明新學報 32 期 pp.203-219
Volume 32, Minghsin Journal, August 2006

An Efficient FIFO-based Buffer Management Strategy for TCP over GFR

Shuo-Cheng Hu[1*] Sen-Ren Jan[2]

[1]Department of Information Management, Shih-Hsin University

[2]Department of Information Management, Minghsin University of Science and Technology

Abstract

The Guaranteed Frame Rate（GFR）service is positioned as an intermediate alternative between Available Bit Rate（ABR）service, which uses a closed-loop feedback control framework, and Unspecified Bit Rate（UBR）service, which provides no assurance whatsoever. The GFR service is expected to be useful for TCP/IP based applications. The aim of the GFR service is to provide each ATM VC with a minimum service rate guarantee at the frame level and a fair access to excessive bandwidth. In this paper, we propose an efficient buffer management approach called Virtual Buffering Strategy（VBS）that provides both Minimum Cell Rate（MCR）guarantees and fair sharing of GFR VCs in a FIFO queue. VBS is composed of an emulation of per-VC queueing with Weighted Round Robin（EWRR）and the Differential Packet Discarding（DPD）buffer management scheme. From the simulation results, we demonstrate that VBS fulfills the requirements of GFR service as well as improves the TCP throughput.

Keywords： TCP/IP, GFR, FIFO queue, VBS

在結合 TCP 與 GFR 下一個有效率的以先進先出方式服務之緩衝區管理方法

胡碩誠[1] 詹森仁[2]

世新大學資訊管理系

明新科技大學資訊管理系

摘要

保證框架速率服務(GFR)是在使用封閉迴路回饋控制架構之可利用位元速率服務(ABR)與不提供任何保證的非指定位元速率服務(UBR)間，另一種可選擇的傳輸服務。保證框架速率服務被期望能用於以 TCP/IP 爲基礎的應用。保證框架速率服務的目標在於針對每一個 ATM 虛擬傳輸線路都能提供在框架層次的最小保證傳輸速率以及公平分享多餘的頻寬。在本篇論文中，將提出一個有效率的緩衝區管理方法，稱爲虛擬緩衝策略。它能在以先進先出方式服務的佇列中對每一個保證框架速率的虛擬連線提供最低速率保證與公平的頻寬分享。虛擬緩衝策略是結合模擬在分別虛擬線路佇列下利用之加權式循環排程方法

* 資訊管理系助理教授　地址：台北市木柵路一段 17 巷 1 號　Tel：02-22368225
E-mail：schu@cc.shu.edu.tw

(WRR)與差別封包丟棄式緩衝區管理方法。從模擬實驗中能證明虛擬緩衝策略不僅能滿足保證框架速率服務的要求，同時也能提升 TCP 的有效傳輸量。

關鍵字：TCP/IP，保證框架速率服務，先進先出方式服務的佇列，虛擬緩衝策略

1. INTRODUCTION

The Asynchronous Transfer Mode（ATM）networks have been deployed as backbones over various parts of the Internet. The issues regarding how TCP/IP can operate efficiently on top of an ATM infrastructure are crucial. The Traffic Management group in ATM Fourm has been fairly focused on developing a new service called Guaranteed Frame Rate（GFR）[1], [2]. This service allows applications which are neither able to specify the traffic parameter values, nor able to comply with the behavior rules required by existing ATM service. IP routers connected via ATM networks can set up GFR VCs between them for data transport so that TCP/IP traffic can benefit from ATM GFR services. In GFR service, traffic shaping/marking is frame-based, and complete frames are either accepted or discarded. The goal is to minimize policing and signaling process, to avoid feedback, and to achieve the minimum rate guarantees through simple buffer management and scheduling at switches.

On the establishment of a GFR connection, the ATM end-system specifies a Peak Cell Rate（PCR）and a Minimum cell rate（MCR）which is defined along with a Maximum Burst Size（MBS）and a Maximum Frame Size（MFS）. The GFR service can be specified with a MCR of zero. The user may always send cells at a rate up to PCR, but the network only commits to carry cells in complete frames at MCR. Traffic beyond MCR will be delivered under the restriction of available resources. There are two versions of GFR, GFR.1 and GFR.2 [1]. With GFR.1, the network conveys the CLP（cell loss priority）bit transparently for frames that consist entirely of CLP=0 cells or of CLP=1 cells. Tagging is not applicable. With GFR.2, the identification of frames to which the service guarantee applies is based on the F-GCRA test [1]. Non-eligible cells of a frame are tagged through the F-GCRA. In this article, we only consider the GFR.2 service definition.

A desirable implementation of GFR is to use a single queue for all GFR VCs. In this study, we propose a GFR traffic control scheme called Virtual Buffering Strategy（VBS）to provide QoS for GFR service through FIFO queueing discipline. VBS is composed of two major components, the Emulated Weighted Round Robin（EWRR）scheduling algorithm and the Differential Packet Discarding（DPD）buffer management scheme. EWRR maintains the per-VC virtual buffer occupancy which is used by DPD to fairly allocate buffer space to GFR VCs. The merits of VBS are the improvements in both transmission efficiency and fair sharing of network resources. We demonstrate its excellence through simulation.

The organization of this paper is as follows. Section 2 gives an overview of GFR implementation alternatives and an insight into the problems of the proposed schemes. In Section 3, we present the functional approaches of our proposed strategy. We then present the simulation results with TCP traffic over LANs interconnected via an ATM network in Section 4. Finally, Section 5 gives concluding remarks.

2. OVERVIEW OF BUFFER MANAGEMENT SCHEMES

2.1 GFR Implementation Alternatives

Several mechanisms have been proposed in the literature to support the GFR service in ATM switches [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [17]. The proposed mechanisms can be grouped in three main categories. The most complicated mechanisms are those utilizing both per-VC accounting and per-VC scheduling. An example from this group is the mechanism proposed in [3]. It maintains one logical queue for each GFR VC and these queues are served with a rate of at least their MCR through a WFQ-like scheduler [16]. In addition to the per-VC scheduling, it also uses a buffer acceptance algorithm to allocate buffer space among competing VCs. The main drawback of this scheme is the high processing overhead, especially the per-VC queueing, which complicates the switching system design.

