論文 98-23-5-7

Design of Flow Control Function for ABR Service in ATM Networks

Hun Kang*, Soong Hee Lee**, *Regular Members*

ATM 망에서 ABR 서비스 제공을 위한 흐름 제어 기능 설계

正會員 康 薰*, 李 崇 熙**

ABSTRACT

We survey several issues for implementing the end-to-end flow control function for ABR service and propose a structure for implementation. In the proposed design, control parameters are calculated and maneuvered by software for easy implementation and efficient adaptability for the changes of parameters while the cells are emitted at the rate according to the allowed cell rate with the hardware implementation for real time processing. The proposed structure can cope with the changes of the specifications including the calculation of the control parameters and their application. Hence, the structure is expected to contribute to providing the ABR service in the future ATM networks.

요 약

본 논문에서는 ATM 망에서 ABR 서비스를 제공하기 위하여 ATM 포럼에서 표준화를 진행하고 있는 망 종단간 흐름제어 기능을 설계하기 위한 제반 고려 사항들을 살펴보고 구현을 위한 설계 구조를 제시한다. 제시한 설계 구조에서는 구현의 용이성과 제어 변수의 효율적인 가변을 위하여 제어 파라미터를 계산 및 조정하고 자원 관리 셀을 처리하는 부분은 소프트웨어로, 실제 셀을 허용된 속도로 송출하는 부분은 하드웨어로 구현하여 실시간으로 송출 트래픽의 흐름 제어가 가능하도록 하였다. 제안한 설계 구조는 추후 표준화의 진행으로 인한 제어 파라미터의 산출 방식 및 적용 방법의 변경에도 쉽게 대처할 수 있고 실시간으로 제어하는 구조가 단순하여 구현 비용을 절감할 수 있는 구조로서 추후 구축될 ATM 망에서 ABR 서비스를 제공하는데 기여할 수 있을 것으로 기대된다.

Ⅰ. Introduction

Asynchronous transfer mode(ATM) technology has been applied to private networks including local area networks(LANs) rather than public networks although it was adopted as a main technology for establishing B-ISDN. In addition, the standardization process for the deployment of ATM technology into the current LAN and WAN has been carried out consistently around the ATM forum. ATM is based on the virtual

*한국전자통신연구원 차세대 인터넷연구실 실장
**인제대학교 정보통신공학과 전임강사
論文番號:96315-1005
接受日字:1996年 10月 5日

circuits and does not have error control procedures, which are general in the legacy LAN, to transfer high speed cells efficiently, which results in a possibility of degradation of quality of service(QoS) caused by the congestion. The deployment of ATM into the LAN carrying bursty data traffic may lead this problem as a significant one.

ATM forum specified available bit rate(ABR) service for reducing such a QoS degradation and maximizing the usable bandwidth by deploying an end to end flow control and has carried out its specification process[1]. ITU-T has also accepted ABR as one of the transfer capabilities in the ATM-based B-ISDN and has proceeded with the standardization[2].

The study of the ABR service, however, has been mainly focused on the performance evaluation of the modified details of the algorithm using analytical means or computer simulation so far[3~11]. The method for actual implementation of the algorithm has not been deeply explored for the establishment of ATM networks supporting ABR service. Thus, there still remains many problems to be solved for actual implementation of ABR service. For example, the behaviors of the source end system(SES) and destination end system(DES) are described separately in the current specification of the traffic management working group of ATM forum. These are to be included in the same end system, which requires the integrated solution for both of the behaviors. Of course, there are more issues left for actual implementation of the end to end flow control for ABR service.

This paper will survey requirements for the design of the end to end flow control function according to the specification of ATM forum and propose a designed structure for actual implementation considering these requirements. Section II includes the discussions about these requirements. The designed structure for the flow control is proposed in the third section as a solution of the derived problems in the second section. The fourth section will describe the details of each block in the proposed design. We will get a conclusion in the section IV.

Ⅱ. Requirements for Implementation of ABR Flow Control

1. Introduction to the end to end flow control for ABR service

The end to end flow control for ABR service establishes a closed loop control system including SES, DES, and switches, as shown in Fig. 1 according to the specification of ATM forum[1]. Control information in this closed-loop system is conveyed by resource management(RM) cells. SES initiates the control by generating forward RM cells. Switches may modify these RM cells passing through themselves. DES integrates the information from the received forward RM cells and the internal state to send backward RM cells with the integrated information in response to the received forward RM cells. Congestion control functions in the switches are classified into two types of explicit forward congestion indication (EFCI) and explicit rate(ER). There have been several proposals for the ER method[1][12~14]. In this paper, only the flow control functions in the end systems (SES, DES) will be regarded.

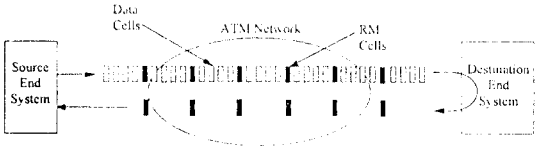


Fig. 1 Configuration of the end to end flow control loop for ABR service.

In actual implementation, end systems must include integrated functions of both of SES and DES. The integrated functions of the end systems are summarized as follows:

i) calculation of control parameters including allowed cell rate

ii) extraction and analysis of RM cells and EFCI from input cell stream

iii) control of cell emission rate(or interval)

iv) generation of RM cells and insertion into output cell stream

The functions of i), ii), and iv) require a complex processing like analysis or calculation of the parameters while the function of iii) requires a simple and prompt processing due to the real time requirements of the cell rate control. The functions of ii), iv) also require a real-time processing for extraction and insertion of RM cells. The function of i) must be performed fast enough not to influence the real time processing of iii). However, the requirement for i) is not so strict compared with those of ii) and iv). Hence, it is one of the possible choices that the implementation of the function of i) be software oriented and the implementation of the rest be hardware oriented as shown in Fig. 2, which leads to an easier implementation.

