Computer Communications 23 (2000) 1389–1399

computer communications

www.elsevier.com/locate/comcom

An intelligent buffer management approach for GFR services in IP/ATM internetworks

Pi-Chung Wang[a,*], Chia-Tai Chan[b], Yaw-Chung Chen[a]

[a]Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan, ROC
[b]Telecommunication Laboratories, Chunghwa Telecom Co. Ltd, Taipei, Taiwan, ROC

Abstract

In ATM networks, the Guaranteed Frame Rate (GFR) service has been proposed to accommodate non-real-time traffic with packet size not exceeding a maximum length, such as TCP/IP based traffic. The GFR service not only guarantees a minimum throughput at the frame level, but also supports a fair share of available resources. In this article, we propose a buffering strategy called Intelligent Fair Buffer Allocation (IFBA) that provides minimum cell rate (MCR) guarantees and fair sharing to GFR virtual circuits (VCs). From the simulation results, we demonstrate that IFBA fulfills the requirements of GFR service as well as improves the TCP throughput using the FIFO scheduling. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Guaranteed frame rate; TCP/IP; FIFO scheduling; Virtual push-out

1. Introduction

The Guaranteed Frame Rate (GFR)/UBR + (GFR hereafter) service category is intended for user applications which are neither able to specify the range of traffic parameter values, nor able to comply with the behavior rules required by existing ATM services [1–5]. The goal of the GFR service is to bring the features of ATM performance and service guarantees to users who are currently unable to take these advantages. The GFR service requires minimal interactions between users and ATM networks, but still provides certain level of service guarantees. Clearly, the simplicity of the service specification does come with the cost in terms of the complexity imposed on the network. GFR service provides a minimum packet rate guarantee for traffic streams with maximum packet size no larger than a given value, this is an important issue in GFR specification. In addition, the service also specifies that the excessive traffic of each user should fairly share the available resource.

We define the goodput as the valid throughput in terms of the higher layer protocol. The goodput of TCP over ATM networks may be quite low due to the wasted bandwidth for transmitting cells of corrupted packets, in which at least one cell is missed. Here we assume that the number of error cells can be ignored. The packet fragmentation can result in wasted bandwidth due to invalid packet transmission caused by the cell loss. It has been demonstrated that the packet-discard strategies, i.e. early packet discard (EPD) and partial packet discard (PPD), alleviate packet fragmentation problem and restores goodput [8,9]. It can be found that PPD improves performance to a certain degree, but its goodput is still not optimal. To further improve the performance, EPD has been proposed. In EPD, the entire higher-layer protocol data unit will be dropped when the buffer occupancy reaches a certain threshold. Although the goodput of EPD is better than PPD, it is difficult to decide the value of threshold in EPD. For the small threshold value, the goodput can be improved, but the buffer utilization is inferior. On the other hand, if we use a large threshold, the buffer utilization is high, but it also leads to more corrupted packets.

The GFR service intends to support non-real-time applications. Up to now, the most of multimedia services on Internet are web-based. The expected real-time multimedia services, such as video conference, IP telephony and video on demand, are not matured yet. The GFR service might be a solution ready for current web-based multimedia applications. The identification of these service-guaranteed packets is based on the modified GFRA (F-GCRA) [13]. Ref. [2] defines the F-GCRA as a token counter which is incremented at the minimum cell rate (MCR) rate up to maximum burst size (MBS). MBS is set to 2 × MFS, where MFS is the maximum frame size. When the first cell of a packet arrives,

* Corresponding author. Tel.: + 886-35-731851; fax: + 886-35-727842.
E-mail address: pcwang@csie.nctu.edu.tw (P.-C. Wang).

0140-3664/00/$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.
PII: S0140-3664(00)00184-5

if the token counter is no less than MBS/2, all cells of the packet are deemed eligible ($CLP = 0$) for service guarantees and each of them consumes a token when forwarded. Otherwise, these cells are considered non-eligible ($CLP = 1$) and they will not consume any token.

The packet-based push-out buffering mechanism [14] improves the total TCP throughput and fairness, but it complicates the buffer management. In this study, we propose a GFR traffic control scheme called Intelligent Fair Buffer Allocation (IFBA) that improves our previous work [14] by reducing the implementation complexity of switching system design. Our scheme demonstrates that it is feasible and efficient to support QoS of GFR service through FIFO queuing discipline. The organization of this article is as follows: Section 2 gives an overview of GFR implementation alternatives. In Section 3, we present the proposed IFBA strategy. The simulation model and numerical results are discussed in Section 4. Section 5 concludes the work.

2. Overview of GFR schemes

2.1. TCP flow-control behavior

TCP uses a window-based flow control mechanism, its window-adjustment algorithm consists of two phases. A connection begins with the slow-start phase. When a new connection is established, its congestion window (CWND) is initialized to one segment. Upon receiving an ACK packet, the CWND is increased by one segment. This process continues until it reaches a slow-start threshold (SSTHRESH, typically 65,535 bytes). The sender can transmit up to either CWND or receiver's advertised window, the smaller one will be chosen. It can be shown that CWND increases exponentially every round-trip time. When congestion occurs (indicated by a timeout or by reception of duplicated ACKs), one half of the current window-size value (the smaller value between the CWND and the receiver's advertised window, with a minimum of two segments) is saved in SSTHRESH. Additionally, if the congestion is indicated by an expired timer, the CWND will be set to one segment. If CWND were no greater than SSTHRESH, TCP is in slow-start phase; otherwise it is in congestion avoidance phase. In the latter case, CWND is increased by ((segment size × segment size)/CWND) whenever an ACK is received, and it results in a linear increase of CWND. TCP Reno implements the fast retransmit and recovery algorithms [6] that allow the connection to quickly recover from isolated segment losses.

It is known that fast retransmit and recovery cannot recover multiple packet losses but only cause the exponential increment phase to last a very short time, and the linear increment phase to begin with a very small window. As a result, the TCP operates at a very low rate and loses a certain amount of throughput. TCP new Reno is a modification to fast retransmit and recovery. In TCP new Reno, the sender can recover from multiple packet losses without having to time out [7].

2.2. Related packet-discard strategies

An efficient GFR service scheduling strategy should allow each flow passing through a network node to get a fair resource share. The fair queuing scheduling serves flows in proportion to certain predetermined shares as well as protects the service deterioration of ill-behaved sources. Several fair queuing service disciplines have been discussed extensively in the literature, examples are, weighted fair queuing (WFQ) [16], virtual clock [17], packet-by-packet generalized processor sharing (PGPS) [18,19], and self-clocked fair queuing (SCFQ) [20]. However, the drawback of these dynamic time-priority schemes are the high processing overhead required for tracking the progress of tasks and scheduling the time-stamped packets. It has been suggested in Ref. [4] that a simple rate-guaranteeing discipline (i.e. Weighted Round Robin) with per-VC (virtual circuit) queuing is indeed necessary to ensure GFR service. However, the per-VC queuing would greatly complicate the switching system design.