Double FPD proposed in [3] is the simplest mechanism that neither uses per-VC accounting nor uses per-VC scheduling, but it designates two thresholds based on the packet priorities. When the global queue length exceeds the first threshold, packets tagged as lower priority（CLP=1）are dropped. When the queue occupancy exceeds the second threshold, EPD [13], [15] is performed on all packets. The scheme neither provide per-connection throughput guarantee nor be fair in allocating bandwidth to competing VCs.

The Differential Fair Buffer Allocation （DFBA）buffer management scheme proposed in [7] requires per-VC accounting instead of per-VC scheduling. It maintains two discard thresholds, tagged packets are discarded when the buffer occupancy exceeds the first threshold *LBO*, and EPD is performed on all packets when the buffer occupancy overpasses the second threshold *HBO*. Between the two thresholds, the incoming untagged packets（CLP=0）of VC_i are dropped according to a probabilistic manner if VC_i has larger buffer occupancy than its fair share. The simulation in [7] shows that it can provide MCR guarantee to GFR VCs, however, excessive bandwidth cannot be shared in proportional to MCR.

2.2 Observation

In ATM networks, data traffic supported by GFR is usually mixed with other higher priority CBR/VBR traffic. Consequently, the available bandwidth allocated to GFR VCs varies dynamically. However, it is expected that the network administrator should map out a minimum bandwidth C_{GFR}, which is available for GFR services. The GFR traffic should be served in a manner that achieves the configured service rate over a time scale TGFR. Since GFR guarantees a minimum cell rate for each VC, we shall assume that before admitting a connection, the network will perform certain admission control and bandwidth reservation procedure to ensure that the available bandwidth C_{avail} （in cells/sec, $C_{avail} \geq C_{GFR}$）is always no less than the sum of the MCR of every VC, i.e., $C_{avail} \geq \sum_{i=1}^{N} \mathrm{MCR}_i$, where MCR_i （in cells/sec）is the minimum cell rate of VC_i, and N is the number of GFR VCs.

Most of the previous studies have examined TCP traffic with a single TCP per VC. However, in the real network the edge switch of the IP-ATM network will multiplex multiple TCP connections over a single VC. The traffic control on VC level is unaware of each TCP in the VC. The proposed scheme will assume that each VC carries multiple TCP connections.

DFBA drops all tagged packets when the buffer occupancy exceeds the first threshold. The simulation result in [9] has demonstrated that the discarding policy would cause the global synchronization syndrome and as a consequence, lowering the TCP throughput. Furthermore, a tagged packet is dropped regardless the actual buffer occupancy of its associated VC. It causes the scheme failed to allocate excess bandwidth fairly, especially

when the GFR VCs have a wide range of round-trip times.

3.THE VIRTUAL BUFFERING SCHEME

Two fundamentals of a robust GFR implementation are the provisions of fair bandwidth sharing and efficient frame-level throughput. A fair queueing strategy, which determines how cells are scheduled at the egress of the queue, can guarantee fair sharing of available bandwidth. Furthermore, a buffer management policy which controls the entry of cells as well as features both fair buffer-space allocation and high buffer utilization, is conducive to reach the ultimate goals. The integration of above two traffic control techniques is usually implemented on the per-VC queueing architecture. In this section, we describe the Virtual Buffering Scheme（VBS）, which is based on a FIFO queue. VBS integrates a fair scheduling algorithm with an efficient buffer management scheme.

3.1 The Emulated Weighted Round Robin Scheduling Algorithm

The Virtual Queueing（VQ）technique is proposed in [11], [12] to emulate the roundrobin service of per-VC queueing on a FIFO queue for improving the performance of TCP over ATM-UBR. An enhancement to VQ, called MCR scheduling [11], proposes the emulation of a weighted scheduler to provide MCR guarantees to ATM connections. A difference between the algorithm in [11] and our proposed method is that the former complies with *MCR plus equal share* fairness criteria, while the latter conforms to *allocation proportional to MCR* fairness criteria [1]. Besides, the implementation of MCR scheduling [11] involes the update of the weighted variable for each VC after every time interval *T*, and this increases the complexity of the scheme. The proposed method uses a fixed and predetermined weight for each VC and consequently, is feasible to implement.

The weighted round robin（WRR）scheduling based on per-VC queueing selects a fixed time frame T（in cell-time-slot）and attaches a weight （W_i）to each VC_i, $1 \le i \le N$. The sum of these N weights satisfies $\sum_{i=1}^{N} W_i = T$. The scheduler will serve W_i cells for each VC_i in each period *T*. We now apply the virtual queueing technique to emulate the per- VC WRR scheduler. Since the available bandwidth allocated to GFR VCs would vary with time, the number of cells to be served during different period *T* is inconsistent. Instead of selecting a fixed time interval *T*, we define $\tilde{T}$ to be a number of service chances. That is, the total number of cells from the GFR VCs that can be served throughout $\tilde{T}$ service chances is $\tilde{T}$ and $\tilde{T} \ge T_{GFR} \times C_{GFR}$. The per-VC weighted variable（W_i）is defined as follows:

$$\mathrm{W_i} = \tilde{T} \times \frac{MCR_i}{\sum_{j=1}^{N} MCR_j} . \qquad (1)$$

Each VC is assigned a credit counter（CC_i）, while a frame counter（FC）is associated with the FIFO queue. Initially, CC_i and FC are set to W_i and $\tilde{T}$ respectively. To simplify the implementation, the value $\tilde{T}$ is selected such that all W_is are integer. Besides, a state variable $\tilde{X}_i$ is maintained for each VC_i, $\tilde{X}_i$ represents the queue occupancy of VC_i as if per-VC queueing and weighted round robin scheduling are implemented. Note that $\tilde{X}_i$

does not correspond to the actual queue occupancy. Moreover, *SL* denotes a service list containing the identifiers of the eligible VCs. We define VC_i as eligible if both $\tilde{X}_i > 0$ and $CC_i > 0$ are satisfied. When the head-of-line cell of the FIFO queue is transmitted, *FC* decrements by one, and the first connection identifier in *SL*, says *j*, is removed from *SL*. Subsequently, our algorithm reduces $\tilde{X}_j$ to $\tilde{X}_j - 1$ and CC_i to $CC_i - 1$, regardless of which VC the transmitted cell actually belongs to. When *FC* is equal to zero or *SL* becomes empty, reset FC to $\tilde{T}$ and CC_i to W_i, where $i = 1,2,\ldots,N$. Meanwhile, eligible connections are appended to the tail of *SL* in sequence.