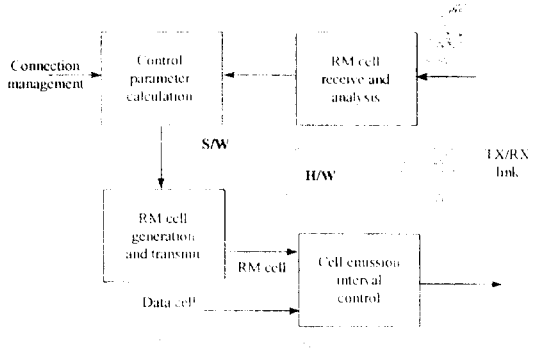


Fig. 2 Composition of the necessary functions for the flow control in the end system.

2. Constraints in the end to end flow control

One of the basic requirements for the end to end flow control is that there exists a congestion control method in the switch at least. Most switches in the current market have EFCI as a congestion control method, a binary feedback congestion control. EFCI, however, has been widely informed to have a problem of unfairness[13]. In addition, the rate-based method, the final choice of ATM forum, has been proved as sensitive to the increased round trip delay time(RTT) [7]. A segmentation strategy using virtual source(VS) and virtual destination(VD) concepts was raised to cope with this problem in the long distance link having a large RTT value especially[1][13][15]. Flow control functions in VS/VD may have different requirements from a case of ATM terminal as an end system. A terminal does not generally require many number of virtual connections for ABR at the same duration. A switch with VS/VD may require a concurrent processing of many virtual connections passing through at the same time. This means that there may exist a number of virtual connections for ABR waiting for the concurrent processing.

Lastly, a constraint for the applicable category of ABR service must be considered. Most cases assume ABR and UBR services to use the remaining bandwidth after CBR and VBR services. ABR service, however, can be applied in the application for LAN interconnection as shown in Fig. 3 for maximal utilization of the network resources. ABR service is expected to work as a main service to contribute to give more chances of access to the users of the LAN in this case. This may require a careful consideration on the

Now, we need to analyze the conditions for implementing to maintain ACR satisfying required QoS. If we let the interval between the kth cell and the next lth cell as I_{kl} [slots], link bandwidth as K[bits/sec], cell size as L_c[bits], and the allowed cell rate of the virtual connection n as ACR_n[cells/sec], the condition to maintain the cell emission rate within the allowed cell rate at the case of Fig. 4(a) is given as follows:

$$\max(I_{kl}) = \frac{K}{ACR_n * L_c} . \quad (1)$$

And if we let the capacity of a FIFO chip storing cells of a specified virtual connection n as X_n [cells], the condition for no cell loss in the connection due to

the excessive delays is given as follows:

$$\max(I_{kl}) \leq X_n . \tag{2}$$

From (1) and (2), the relation between allowed cell rate of each virtual connection and the capacity of each FIFO chip is

$$\frac{K}{ACR_n * L_c} \leq X_n . \tag{3}$$

As K/L_c is a fixed constant, the FIFO chip for each virtual connection must have the larger capacities than the maximum allowed interval between cells which is the product of K/L_c and the inverse of the allowed cell rate. For example, if a virtual connection wants to use the full bandwidth of 149.76 Mbps, i.e. 353,207 cells/sec, the necessary capacity for the FIFO chip is only a small amount approaching zero. However, if the connection only wants to use a small bandwidth approaching minimum cell rate(MCR) which might be zero, the necessary capacity for the FIFO chip increases very high. Hence, this way of using a separated FIFO chip for each virtual connection is not efficient in the point of the utilization of the memory space. In fact, it is not easy to get such a FIFO chip having a large capacity required for this purpose. Additionally, the capacity of the FIFO chips in the actual implementation, X, must satisfy for all FIFO chips so that the necessary capacity for virtual connection n, X_n, is always smaller than X, i.e. $X \geq \max(X_n)$. Consequently, if we let the number of concurrently active virtual connections as p, the method using separated FIFO chips has the following constraint.

$$\frac{K}{p * L_c} \max\left(\frac{1}{ACR_n}\right) \leq X. \tag{4}$$

Another method using the shared memory of Fig. 4(b) also requires the same condition with that of Fig. 4(a) to maintain the cell emission rate within the allowed cell rate as shown in the relation of (1). And if we let the capacity of a memory area storing cells of a specified virtual connection n as Y_n [cells], the condition for no cell loss in the connection due to the excessive delays is given as follows:

$$\max(I_{kl}) \leq Y_n . \tag{5}$$

From (1) and (5), the relation between allowed cell rate of each virtual connection and the capacity of each memory area is

$$\frac{K}{ACR_n * L_c} \leq Y_n . \tag{6}$$

Then, if we let the total capacity of the shared memory as Y and the number of concurrently active virtual connections as p, the capacity of a memory area storing cells of a specified virtual connection n, Y_n is related with these as follows:

$$\sum_{i=0}^{p-1} Y_i \leq Y. \tag{7}$$

Hence,

$$\frac{K}{L_c}\left(\sum_{i=0}^{p-1} \frac{1}{ACR_i}\right) \leq Y. \tag{8}$$

Now we can compare the required memory capacities of these two methods from (4) and (8). The summation for method for applying ABR service in a LAN/WAN interworking equipments like ATM routers. It is generally needed to have a large cell queue and its queue management per virtual connection for this kind of interworking at ATM routers. A similar requirement