Several approaches have been proposed to provide rate guarantees to TCP sources through FIFO queuing in ATM networks. The Selected Drop (SD) and Fair Buffer Allocation (FBA) schemes use per-connection accounting to maintain the current buffer utilization for each UBR + VC [10,11]. Since these two schemes compare the per-connection queue lengths with a global threshold, they can just guarantee equal throughput but not specific rate for competing VCs. Besides, they cannot support packet priority, which is based on tagging. Another scheme to implement GFR service through tagging and FIFO queuing has been proposed in Ref. [12], which designates threshold control based on the packet priorities. When the global queue length exceeds the first threshold, packets tagged ($CLP = 1$) as lower priority are dropped. When the queue occupancy exceeds the second threshold, EPD will be performed on all packets. The scheme neither provides per-connection throughput guarantee nor achieves fairness in allocating bandwidth to competing VCs.

Goyal et al. [13] proposed a buffer management scheme called Differential Fair Buffer Allocation (DFBA). It maintains two discard thresholds, the low threshold ($L_{\text{threshold}}$) and the high threshold ($H_{\text{threshold}}$). Tagged packets are discarded when the buffer occupancy exceeds $L_{\text{threshold}}$ and EPD is performed on all packets when the buffer occupancy exceeds $H_{\text{threshold}}$. Between $L_{\text{threshold}}$ and $H_{\text{threshold}}$, the incoming untagged packets ($CLP = 0$) of VC_i are dropped in probabilistic manner if VC_i has larger buffer occupancy than its fair share. The simulation result in Ref. [13] shows that it can provide MCR guarantees to GFR VCs, however, excessive bandwidth cannot be shared in proportional to MCR. Besides, it is difficult to determine

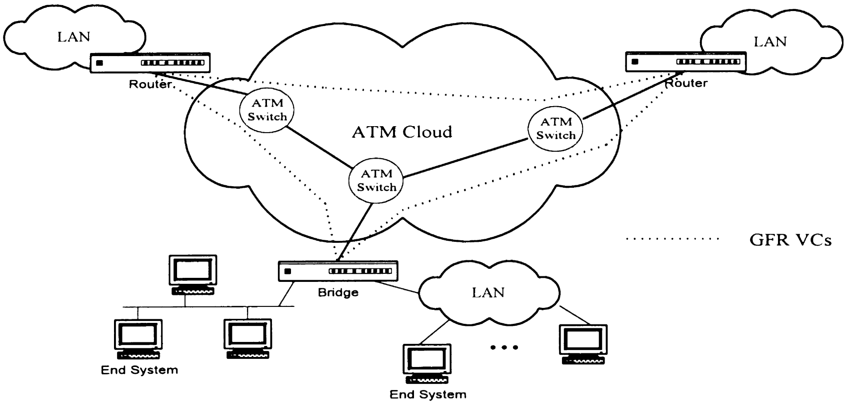

Fig. 1. GFR services in IP/ATM internetworks.

the value of thresholds in DFBA. With a small low-threshold value, tagged packets are dropped unnecessarily, and this results in low TCP throughput. If the low threshold is set to a large value, the TCP throughput can be improved. Nevertheless, it may cause poor fairness. On the other hand, a small high threshold value reduces TCP throughput, while a large value decreases the goodput. In the next section, we propose an intelligent and fair buffer allocation to fulfill the GFR service requirements.

3. Proposed GFR traffic control

GFR can be used between ATM edge devices. For example, IP routers connected through an ATM network can set up GFR VCs between them for data transfer. Fig. 1 illustrates a configuration where the ATM cloud connects LANs and routers. ATM end systems may also establish GFR VC connections between them to obtain a minimum throughput guarantee.

To provide GFR service guarantees, we propose an intelligent fair buffer allocation (IFBA) approach which is constituted by the selective packet-discard associated with a virtual packet-based push-out buffering scheme. In this work, IFBA supports GFR service according to the GFR.2 conformance definition [12]. The conformance tests of the F-GCRA are intended to allow the network to satisfy the GFR service guarantee. By keeping the throughput of untagged cells, it can achieve MCR for GFR service. However, the amount of untagged packets is always insufficient to achieve MCR throughput due to the bursty nature of TCP traffic. In TCP new Reno, when the network suffers congestion, the TCP CWND will decrease to CWND/2 and wait for one-half the round-trip time to retransmit the lost

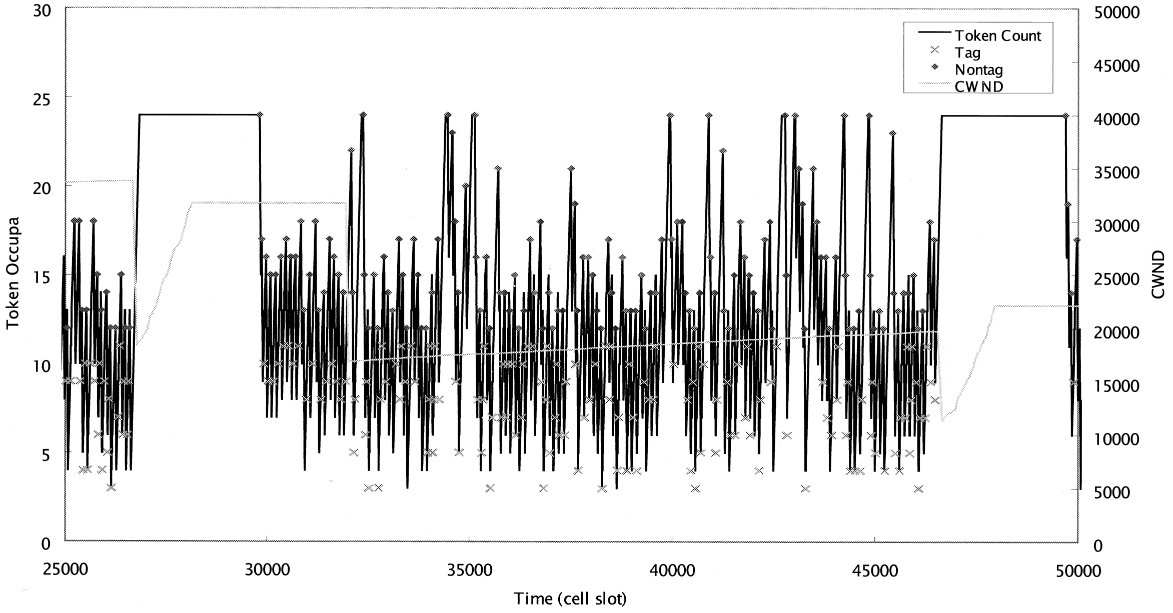

Fig. 2. The relation between TCP CWND and token occupancy.

segments. The tokens generated during this time interval may cause the bucket overflow, as shown in Fig. 2.