3.2 The Differential Packet Discarding Buffer Management

Since TCP reacts to packet losses by reducing its congestion window and hence its sending rate, TCP throughput can be controlled by appropriately dropping packets. In the following, we present the Differential Packet Discarding（DPD）buffer management scheme based on EWRR. DPD controls TCP rates based on both per-VC accounting and threshold. The scheme stores the *virtual* buffer occupancy（$\tilde{X}_i$）as the state variable to indicate the number of cells of each active VC（VC_i）, where *active* VCs are those satisfy $\tilde{X}_i > 0$. Note that we only consider the GFR.2 conformance definition with DPD.

The Differential Packet Discarding policy uses the current buffer occupancy as an indication of network load. The scheme tries to maintain an optimal load so that the network is efficiently utilized, yet not congested. Figure 1 [14] illustrates the performance of a network in response to increasing load. The shadowed portion represents the operation region of VBS. When the load is low, increasing the load increases the throughput, but it does not significantly increase the delay. When the load reaches the capacity of the network, further increasing the load increases delay but not throughput. The increase in delay is due to queueing in the network. An excessive increase in load results in a decrease of effective throughput and a drastic increase in delay. This is because of queue overflow which leads to packet loss and thus retransmission, and a decrease in throughput will be experienced by the higher layer user. In the graph, the point on which the load equals to the capacity is called the knee. The point where the throughput falls is called the cliff [19]. Figure 2 illustrates three global thresholds on the output queue, where *L*, *M* and *H* represent the low threshold, the middle threshold and the high threshold respectively and *B* is the queue size. The high threshold（*H*）and the low threshold（*L*）represent the cliff and the knee of the load corresponding to the delay/throughput graph respectively. When the total buffer occupancy（*X*）falls below *L*, the network is unlikely to be congested and all incoming packets are admitted. When *X* grows to higher than *H*, *EPD* is performed for congestion control. When *X* is between *L* and *H*, the policy drops tagged（CLP=1）packets according to the total buffer occupancy, the per-VC dynamic thresholds and the virtual buffer occupancy of the VC. In order to provide MCR guarantees to GFR VCs, untagged（CLP=0）packets are always accepted unless *X* exceeds *H*.

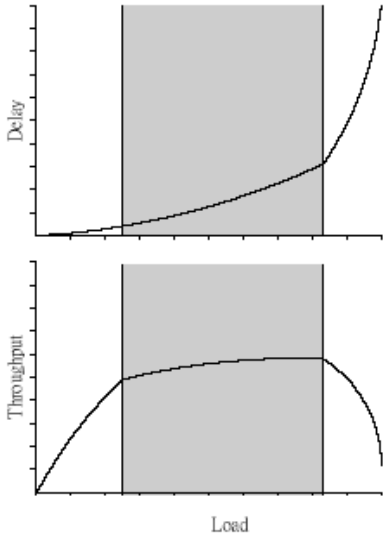


Fig. 1. Load versus delay and throughput.

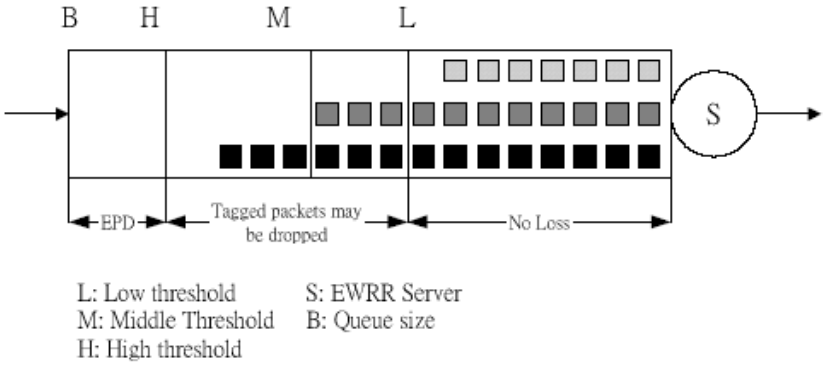


Fig. 2. The three thresholds for the output queue.

Among the active flows, a VC is assigned one of the three levels of drop precedence（DP0, DP1, DP2）. To ease the description, we denote DP_i, which is an integer and $0 \le DP_i \le 2$, as the drop precedence of VC_i, In addition, TH_i represents the per-VC dynamic threshold for VC_i, defined as follows:

$$TH_i = H \times \frac{W_i}{\sum_{j=1}^{N_a} W_i} \qquad (2)$$

where N_a is the number of active VCs. Assume F_i is a flag bit, the combinations of DP_i and F_i form four different states, as shown in Figure 3. VC_i may fall into one of the four states and the state transition only occurs at the arrival of a new packet. Initially, VC_i is in state 1（$DP_i = 1, F_i = 1$）. It would transit to state 2（$DP_i = 2, F_i = 0$）if $\tilde{X}_i \ge Z \times TH_i$, where Z is a parameter and $0 \le Z \le 1$, and to state 3（$DP_i = 0, F_i = 1$）if $X \ge H$. In case that

VC_i is in state 2 and $\tilde{X}_i < Z \times TH_i$, it gets back to state 1. On the other hand, it changes VC_i's state from state 3 to state 4（$DP_i = 1, F_i = 0$）supposing that $\tilde{X}_i < Z \times TH_i$. Subsequently, VC_i in state 4 transits to state 1 if $X \geq M$ or to state 2 if $\tilde{X}_i > TH_i$. Now we can discuss the packet discarding policy if X is between L and H. When $L \leq X < M$, the network is in mild congestion, discarding the newly arriving tagged packets from VCs in state 2. When $X \geq M$, the previous discarding operation cannot relieve congestion, hence the tagged packets from VCs in either state 2 or state 4 will be dropped.

