Journal of High Speed Networks 15 (2006) 111–122
IOS Press

Neural networks based variable bit rate traffic prediction for traffic control using multiple leaky bucket

Yen Chieh Ouyang [a], Ching-Wen Yang [b,c,*] and Wei Shi Lian [a]
[a] *Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, ROC*
[b] *Computer Center, Taichung Veterans General Hospital, Taichung, Taiwan, ROC*
[c] *Department of Management Information Systems, Chung Tai Institute of Health Sciences and Technology, Taichung, Taiwan, ROC*

Abstract. This work presents a novel feedback rate regulator using the multiple leaky bucket (MLB) for variable bit rate (VBR) self-similar traffic that is based on the traffic load prediction by time-delayed neural networks in ATM networks. In the MLB mechanism, the leak rate and buffer capacity of each leaky bucket (LB) can be dynamically adjusted based on the buffer occupancy. A finite-duration impulse response (FIR) multilayer neural network is used to predict the incoming traffic load and pass the information to the feedback rate regulator. Ten real world MPEG1 and ten synthesized traffic traces are used to validate the performance of the MLB and the MLB with an FIR prediction mechanism. Simulation results demonstrate that the cell loss rate using MLB and MLB with an FIR filter-based predictor can be significantly reduced compare to the conventional leaky bucket method.

Keywords: ATM networks, multiple leaky bucket (MLB), neural networks, self-similarity, traffic control

1. Introduction

ATM technology is based on a small and fixed size packet, to be called a cell. These cells permit switching to be sufficiently rapid so that multiple isochronous data can be statistically multiplexed and physical resources can be maximally utilized. To maintain the quality of service (QoS) perceived by network users, the users must make contracts with networks where a policing mechanism acts accordingly to protect all well-behaved sources. In ATM networks, the need for multimedia and real-time applications has increased rapidly. To elucidate the characteristics of traffic, some rate regulators based on neural networks have been proposed [1,2,13,15,16]. These mechanisms can increase bandwidth utilization and decrease cell loss rate while integrating with a rate-based feedback control scheme. Therefore, developing a simple and feasible feedback rate regulator for real-time service is highly desirable.

Many efforts have devoted to the leaky bucket control scheme to resolve a flow control problem [1,3,4], whereas several policing mechanisms have been also proposed as well [1,5]. In fact, determining an appropriate leak rate and buffer capacity for a leaky bucket is a challenging issue. For an MPEG traffic source, it may generate cells at a near-peak rate for a short time interval and become silent afterwards. Selecting a leak rate close to the source's peak rate may waste the bandwidth. By contrast, the cell loss rate (CLR) may be extremely high if a leak rate is close to the source's mean rate. It is generally known that the cell loss rate can be reduced by increasing the buffer capacity. However, if the leak rate is too low, the cell loss rate cannot be greatly improved by merely increasing buffer capacity.

*Corresponding author.

In light of above developments, we develop a source rate regulator (SRR) by adopting the concept of reactive congestion control and an FIR multilayer neural network to predict the incoming traffic load [6,14–16]. The weakness of reactive congestion control comes from high bandwidth-delay product, but it is still valid when the control method is applied to monitor a single MPEG traffic as well as multiple self-similar traffics.

The remainder of this paper is organized as follows. Section 2 briefly reviews an FIR multilayer network to be used in this paper along with its learning algorithm. Section 3 describes the characteristics of self-similar traffic. Section 4 present a novel policing mechanism, called multiple leaky buckets (MLB) for controlling the self-similar traffic. Section 5 conducts experiments and analysis based on ten real-world MPEG video traces and synthesized self-similar traffic to validate the performance of MLB. Finally, Section 6 summarizes conclusions and future research.

2. FIR multilayer network for traffic predictor

2.1. FIR multilayer network

Multilayer feed-forward neural networks have been studied extensively in many applications, such as pattern recognition, character segmentation, or diagnosis. However, one of main drawbacks suffered from the standard multilayer neural network is that the input-output mapping from which the neural networks learn is static. The form of such static input-output mapping is well suited for spatial applications, but may not be adequate for temporal applications such as prediction by time series. By including time-delay units in hidden neurons and output neurons, a neural network can be adapted to temporal applications. On such neural network is called time-delay neural network (TDNN) [7]. The TDNN topology is embodied in a multilayer network in which each synapse is represented by a finite-duration impulse response (FIR) filter as shown in Fig. 1. The weight w_{j0} connected to the fixed input $x_0 = -1$ represents the threshold value.