In Fig. 2, we set the maximum TCP segment size to 512 bytes, which will be divided into 12 cells of payload. The bucket size is twice the segment size (24 cells), and the token generation rate is MCR (10 Mbps). We can find that once a segment gets lost, TCP CWND will be reduced and the transmission will be halted. Then the bucket may overflow after the time interval (i.e. available bucket size/MCR). Obviously, the throughput of untagged cells cannot satisfy the MCR guarantee through F-GCRA. The implementation of the F-GCRA must be more stringent than that in Ref. [2].

Using FIFO queuing, it needs a specific buffering scheme to police the incoming packets to guarantee the GFR service. A flow can share more service by sending more traffic to keep a higher occupancy in the FIFO queue. To guarantee the fair sharing of the link capacity, an arriving packet should be either admitted or discarded according to the fairness rules. A switch can use per-VC accounting to realize the dynamic sharing of the buffer space among all TCP flows. Let C_{GFR} be the current bandwidth available for GFR traffic in an output port, and r_i be the requested service rate of VC_i, where $1 \leq i \leq n$. In a FIFO service discipline, the service rate r_i can be achieved if VC_i maintains an average buffer occupancy of b_i cells in the FIFO queue, where

$$b_i = \frac{r_i}{C_{\mathrm{GFR}}} \times \sum_{j=1}^{n} b_j \qquad (1 \leq j \leq n)$$

Then VC_i can obtain a service rate r_i ($r_i \geq C_{\mathrm{GFR}} \times (b_i/B)$). That means the throughput experienced by a connection VC_i is proportional to its average fraction of buffer occupancy, which can be preset as a threshold TL_i. Hence, if the buffer occupancy of VC_i is maintained at a desired level, its service rate can also be controlled. In this work, each VC first allocates its MCR and the rest of the available bandwidth is through weighted sharing based on the proportion of MCR. This is to conform the requirement of fairly sharing any excess bandwidth in GFR [21]. We assign an appropriate TL_i to represent the fair share of buffer occupancy for VC_i packets as follows:

$$\mathrm{TL}_i = \frac{\mathrm{MCR}_i + \left(C_{\mathrm{GFR}} - \sum_{i=1}^{n} \mathrm{MCR}_i\right) \times \left(\mathrm{MCR}_i / \sum_{i=1}^{n} \mathrm{MCR}_i\right)}{W} \times B$$

Assume that the output port in an ATM switch uses a FIFO queue of size B (in cell units) for n GFR connections, and the maximum frame size (MFS) for VC_i is L cells ($1 \leq i \leq n$). The switch maintains a state variable B_{abs} to track the available buffer space. Three variables are needed for each VC_i: PL_i is used to count the incoming packet length, NC_i is the actual number of cells of VC_i in the buffer, and TC_i is the total number of tagged cells of VC_i in the buffer.

3.1. Selective packet-discard strategy

Given an ATM switch, assume an output port has a FIFO queue of size B (in cell units) allocated for n GFR connections. Also, we assume that the maximum packet size is L cells.

EPD strategy is based on either a static- or dynamic threshold. In the latter, the switch sets the threshold according to the number of active VCs. Our proposed method also uses a dynamic threshold based on the tracking of the available buffer space. The switch maintains a state variable B_{abs} to estimate the available buffer space and a static threshold TH to perform EPD.

The proposed method operates as follows:

1. Initially, the switch sets the value of B_{abs} to B and PL_i to zero ($1 \leq i \leq n$).
2. If B_{abs} is no less than one, the first cell of a packet arriving at the output buffer will be admitted into the buffer and B_{abs} will be set to $B_{\mathrm{abs}} - L$.
3. Otherwise, check the buffer occupancy. If the buffer occupancy is less than TH, only the untagged packet will be admitted and B_{abs} will be updated to $B_{\mathrm{abs}} - L$. Else, the switch will drop the first arriving cell and all subsequent cells belonging to the same packet.
4. Whenever a cell is served, B_{abs} is incremented by one if it is less than B (i.e. buffer is not empty). When the first cell of a packet from connection i is admitted into the buffer, the PL_i is incremented by one, and the switch starts to count the length of this incoming packet until its last cell is received, then B_{abs} is updated to $B_{\mathrm{abs}} + L - \mathrm{PL}_i$.

Note that there may be multiple cells arriving within a cell-slot time, in such case, the cell service sequence will follow the predetermined order. The switch can identify the last cell of the incoming packet by checking the ATM-layer user-to-user (AUU) indication bit specified in the cell header. Once a switch discards the first cell of a packet, it continues to discard the subsequent cells of the same incoming packet.

By tracking the available buffer space, the dynamic threshold packet-discard method efficiently improves the buffer utilization. Also, the proposed method based on dynamic threshold can achieve almost 100% goodput. Moreover, it improves the TCP throughput to a near-optimal level.

3.2. Virtual packet-based push-out buffering scheme

The push-out buffering scheme ensures the fair sharing of network resource, but it complicates the buffer management. To reduce the complexity, we propose the virtual push-out process (VPOP) which is different from the push-out operation. The main idea of virtual push-out is based on dropping packet from front and re-tagging the packet. Briefly, in our method, if the untagged packet is dropped due to buffer overflow, then the VPOP will

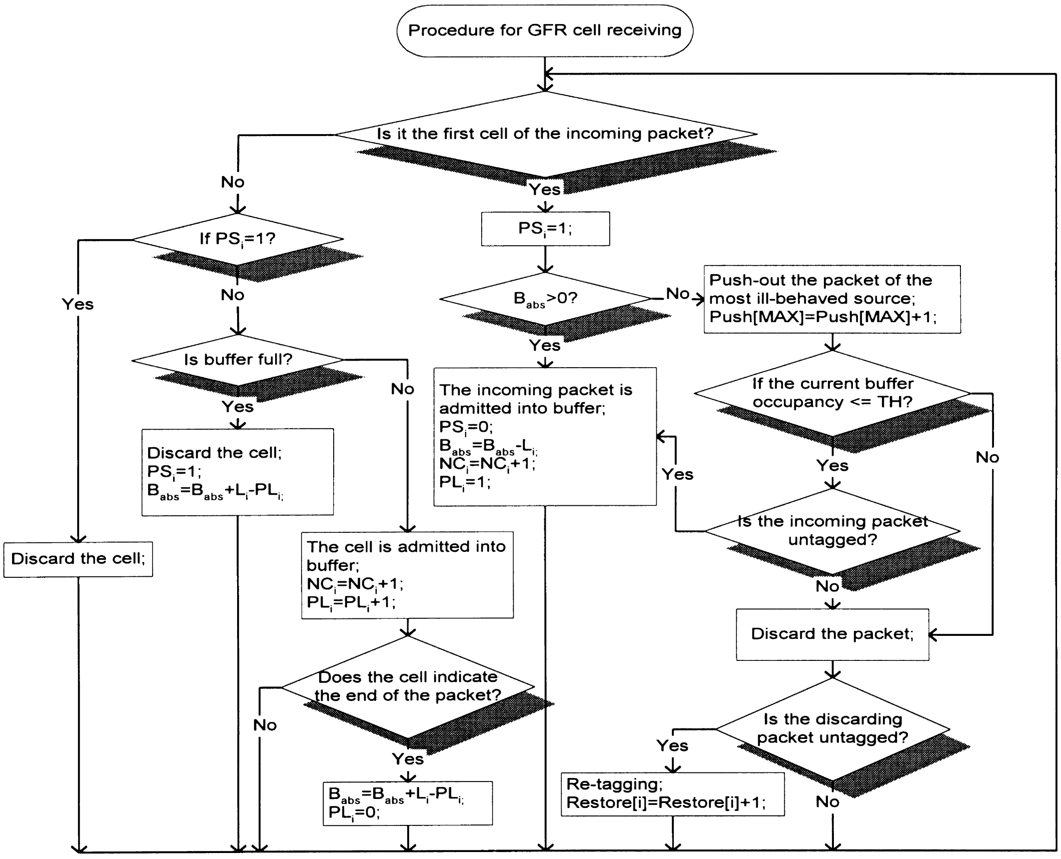
Procedure for GFR cell receiving
Is it the first cell of the incoming packet?
No
Yes
If PS_i=1?
PS_i=1;
Yes
No
B_abs>0?
No
Push-out the packet of the most ill-behaved source; Push[MAX]=Push[MAX]+1;
Is buffer full?
Yes
No
Yes
The incoming packet is admitted into buffer; PS_i=0; B_abs=B_abs-L_i; NC_i=NC_i+1; PL_i=1;
If the current buffer occupancy <= TH?
Discard the cell; PS_i=1; B_abs=B_abs+L_i-PL_i;
Yes
Yes
No
Is the incoming packet untagged?
Discard the cell;
The cell is admitted into buffer; NC_i=NC_i+1; PL_i=PL_i+1;
No
Discard the packet;
Does the cell indicate the end of the packet?
No
Is the discarding packet untagged?
Yes
Yes
B_abs=B_abs+L_i-PL_i; PL_i=0;
Re-tagging; Restore[i]=Restore[i]+1;
No

