conforming cells entering the network is no more than the tokens which are generated in the time interval. Therefore, the SCR

will be limited to the token generating rate R. The parameter M is used to determine the MBS, which represents the peak rate duration. As shown in [6], the MBS will be limited to:

$$\text{MBS} = \lceil \frac{\text{MBS} - 1}{\text{PCR}} \cdot \text{R} + \text{M} \rceil \qquad (1)$$

The connection controlled by the original leaky bucket cannot send data with a rate faster than R, which will not change before the connection is terminated. If the the source sends traffic with a constant bit rate and does not fluctuate during transmission, R can be set as the constant bit rate. In this case, when a data cell passes through the leaky bucket, there is a corresponding token generated for that cell because they operate at the same rate. However, if the traffic is bursty in nature, the source might take away tokens with a peak rate at bursty duration; and if the peak rate is higher than R, the excessive data cells will be marked as violations after the token buffer depletes. From Eq. (1), there are two possible approaches to reducing the violation probability. One is to increase the size of the token buffer M; the lager token buffer can store more tokens to reduce the violation of data cells. However, large token buffer increases the MBS, which is undesirable. The other is to increase the token generating rate R; this will admit more data cells to the network. Often time, to limit the violation probability, R is set close to the transmitting peak rate [3]. However, most of the time, the source sends data with a rate less than the peak rate. Setting R close to the peak rate means that the network should allocate more bandwidth for the connections. As a result, the bandwidth utilization is reduced.

We use Fig. 2 to illustrate the relationships of the violation probability, token generating rate R and token buffer size M. In Fig. 2, the whole shaded area (A+B+C) is the traffic generated at a source. Assume that initially, the token buffer is empty and the token generating rate R is set to be less than the peak rate. In this case, area D represents the token not initially used and are stored in the token buffer. Area A is below the token-generating rate and therefore the cells in this area are conforming cells. Area B is above the token generating rate, and therefore consumes the tokens stored in the token buffer. Area C represents the violated cells after the token buffer depletes. Area E again represents the tokens not used by the data cells. To reduce the area C, there are two solutions: The first is to increase the token generating rate so that the area A (for the conforming data) increases and the area C reduces accordingly. The second is to increase the token buffer size so that the maximum size of area B is larger, reducing the area C; Smaller area C will result in smaller violation probability.

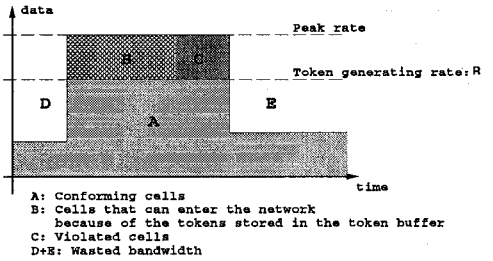

Figure 2: The traffic diagram using the leaky bucket mechanism

For a group of connections, if each connection employs only its original leaky bucket mechanism, the tokens cannot be shared among the connections in the same group. Because of traffic fluctuation, it is possible that one connection lacks the tokens to send data, but the token buffers of other connections are full and the incoming tokens are tossed away. Take Fig. 3 for example, each connection is bounded by a fixed bandwidth restriction and the white-space area in the figure stands for the unused bandwidth. We should be able to take advantage of the unused bandwidth to improve the overall bandwidth utilization.

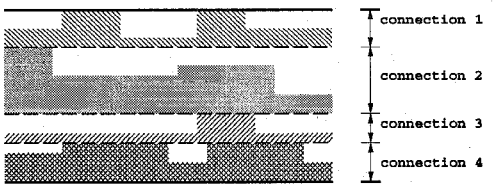

Figure 3: The group connections using the leaky bucket mechanism

3 The Token-Bank Leaky Bucket Mechanism

As discussed in Section 2, the original leaky bucket mechanism needs to set its token generating rate close to the peak rate to obtain the acceptable violation probability. The mechanism will result in lower bandwidth utilization. Based on the observations in Section 2, we propose to use a new mechanism, called the Token-Bank Leaky Bucket (TBLB) mechanism, to improve bandwidth utilization and reduce violation probability for group connections.