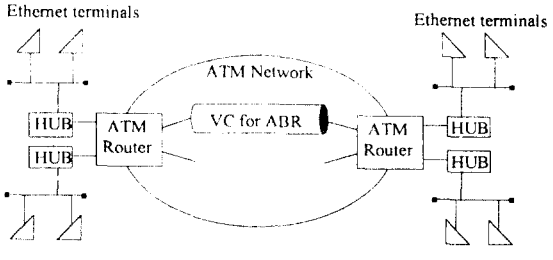


Fig. 3 Provision of ABR service in the interworking between the legacy LAN(Ethernet) and the ATM network.

may also be needed for the case of VS/VD. This requirement is very different from the case of an ATM terminal as an end system of ABR control loop, which only uses host memory for internal cell processing without the need of additional memory management.

3. Requirements for implementation

3.1 Store and extraction of cells per virtual connection

We have problems to solve for implementing end to end flow control for ABR service as the followings. First, a method must be provided to support multiple virtual connections for ABR service at the same duration. Sufficient number of virtual connections must provide ABR service concurrently in the case of interconnection between the legacy LAN and ATM LAN/WAN as shown in section II. The flow control function must be designed so that the interconnection can accept the concurrent multiple requests of the legacy LAN users. Second, in the viewpoint of actual implementation, this flow control function is a kind of traffic shaping function at the cell output part. We must maintain intervals between cell outputs during the valid time for the virtual connection according to the allowed cell rate(ACR). This requires that we have a buffer for each virtual connection, store the cells for the virtual connection into the specific area of the buffer, and read out the cells according to the allowed interval for the virtual connection. Using a FIFO memory chip per each virtual connection can be a simple way to implement this function as shown in Fig. 4(a), which has a difficulty in maintaining the allowed interval between cells without cell loss due to the limit of the memory capacity in each FIFO. Another way is using a shared memory for multiple virtual connections as shown in Fig. 4(b). This method has little limit for memory capacity for each virtual connection within the capacity of the implemented memory chip, whereas it requires additional memory management for manipulating the memory area per each virtual connection. This method can utilize the memory space more efficiently and can easily expand the number of concurrently active virtual connections.

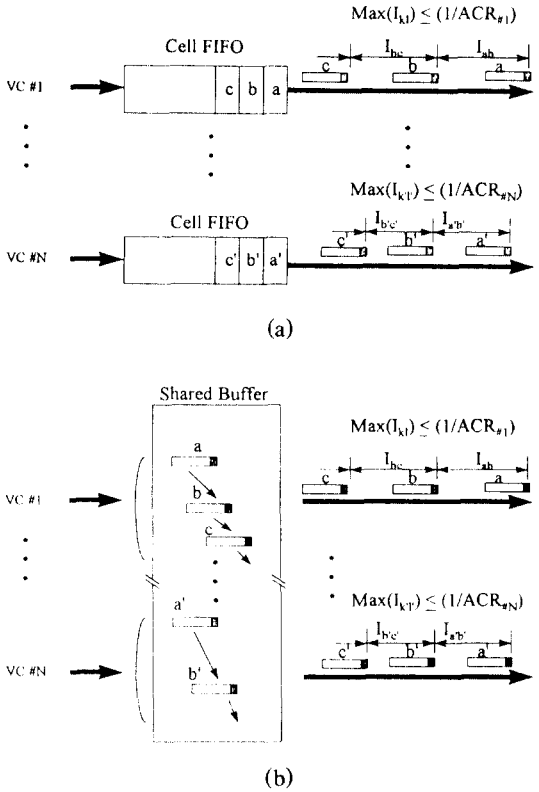


Fig. 4 Methods for store and release of a cell per each virtual connection (a) using separate FIFO chips and (b) using shared memory

all virtual connections, i.e. $Y=\frac{K}{L_c}\left(\sum_{i=0}^{p-1}\frac{1}{ACR_i}\right)$, can not be larger than the p times of $X=\frac{K}{L_c}\max\left(\frac{1}{ACR_n}\right)$. Thus the result of the comparison is as follows:

$$p * X \geq Y. \tag{9}$$

This shows that the method using separated FIFO chips can not have smaller requirements for the required memory capacity than that using shared memory. For example, assuming the allowed emission

interval for a virtual connection to be 1,000 slots and that for another connection to be 20,000 slots, the total amount of the required memory capacity in the method using FIFO chips has to be more than 40,000 cells (the product of $p=2$ and $\max(K/(ACR_n * L_c))$ at (4)) as it may not have different capacities for different connections. This amount is hard to be satisfied with the FIFO chip in the current market. On the contrary, the method using the shared memory only requires the total amount of 21,000 cells (Y of (8)), which means it is more economic and easy to be implemented with the general memory chips. With the method using shared memory, we can use general DRAM or SRAM chips to store and release cells of multiple virtual connections and can simply expand the number of concurrently active virtual connections and the duration of the allowed emission interval only by additional allocation of the memory or the expansion of the size of the memory chips. This merit will grow as the number p increases.