Fig. 3. The procedure of cell receiving in IFBA control approach.

re-produce an untagged packet by re-tagging the cell and punish the ill-behaved source by dropping its packet from the head of FIFO.

Two state variables are needed for each VC: Restore[*i*] is the number of discarded untagged packet of VC_j ($1 \leq i, j \leq n$), and Push[*j*] is the number of tagged packets of VC_i to be dropped. Since the untagged packets are dropped only when $B_{\mathrm{abs}} \leq 0$ and the buffer occupancy is larger than TH.

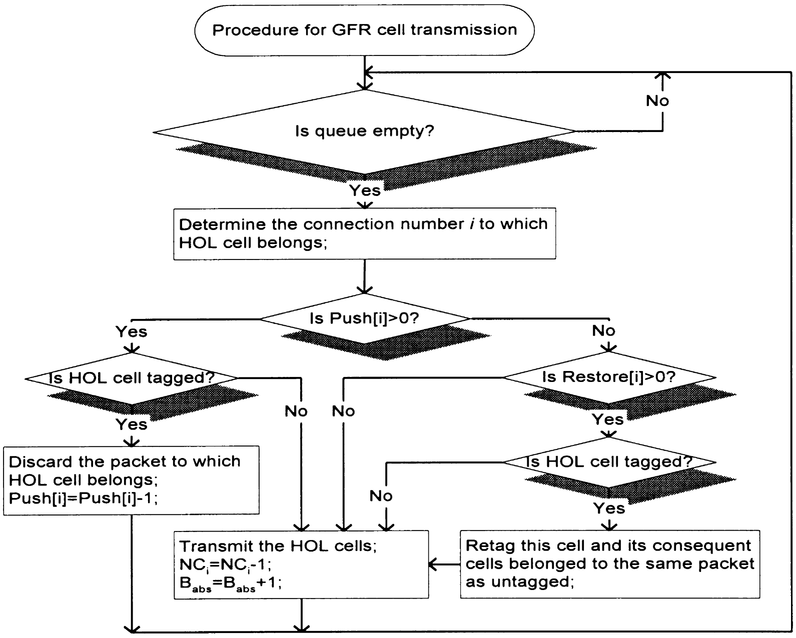
Procedure for GFR cell transmission
No
Is queue empty?
Yes
Determine the connection number i to which HOL cell belongs;
Is Push[i]>0?
Yes
No
Is HOL cell tagged?
Is Restore[i]>0?
No
No
Yes
Yes
Discard the packet to which HOL cell belongs; Push[i]=Push[i]-1;
Is HOL cell tagged?
No
Yes
Transmit the HOL cells; NC_i=NC_i-1; B_abs=B_abs+1;
Retag this cell and its consequent cells belonged to the same packet as untagged;

Fig. 4. The procedure of cell transmission in IFBA control approach.

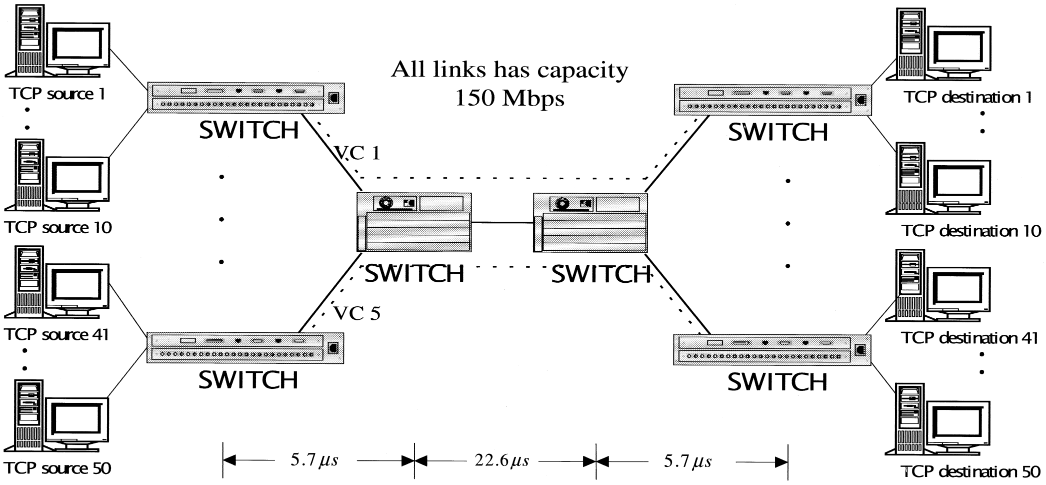

Fig. 5. The simulation configuration.

When the first cell of the incoming untagged packet of VC_i is rejected and $NC_i + L$ is less than or equal to its threshold TL_i, the VPOP will select the maximum ratio of TC_j to TL_j (i.e. the most ill-behaved source) as a candidate. Then, the VPOP increases both Restore[i] and Push[j] by 1.