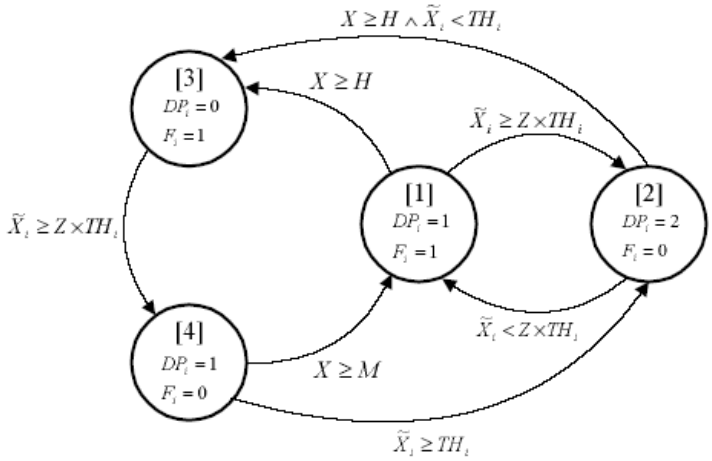


Fig. 3. The state transition diagram of VC_i

The main application of the GFR service is to carry TCP traffic over ATM. It is a well-known fact that when TCP flows with different round-trip times（RTTs）traverse the same link, flows with larger RTT flows get less than their fair share of link resources. We say VC_i gets less buffer space than its fair share if VC_i is in either state 1 or state 3. As described in above paragraphs, VCs in state 1 and state 3 are given precedence in buffer allocation over others. DPD will admit more tagged packets from those VCs to achieve better fairness. As a consequence the effect of different RTTs can be mitigated to a certain extent.

3.3 VBS Thresholds and Parameter

The operation of VBS is based on three global thresholds（L, M and H）, the per-VC dynamic thresholds TH_i and the parameter Z. Among the thresholds and parameter, H is a static EPD threshold and TH_is are derived from Equation 2. L, M and Z are assigned dynamically to maintain high throughput as well as good fairness.

Before describing how to determine L, M and Z, we propose the so-called time-sliding window rate estimator（TSWRE）, which provides a smooth estimate of the arrival rate and the service rate. TSWRE estimates the service rate upon the departure of every Q_{out} cells and estimates the arrival rate upon the arrival of every Q_{in} cells. To estimate the average service rate, TSWRE maintains two state variables - C_{avg}^{out}, the average service rate, and T_{pre}^{out}, which is the time of last Q_{out} cells departure. While three variables are maintained for the average arrival rate estimation - C_{avg}^{in}, the average arrival rates, $C_{agg\_avg}^{in}$, the aggressive average arrival rate and T_{pre}^{in},

which is the time of last Q_{in} cells arrival. The length of the time-sliding window *Win_length*, which is measured in units of cell time slots, is a constant. The TSWRE estimates the arrival rate as shown in the following algorithm. We omit the service rate estimation algorithm because it can be figured out from the arrival rate estimation algorithm.

Initially:

Win_length is a constant;

$C_{initial} = MAX\left(C_{GFR}, \sum_{i=1}^{N} MCR_i\right);$

$C_{agg\_avg}^{in} = C_{avg}^{in} = C_{avg}^{out} = C_{initial}$;

$T_{pre}^{in} = T_{pre}^{out} = 0$;

Upon every Qin cells arrival, the state variables are updated as follows:

$Cells\_in\_Win\_1 = C_{avg}^{in} \times Win\_length$;

// The following two lines are not required for the service rate estimation //

$Cells\_in\_Win\_2 = C_{agg\_avg}^{in} \times Win\_length$;

$Temp = C_{avg}^{in}$;

$C_{avg}^{in} = MAX\left(C_{initial}, \left(Cells\_in\_Win\_1 + Q_{in}\right) / \left(now - T_{pre}^{in} + Win\_length\right)\right);$

// The following two lines are not required for the service rate estimation //

If $\left(C_{avg}^{in} > Temp\right)$;

$C_{agg\_avg}^{in} = \left(Cells\_in\_Win\_2 + Q_{in}\right) / \left(now - T_{pre}^{in} + Win\_length\right)$;

$T_{pre}^{in} = now$;

Assume the current buffer occupancy is *X* cells, the arrival rate should be reduced if the buffer is to be overflow before *X* cells can be served. That is

$$\left(\frac{X}{service\_rate}\right) \times arrival\_rate \le B \Rightarrow X \le B \times \frac{service\_rate}{arrival\_rate}$$

Therefore, we define the thresholds *L* and *M* as follows:

$$L = B \times \frac{C_{avg}^{out}}{C_{agg\_avg}^{in}} \quad \text{and} \quad M = B \times \frac{C_{avg}^{out}}{C_{avg}^{in}} \tag{3}$$

The parameter *Z* is defined as

$$Z = \frac{C_{avg}^{out}}{C_{agg\_avg}^{in}} \times \frac{B}{H} \tag{4}$$

Both the dynamic thresholds *L* and *M* are inverse proportional to the estimated traffic loads which are derived from the results of TSWRE. Intuitively, TSWRE remembers *Win_length* of past history. Both the average service rate $\left(C_{avg}^{out}\right)$ and the average arrival rate $\left(C_{avg}^{in}\right)$ are smooth estimates of the service rate and the arrival rate respectively over a period of time. The estimated load $\frac{C_{avg}^{in}}{C_{avg}^{out}}$ can smooth out the fluctuation in traffic load. Persistent congestion in the switch is reflected by a high average load. The aggressive average arrival rate $C_{agg\_avg}^{in}$ averages the peaks of the average arrival rate sawtooth, $C_{agg\_avg}^{in} \geq C_{avg}^{in}$. $C_{agg\_avg}^{in}$ can be viewed as a conservative estimation of the average arrival rate. We use $C_{agg\_avg}^{in}$ to calculate L in order to prevent an over-descent of the low threshold because of a long-term small arrival rate.

In sum, DPD starts to drop some tagged packets earlier under excessively heavy load, and in consequence more buffer space is available for untagged packets and for tagged packets from VCs in state 1 and state 3. On the contrary, when the traffic load is just slightly heavy, DPD can improve the throughput by admitting more tagged packets from all VCs. The flowchars of DPD and EWRR are illustrated in Figure 4 and Figure 5 respectively. PS_i is the packet state bit of VC_i and the incoming cell except the first cell of a packet of VC_i is to be dropped if $PS_i = 1$.

