A FIFO-Based Buffer Management Approach for the ATM GFR Services

Chia-Tai Chan, Shuo-Cheng Hu, Pi-Chung Wang, and Yaw-Chung Chen, *Member, IEEE*

***Abstract*—We investigate the TCP packet transmission over the ATM Guaranteed Frame Rate (GFR) service using a selective packet-discard with buffer vacancy tracking strategy and a packet-based push-out buffering scheme. Our approach is based on FIFO queueing discipline, it features fair sharing of available resources as well as feasibility in hardware implementation.**

***Index Terms*—Packet-based push-out buffering scheme, selective packet-discard.**

I. INTRODUCTION

GUARANTEED Frame Rate (GFR) has recently been standardized by the ATM Forum. It is expected to be useful for TCP/IP based application. The GFR is intended to provide a minimum rate guarantee to VC's at the frame level. In addition, a connection sending in excess of the minimum rate should receive a fair share of any unused network capacity. The design of a simple and efficient buffer management approach for accommodating GFR service requirements is an important issue toward the successful development of GFR.

Fair queueing disciplines [1] such as weighted fair queueing (WFQ), virtual clock, packet-by-packet generalized processor sharing (PGPS), and self-clocked fair queueing (SCFQ) have high processing overhead for tracking the progresses of tasks. It has been suggested in [8] that a simple rate-guaranteeing discipline (i.e., Weighted Round Robin) with per-VC queueing is indeed necessary to ensure GFR service. However, the per-VC queueing (i.e., each virtual connection uses separate queue) greatly complicates the switching system design. The effective throughput of TCP over ATM networks may be unsatisfactory because packet fragmentation may result in wasted bandwidth for transmitting those cells of corrupted packets, in which at least one cell is dropped, and the packet has to be retransmitted [2]–[4]. The Fair Random Early Drop (FRED) [6], a modified version of RED, uses per-active-flow accounting to provide better protection than RED. However, FRED does not address the packet fragmentation problem. It has been demonstrated that the packet-discard strategies, i.e., early packet discard (EPD) and partial packet discard (PPD), prevent packet fragmentation problem. In this work, we propose a feasible FIFO buffering scheme that provides the requirements of GFR service. The merit of the proposed approach is the efficient and fair sharing of network resources with a feasible implementation method for ATM GFR services. The organization of this paper is as follows. In Section II, we present the functional approach of the proposed scheme. The implementation architecture is discussed in Section III. Section IV concludes the work.

Manuscript received July 13, 1999. The associate editor coordinating the review of this letter and approving it for publication was Prof. N. B. Shroff.

C.-T. Chan is with the Telecommunication Laboratories, Chunghwa Telecom Company, Ltd, Taipei 106, Taiwan, R.O.C. (e-mail: ctchan@ms.chttl.com.tw).

S.-C. Hu, P.-C. Wang, and Y.-C. Chen are with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. (e-mail: {schu@csie.nctu.edu.tw; pcwang@csie.nctu.edu.tw; ycchen@csie.nctu.edu.tw).

Publisher Item Identifier S 1089-7798(00)05693-3.

II. PACKET-DISCARD PUSH-OUT CONTROL APPROACH

Our approach consists of a selective packet-discard with buffer-vacancy tracking strategy and a packet-based push-out buffering scheme (abbreviated as PDPO here after). The latter ensures the fair sharing of the network resources and prevents misbehaved connections from occupying excessive buffer space, while the former avoids the packet fragmentation problem and improves the TCP throughput [5].

A. Selective Packet-Discard with Buffer-Vacancy Tracking Strategy

Assume that an ATM switch has an output FIFO queue of size B(in cell units), and the maximum packet size is L cells. The switch maintains a variable C_{bv} to track the buffer vacancy, and a variable PL_i to count the incoming packet length for the virtual connection VC_i. Initially, the switch sets C_{bv} to B and PL_i ($1 \leq i \leq n$) to zero, where n is the number of VC's. Whenever a cell is transmitted, C_{bv} will be incremented by one. If $C_{bv} > 0$, the first cell of an arriving packet will be admitted and C_{bv} will be set to $C_{bv} - L$. Otherwise, the cell and all subsequent cells of the same packet will be dropped. When the first cell of a packet from VC_i is admitted, the switch starts to count up PL_i until the last cell is received, then C_{bv} is updated to $C_{bv} + L - PL_i$. The switch can identify the last cell of an incoming packet by checking the ATM-layer user-to-user (AUU) indication bit. By tracking the buffer vacancy, the buffer utilization can be improved and the packet fragmentation problem can be avoided.

B. Packet-Based Push-out Buffering Scheme

A flow can acquire larger service rate by sending more traffic to keep higher occupancy of the FIFO buffer. To guarantee fair bandwidth sharing, it should either admit or discard the arriving packet according to the fairness requirement. A switch can use per-VC accounting (i.e., keep a separate record for each VC) to realize the dynamic buffer sharing. Let W be the bandwidth available in an output port, and r_i be the requested service rate of VC_i, where $1 \leq i \leq n$. Using FIFO queueing, the service rate

r_i can be achieved if VC_i keeps an average buffer occupancy of b_i cells, where

$$r_i = W \times \frac{b_i}{\sum_{j=1}^{n} b_j} \geq W \times \frac{b_i}{B}(1 \leq i \leq n). \tag{1}$$

That means the service rate r_i experienced by VC_i will be at least $(b_i/B) \times W$, therefore we set a threshold TL_i for b_i, $TL_i = (r_i/W) \times B$.