Consequently, VPOP will perform the virtual packet pushout by dropping the tagged packet from the front. If an HOL cell is the first cell of the tagged packet and Push[j] ≥ 1, it is dropped and all the subsequent cells of the same packet are also discarded. In the mean time, the Push[j] is updated to Push[j] $- 1$. The VPOP is unable to achieve fair buffer sharing immediately, which is different from the push-out operation. But dropping packets from the front will cause duplicate acknowledgements to be sent for whole buffer-drain-time earlier than the case of dropping from the tail.

As described above, correspondingly receiving earlier duplicated acknowledgements in sources cause earlier reduction in window sizes [15]. In all TCP versions, the window is at least halved upon the detection of a loss. Halving the window causes the source to stop sending packets for approximately half a typical round-trip time. The drastic rate reduction from these sources allows cells from other sources to successfully enter the buffer. Thus the congestion can be alleviated earlier and this can reduce or even eliminate the later over-reaction by sources.

If an untagged packet is dropped, the VPOP will record it and try to make up the number of untagged packets by updating tagged packets (CLP $= 1$) as untagged packets (CLP $= 0$). Re-tagging a tagged packet can result in the same effect as pushing out a tagged packet and admitting

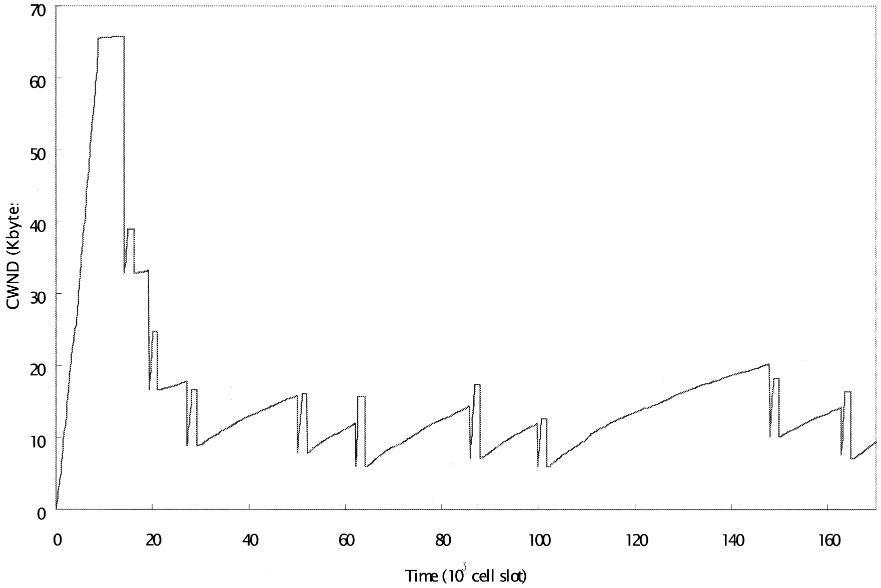

Fig. 6. Congestion-window variation of a TCP connection.

Table 1
Comparison of DFBA and IFBA (50% MCR allocation, Buffer Size = 25K cells)

	DFBA	IFBA
Total throughput	141.553	148.296
Fairness index	0.961566	0.99349

Table 2
Comparison of DFBA and IFBA (85% allocation, Buffer Size = 25K cells)

	DFBA	IFBA
Total throughput	146.04	149.533
Fairness index	0.994637	0.998971

an untagged one. If the number of untagged packets is sufficient to achieve MCR throughput, the proposed method obviously can fulfill the requirements of GFR service.

Assume that the untagged packet with length L is dropped and the length of re-tagged packet is L. A problem of virtual push-out scheme is the variation of packet size. If $L < L$, then the size of the re-tagged packet will be too small to satisfy MCR. Since a block of data transferred to the transport layer is mostly divided into several packets of a fixed size. Only the size of the last packet will be smaller than the maximum segment size L. In addition, it is possible that the re-tagging process increases MCR. The re-tagging process would not affect the guaranteed MCR for the GFR service.

The detailed flowcharts for the procedure of cell receiving and transmission in IFBA control approach are illustrated in Figs. 3 and 4. PS$_i$ is the packet state bit of VC$_i$ and the incoming cell of VC$_i$ will be discarded if PS$_i$ is set to 1. Initially, NC$_i$ and PL$_i$ are set to 0 and B_{abs} is set to the total buffer size.

4. Simulation and performance evaluation

4.1. Simulation environment

A simple network topology shown in Fig. 5 is used to illustrate the effect of our proposed strategy. As shown in the figure, five local IP/ATM edge switches connected to backbone ATM switches with each edge switch carrying traffic from 10 TCP sources. Each TCP source follows the new Reno flow control, the congestion-window variation of a TCP connection is shown in Fig. 6. The backbone link carries five GFR VCs, each comes from a local network. Each GFR VC carries traffic from 10 TCP connections, which are all greedy sources. The allocated GFR bandwidth for each VC are 10, 20, 30, 40 and 50 Mbps, respectively. All network links are OC-3 and therefore the cell transmission time is about 2.78 μs which is defined as a cell slot time. The propagation delay is 2 slot times between adjacent backbone switch, and 8 slot times between backbone switches. We first use a buffer size of 25K cells in the backbone switch and fix the TH at $\text{Buffer Size} - 2 \times L$. We also evaluate IFBA performance with buffer sizes of 6K and 3K cells.

Table 3
Effect of buffer size (50% MCR allocation, Buffer Size = 6K cells)

	DFBA	IFBA
Total throughput	132.83	144.122
Fairness index	0.961005	0.992936

Table 4
Effect of buffer size (85% MCR allocation, Buffer Size = 6K cells)

	DFBA	IFBA
Total throughput	143.149	148.51
Fairness index	0.963828	0.999801

Table 5
Effect of buffer size (50% MCR allocation, Buffer Size = 3K cells)

	DFBA	IFBA
Total throughput	132.771	140.584
Fairness index	0.949308	0.992425

The GFR capacity is fixed at the link rate. Two sets of MCR allocations are employed for per-VC MCR allocation. In the first set, the MCR values for VC1 to VC5 are approximately 5, 10, 15, 20 and 25 Mbps, respectively, giving a total MCR allocations be 50% of the GFR capacity. In the second set, the MCR values for VC1 to VC5 are approximately 8.5, 17, 25.5, 34 and 42.5 Mbps, respectively. This results in a total MCR allocation of 85% of the GFR capacity.

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 produce 12 cells at the ATM layer. On the receiving side, the received cells are re-assembled and into the IP datagram, and then passed to the TCP layer.

In the simulations, we investigate and compare the performance characteristics of our proposed scheme with DFBA scheme [13]. The two thresholds of DFBA, $L_{\mathrm{threshold}}$ and $H_{\mathrm{threshold}}$, are set to 0.5 and 0.9 of the buffer capacity, respectively. The major performance measures considered here are the TCP throughput and the fair sharing of the network resource.