3.2 Cell emission interval control per virtual connection

The emission of cells according to the allowed emission interval just as in Fig. 4 requires a function of measuring time duration between the latest cells transmitted and the cell to be transmitted on a virtual connection, which is to prevent any violation of the allowed emission interval. A way to do this is to use a counter for each virtual connection as shown in Fig. 5(a), which starts the counter at the instance of a cell emission and outputs a signal for permission of the next cell emission at the instance of the expiry of the counter after the elapse of the allowed emission interval of the virtual connection.

In this method, if we let the output of the counter showing the elapsed time after the latest cell emission of the virtual connection i as C_i, the time instance of the latest emission as D_i, the link bandwidth as K [bits/sec], the cell size as L_c[bits], and the allowed cell rate as ACR_i, the cells on the virtual connection i must accord with the following relation for the counter method to work properly.

$$C_i - D_i < \frac{K}{ACR_i * L_c} \tag{10}$$

This method is simple for a single virtual connection, whereas the complexity increases in proportion to the increasing number of virtual connections to be processed concurrently. A way to solve this problem is shown in Fig. 5(b), which uses the memory as the counter for counting the current elapsed time of each virtual connection. The memory area is divided into segments for storing and updating the time instance of the cell departure(TCD) and allowed emission interval(AEI) of each virtual connection. AEI is updated in accordance with the changing ACR value for the ABR virtual connection. TCD is initialized to zero at the instance of cell departure of the ABR connection. The stored TCD is to be updated and compared with the stored AEI at every cell transmission time. The comparison is performed by subtracting the stored AEI from the updated TCD and checking whether the result matches to zero. If it matches to zero, a signal permitting the emission of the cell on the connection is issued. These update and comparison, however, must be finished within one cell transmission time as all the memory segments for multiple connections must be updated in each transmission time of cells. This causes a limit to the number of virtual connections to be processed concurrently by the limited time of each cell transmission. We let the memory access time including setup time and hold time as t_m, the number of memory accesses during each transmission time of cells as Θ, and the time required for updating the stored TCD as t_p. Then t_p is determined as follows:

$$t_p = \Theta * t_m \tag{11}$$

We can regard all the virtual connections have the same value for t_p, and the update of the memory

segments must be finished within each transmission time of cells. Hence, if we let the transmission time of cells as T_c (2.83 μs for the available link speed of 149.76 Mbps), the following relation must be maintained for a proper operation.

$$\Theta * t_m < T_c \tag{12}$$

In actual implementation, the value of Θ is the number of possible virtual connections to be processed concurrently. This means that a large value of Θ is needed enough for the multiple virtual connections to be processed concurrently. The value of Θ, however, can not exceed a fixed limitation due to the fixed constants, t_m and T_c as shown in the relation of (12).

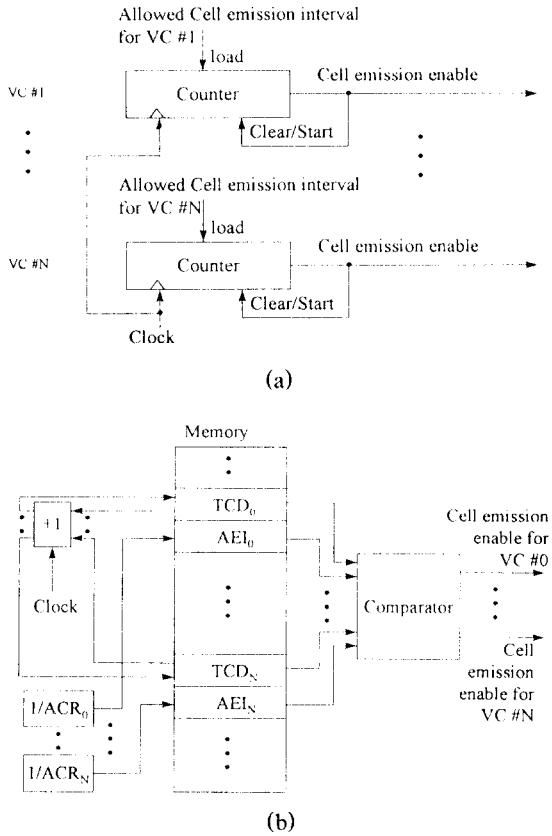


Fig. 5 Methods for cell emission interval per each virtual connection (a) using separated counters and (b) using a memory

Even with this problem, the method using memory has advantages compared with the method using counters in the point of implementation. The method using counters does not reuse the counter logic for multiple connections, which causes the proportional increase of complexity with the increase of ABR virtual connections to be processed concurrently. On the contrary, the method using memory reuses the logic and memory as far as the relation of (12) is guaranteed. This means that the complexity of the method using memory will not increase proportionally with the increase of ABR virtual connections to be processed concurrently. If the increased number of ABR connections makes Θ violate the relation of (12), it is inevitable to add the additional modules having the same logic with the explained scheme using memory.

Ⅲ. Designed Structure of the Flow Control Function

1. Structure for implementation

We design the flow control function to provide ABR service from the requirements acquired in section II. First, we decide to implement with full software for the function of i) at II. 1, i.e. calculation of control parameters including allowed cell rate. This function requires a CPU including a coprocessor for floating point calculation. The function of iii) at II. 1 requires the implementation using full hardware for real time processing of cells according to the allowed emission interval. The functions for RM cells, ii) and iv) can be constructed with a combination of hardware and software. In addition, an adaptation function to be coupled with ATM layer functions and the PHY/ATM layer functions for actual transmission of the cells are needed. More detailed design of the required functions needs to be confirmed through the mapping of the functions into the blocks.