Let NC_i be the number of cells of VC_i in the FIFO buffer. When the first cell of a VC_i packet arrives, it is admitted if $C_{bv} > 0$. If $C_{bv} \leq 0$ and $NC_i + L \leq TL_i$, it activates the push-out process. The packet discard controller selects a VC with the maximum value of $NC_j - TL_j$ $(1 \leq j \leq n)$ and pushes out its last packet, whose length is then added to C_{bv}. The push-out process repeats until $C_{bv} > 0$. By using the push-out scheme, a switch can ensure fair sharing of the bandwidth through FIFO queueing.

In [5], it has been demonstrated that the proposed strategy not only provides the service requirements of GFR, but also brings the TCP throughput to its optimal level. Although the push-out process complicates the buffer management, it achieves fair sharing of the network resource and improves the total TCP throughput because it reduces the number of packet retransmissions. Moreover, the selective packet-discard strategy using buffer-vacancy tracking increases the buffer utilization that leads to the improvement of TCP throughput. The detailed discussion of the resulting performance has been shown in our previous work [5].

III. REALIZATION OF PDPO

The proposed PDPO approach can be realized using an architecture consisting of four major function components, a packet-discard controller (PDC), a cell dispatcher (CD), a stack controller (SC) and a push-out controller (POC), as shown in Fig. 1. The PDC performs two tasks: discarding the incoming packets when there is no buffer vacancy and manipulating the control variables NC_i $(0 \leq i \leq n)$ and C_{bv}. The POC consists of many modules for storing the cells' PA's, and performs FIFO queueing and put-out process. The purpose of the SC is to maintain the address of illegal packets of VC's. An arriving packet of VC_i is defined as **legal** if $NC_i + L \leq TL_i$. It will be accepted if there is enough buffer space for the whole packet (i.e., $C_{bv} \geq 1$), otherwise, it will be discarded. The variable NC_i is used to activate/deactivate both POC and SC for VC_i. Once the first cell of a new packet arrives, if $NC_i > TL_i - L$, the physical addresses (PA's) of all packet cells must be stored into the associated stack for possible push-out operation later, because the packet may be illegal. To maintain stacks, the PDC immediately activates the SC to clear the corresponding stack if $TL_i > NC_i \geq TL_i - L$, because these associated cells become legal and therefore cannot be discarded.

An arriving cell is classified by the PDC, then written into the cell pool. Prior to this process, the cell's PA is stored into the POC. If the arriving cell is illegal, the associated stack number (SN) will be extracted by the CD and its PA is pushed into the associated stack. Once the POC receives a push-out signal, it triggers the SC to pop packets out from the stack which has the largest number of illegal packets. Each time the push-out process is activated, at least one packet will be popped out and the total popped-out cells must be no less than L cells, then PA's of these popped-out cells will be released to the idle-address pool. A packet indication bit for identifying the last cell's PA is attached to each entry in a stack.

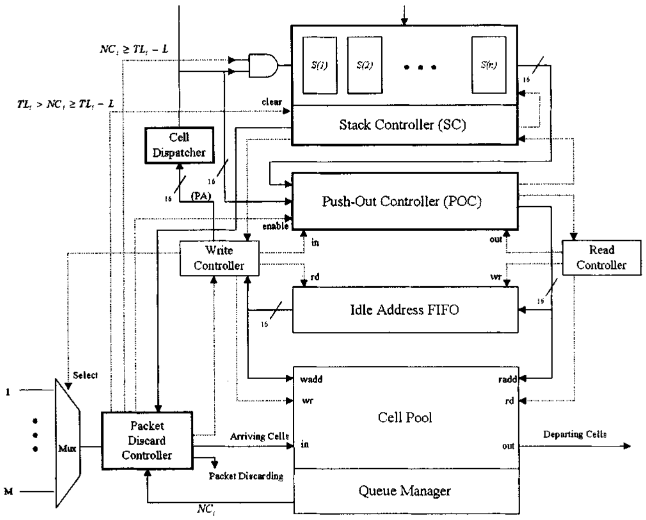

Fig. 1. Implementation architecture of the proposed traffic controller.

Once the SC pops the last cell of a packet, it continues to pop other cells of the same packet. The number of popped cells must be sent to the PDC to update the NC value. In addition, it is possible to pop out a partial packet because the remaining packet cells may be still on the way. The SC must inform the PDC to discard the subsequent cells of the popped packet and update C_{bv}. Since SC needs $O(n)$ comparisons to determine the maximum value of $NC_j - TL_j$, to reduce the complexity, SC maintains the Max_Stack information of the longest stack. If the cell's PA of an arriving packet were stored into its associated stack, the stack length is then increased and compared with the Max_Stack. Each time the SC pops the stack which has the maximum length, it updates the Max_Stack.