4.2. Numerical results

The throughput and fairness comparisons between DFBA and the proposed IFBA under different buffer sizes and

Table 6
Effect of buffer size (85% MCR allocation, Buffer Size = 3K cells)

	DFBA	IFBA
Total throughput	142.882	147.844
Fairness index	0.990279	0.997619

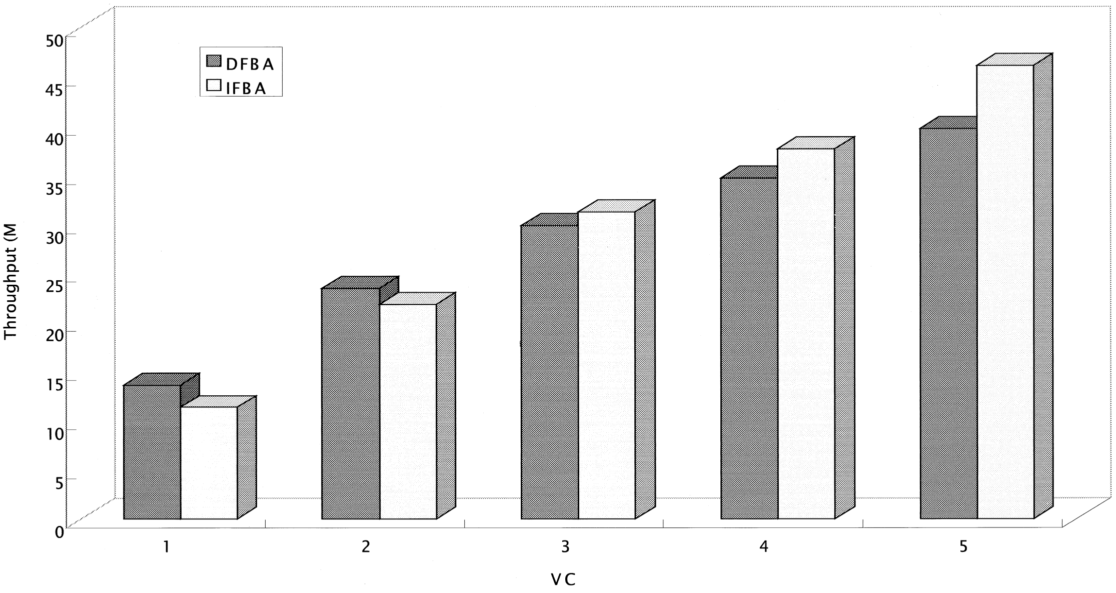

Fig. 7. Per-VC throughput (50% MCR allocation, buffer size = 25K cells).

per-VC MCR allocations are shown in Tables 1–6. The fairness measure is based on the Fairness Index, which is defined as

$$\frac{\left(\sum_{i=1}^{n} x_i/f_i\right)^2}{\left(n \times \sum_{i=1}^{n} (x_i/f_i)^2\right)}$$

where x_i is the actual throughput of VC$_i$ and f_i is the GFR capacity of VC$_i$.

Figs. 7 and 8 illustrate the performance of DFBA and IFBA when 50 and 85% of the GFR capacity are allocated as the MCR values, respectively. Although both schemes can satisfy MCR guarantees, our proposed scheme provides better throughput than DFBA. As shown in Tables 1 and 2, our scheme improves the total throughput by approximately 6.7 and 3.5 Mbps, respectively. IFBA is based on a dynamic threshold instead of static threshold in DFBA. IFBA discards or pushes out some tagged packets only when the network is under potential congestion condition (i.e. $B_{\mathrm{abs}} \leq 0$). However, DFBA drops all incoming tagged packet when buffer occupancy grows larger than $L_{\mathrm{threshold}}$. As a result,

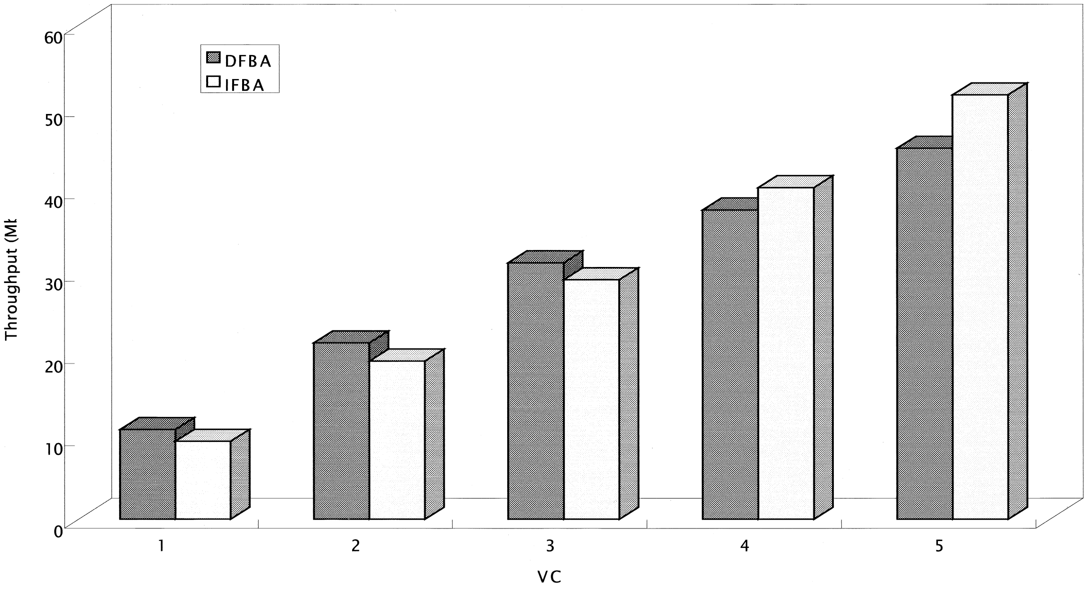

Fig. 8. Per-VC throughput (85% MCR allocation, buffer size = 25K cells).