The functions discussed in section II can be mapped into four blocks: CRCB(cell rate control block), AICB (allowed cell interval calculation block), RCIB(RM

cell input block), and RCOB(RM cell output block) in the flow control function module as shown in Fig. 6. We will discuss the details of these blocks in the next sections.

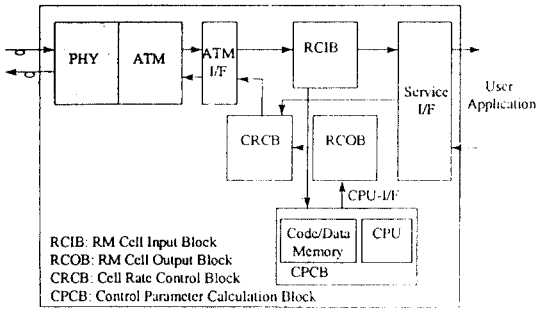


Fig. 6 Block diagram of the flow control function module.

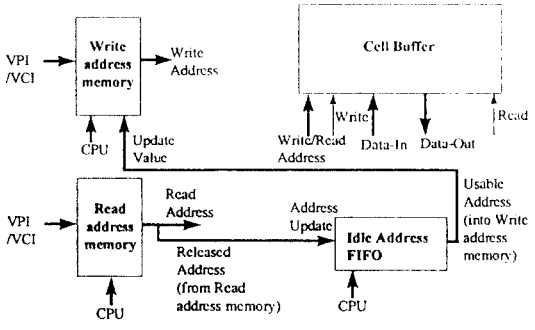


Fig. 7 Structure of the part to store and release cells per virtual connection.

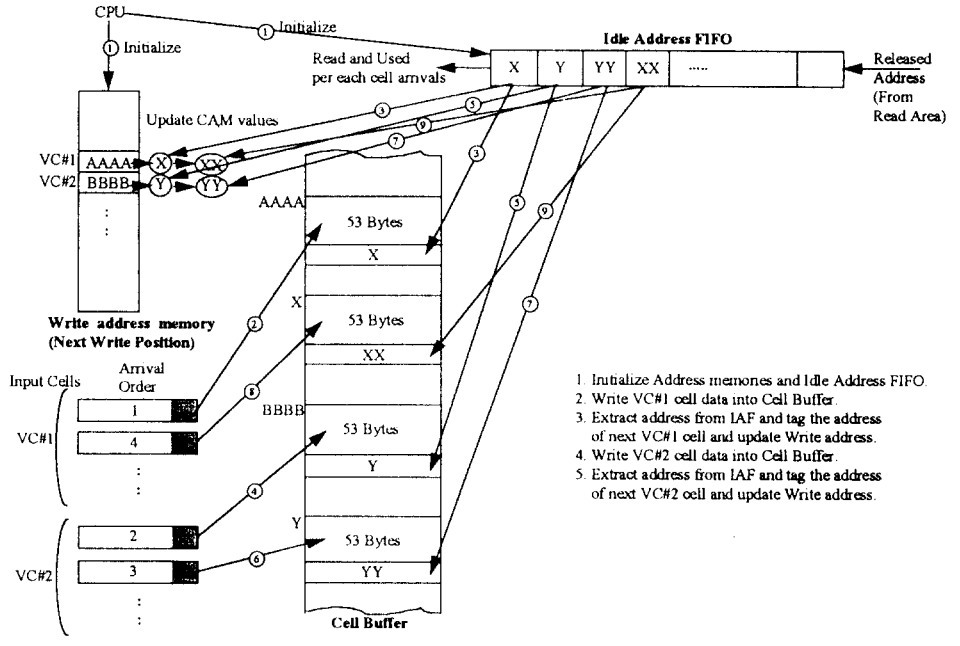
2. Cell rate control block (CRCB)

CRCB consists of a part to store and release cells and another part to decide the instance of cell emission per virtual connection. We design the part to store and release cells as shown in Fig. 7.

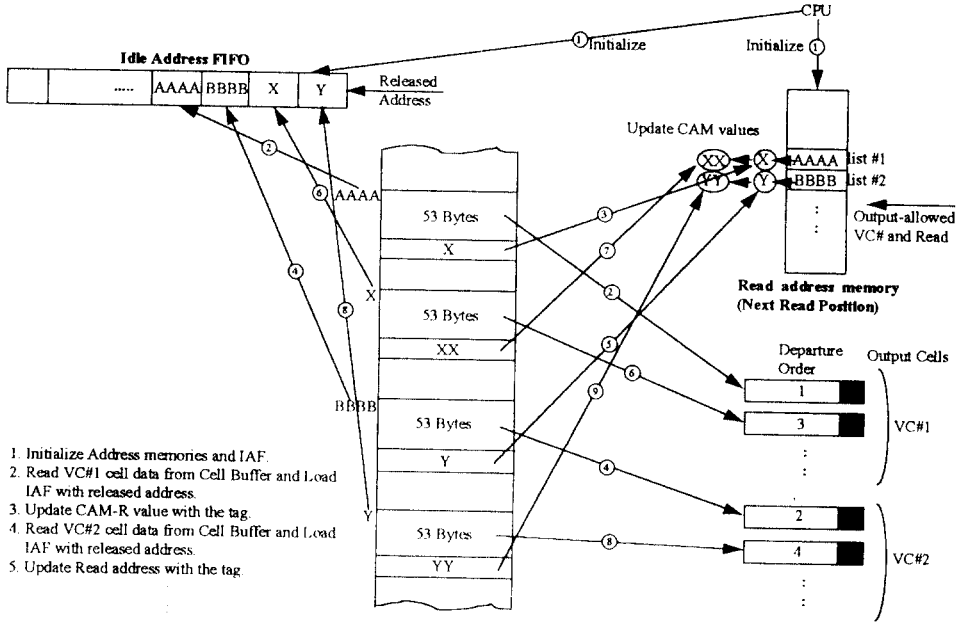
In Fig. 7, an input cell is temporarily stored in 1-cell FIFO for VP/VC identifier extraction. Extracted identifier is translated to the list number of cell queue. The list number is transferred to write address memory and the write address memory outputs the correspondent *write address* of the input cell from the list number. The write address is to be used as an access address of the cell queue in the shared memory. The writing of the cell stored in the 1-cell FIFO into the appropriate area of the write address in the shared memory is performed by the *store* command which was activated by the input of the cell. The read address memory stores the list number of cell queue to be released, received from the cell rate control block. The shared memory for cell queueing is shared by multiple virtual connections and read and written with the address from the write address memory and read address memory. The addresses used for the read of cells are stored into the idle address FIFO and reused for the write of the cells. The initial values of these write address memory, read address memory, and idle address FIFO are written by the software in the control parameter calculation block. The detailed description of the operation for the cell store and forward is shown in (a), (b) of Fig. 8.