3.1 The Architecture of the Token-Bank Leaky Bucket

As illustrated in Fig. 4, the mechanism consists of a token buffer of size one and a counter E_i for each

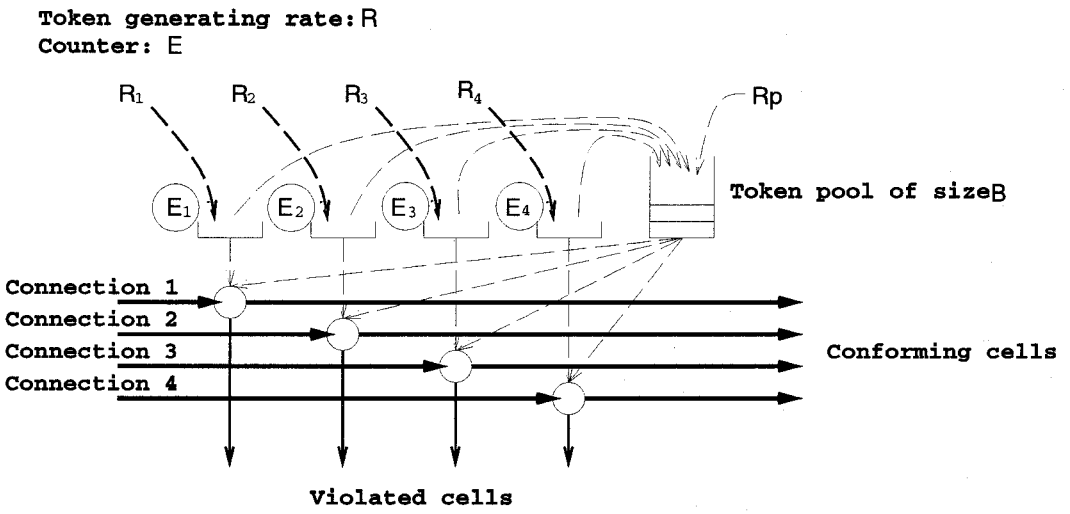

Figure 4: The Token-Bank Leaky Bucket mechanism

connection i in the group, and a token pool with size B shared by all the connections in the group. The counter E_i is used to keep track of the "extra" number of the tokens that the connection i has obtained from the shared token pool in the current cycle. (It records the current "debt" of the connection i. Note that negative E_i represents "credit".) When the connection i stores a token in the shared token pool, the E_i is decremented by one; and when the connection i obtains a token from the token pool, the E_i is incremented by one. The TBLB uses tokens to control data rate. The total number of the data cells of all connections in the same group will be limited to be less than or equal to the tokens which the TBLB generates. Each connection has an original leaky bucket with the token buffer size of one as the policing mechanism. Each leaky bucket generates tokens with a separate token generating rate R_i. Since the size of the individual token buffer is one, the TBLB stores excessive tokens generated for individual connections in the shared token pool, in contrast to the original leaky bucket where the excessive tokens are tossed away.

If a connection i sends data with a rate less than or equal to the corresponding token generating rate R_i, the data cells can always get their tokens from its individual token buffer. However, if source i send data with a rate faster than R_i, it is possible that there is a data cell arriving with no token in the corresponding token buffer. The connection i can then request tokens from the shared token pool if the pool is not empty and the E_i is less than the credit limit L_i.

3.2 Protection against Malicious Sources

The credit limit L_i is the upper bound of the "debt" allowed for the connection i. The connection i can obtain tokens only if the E_i is less than the L_i. Since the source i has a corresponding token generating rate R_i to generate enough tokens for data cells, for a well-behaved source i, E_i should be close to zero after a a long interval. The L_i is used to limit the maximum "debt" the connection i can have to ensure that other connections will not be disturbed. With a credit limit, no connection can obtain extra tokens indefinitely without putting back tokens in the token pool.