An FIR multilayer network is a feed-forward neural network whose hidden neurons and output neurons are replicated across a finite duration of time. The temporal backpropagation learning algorithm is generally used and described by [7]:

$$\boldsymbol{w}_{ij}(n+1) = \boldsymbol{w}_{ij}(n) + \eta\delta_i(n)x_j(n), \tag{1}$$

$$\delta_i(n) = \begin{cases} e_i(n) \cdot f(v_i(n)) & i \text{ is in the output layer} \\ f(v_i(n)) \cdot \sum\limits_{q\in\Psi}\sum\limits_{n=0}^{M} \delta_q(n+\chi) \cdot w_{qi}(n) & i \text{ is in a hidden layer,} \end{cases} \tag{2}$$

where $\boldsymbol{w}_{ij}, \eta, e_i, f(\cdot)$, M and Ψ represent the connection weight, the learning rate, the difference between the desired output the neural network output, the activation function, the number of taps in the current layer, and the set of all neurons whose inputs are fed by neuron i via (2) in a feed-forward manner respectively. The FIR algorithm is composed of two phases: an off-line learning phase and an on-line processing phase. The off-line learning phase is mainly used for training the FIR multilayer network, and is not used for predicting the number of incoming cells at the next period. Once the off-line learning phase is completed, the on-line processing phase is then started where an FIR multilayer network is used to predict the number of incoming cells in the next period, and the cell discarding function is implemented based on the information provided by the output of the FIR multilayer network. Note that when an accumulated prediction error is greater than a pre-determined threshold value ε, the FIR multilayer network must be trained again.

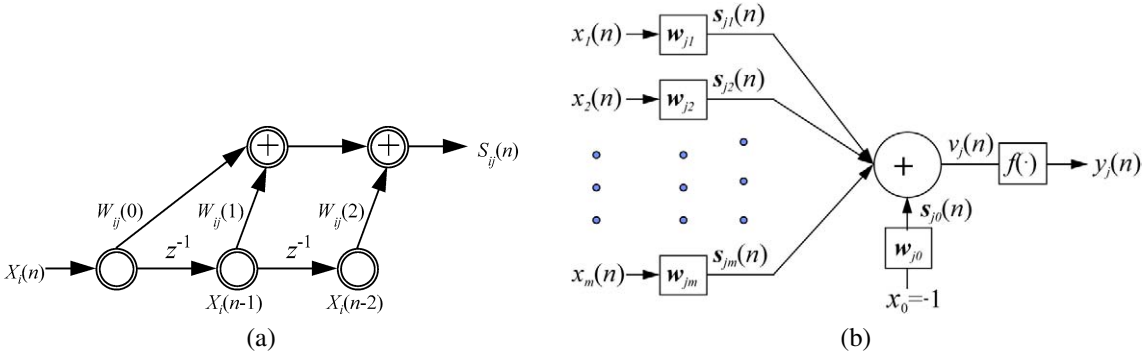


Fig. 1. (a) Signal flow graph of a FIR filter. (b) Spatial-temporal model of a neuron, incorporating synaptic FIR filter.

2.2. Implementation complexity

In order to train the FIR multilayer network in the off-line learning phase, the incoming cells in the first K periods can be collected and used as the desired training data set (*TDS*), denoted by *TDS*(1, *K*) = $\{G(1), G(2), \ldots, G(K)\}$.

Since there are no rules to follow in designing the topology of the FIR multilayer neural network for a specific problem, different topologies of an FIR network are trained by using the training set *TDS*(1, *K*). The network with the smallest accumulated error is chosen to perform prediction in the on-line processing phase. The accumulated error should be smaller than ε. If it is not, a larger network is required to be used for training. Note that in the off-line learning phase, the learning rate η in (1) has impact on the convergence speed of the training process. For a multilayer neural network, the speed of the weight update in earlier iterations is faster than that in later iterations, therefore a large value η at the beginning of the training process is used and will be replaced with a smaller value after a certain number of iterations.

In the on-line processing phase, the trained FIR multilayer network is used to predict the number of incoming cells in the next period. If the predicted number of incoming cells in the next period does not exceed the buffer capacity, then all the incoming cells during the next period are admitted to enter the buffer. It should be noted that it is possible that the FIR multilayer network may predict no overflow for the next period, but the overflow does happen at the next time slot. Under this circumstance, the feedback traffic rate regulator is then used to adjust the source cell rate.

In order to use real time VBR traffic prediction, it would be physical if the amount of computation time of FIR neural networks can be estimated. From Fig. 1(a) and (b), we have seen the neuron has m primary connections. The FIR multilayer network is a feed-forward neural network whose hidden neurons and output neurons are replicated across finite duration of time. Figure 2 shows the FIR multilayer network with 2 taps in the first hidden layer and one tap in the second hidden layer. Each input consists of a linear discrete time filter implemented in the form of an FIR filter of order i. Each primary connection has $(i+1)$ secondary connections that are connected to its respective input and the memory taps of its FIR filter. The total number of connecting weights in the structure is $m(i+1)$. An FIR network with 2nd order taps for all connections can be unfolded into constrained static network. For example, in our case, a $6 \times 8 \times 3 \times 1$ structure FIR neural network traffic predictor is used with $6 \times 8 \times 3 \times 1$ indicating the four-layer FIR predictor having 6 input nodes, 8 first hidden layer, 3 second hidden layer and one output neuron. Since the number of computations is highly correlated with the connection weights, we can simplify the number of computations required in the off-line training phase to approximately $1/\eta \cdot$ (Number of _Input neuron) $\cdot$ (Number of first hidden neuron) $\cdot$ (Number of second hidden neuron) $\cdot$ (Number of Output neuron) $\cdot$ (Number of Synapses) basic operation, where η represents the learning rate. The training process was conducted 1000 iterations with learning rate $= 0.1$.