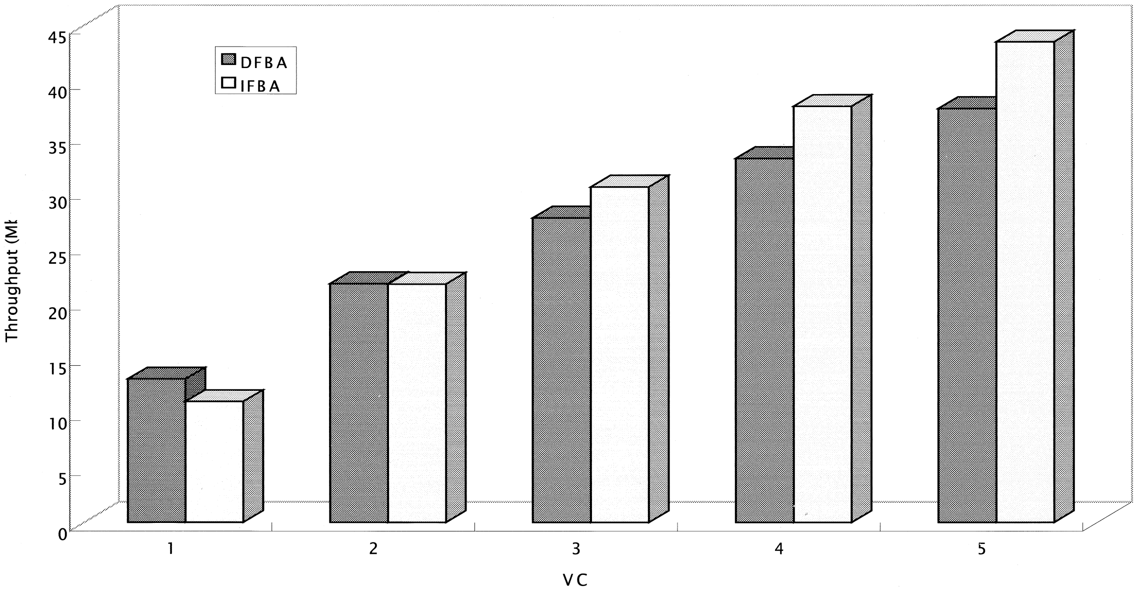

Fig. 9. Per-VC throughput (50% MCR allocation, Buffer Size = 6K cells).

IFBA increases the buffer utilization and reduce the number of dropped packets, and consequently, increases the total throughput.

The performance of DBFA is better for the 85% MCF allocation than that for the 50% MCR allocation. The main reason is because DBFA performs EPD on tagged packets when the buffer occupancy exceeds $L_{\text{threshold}}$. Since F-GCRA is used to tag non-eligible AAL5-PDUs at the edge switch, there are more tagged packets for 50% MCR allocation. Therefore, more packets are dropped. This results in inferior throughput. Besides, there are more untagged packets for 85% MCR allocation hence DFBA can derive better fairness under this case.

The fairness index shown in the bottom rows of Tables 1 and 2 indicates that the fair share of buffer space can be maintained by the virtual push-out process with IFBA. The most unfair case in DFBA is that the buffer space below $L_{\text{threshold}}$ are occupied by ill-behaved sources. Although the DFBA will drop packets from ill-behaved sources once the buffer occupancy gets larger than $L_{\text{threshold}}$, however, it may cause losses of tagged packets from well-behaved sources (i.e. $\text{TC}_i < \text{TL}_i$). The throughput of the VC with the highest allocated MCR will be decreased due to the unfair buffer allocation. From Table 1, the improvement of fairness in IFBA is more clear as the total MCR allocation gets lower. With fewer MCR allocation, the lack of untagged

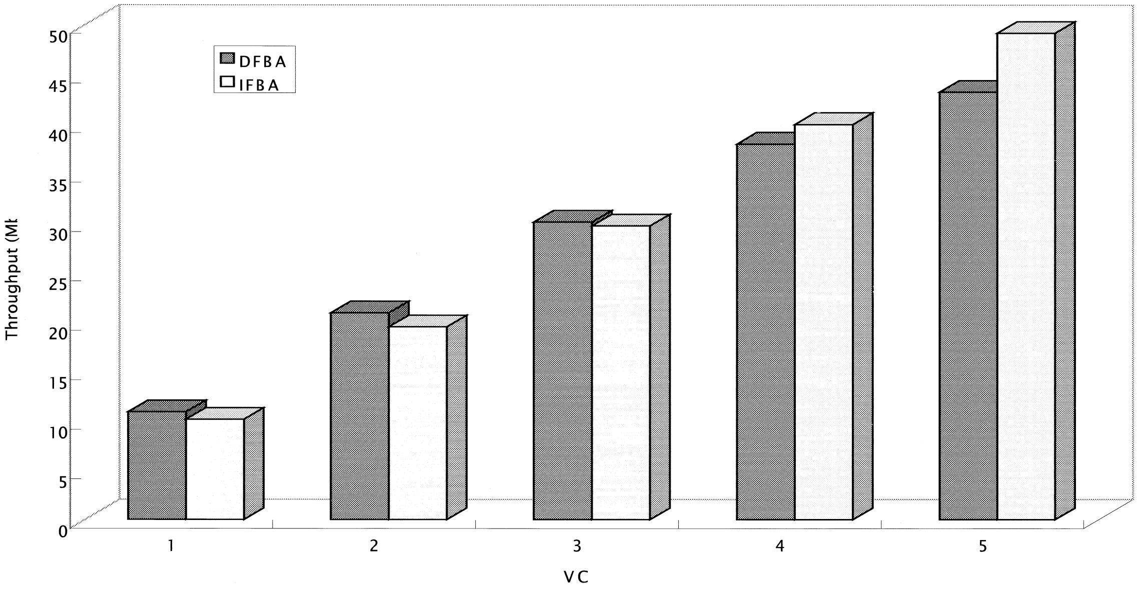

Fig. 10. Per-VC throughput (85% MCR allocation, Buffer Size = 6K cells).

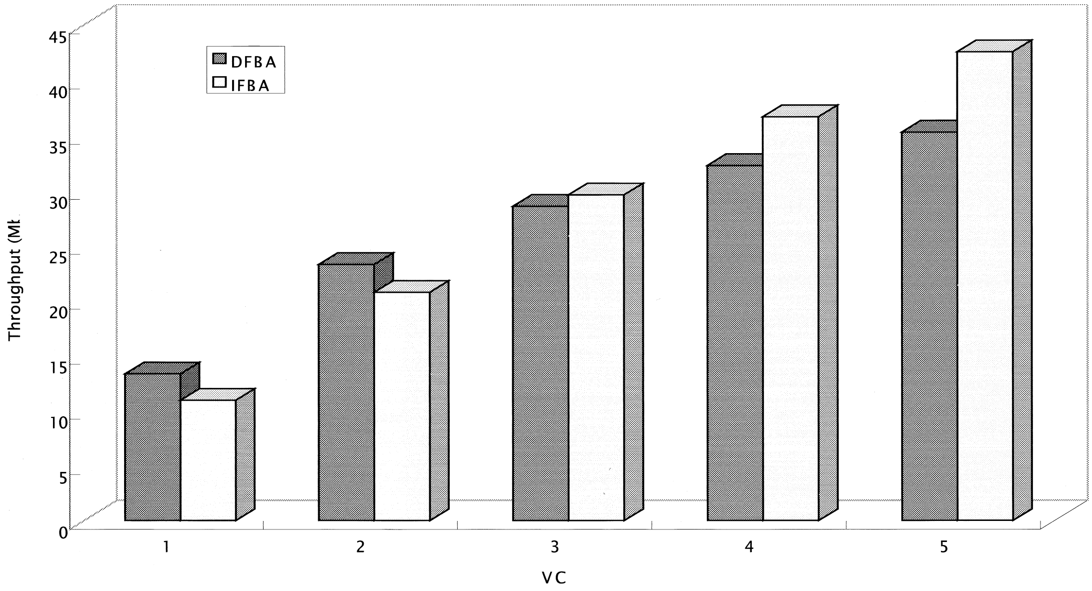

Fig. 11. Per-VC throughput (50% MCR allocation, Buffer Size = 3K cells).

packets will cause unfairness due to the discarding of tagged packets.