3.3 Priority Rules for Fairness

In order to achieve fairness among the group connections, when the tokens in the shared token pool is less than the total tokens the connection group actually need, the TBLB mechanism uses a priority rule to determine which connections can get the tokens. The priority for connection i is determined at the beginning of each cycle time T by the values of the counter E_i and the token generating rate R_i. We choose to determine the priority for each connection at the beginning of the cycle time T because: 1) calculating the priorities at each cell time will be computation-intensive and 2) if the priority is set for the connection duration, it would be inflexible and unfair. At the beginning of the cycle time T, the TBLB mechanism calculates the fraction of $\frac{E_i}{R_i}$ and sets the priority for connection i accordingly. Note that a smaller $\frac{E}{R}$ ratio represents higher priority. As a result, the total bandwidth can be distributed to all connections fairly.

To reduce the violation probability of all the con-

nections in the same group, the TBLB mechanism increases optional Rp for the shared token pool so that the violation probability can be reduced evenly among all connections.

3.4 Freedom from Starvation

In the TBLB mechanism, there is an individual token buffer for each connection i and tokens will be generated and deposited into the token buffer with rate R_i. The tokens stored in the individual token buffer is to be used by the corresponding connection only when the R_i is greater than the data cell rate of connection i and the excessive token will be stored in the shared token bank. Therefore, the TBLB mechanism can ensure that each connection can send data with a rate no less than the individual token generating rate R_i; as a result, there will be no starvation for any connection.

3.5 Configuring the TBLB

To ensure that the connection i can use data no less than R_i, R_i is set as the SCR of the connection. The PCR of the group connections can be set as the sum of the PCRs of all the connections, and the SCR can be set as the sum of the SCRs of all the connections. With the predetermined MBS of the combined traffic, if there are n connections in the same group, the relationship between the maximum value B of the shared token pool size and the MBS is as follows:

$$\mathrm{MBS} = \lceil \frac{\mathrm{MBS}-1}{\sum_{i=1}^{n}(\mathrm{PCR}_i - \mathsf{R}_i)} \cdot (\sum_{i=1}^{n} \mathsf{R}_i + \mathsf{Rp}) + \mathsf{B} \rceil \quad (2)$$

The size of the shared token buffer B for the group connections is configured by the user specified MBS. The extra token generating rate Rp is an optional parameter. If the violation probability of the combined traffic is too high, the Rp can be incremented to reduce violation probability.

The credit limit L_i for connection i is determined by the token generating rate R_i and the specific cycle C_i of the connection i. A traffic source can be specified to transmit data with a set of bit rates within a window of time interval. This interval windows is called *cycle* [5]. For example, an HDTV source transmits on average at a high rate (135Mbps) of 0.1 second and a low rate (35Mbps) of 2.0 seconds out of a 2.1 seconds interval [5]. Therefore 2.1 seconds is the cycle. The credit limit L_i can be calculated as:

$$L_i = R_i \times C_i$$

4 Simulation Results

One of the traffic characteristics of the multimedia traffic is burstiness. In our simulation, the discrete-time Interrupted Poisson Process (IPP) [8] is used to model a bursty traffic source. While the IPP was originally introduced in a continuous-time domain to model overflow traffic in telephone trunks, it also describes bursty traffic in ATM networks quite nicely.

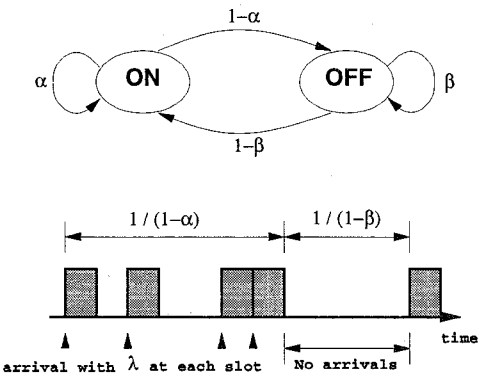

Figure 5: The discrete-time Interrupted Poisson process

As shown in Fig. 5, the IPP changes from the ON state to the OFF state with the probability $1-\alpha$ per cell time, and changes the OFF state to the ON state with the probability $1-\beta$ per cell time. At the ON state, the source sends data cells with a rate λ per cell time, and at the OFF state, there are no data cells generated. The length of the duration that the IPP will stay in the ON [OFF] state is geometrically distributed with the mean $1/(1-\alpha)$ $[1/(1-\beta)]$ and the variance $(1-\alpha)/\alpha^2$ $[(1-\beta)/\beta^2]$. The average number of cells per slot, ϕ, is given in Eq. (3):

$$\phi = \frac{\frac{\lambda}{(1-\alpha)}}{\frac{1}{(1-\alpha)} + \frac{1}{(1-\beta)}} \quad (3)$$