The feedforward artificial neural network (FANN), propose by Liu and Douligeris [1], used to predict possible future cell loss rates is a feedforward error backpropagation network. In order to investigate the efficiency of the

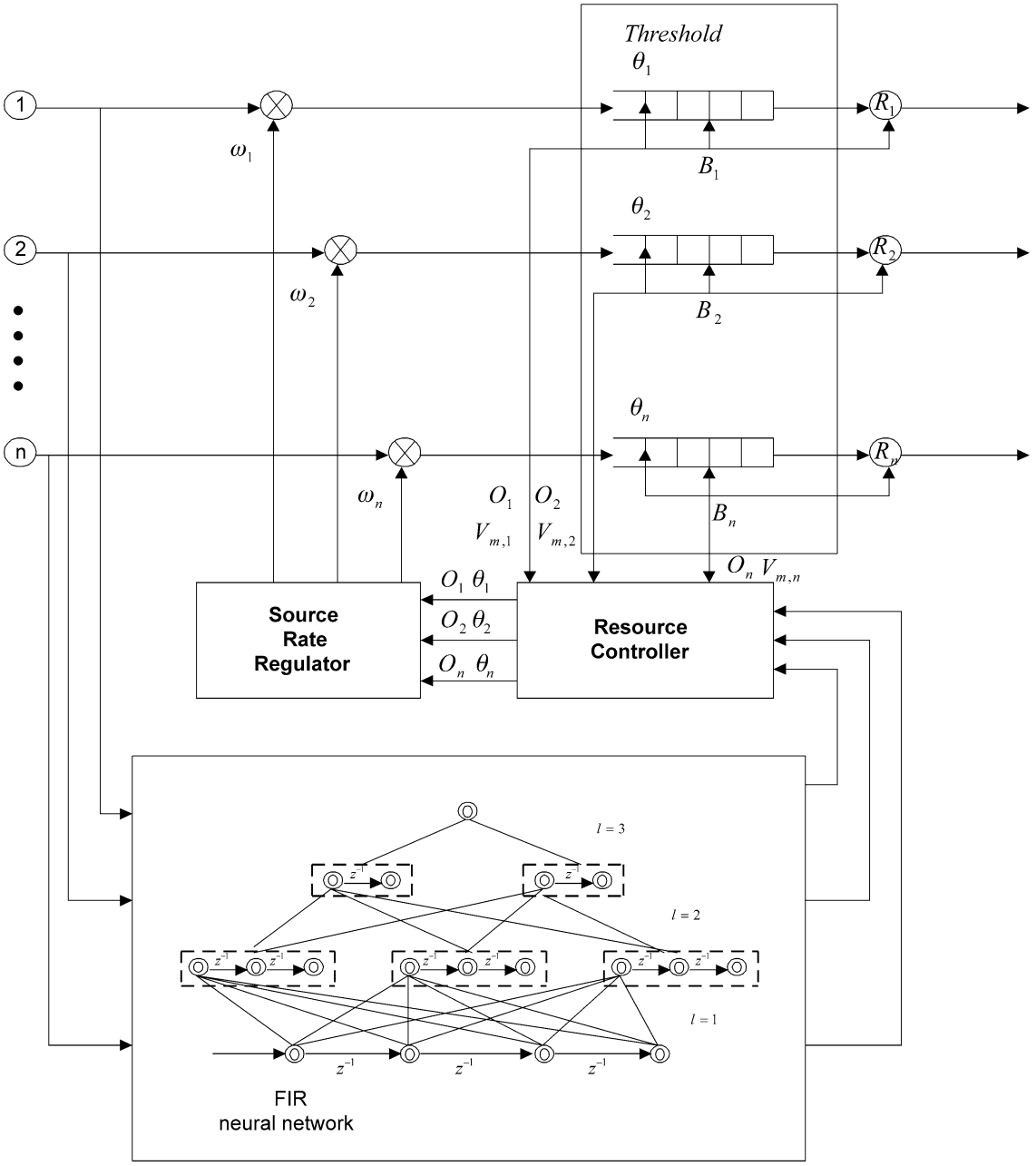
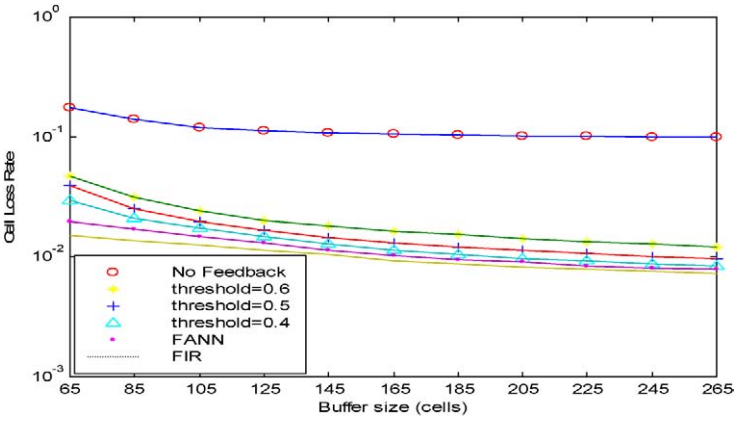
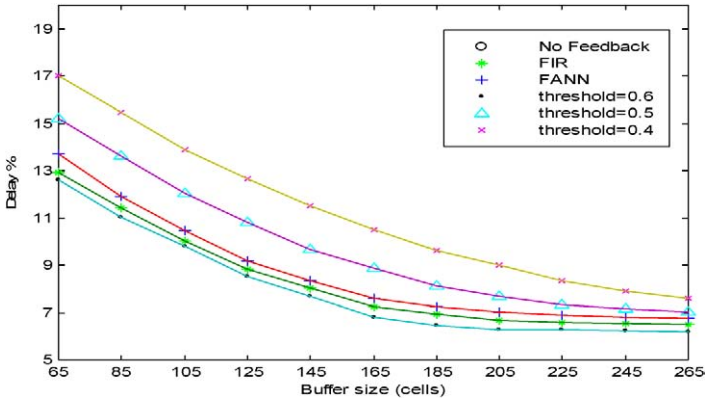