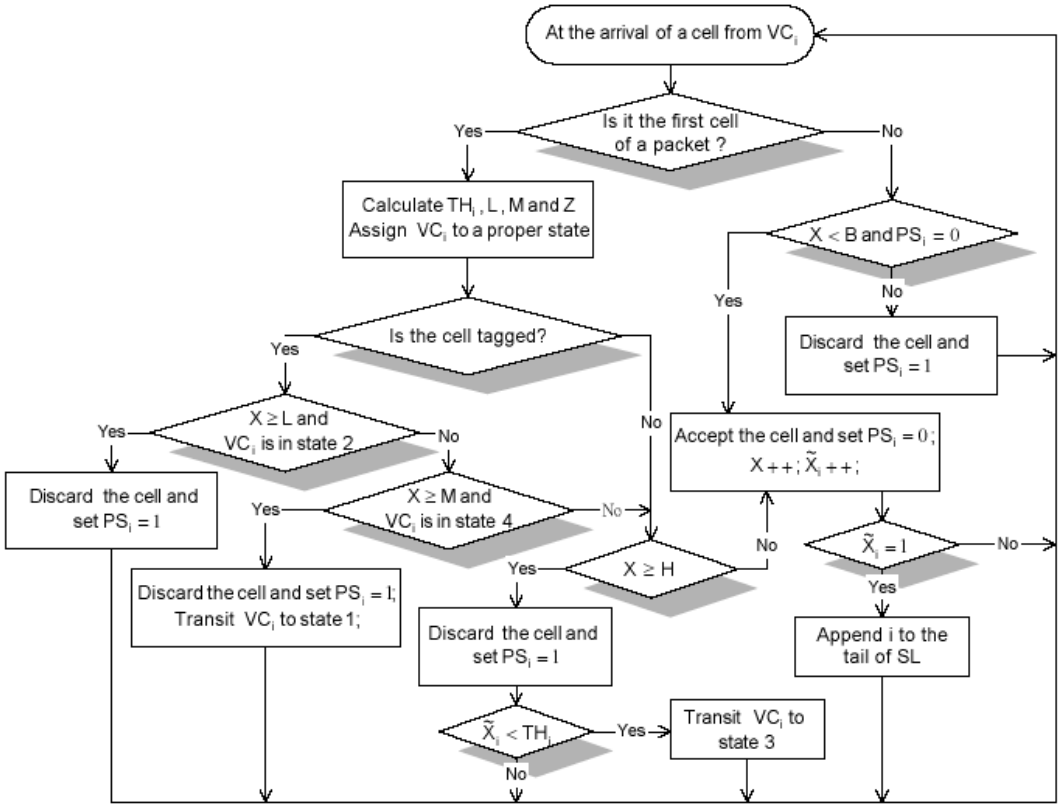


Fig. 4. The flowchart of DPD.

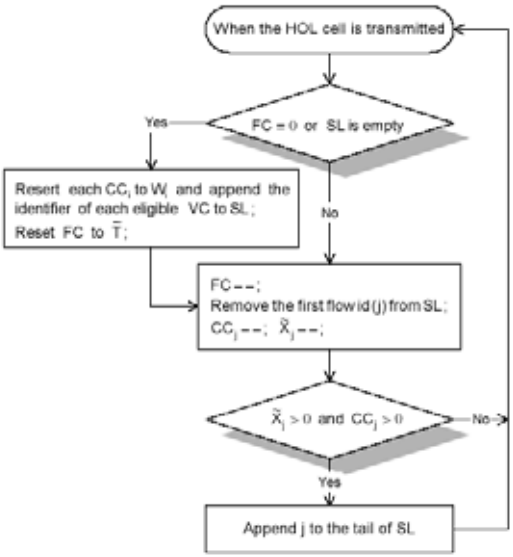


Fig. 5. The flowchart of EWRR.

4. SIMULATION AND PERFORMANCE EVALUATION

4.1 Simulation Environment

A simple network configuration is chosen to illustrate the effect of our proposed strategy, it is shown in Figure 6. The figure shows that 5 local IP/ATM edge switches connected to backbone ATM switches. Each edge switch carries traffic of 10 TCP connections as shown in the figure. The backbone link carries 5 GFR VCs, one from each local network. Each GFR VC carries traffic from 10 TCP connections. All sources are greedy TCP sources. The version of TCP used is New-Reno TCP. All network links are 150 Mbps and therefore the cell transmission time is about 2.83 μs which is defined as a cell time slot.

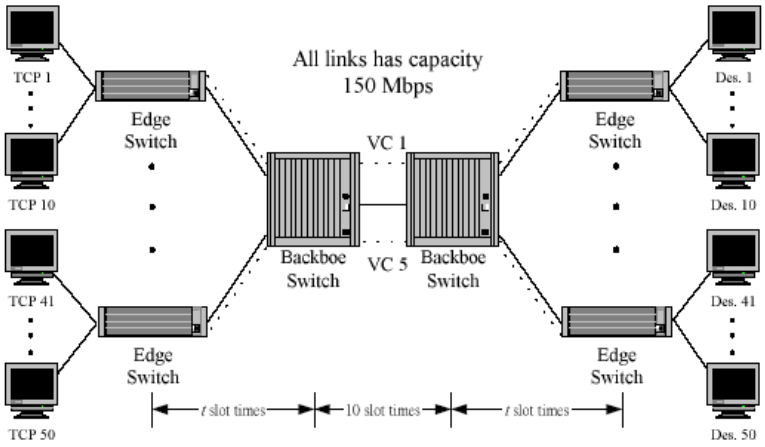


Fig. 6. The simulation configuration.

All required TCP/IP and ATM layer are implemented in our simulation. The maximum TCP segment size is 512 bytes which results in 568 bytes of AAL5 PDU. The AAL5 PDU is padded to produce12 cells at the ATM layer. On the receiving side, the received cells are re-assembled and restored into IP packets, and then passed to

the TCP layer.