In our simulation, we assume an IPP traffic source $\mathcal{S}$ with parameters $\alpha = 0.9975$, $\beta = 0.99875$, and $\lambda = 1.0$. With these parameters defined, the traffic source will produce an ON and OFF durations with average lengths 400 and 800 respectively. The length of the ON and OFF durations are measured with time unit being one cell slot time.

One of the major factors which will influence the performance of the LB and TBLB mechanisms is the burstiness of the traffic behavior. The burstiness, ρ, is the peak-to-average ration. It implies the range of

fluctuation of the traffic and can be defined as follows:

$$\rho = \frac{\text{Peak Rate}}{\text{Average Rate}} \tag{4}$$

A traffic source with a large ρ means that its peak rate is much higher than the average rate. As discussed in Section 2, the original LB mechanism performs better when the source sends data with a rate closer to a constant. In other words, the LB mechanism is better suited for traffic sources with a small ρ. For the IPP model, the peak rate υ is defined as:

$$\upsilon = \frac{\frac{\lambda}{(1-\alpha)}}{\frac{1}{(1-\alpha)}} = \lambda \tag{5}$$

and ρ can be represented as:

$$\rho = \frac{\upsilon}{\phi} = \frac{(1-\alpha)+(1-\beta)}{(1-\beta)} \tag{6}$$

As shown in Eq. (6), in the IPP model, ρ is only a function of parameter α and β. For simplification, we assume $\lambda = 1.0$.

Ideally, a source policing mechanism should protect the well-behaved sources against the malicious traffic source. In addition, the policing mechanism should not cause the conforming cell to be marked as violation. With the same traffic parameters, a better policing mechanism should result in lower violation probability. Therefore, we can use violation probability as a measure for the effectiveness of the source policing mechanisms. We define VP_i as the violation probability for source i as follows:

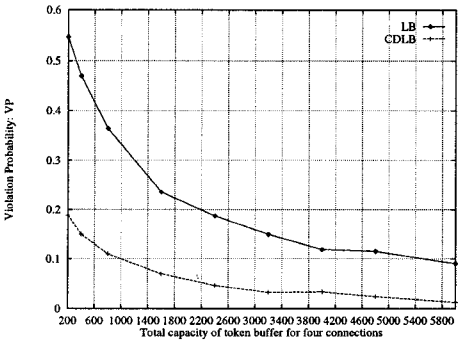

Figure 6: VP v.s. Token buffer size

$$\mathsf{VP}_i = \frac{\#\text{ of violated cells from source } i}{\text{Total }\#\text{ of cells arrive at source } i} \tag{7}$$

First, we examine the VP of the traffic source $\mathcal{S}$ as it is policed by the LB and TBLB mechanisms while varying the size of the token buffer M. In this simulation, the number of connections in the group is four. In Fig. 6, the X-axis denotes the total number of token buffer sizes, M in the group.

As can been seen in Fig. 6, the VP reduces as M increases. The reduction is more obvious when M is small. The difference of VP between the two mechanisms diminishes as M increases from 50 to 1500.

Another factor affecting the VP is the token generating rate, R. R is used to enforce the connection not to send unconforming data to the network. It is often set as the average rate of the traffic. The connection policed by the LB mechanism is limited to send data with a average rate less than R because the conforming cells are bounded by the total tokens that can be generated.

As illustrated in Fig. 7, the VP of the TBLB mechanism is always lower than that of the LB mechanism for the same R. For example, for VP=0.1, R can be set to be about the average rate of the connections in the group if TBLB is used, but it needs about 1.2 times of the average rate if the LB is used. Note that a higher token generating rate demands more network bandwidth and is undesirable.