Fig. 2. An ATM traffic control model with FIR multi-layer neural network using source rate regulator (SRR) on the multiple leaky bucket with buffer sharing.

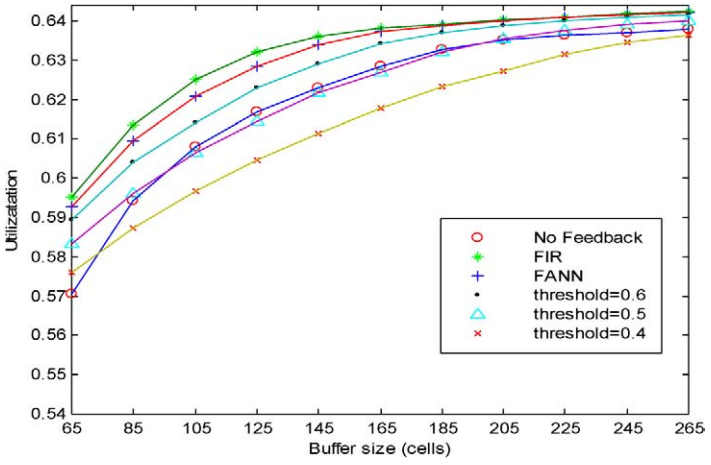
FIR neural network, we have conducted a similar experiment in [1] using both FIRNN and FANN to train for the traffic prediction. Figure 3(a–c) shows the simulation results between the comparison FIR and FANN for single MPEG trace ($N = 1$). A single video trace (trace-1) is fed to the leaky bucket (LB), cells are removed from the LB at a constant rate μ which is equal to 757 cells/s. The cell loss rate of using the FIR neural network shows a better improvement compared to the case that $Q_{threshold} = 0.4$, where $Q_{threshold}$ is threshold value and was defined as



(a)



(b)



(c)

Fig. 3. Single source and leaky rate = 757 cells/s. (a) Cell loss rate versus buffer size, single source, trace-1. (b) Delay versus buffer size, single source. (c) Utilization versus buffer size.

same as in [1]. Also the delay is smaller than $Q_{threshold} = 0.6$, and the utilization is better than $Q_{threshold} = 0.4$ and 0.5. We have learned that using the FIR can have better performance on both delay and bandwidth allocation. From these figures we can see that using the FIR neural network performed better than the FANN.

3. Characteristics of self-similar traffic

Recently, it has been confirmed in [9,10] that self-similar processes in real-world networking applications aptly described traffic patterns. For example, Ethernet traffic, World Wide Web (WWW) traffic, and video traffic transmitted over ATM networks are self-similar. In what follows, we introduce the definition, measurement, and major characteristics of self-similar traffic.

The fact that most traffic patterns have self-similarity indicates the need of an effective bandwidth formula for self-similar traffic. Norros [9] was the first to look into this issue. Assume that a self-similar traffic source is given along with the mean bit rate m (in bits/sec), the Hurst parameter H, and the variance coefficient a (in bit-sec) where the variance coefficient is the ratio of the variance over one second interval to the mean bit rate of the traffic stream.

For a given buffer size, B and desired cell loss ratio, ε, the effective bandwidth of a self-similar traffic, C, can be expressed as:

$$C = m + (k(H)\sqrt{-2\ln\varepsilon})^{1/H} a^{1/2H} B^{(H-1/H)} m^{1/2H}, \tag{3}$$

where $k(H) = H^H(1-H)^{(1-H)}$.

By virtue of the Norros formula, the network resource requirement for variable bit rate (VBR) video traffic can be estimated. In other words, the Norros formula can be used to estimate the effective bandwidth requirement for self-similar traffic. In [10], it was demonstrated that a VBR video source transmitted over ATM networks was self-similar traffic.