In the simulations, we investigate and compare the performance characteristics of our proposed strategy with DFBA scheme. The GFR capacity is fixed to the link rate. We use TSWRE to estimate the arrival rate and *L*, *M* and *Z* can be derived hereby. The high threshold *H* is set to $B-3\times PL$, where *PL* is the packet size. The time-sliding window has length of 1,000 cell time slots and Q_{in} is chosen to be 36 cells. The parameters used in DFBA are based on the simulation model proposed in [7]. The peak cell rate（PCR）for each VC is 50 Mbps for both schemes. The major performance measures considered here are the throughput and the fair sharing of the network resource when we adjust the following parameters:

- **Per-VC MCR allocations:** Two sets of MCR allocations are employed for per-VC MCR allocation. In the first set, the MCR values for VCs 1…5 are approximately 5,10,15,20 and 25 Mbps respectively, giving a total MCR allocations of 50% of the GFR capacity. In the second set, the MCR values for VCs 1…5 are approximately 8.5,17,25.5,34 and 42.5 Mbps respectively. As a consequence, it results in a total MCR allocation of 85% of the GFR capacity.
- **Buffer size:** We first used a large buffer size of 25K cells in the bottleneck backbone switch. We also compare the performance of DFBA and VBS with buffer size of both 6K cells and 3K cells.
- **Heterogeneous round trip time:** In most cases, we test the performance of DFBA and VBS for VCs with identical RTTs. All RTTs are 40 slot times（$t = 5$）. Since TCP throughput is inherently dependent on the RTT, we test the performance of VBS for VCs having heterogeneous RTTs, by increasing the RTT of VC5 to 220 slot times（$t = 50$）

4.2 Simulation Results

The throughput and fairness comparisons between DFBA and proposed VBS under different buffer sizes and different per-VC MCR allocations are shown in Figure 7 through Figure 12 and Table 1. The fairness measure is based on the Fairness Index defined as $\left(\sum_{i=1}^{N} c_i / f_i\right)^2 / \left(n \times \sum_{i=1}^{N} (c_i / f_i)^2\right)$, where c_i is the achieved throughput of VC_i and f_i is the fair share of VC_i throughput. The f_is are 10 Mbps, 20 Mbps, 30 Mbps, 40 Mbps and 50 Mbps for VC_1 through VC_5 respectively.

4.2.1 Comparison of throughput and fairness with different buffer sizes and MCR allocations

Figure 7(a) and Figure 8(a) illustrate the performance of DFBA and VBS when 50% and 85% of the GFR capacity is allocated as the MCR values respectively. Although both schemes can achieve MCR guarantees, our proposed scheme provides better throughput than DFBA. As shown in Figure 7(b) and Figure 8(b), VBS improves the total throughput by approximately 6.74 Mbps and 3.67 Mbps respectively. It is attributed to the fact that VBS discards some tagged packets when the buffer occupancy overpasses L, which is adaptive to the traffic load. However, DFBA drops all incoming tagged packet when the buffer occupancy exceeds *LBO*. Therefore, VBS enhances the buffer utilization and reduce the number of retransmitted packets, and consequently, increases the total throughput.

The performance of DFBA with 85% MCR allocation is much better than that with 50% MCR allocation. The main reason is that DFBA performs EPD on tagged packets when the buffer occupancy exceeds *LBO*. Since

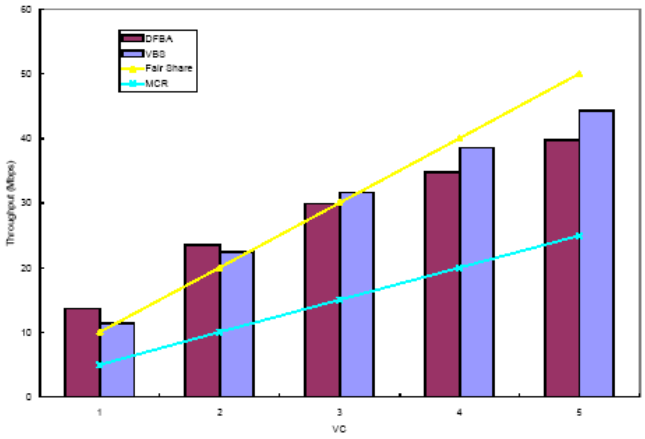
F-GCRA is used to tag those non-eligible AAL5-PDUs at the ingress of the ATM backbone, there are more tagged packets for 50% MCR allocation. Therefore, more packets are dropped and this results in inferior throughput.

The fairness index shown in the bottom rows of Figure 7(b) and Figure 8(b) indicates the VBS can maintain the fair share of buffer space. We attribute the merit to the combined action of EWRR, which maintains the per-VC virtual buffer occupancy, and DPD, which fairly allocate buffer space to competing VCs based on the virtual buffer occupancy and the dynamic thresholds. As to DFBA, it admits all incoming packets when the buffer occupancy is below *LBO* and discards all tagged packets once the buffer occupancy exceeds *HBO*, regardless the actual buffer occupancy of each VC. When the buffer occupancy is between *LBO* and *HBO*, DFBA intends to maintain fairness by dropping untagged packets in a probabilistic manner. When there is no enough amount of untagged packets, DFBA will suffer inferior fairness. It is clear that DFBA achieves better fairness for 85% MCR allocation than for 50% MCR allocation, as shown in Figure 7(b) and Figure 8(b).

By setting the buffer size of the bottleneck backbone switch to 6K cells and 3K cells, Figure 9 through Figure 12 show the effect of buffer size on the performance of both schemes. These four figures show that both DFBA and VBS can provide MCR guarantees even when the bottleneck backbone-switch has small buffer. Obviously, the total throughput decreases as the buffer size is reduced. From Figure 9(b) and Figure 11(b), the total throughput degradation is approximately 8.7 Mbps and 8.8 Mbps in DFBA, and 3.7 Mbps and 5.4 Mbps in VBS respectively. It is clear that DFBA has more serious throughput degradation than VBS when the buffer size is reduced from 25K cells to 6K cells and 3K cells. Similar results can be observed in Figure 10(b) and Figure 12(b), in which the total MCR allocation is 85% of the GFR capacity. The proposed VBS scheme has smaller throughput degradation for 50% MCR allocation and the difference is much smaller for 85% MCR allocation.

Table 1. Effect of heterogeneous RTTs（50% MCR allocation, Buffer size=6K cells）.