Numbers in the circles of Fig. 8(a) indicates the sequences of the operation in the part for cell store. The behavior in each step is as follows:

1. Initial values of write address memory, read address memory, and idle address FIFO are established.
2. The cell of the list #1 is stored in the address of 'AAAA' according to the contents of the write address memory(AAAA).
3. The address from idle address FIFO is written as the address of the next write location(X) for the list #1 into the shared memory and write address memory.
4. The cell of the list #2 is stored in the address of 'BBBB' according to the contents of the write address memory(BBBB).
5. The address from idle address FIFO is written as the address of the next write location(Y) for the list #2 into the shared memory and write address memory.
6. The next cell of list the #2 is stored in the address of 'Y' according to the contents of the write ad-



(a)



(b)

Fig. 8 Detailed procedure of (a) cell store and (b) release in the shared memory.

dress memory(Y).

7. The address from idle address FIFO is written as the address of the next write location(YY) for the list #2 into the shared memory and write address memory.
8. The next cell of the list #1 is stored in the address of 'XX' according to the contents of the write address memory(XX).
9. The address from idle address FIFO is written as the address of the next write location(XX) for the list #1 into the shared memory and write address memory.

The procedures of steps 4-5, 6-7, and 8-9 are the repeats of the steps 2-3. These procedures also apply to the lists other than #1 and #2. Numbers in the circles of Fig. 8(b) indicates the sequences of the operation in the part for cell releases. The behavior in each step is as follows:

1. Initial values of read address memory and idle address FIFO are established.
2. Cells of the list #1 are extracted from the memory location(AAAA) indicated by read address memory and the used address(AAAA) is written into idle address FIFO.
3. The memory location of the list #1 in the read address memory is updated with the next write address(X).
4. Cells of the list #2 are extracted from the memory location(BBBB) indicated by read address memory and the used address(BBBB) is written into idle address FIFO.
5. The memory location of the list #2 in the read address memory is updated with the next write address(Y).
6. Cells of the list #1 are extracted from the memory location(X) indicated by read address memory and the used address(X) is written into idle address FIFO.
7. The memory location of the list #1 in the read address memory is updated with the next write address(XX).
8. Cells of the list #2 are extracted from the memory location(Y) indicated by read address memory and the used address(Y) is written into idle address FIFO.
9. The memory location of the list #2 in the read address memory is updated with the next write address(YY).

The procedures of steps 4-5, 6-7, and 8-9 are the repeats of the steps 2-3. These procedures also apply to the lists other than #1 and #2.

Functions for the decision of cell emission instance in CRCB consists of the parameter memory per virtual connection, access controller, adder, comparing logic, and initializer as shown in Fig. 9. The parameter memory per connection includes lists of the control parameters per each virtual connection. The access controller allows cyclic accesses of the parameter memory. The number of accesses within one cell transmission time is limited to N which is specified by the way of implementation. N can be increased with the advances of the processing speed of the logic devices. The timing diagram of Fig. 9 shows N times accesses within one cell transmission time. The longer one cell transmission time is and the faster processing speed of the devices is, the larger the value of N becomes. During the active duration of access permission signal(shown as zero level in Fig. 9), access controller reads the current emission interval to transfer to the adder and reads the allowed cell emission interval to transfer to the comparing logic. The comparing logic compares the resulted value of the adder and the allowed emission interval. If the compared values are matched, the initializer will be activated. The result of the adder also updates the current cell interval in the list of the parameter memory. The zero value from the activated initializer clears the current emission interval in the list of the parameter memory and the list number of the same list is issued as the number of the

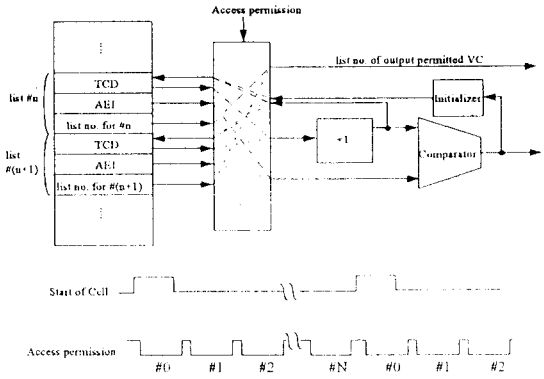


Fig. 9 Structure for determining cell emission interval for each virtual connection and its timing.