4. Feedback rate regulator for self-similar VBR traffic

4.1. Multiple leaky buckets

In ATM networks, each virtual path (VP) may contain several traffic sources. Therefore, several independent leaky buckets may be available and can be used to monitor every traffic source. If these leaky buckets from all sources collaborate among themselves, the bandwidth and buffer spaces can be effectively used, thereby reducing the cell loss rate (CLR) significantly.

Ho [11] first proposed the above idea and applied it to a policing mechanism which implemented the CLB. The CLB differs from the LB mainly in that if a leaky bucket becomes empty, its leak rate is distributed to other buckets. To improve the performance of CLB, our results indicated that there was no need to distribute leak rate to other buckets unless its buffer becomes empty. Besides, the LB not only can share the leak rate, but also can share the buffer capacity.

However, when all the leaky buckets in the same virtual path are integrated, how to distribute the total leak rates and buffer spaces over all the sources is a major issue. In order to mitigate this dilemma, we dynamically adjust the leak rate and buffer capacity for each LB according to their buffer occupancies.

Assume that there are n traffic sources and n leaky buckets, each of which monitors one traffic source. For the ith traffic source and ith LB the following parameters are used

- M_i: the negotiated mean cell rate of ith traffic source (in cell/sec),
- P_i: the negotiated peak cell rate of ith traffic source (in cell/sec),

- $R_i(t)$: the leak rate of ith LB at time t,
- $B_i(t)$: the buffer capacity of ith LB at time t,
- $R_i(0)$: the initial value of $R_i(t)$ (in cell/sec),
- $B_i(0)$: the initial value of $B_i(t)$ (in cell),
- $K_i(t)$: the number of cells in the ith buffer at time t,
- $O_i(t)$: the occupancy of ith buffer and $O_i(t) = K_i(t)/B_i(t)$,
- ε: the desired cell loss rate.

First of all, the initial values of the leak rate, $R_i(0)$, and buffer capacity, $B_i(0)$, need to be determined, where $R_i(0)$ and $B_i(0)$ are given by

$$R_i(0) = \gamma \cdot M_i, \tag{4}$$

$$B_i(0) = \beta \cdot P_i, \tag{5}$$

where γ and β are constant. Furthermore, the aggregate leaky rate and buffer capacity, denoted by R_0 and B_0 are defined respectively by

$$R_0 = \sum_{i=1}^{n} R_i(0), \tag{6}$$

$$B_0 = \sum_{i=1}^{n} B_i(0). \tag{7}$$

Let ΔT denote the time interval. After the initial values of all relevant parameters are determined, the policing function must obtain the value for the buffer occupancy $O_i(t)$ every ΔT seconds. For a given $O_i(t)$, the new leak rate and new buffer capacity can be calculated. The new leak rate of ith LB is the further defined by

$$R_i(t+1) = R_0 \cdot \frac{R_i(0) \cdot O_i(t)}{\sum_{i=1}^{n} R_i(0) \cdot O_i(t)}. \tag{8}$$

If we assume that the buffer spaces are shared, we define the reservation ratio, δ, for each LB as follows

$$\delta = \frac{\textit{reserved buffer size}}{\textit{total buffer size}}. \tag{9}$$

The δ' used in FIR multilayer neural networks can be express as

$$\delta' = \delta \cdot (1 + (n_C/n_T)), \tag{10}$$

where n_C is the cell loss number and the n_T is the total cell transmitted number.

It is clear that when the buffer reserved size is too large (i.e., δ increases) the benefit resulting from buffer sharing may vanish. On the other hand, if the buffer reserved size is insufficient (i.e., δ decreases) and if the source transmission rate is close to the peak rate, then buffer may quickly become full, which results in a large number of cells being discarded. As a consequence, a new buffer capacity of the ith LB should be defined to accommodate this situation as follows.

$$B_i(t+1) = (1-\delta') \cdot B_0 \cdot \frac{B_i(t) \cdot O_i(t)}{\sum_{i=1}^{n} B_i(t) \cdot O_i(t)} + \delta' \cdot B_i(0), \tag{11}$$

where the occupancy $O_i(t)$ is an indicator. If $O_i(t)$ exceeds the average value, $1/n \sum_{j=1}^{n} O_j(t)$, then the leak rate, $R_i(t)$, and buffer capacity, $B_i(t)$, are reassigned higher values than their initial ones.

This mechanism has an advantage. It can effectively distribute the leak rate and buffer capacity to reduce the CLR. If the counter is close to the maximum allowable value, cell loss would likely occur. In addition, increasing the leak rate and buffer capacity before cell loss occurs can significantly reduce the CLR.

4.2. Multiple leaky buckets with feedback control

The feedback traffic control is used to reduce cell loss rate by regulating source transmission rate and provides better protection to well-behaved sources. The ATM traffic control model using a source rate regulator with an FIR multilayer neural network is shown in Fig. 2.