VCs	Achieved Throughput (Mbps)			
	DFBA		VBS	
	Identical RTTs	Different RTTs	Identical RTTs	Different RTTs
1	12.993	13.12	11.114	11.627
2	21.721	22.631	21.779	22.359
3	27.565	29.138	30.94	31.27
4	32.96	34.159	37.92	38.117
5	37.591	33.81	42.814	41.172
Total	132.83	132.858	144.567	144.545
Fairness	0.961005	0.953029	0.991321	0.986047

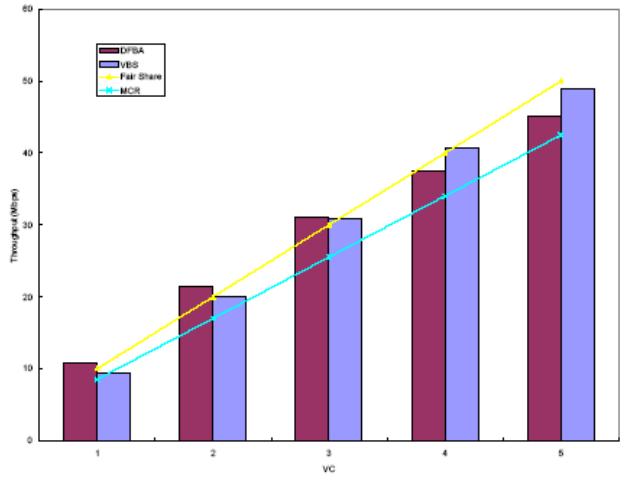


(a)

	DFBA	VBS
Total Throughput	141.553	148.29
Fairness Index	0.961566	0.99149

(b)

Fig. 7. (a) Per-VC throughput. (b) Comparison of DFBA and VBS.（50% MCR allocation, Buffer size=25K cells）

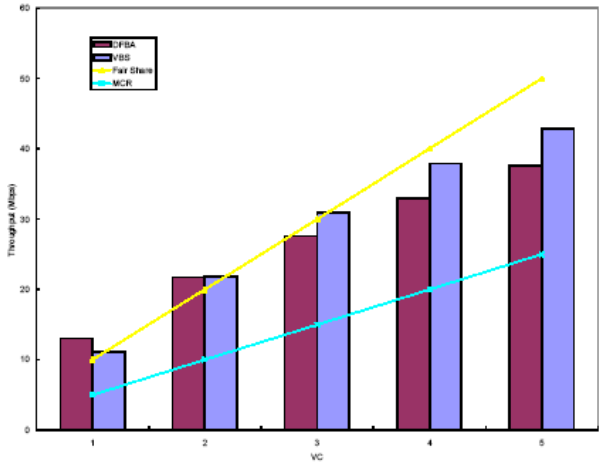


(a)

	DFBA	VBS
Total Throughput	146.04	149.702
Fairness Index	0.994637	0.999029

(b)

Fig. 8. (a) Per-VC throughput. (b) Comparison of DFBA and VBS.（85% MCR allocation, Buffer size=25K cells）

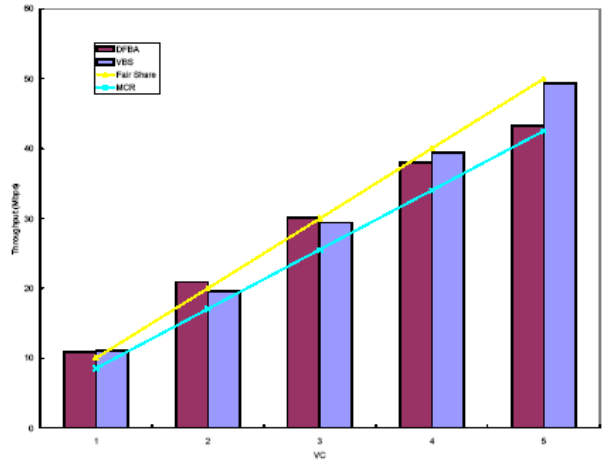


(a)

	DFBA	VBS
Total Throughput	132.83	144.567
Fairness Index	0.961005	0.991321

(b)

Fig. 9. (a) Per-VC throughput. (b) Comparison of DFBA and VBS.（50% MCR allocation, Buffer size=6K cells）

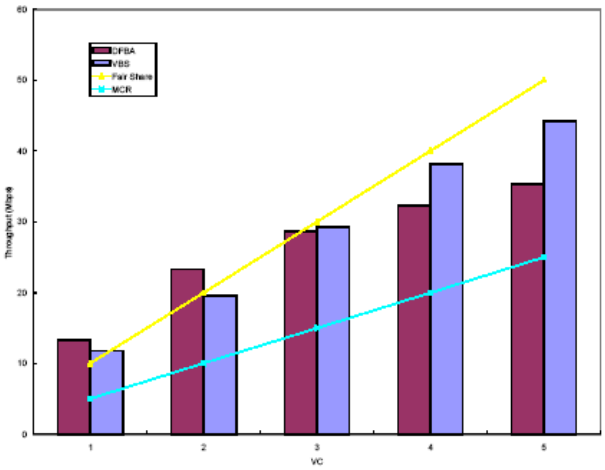


(a)

	DFBA	VBS
Total Throughput	143.149	148.636
Fairness Index	0.993828	0.997768

(b)

Fig. 10. (a) Per-VC throughput. (b) Comparison of DFBA and VBS.（85% MCR allocation, Buffer size=6K cells）

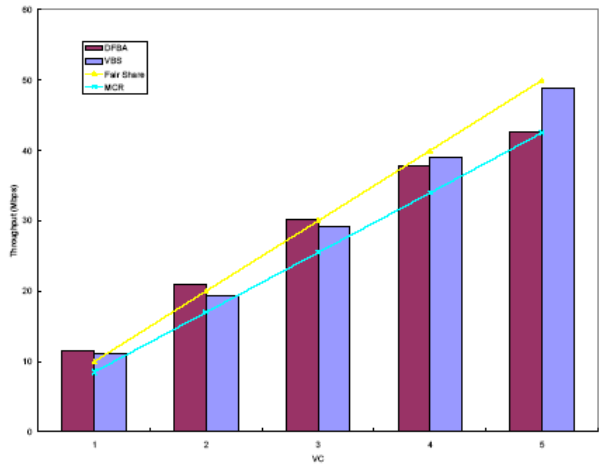


(a)