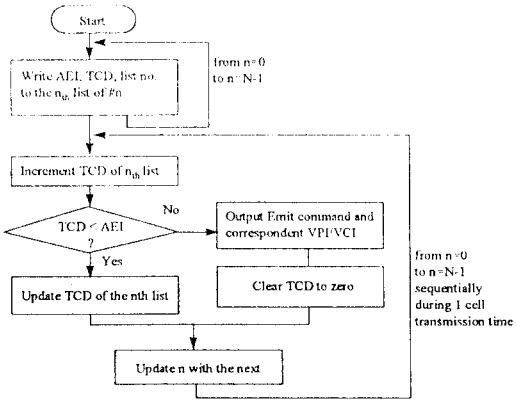


Fig. 10 Flow of the cell emission instance determination.

list allowed to emit a cell.

Fig. 10 shows the control flow for the decision of cell emission instance in the CRCB. This flow assumes there exist N lists in the parameter memory for N concurrently active virtual connections. After the initialization of control parameters, the current cell emission interval values of the connections are incremented by one and compared with the correspondent allowed cell emission interval value with the elapse of one cell transmission time. If matched, the number of the matched list will be issued and the current cell emission interval value will be cleared to zero.

3. Control parameter calculation block(CPCB)

CRCB consists of three software modules to calculate the control parameters including ACI and to report to the other blocks as shown in Fig. 11. An interrupt indicating RM cell receive intrigues RM cell analyzing module(RCAM) to report the analyzed results to the ACI calculation module(AICM). RCAM converts the 16 bits of CCR, ER into the real values to be used for the calculation of the ACI or new values for CCR, ER. AICM calculates control parameters integrated status information from the received control parameters of the correspondent virtual connection and the current state of the network from RCAM. AICM sends the calculated ACI, actually the number of slots permitted between sending of the cells, to CRCB. The calculated parameters used as elements of the RM cell field are sent to RCOB. CPU chip used for CRCB must have a coprocessor for the floating point calculation and be fast enough to work well with the other hardware blocks.

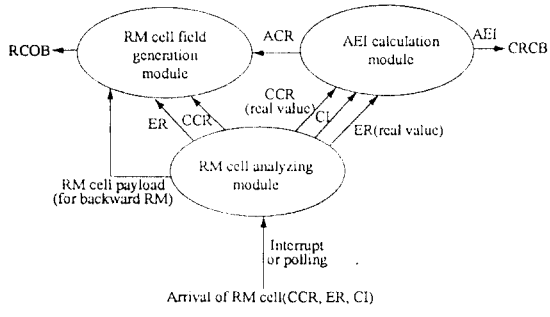


Fig. 11 Structure of CRCB.

4. RM cell input block(RCIB)

RM cells can be extracted by detecting if the value of PT field of the input cell is 110. Most of ATM layer chips have the function of extracting OAM and RM cells from input cell stream. Hence, this function block can be implemented with the ATM layer chips having this extracting function. The extracted RM cell is read and written by CPU to analyze and up-

date its contents. RCIB can report to the CPCB that an RM cell is received using an interrupt signal. RCIB only extracts the RM cells on the correspondent virtual connection and validates it to CPCB. The analysis and update of the received RM cell is performed in CPCB as discussed in the last paragraph.

5. RM cell output block(RCOB)

This block performing the function of iv) in II.2 consists of software for generating or updating RM cells and hardware for sending RM cells with CRC-10 generation. RM cells are generated by the successive updates of the RM cell fields received from CPCB. The forward and backward RM cells are stored and released in the separated FIFO chips in RCOB. Contents of the backward RM cells are updated with the information from the latest forward RM cell and the changes of EFCI bit of the received data cells. The change of the EFCI bit is reported to the software for generating and updating RM cells via polling or interrupt. The fields of the forward RM cell are updated whenever there occurs changes in the control parameters related with the fields of the forward RM cell.

A backward RM cell must be sent as fast as possible after receiving a forward RM cell while a forward RM cell must be sent after sending *Nrm* in-rate cells or the expiry of the specified time of *Trm*. This requires monitoring of the number of in-rate cells between sending forward RM cells and sending a forward RM cell after *Nrm*-1 in-rate cells are sent after sending the last forward RM cell. If at least *Mrm* in-rate cells have been sent and *Trm* has been elapsed, a forward RM cell must be sent. RCOB is designed to satisfy these requirements as shown in Fig. 12.

Applying the list number belonging to the virtual connection allowed to emit a cell, the number of already sent in-rate cells, *Ndt*, is read from the RM control memory and sent to the adder. *Nrm* is also read from the RM control memory and sent to the comparing logic to be compared with the incremented *Ndt* value from the adder. Comparing logic activates the initializer and issues the list number to send a forward RM cell when incremented *Ndt* is same with *Nrm* to clears the *Ndt* value to zero. If not activated, the initializer updates *Ndt* with the incremented *Ndt* from the adder. Procedures for *Mrm* and *Trm* can be established with the same way of the explained about *Nrm*. Forward RM cells are sent on the condition satisfying the constraints of *Nrm* or those of *Mrm* and *Trm*.