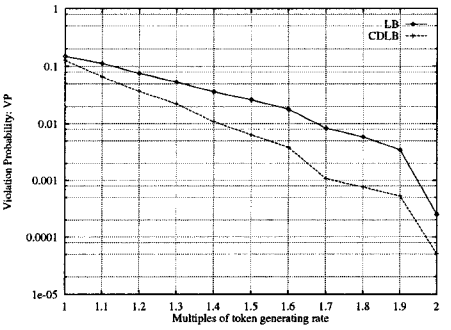

Figure 7: VP v.s. Token generating rate

Since the TBLB mechanism is designed for group connections, we are also interested to know how effective the mechanism is for different numbers of sources in the group connections. As Fig. 8 shows, the VP drops as the number of sources increases. As the number of connections increases, more tokens are generated from all the connections and there are a higher probability for one connection to obtain tokens. So that the more sources in the group, the lower the VP. The VP drops moderately when the number of sources increases from 32 to 64. This shows that the effect of statistical multiplexing is significant even when the number of sources in the group is small.

The TBLB mechanism sets up a credit limit L_i for the connection i to cope with the malicious sources.

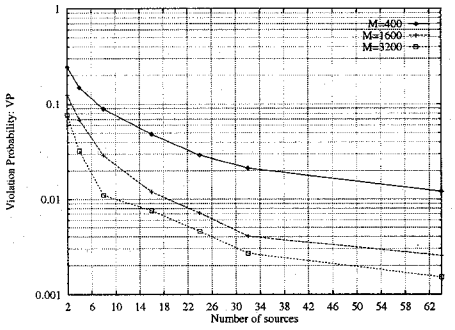

Figure 8: VP v.s. Number of sources

In the simulation, connection 1, 2, 3, and 4 have the same negotiated traffic parameters. Assume that the connection 4 acts as a malicious source, i.e., it will send more data than specified in the negotiated parameters. We will examine the impact to connections 1, 2, and 3 when the connection 4 increases traffic from 1.0 to 2.0 times of the negotiated average rate, while connections 1, 2, and 3 stay the same. Their average rates stay the same as the negotiated average rate in the simulation.

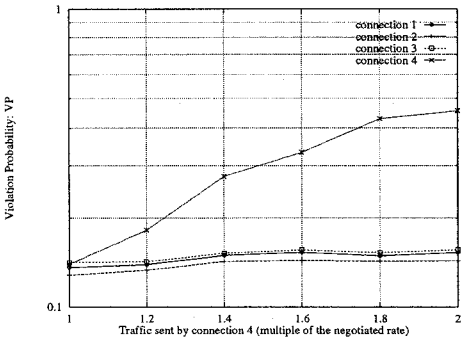

Figure 9: Effects of a malicious source to VP

As can be seen from Fig. 9, the connection 4 increases the VP as its average rate increases. Since the credit limit L_4 limits the tokens that connection 4 can obtain, lots of data cells from connection 4 are marked as violations, resulting in higher violation probability. Although the VPs of the connections 1, 2, and 3 also increase as the average rate of the connection 4 increases, they are confined to an acceptable level. As a result, the TBLB mechanism can effectively protect the well-behaved sources against the malicious sources.

The last set of the simulations are to examine the effectiveness of grouping traffic sources that have different patterns of burstiness. With high burstiness, most of data cells are concentrated in long bursts, and then followed by long silent periods. Consequently, the effect of statistical multiplexing diminishes.

Table 1: Traffic sources

	α	β	λ	ρ
$\mathcal{S}$	0.9975	0.99875	1.0	3.0
$\mathcal{U}$	0.996	0.996	1.0	2.0
$\mathcal{V}$	0.995	0.99	1.0	1.5

For illustration propose, three traffic sources are used in the simulation. Their parameters for $\mathcal{S}$, $\mathcal{U}$, and $\mathcal{V}$ are shown in Table 1.

As shown in Fig. 10, the traffic source $\mathcal{S}$ has the highest burstiness among the three. It is shown that its violation probability is the largest among the three traffic sources.

In the TBLB mechanism, the optional Rp can be increased to reduce the violation probability for the group connections. Figure 11 shows the relationship between VP and Rp. Note that Rp is in the multiple of the average rate of $\mathcal{S}$.