A feasible architecture to implement the POC is shown in Fig. 2 that is modified from [7]. The service discipline of POC is simply FIFO and a cell's PA is stored in a 16-bit register. The PA of an incoming cell is appended to the POC FIFO. When a head-of-line(HOL) cell is scheduled for transmission, its PA is retrieved from POC to identify the cell location. Once a cell is transmitted, the content of all 16-bit registers will be shifted one position to the right. In the mean time the PA of the transmitted cell is returned to the idle-address pool.

In the push-out process, when a cell's PA is to be removed, it is broadcasted to all modules. By comparing the PA value, Z_{0-15}, with the value, Q_{0-15}, of module j $(j = 1 \cdots N)$, the control logic is able to decide whether to retain or to shift-right the register content. When the control logic receives a newly broadcasted value, it changes to the shift state. Then it evaluates both values of b_{out} and b_{in}, and shifts right the contents of those modules with $b_{\mathrm{out}} = 1$ because their positions must be to the left of the pushed-out cell. After the shifting process, the popped PA value can be extracted and written into the idle-address pool.

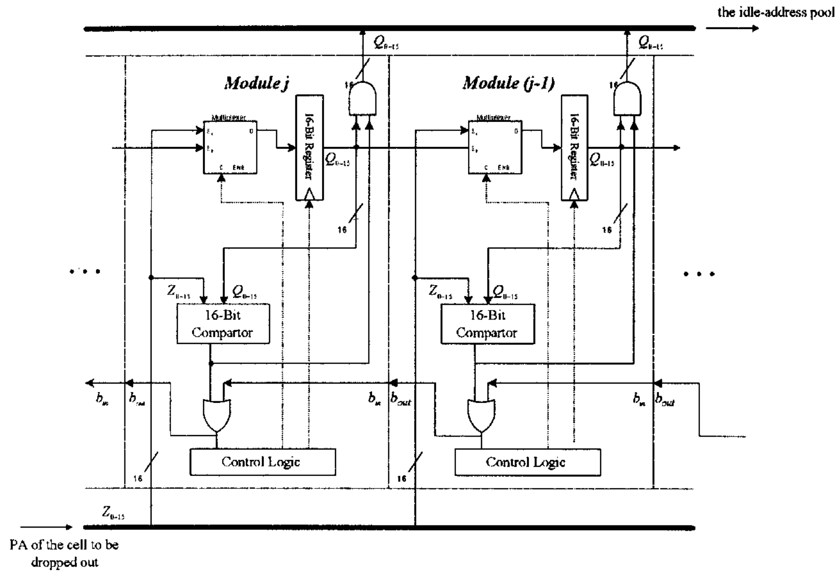

Fig. 2. The architecture of the push-out controller.

The proposed implementation architecture has a regular structure that well suits to the VLSI design. The number of virtual connections can grow flexibly since a large cell buffer capacity can be accommodated by cascading the chips. Moreover, by connecting chips in parallel, a large cell pool can be supported. With the implementation proposed in [7], the realization approach should be able to accommodate the real-time cell streams.

IV. Conclusions

A major concern to transmit TCP traffic over ATM networks is the implementation cost. Although the per-VC queueing may support the quality of service, it complicates the switch design. In this work, we show that it is feasible to achieve the requirements of GFR service through FIFO queueing. Our proposed PDPO scheme consists of a selective packet-discard strategy and a packet-based push-out buffering scheme, it ensures the fair sharing of network resource and prevents a misbehaved VC from occupying excessive FIFO buffer space, as well as improves the TCP throughput by avoiding the packet fragmentation problem.

References

[1] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. IEEE*, vol. 83, pp. 1374–1396, Oct. 1995.

[2] A. Romanow and S. Floyd, "Dynamic of TCP traffic over ATM networks," *IEEE J. Select. AreasCommun.*, vol. 13, pp. 633–641, May 1995.

[3] N. Yin and S. Jagannath, "The end-to-end traffic management in IP/ATM Internetworks,", ATM Forum Cont. 96-1406, Oct. 1996.

[4] D. E. Knuth, "Packet loss performance of selective cell discard scheme in ATM switches," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 903–913, June 1997.

[5] C. T. Chan, Y. C. Chen, and P. C. Wang, "An efficient traffic control approach for GFR services in IP/ATM Internetworking," in *IEEE GLOBECOM'98*, Nov. 1998, pp. 2446–2451.

[6] D. Lin and R. Mottis, "Dynamic of random early detection," in *SIGCOMM'97*, pp. 127–137.

[7] H. J. Chao and N. Uzun, "An ATM queue management handling multiple delay and loss priority," *IEEE/ACM Trans. Networking*, vol. 3, pp. 652–659, Dec. 1995.

[8] D. Basak and S. K. Pappu, "GFR implementation alternatives with fair buffer allocation scheme,", ATM Forum Cont. 97-0528, July 1997.