By setting the buffer size of the bottleneck backbone-switch to 6K- and 3K cells, we show the effect of buffer size on performance of both schemes in Figs. 9–12 and Tables 3–6. These four figures show that both DFBA and IFBA can provide MCR guarantees even when the bottleneck backbone-switch has small buffers. Obviously, the total throughput decreases as the buffer size is reduced. From Tables 3 and 5, the total throughput degradation is approximately 8.7 and 8.8 Mbps in DFBA, and 4.2 and 7.7 Mbps in IFBA, respectively. It is clear that IFBA experiences less serious throughput degradation than DFBA. Similar results can be observed in Tables 4 and 6 when the total MCR allocation is 85% of the GFR capacity. The proposed IFBA scheme has smaller throughput degradation for 50% MCR allocation and the difference is much smaller for 85% MCR allocation. These numerical results demonstrate that IFBA can achieve higher throughput as well as better fairness with smaller buffer. We can conclude that IFBA is more efficient that DFBA.

5. Conclusions

The Guaranteed Frame Rate service has been designed to

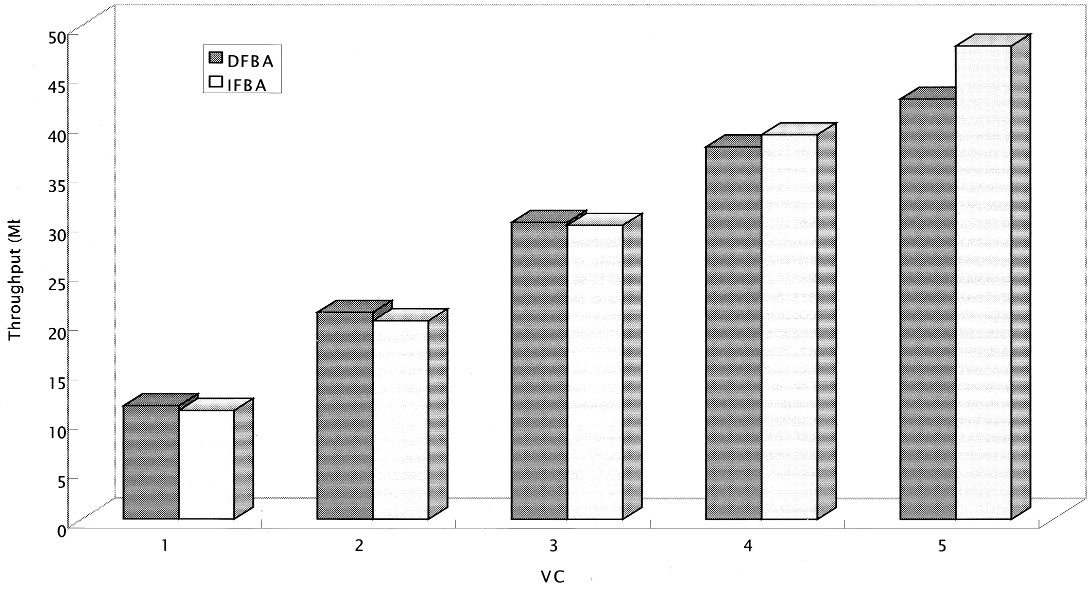

Fig. 12. Per-VC throughput (85% MCR allocation, Buffer Size = 3K cells).

provide the TCP traffic through ATM networks. In this work, we have shown that the proposed control approach is able to fulfill the requirements of GFR service through FIFO queuing service. The selective-discard scheme combined with virtual packet-based push-out mechanism provides a nearly optimal throughput and remarkable fair share of network resource among competing connections. The virtual packet-based push-out scheme reduces the implementation complexity of buffer management as well as improves the total TCP throughput and fairness.

References

[1] N. Yin, S. Jagannath, End-to-End Traffic Management in IP/ATM Internetworks, ATM Forum Cont., 96-1406, October 1996.

[2] R. Guerin, J. Heinanen, UBR + Service Category Definition, ATM Forum Cont., 96-1598, December 1996.

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

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

[5] R. Goyal, R. Jain, S. Fahmy, B. Vandalore, S. Kalyanaraman, GFR-Proving Rate Guarantees with FIFO Buffer to TCP Traffic, ATM Forum Cont., 97-0831, September 1997.

[6] W. Stevens, TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, Internet RFC 2001, January 1997.

[7] R. Goyal, R. Jain, S. Fahmy, S. Fahmy, B. Valdalore, X. Cai, Selective Acknowledges and UBR + Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks, ATM Forum Cont., 97-0423, April 1997.

[8] A. Romanow, S. Floyd, Dynamic of TCP traffic over ATM networks, IEEE Journal on Selected Areas in Communication 13 (4) (1995) pp. 633–641.

[9] M. Casoni, J.S. Turner, On the performance of early packet discard, IEEE Journal on Selected Areas in Communication 15 (5) (1997) pp. 892–902.

[10] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, Seong-Cheol Kim, UBR + : improving performance of TCP over ATM-UBR service, Proceedings of the ICC’97, June 1997.

[11] J. Heinanen, K. Kilkiki, fair buffer allocation scheme, Computer Communications 21 (1998) 220–226.

[12] J. Kenney, Traffic Management Working Group Baseline Text Document, ATM Forum Document BTD-TM-01.03.

[13] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, B. Vandalore, Improving the performance of TCP over the ATM-UBR service, Computer Communications 21 (10) (1998) 898–911.

[14] C.T. Chan, Y.C. Chen, P.C. Wang, An efficient traffic control approach for GFR services in IP/ATM internetworks, Proceedings of the IEEE Globecom’98, pp. 1112–1117.

[15] T.V. Lakshman, A. Neidhardt, J. Teunis, J. Ott, The drop from front strategy in TCP and in TCP over ATM, Proceedings of the IEEE INFOCOM’96, pp. 1242–1250.

[16] A. Demers, S. Keshav, S. Shenker, Analysis and simulation of a fair queueing algorithm, Journal of Internetworking Research and Experience October (1990) 3–26.

[17] L. Zhang, Virtual clock: a new traffic control algorithm for packet switching networks, Proceedings of the ACM SIGCOMM’90, Philadelphia, PA, September 1990, pp. 19–29.

[18] A. Parekh, R. Gallager, generalized processor sharing approach to flow control in integrated services networks: the single-node case, IEEE Transactions on Networking 1 (3) (1993) 344–357.

[19] A. Parekh, R. Gallager, generalized processor sharing approach to flow control in integrated services networks: the multiple-node case, IEEE Transactions on Networking 2 (2) (1994) 137–150.

[20] S. Golestani, A self-clocked fair queueing scheme for broadband applications, Proceedings of the IEEE INFOCOM’94, Toronto, Canada, June 1994, pp. 636–646.

[21] R. Goyal, R. Jain, S. Fahmy, B. Vandalore, Buffer Management for the GFR Service, ATM Forum Cont., 98-0405, July 1998.