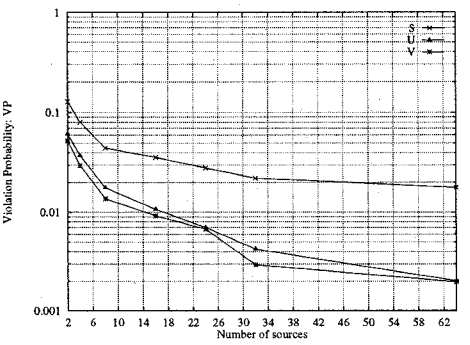

Figure 10: Effects of burstiness to VP

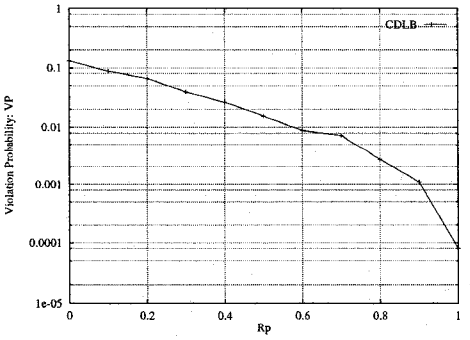

Figure 11: VP as a function of Rp

As shown in Fig. 11, the VP reduces as Rp increases. For a traffic source policed by the TBLB mechanism, if the token generating rate R is specified as the minimum rate it requires and the size of the shared token pool M is set by the predetermined MBS, then increasing Rp becomes the only solution to reduce the VP.

5 Conclusion

The leaky bucket mechanism is a well accepted policing mechanism for ATM networks. The original leaky bucket has been shown to be effective for traffic with a constant bit rate, but reacts poorly to bursty traffic. Although there are some other policing mechanisms in the literature that provide improvements over the original leaky bucket mechanism, most of them enforce the negotiated parameters for individual source traffic and not fully utilize the statistical multiplexing of multiple connections. As a result, the bandwidth utilization is low.

In this paper, we have proposed a new source policing mechanism, called the Token-Bank Leaky Bucket, for group connections in ATM networks. The mechanism consists of a token buffer of size one and a counter for each connection in the group, and a shared token buffer. The individual connections shared in the group can "credit" excessive tokens or request ("debit") tokens from the shared token buffer ("the Bank") for the extra data cells that would otherwise deemed as violations. It provides a priority scheme to allow the unused bandwidth to be shared fairly by the connections in the group. In addition, a credit limit scheme is used to prevent malicious sources from debiting tokens well beyond their negotiated limits. The mechanism was shown by simulations to have a lower violation probability and better bandwidth utilization even for small group connections when compared to the original leaky bucket with a similar configuration.

References

[1] R. Händel, M.N.Huber and S. Schröder, "ATM Networks: Concepts, Protocols, Applications", Second Edition, *Addison-Wesley*, 1994.

[2] The ATM Forum, "User-Network Interface(UNI) Specification version 3.1," September 1994.

[3] J. Sairamesh and N. Shroff, "Limitations and Pitfalls of Leaky Bucket, A Study with Video Traffic," in *Proc. of IEEE IC3N'94*, September 1994.

[4] M. Buttó, E. Cavallero, and A. Tonietti, "Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 3, pp. 335-342, April, 1991.

[5] G.Mayor and J. Silvester, "The Multi-level Leaky Bucket Mechanism," in *Proc. of IEEE IC3N'95*, pp. 380-387, 1994.

[6] I. Khan and V. O.K. Li, "Traffic control in ATM networks", *Computer Networks and ISDN Systems 27*, pp.85-100, 1994

[7] J. S. M. Ho, H. Uzunalioğlu and I. F. Akyildiz, "Cooperating Leaky Bucket for Average Rate Enforcement of VBR Video Traffic in ATM Networks," in *Proc. of IEEE INFOCOM 95*, pp. 1248-1255, April 1995.

[8] M. Murata, Y. Oie, T. Suda, and H. Miyahara, "Analysis of a Discrete-Time Single-Server Queue with Bursty Inputs for Traffic Control in ATM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 447-458, April 1990.