	DFBA	VBS
Total Throughput	132.771	142.945
Fairness Index	0.949308	0.99049

(b)

Fig. 11. (a) Per-VC throughput. (b) Comparison of DFBA and VBS.（50% MCR allocation, Buffer size=3K cells）



(a)

	DFBA	VBS
Total Throughput	142.882	147.65
Fairness Index	0.990279	0.996866

(b)

Fig. 12. (a) Per-VC throughput. (b) Comparison of DFBA and VBS.（85% MCR allocation, Buffer size=3K cells）

4.2.2 Heterogeneous round trip time

In this configuration, VC5 has a RTT of 220 slot times and all other RTTs are with 40 slot times. We use a buffer size of 6K cells and an MCR allocation of 50% of the GFR capacity. The throughput of each VC for both heterogeneous RTTs and homogeneous RTTs are shown in Table 1. Although both DFBA and VBS can provide MCR guarantees and similar total throughput for these two simulations. The table demonstrates that the throughput degradation of VC5 is 3.781 Mbps for DFBA, however, it is only 1.642 Mbps for VBS. The bottom row of Table 1 shows that VBS achieves better fairness. Therefore, VBS can mitigate the effect of different RTTs.

From the simulation results presented above, we can observe that VBS possesses higher throughput as well as better fairness under different simulation environments. We can conclude that VBS is more efficient than DFBA. From an implementation point of view, both VBS and DFBA need to maintain per-VC states. The difference is that VBS keeps track of virtual buffer occupancy and the additional complexity is to maintain the service list as described in section III.A. While VBS leaves out the process of indentifying the VC the newly departure cell belongs to. Another noticeable extra work of VBS is the rate estimator（TSWRE）. However, the design of TSWRE is extremely simple and no per-VC state is required. Consider the advantages of VBS over DFBA, the additional work is cost effective.

5. CONCLUDING REMARKS

The GFR service may be the key solution for providing the TCP traffic over ATM networks. In this work, we have shown that it is possible to fulfill the requirements of GFR service with FIFO buffer. We have presented the Virtual Buffering Strategy that is composed of two major schemes: the Emulated Weighted Round Robin scheme and the Differential Packet Discarding buffer management scheme. VBS uses per-VC accounting and FIFO queueing to provide minimum rate guarantees as well as fair sharing of bandwidth to VCs carrying TCP traffic. VBS can also mitigate the effect of heterogeneous RTTs. The simulation results demonstrate that VBS outperforms DFBA in both throughput and fairness.

In this paper, we concern VCs with several aggregated TCPs because individual TCPs are usually not visible in an ATM network. Therefore, the fair sharing of the VC throughput among the TCPs within the VC is not addressed in this article. As described in [19], the fairness issue can be solved by deploy FRED [18] at the edge of the ATM network and a different instance of FRED would exist for each VC.

REFERENCES

[1] ATM Forum *Traffic Management Specification Version 4.1*, December 1998.

[2] S. Kalyanaraman *An Update on ATM Traffic Management*, IEEE Network, May/June 1998, pp. 5-7..

[3] R. Guerin and J. Heinanen, *UBR+Service Category Definition*, ATM Forum/96-1598, December 1996.

[4] S. K. Pappu and D. Basak, *TCP over GFR Implementation with Different Service Disciplines*: *A Simulation Study*, ATM Forum/97-0310, May 1997.

[5] D. Basak and S. K. Pappu, GFR *Implementation Alternatives with Fair Buffer Allocation Scheme*, ATM Forum 97-0528, July 1997.

[6] R. Goyal, R. Jain, S. Fahmy, B. Vandalore and S. Kalyanaraman, *GFR-Proving Rate Guarantees with FIFO*

Buffer to TCP Traffic, ATM Forum 97-0831, September 1997.

[7] R. Goyal, R. Jain, S. Fahmy and B. Vandalore, *Buffer Management for the GFR Service*, ATM Forum/98-0405.

[8] O. Bonaventure, *Providing Bandwidth guarantees to Internetwork Traffic in ATM Networks*, Proceedings of ATM'98, May 1998, pp. 269-278.

[9] O. Elloumi and H. Afifi, *Evaluation of FIFO Based Buffer Management Algorithms for TCP over Guaranteed Frame Rate Service*, Proceedings of ATM'98, May 1998, pp. 65-74.

[10] D. Wu and H. J. Chao, *TCP/IP over ATM-GFR*, Proceedings of ATM'98, May 1998, pp. 261-268.

[11] K. Y. Siu, Y. Wu and W. Ren, *Vitrual Queueing Techniques for UBR+ Service in ATM with Fair Access and Minimum Bandwidth Guarantee*, Proceedings of Globecom'97, 1997, pp.1081-1085.

[12] Y. Wu, K. Y. Siu and W. Ren, *Improved Virtual Queueing and Dynamic EPD Techniques for TCP over ATM*, Proceedings of ICNP'97, pp. 212-219, 1997.

[13] A. Romanow and S. Floyd, *Dynamic of TCP Traffic over ATM Networks*, IEEE Journal on Selected Area in Communications, Vol. 13, No. 4, May 1995, pp. 633-641.

[14] D. Chiu and R. Jain, *Analysis of Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks*, Journal of Computer Networks and ISDN Systems, June 1989, pp. 1-14.

[15] M. Casoni and J. S. Turner, *On the Performance of Early Packet Discard*, IEEE Journal on Selected Area in Communications, Vol. 15, No. 5, June 1997, pp. 892-902.

[16] H. Zhang, *Service disciplines for guaranteed performance service in packet-switching networks*, Proceedings of the IEEE, Vol.83, No 10, October 1995, pp. 1374-1396.

[17] J. Heinanen and K. Kilkiki, *A Fair Buffer Allocation Scheme*, Computer Communications 21（1998）, pp. 220-226.

[18] D. Lin and R. Morris, *Dynamics of Random Early Detection*, Proceedings of SIGCOMM97, 1997, pp. 127-137.

[19] R. Goyal, *Traffic Management for TCP/IP over Asynchronous Transfer Mode（ATM）Networks*, PhD Dissertation, The Ohio State University, 1999.