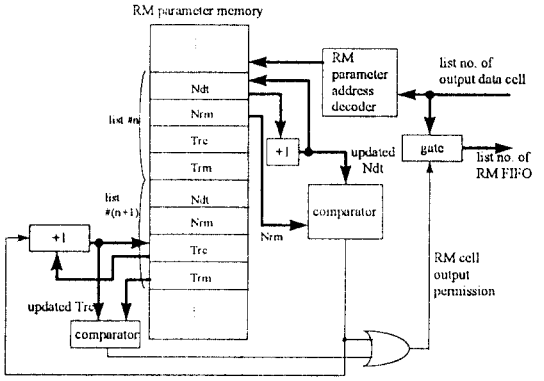


Fig. 12 Structure of RM cell emission control in RCOB.

Ⅳ. Conclusion

We proposed a structure for implementation of flow control for ABR service in ATM networks. In the proposed structure, control parameters are calculated and maneuvered by software for easy implementation and efficient adaptability for the changing procedures of parameters due to the standardization process while the cells are emitted at the rate according to the allowed cell rate with the hardware implementation for real time processing. Detailed description about the updating methods of the control parameters will be given in the future. The proposed structure can cope with the changes of the specifications including the calculation of the control parameters and their application. In the proposed structure, general memory chips are used instead of separate FIFOs or counters

for the simpler implementation and cost reduction.

The structure in this research is expected to contribute to provide the ABR service in the future ATM networks with its adaptability, simplicity and low cost. We have not considered the effects of the detailed characteristics of the cell traffic like the burstiness or cell delay variation, which is left for further study. We expect more studies on this issue for more efficient function implementation.

References

1. S. S. Sathaye, "ATM Forum Traffic Management Specification Version 4.0," *ATM Forum/af-tm-0056 .000*, Anchorage, Apr. 1996.
2. ITU-T SG13, "Traffic control and congestion control in B-ISDN," *ITU-T Rec I.371*, Temporary Document, Mar. 1994.
3. C. Lefelhocz, B. Lyles, S. Shenker, and L. Zhang, "Congestion control for best-effort service: Why we need a new paradigm," *IEEE Network Magazine*, pp. 10-19, Feb. 1996.
4. N. Yin and M. G. Hluchyj "On closed-loop rate controller for ATM cell realy networks," *Proc. INFOCOM'94*, pp. 99-108, Toronto, Jun. 1994.
5. A. Atai, "A rate-based feedback traffic controller for ATM networks," Proc. ICC'94, New Orleans, Louisiana, vol. 3, pp. 1605-1615, May 1994.
6. L. Benmohamed and D. Su, "Analysis of the rate-based traffic management proposal for ATM networks," *Proc. International Conference on Telecommunications Systems, Modeling and Analysis*, Mar. 1995.
7. A. Kolarov and G. Ramamurthy, "End-to-end adaptive rate based congestion control scheme for ABR service in wide area ATM network," *Technical Report of NEC USA Inc.*, Feb. 1995.
8. K. Siu and H. Tzeng, "Intelligent congestion control for ABR service in ATM networks," *ACM SIGCOMM Computer Communication Review*, Jul. 1994.
9. C. Ikeda, H. Suzuki, and M. Murata, "Transient state Analysis and Maximum Buffer Requirements for Enhanced PRCA," *ATM Forum/94-910*, Ottawa, Sep. 25. 1994.
10. S. Lee and H. Choi, "Congestion Control in B-ISDN Using Reactive Access Control," *KICS Journal*, pp. 2868-2881, vol. 20. no. 10., Oct. 1995.
11. S. Lee and H. Choi, "Traffic flow control of B-NT for prevention of congestion in B-ISDN UNI," *KICS Journal*, pp. 1085-1094, vol. 19, no. 6, Jun. 1994.
12. R. Walthall and M. Clement, "Simulation and analysis of the performance of EPRCA in a wide area ATM network consisting of both ER and EFCI switch mechanisms," *Proc. International Conference on Telecommunications Systems, Modeling and Analysis*, Mar. 1996.
13. R. Jain, "Congestion control and traffic management in ATM networks: Recent advances and a survey," *Computer Networks and ISDN Systems*, Feb. 1995.
14. W. Rhee, H. Park, and C. Hwang,, "A study on traffic control for the ABR service in ATM networks," *KICS Journal*, pp. 2346-2361, vol. 20. no. 8., Aug. 1995.
15. L. Roberts, "Enhanced PRCA(Proportional Rate-Control Algorithm)," *ATM Forum/94-0735R1*, Aug. 1994.

康　薰(Hun Kang)　　정회원

1958년 8월 15일생

1980년 : 연세대학교 전자공학과 (공학사)

1989년 : IOWA state university 컴퓨터공학과(공학석사)

1993년 : IOWA state university 컴퓨터공학과(공학박사)

1993년 10월~현재 : 한국전자통신연구원 선임연구원

※주관심분야 : ATM ; LAN, 고속라우터

e-mail : hkang@winky.etri.re.kr

李 崇 熙(Soong Hee Lee) 정회원
1961년 12월 8일생
1987년 2월 : 경북대학교 전자공학과(공학사)
1990년 2월 : 경북대학교 전자공학과(공학석사)
1995년 2월 : 경북대학교 전자공학과(공학박사)
1987년~1997년 : 한국전자통신연구원 선임연구원
1997년 3월~현재 : 인제대학교 정보통신공학과 전임강사
※주관심분야 : ATM/B-ISDN, 통신시스템
e-mail : icshlee@ijnc.inje.ac.kr