A difficult problem with preventive congestion control is to select an adequate set of parameters to describe a source and allocate an appropriate amount of resources for it. In practice, even if several sources generate data at their near-peak rates simultaneously, cell loss may still occur. In order to reduce the possibility that many sources request a large amount of resources during the same time period, the use of the policing mechanism to control the source rate becomes essential.

To regulate the transmission rate, we define an initial threshold value for the ith traffic, denoted by $\theta_i(0)$. This threshold value of the ith traffic source, $\theta_i(t)$, is renewed every ΔT seconds and defined by

$$\theta_i(t) = \frac{\theta_i(0)}{\max\{V_{m,i}(t), 1\}}. \tag{12}$$

In addition, rate regulation can be employed to protect well-behaved sources. It is feasible to reduce the transmission rate for a malicious user that violates the bandwidth contract. A modified version of feedback traffic regulation proposed in [1] is also used. The parameters of LBs are controlled by resource controller (RC). The RC keeps track of the status of each LB in every ΔT seconds and calculates the new leaky rates, new buffer space and new threshold values respectively. The objective of the source rate regulator (SRR) is to decide whether or not to throttle down the source transmission rate. When the occupancy of the ith buffer, $O_i(t)$, exceeds its threshold value, $\theta_i(t)$, the SRR generates a backward RM-cell in which the congestion notification (CI) field is set to 1, and transmits it to the source i. When the source i receives an RM-cell with CI $= 1$, it decreases its transmission rate by multiplying the factor $\omega_i(t)$, given by

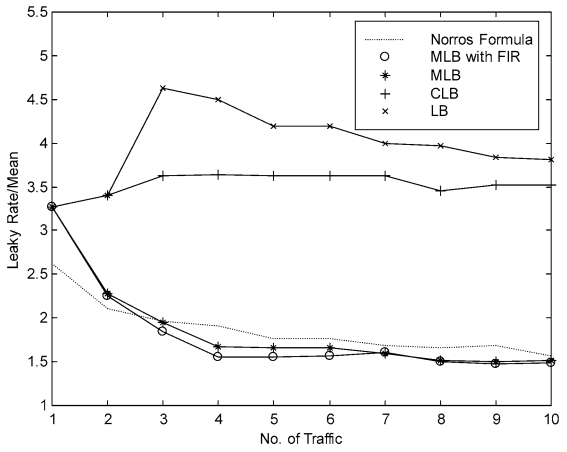
$$\omega_i(t) = \frac{1}{2 \cdot \max\{V_{m,i}(t), 1\}}. \tag{13}$$

Only when the value of $O_i(t)$ is lower then the threshold, $\theta_i(t)$, the source i can receive a backward RM-cell with CI $= 0$ and transmit data at the original transmission rate.

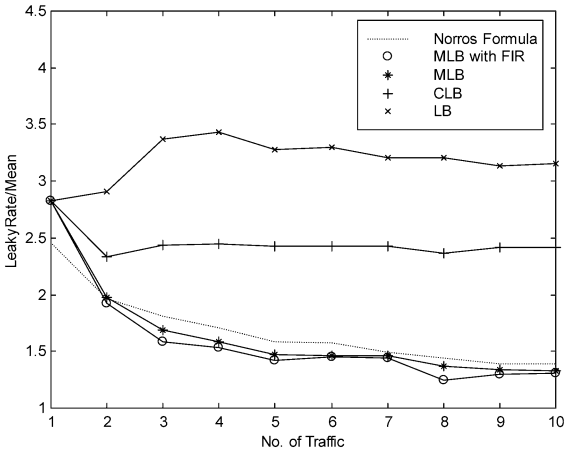
5. Simulation results

Ten MPEG1 frame size traces and synthesized self-similar traffic were used for experiments. The MPEG1 traces, trace1 to trace10, can be found at the website [14]. These frame size traces were extracted from MPEG1 sequences, which have been encoded by the Berkeley MPEG-encoder (version 1.3). Each MPEG video consists of 40 000 frames and is equivalent to approximately half an hour. The pattern of the MPEG1 stream is IBBPBBPBBPBB, and twenty-four frames are coded per second. The synthesized self-similar streams were generated using a fractional ARIMA (Auto-Regressive Integrated Moving Average) process [12], which generated a time series with a specified degree of self-similarity H. A variance-time (VT) plot technique is applied to estimate the Hurst parameters.

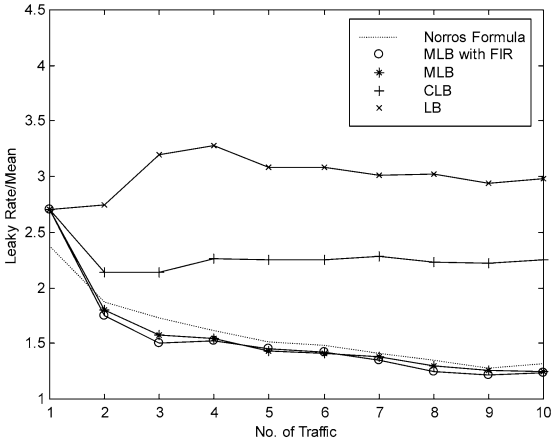
In our simulations, the normalized effective bandwidth for each mechanism is plotted. The desired cell loss rate is set to zero. Figure 4(a–c) indicates that using the leaky bucket has the highest normalized effective bandwidths.



(a)

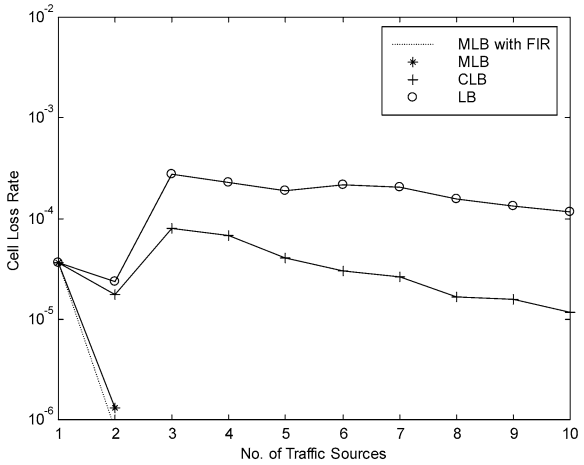


(b)

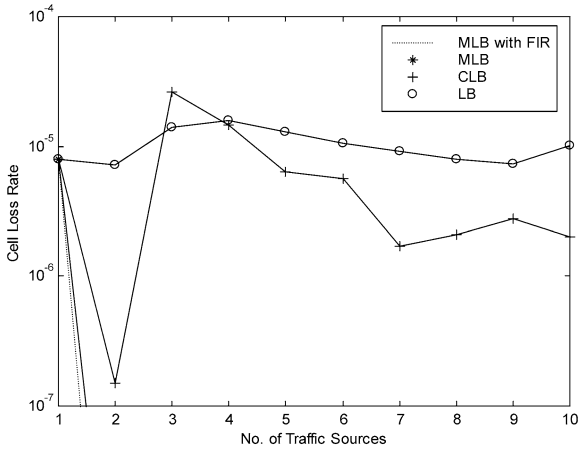


(c)

Fig. 4. Normalized effective bandwidth for aggregated MPEG1 traces. (a) $\beta = 0.1$, (b) $\beta = 0.2$, (c) $\beta = 0.3$.



(a)



(b)

Fig. 5. Cell loss rate for aggregated traffics using feedback control mechanism. (a) MPEG1 traces, $\gamma = 1.3$, $\beta = 0.1$. (b) Synthesized self-similar data series.

However, the curves generated by using the MLB with FIR are close to the curves generated by using the Norros formula which requires the lowest effective bandwidth. These results further demonstrate that the multiplexing gain exists, even for heterogeneous traffic. This observation also implies that a large amount of bandwidth has been wasted when the conventional leaky bucket mechanism is used. This implies that the MLB is a better option.

Figure 5(a–b) indicates that the CLR drops to zero as long as two or more traffic sources are fed into the MLB. This finding suggests that less bandwidth is needed to achieve the desired cell loss rate when a feedback rate regulator is used. Figure 6(a–b) reveals that the MLB with FIR mechanism has the best performance in practice and can reach the upper bound $1/\gamma$ as the number of traffic sources is increased. Each mechanism has the same protection policy. Our proposed mechanism, MLB with FIR traffic prediction, yields the optimum performance in terms of CLR and delay. The cell loss rates for each well-behaved source drops to zero regardless of whether or not the malicious source is fed.

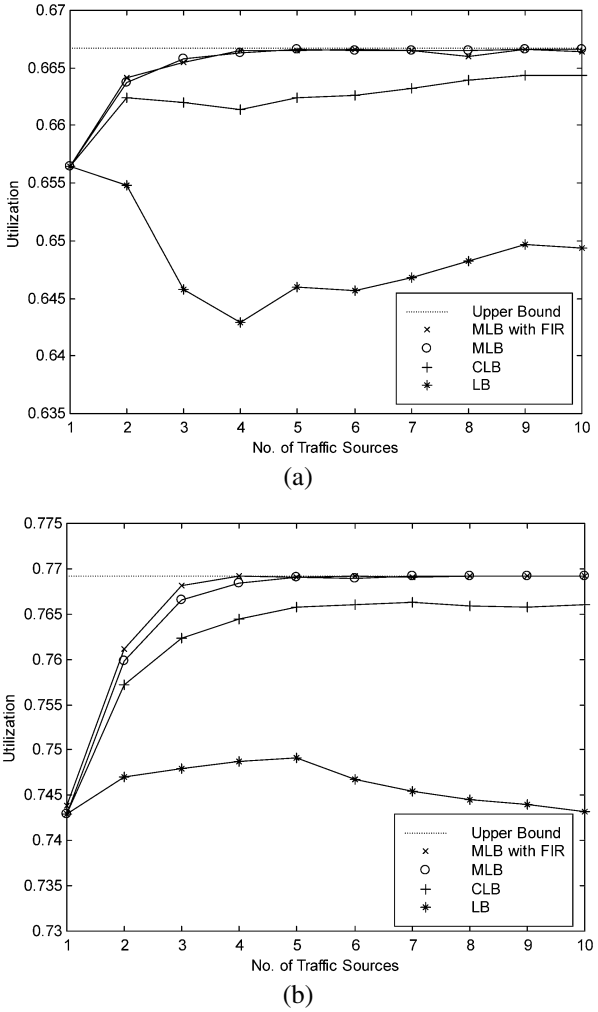


Fig. 6. Utilization for aggregated traffics using feedback control mechanism. (a) MPEG1 traces, $\gamma = 1.3$, $\beta = 0.1$. (b) Synthesized self-similar data series, $\gamma = 1.2$, $\beta = 0.1$.

6. Conclusions

This paper presents multiple leaky buckets (MLB) with an FIR filter-based traffic predictor for the feedback traffic regulator. The proposed mechanism is implemented to monitor and regulate self-similar VBR traffic to avoid congestion. To maximize network utilization, the MLB mechanism integrates all the leaky buckets (LBs) in the same virtual path. Simulation results demonstrate that the MLB with our designed FIR traffic predictor takes advantage of the statistical multiplexing gain of self-similar VBR traffic. Thus, the MLB in conjunction with such an FIR filter can significantly reduce the cell loss rate than the conventional LB mechanism. To achieve the desired cell loss rate, e.g., CLR $= 0$, the MLB requires a less effective bandwidth than the LB and CLB mechanism. Furthermore, it requires to use the Norros formula to estimate the effective bandwidth.

Simulation results demonstrate that when the MLB and MLB with FIR traffic predictors and feedback control mechanism are used, it makes a malicious source difficult to obtain extra network resources. Additionally, the QoS can be maintained for well-behaved sources as well. Our results further demonstrate that using the feedback control

mechanism with an FIR traffic predictor yields better performance not only for MPEG1 traffic, but also for the data streams with higher self-similarity.

Acknowledgement

The authors would like to thank for support funded by National Science Council under NSC 87-2218-E-023.

References

[1] Y.C. Liu and C. Douligeris, Rate regulation with feedback control in ATM networks – a neural network approach, *IEEE JSAC* **15** (1997), 200–208.

[2] P.R. Chang and J.T. Hu, Optimal nonlinear adaptive prediction and modeling of mpeg video in ATM networks using pipelined recurrent neural networks, *IEEE JSAC* **15** (1997), 1087–1100.

[3] V. Anantharam and T. Konstantopoulos, Burst reduction properties of the leaky bucket flow control scheme in ATM networks, *IEEE Trans. Commun.* **42** (1994), 3085–3089.

[4] K.Q. Liao, A. Dziong, L. Mason and N. Tetreault, Effectiveness of leaky bucket policing mechanism, in: *Proc. of IEEE ICC '92*, 1992, pp. 1201–1205.

[5] E.P. Rathgeb, Modeling and performance comparison of policing mechanisms for ATM network, *IEEE JSAC* **9** (1991), 325–334.

[6] H.P. Lin and Y.C. Ouyang, ATM cell discarding policy by FIR neural networks, in: *Proceeding of IEEE/ICNN97* **4** (1997), 2051–2056.

[7] S. Haykin, *Neural Network a Comprehensive Foundation*, Macmillan College Publishing Company, 1994.

[8] E.A. Wan, Time series prediction by using a connectionist network with internal line, in: *Time Series Prediction: Forecasting the Future and Understandard the Past*, A.S. Weigend and N.A. Gershenfeld, eds, Addison-Wesley, 1994, pp. 195–217.

[9] I. Norros, On the use of fractional Brownian motion in the theory of connectionless networks, *IEEE JSAC* **15** (1997), 200–208.

[10] Available 6: 6-info3.informatik.uni-wuerzburg.de Directory: pub/MPEG.

[11] J.S.M. Ho, H. Uzunalioglu and I.F. Akyildiz, Cooperating leaky bucket for average rate enforcement of VBR video traffic in ATM networks, in: *Proc. of IEEE INFOCOM*, 1995, pp. 1248–1255.

[12] G.E.P. Box and G.M. Jenkins, in: *Time Series Analysis: Forecasting and Control*, 1976, pp. 17–19.

[13] I.W. Habib, Applications of neurocomputing in traffic management of ATM networks, *Proc. IEEE* **84** (1996), 1430–1441.

[14] W. Xu and A.G. Qureshi, Adaptive linear prediction of MPEG videotraffic, in: *Proc. 5th Int. Symp. Signal Processing Its Applications*, 1999, pp. 67–70.

[15] A. Sang and S. Li, A predictability analysis of network traffic, in: *Proc. IEEE INFOCOM*, 2000, pp. 342–351.

[16] Y. Liang, Real-time VBR video traffic prediction for dynamic bandwidth allocation, *IEEE Trans. Sys., Man, and Cyber. C* **34**(1) (2004), 32–47.