

FPGA를 이용한 single rate Rate Adaptive Shaper 설계

Design of single rate Rate Adaptive Shaper Using FPGA

박천관*

Chun-Kwan Park*

요 약

본 논문은 RFC2963에서 제안된 single rate Rate adaptive Shaper (srRAS)를 설계한 것이다. srRAS는 RFC2693에서 제안된 하향의 single rate Three Color Marker (srTCM)과 함께 사용된 셰이퍼이다. 그것은 가변 속도로 출력되는 tail-drop FIFO (First Input First Out) 큐이다. srTCM은 srRAS로부터의 IP 패킷 스트림을 측정 후 그 패킷을 green, yellow, 또는 red로 마킹해 준다. 이 셰이퍼는 AF PHB (Per Hop Behavior)를 제공하는 DS (Differentiated Service) 네트워크의 입력에서 사용되도록 제안되었다. 그리고 srRAS는 srTCM의 상향 트래픽의 버스트성을 줄여줄 수 있다. 본 논문은 srRAS의 알고리즘, 구조, 그리고 FPGA 및 관련 기술을 통하여 구현할 수 있는 방안을 언급하였다.

Abstract

This paper has addressed the scheme to design single rate Rate Adaptive Shaper (srRAS) proposed in RFC2963. srRAS is the shaper used in conjugation with downstream single rate Three Color Marker (srTCM) described in RFC269. it is tail-drop First Input First Out (FIFO) queue that is drained at a variable rate. srTCM meters IP packet streams from srRAS and marks its packets to be either green, yellow, or red. This shaper has been proposed to use at the ingress of differentiated services networks providing AF PHB. And then srRAS can reduce the burstiness of the upstream traffic of srTCM. This paper addresses algorithm, architecture of srRAS, and the scheme to implement srRAS using Field-Programmable Gate Arrays (FPGA) and the related technology.

Key words : srRAS, srTCM, FPGA, QoS, DS

I. Introduction

As new application services requiring QoS guarantees, such VoIP and VPN, have appeared, IP QoS problems have become one of the most important issues regarding Internet throughput. Specific requirements for delay time, delay

variation, and loss of packets transmitted though the Internet can be set according to differentiated services. Currently, Internet provides only best-effort service, which treats all packets in the same manner, and thus cannot guarantee meeting the requirements for delay and delay variation. Therefore, a new service model, unlike the

* 국립목포해양대학교 해양전자통신공학부(Division of Marine Electronic & Communication Engineering, Mokpo national Maritime University)

· 제1저자 (First Author) : 박천관

· 접수일자 : 2005년 12월 28일

best-effort service model, is required to guarantee QoS in the Internet.

In IP network, in order to provide all users with the diverse QoS satisfactorily while using network resources effectively, traffic regulation function at network edge is necessary.

The main objective of the shaper is to produce the output traffic that is less bursty than the input traffic. And rate shaping is used to bound, or constrains the unpredictability of a certain traffic class, and requires queues, queue management, and scheduling function. If a flow does not conform to the traffic profile, shaping function can be used to delay non-conforming traffics until they conform to the profile.

srRAS proposed in RFC2963 is the shaper used in conjugation with downstream srTCM. it is tail-drop First Input First Out (FIFO) queue that is drained at a variable rate. srTCM meters IP packet streams received from srRAS and marks its packets to be either green, yellow, or red. This shaper has been proposed to use at the ingress of differentiated services networks providing AF PHB (Per Hop Behavior). And then srRAS can reduce the burstiness of the upstream traffic of srTCM. By reducing the burstiness of the traffic, srRAS increases the percentage of packets marked as green by the downstream srTCM.

srTCM consists of meter and marker. Meter measures the instantaneous properties of the selected packets streams according to a traffic profile specified in a Traffic Conditioning Agreement(TCA), and then passes the metering result and the packet to the marking function to trigger a particular action for each packet, which is either in-profile or out-of-profile. Marker sets the Differentiated Service (DS) field of a packet to a particular codepoint, adding the marked packet to a particular DS behavior aggregate. in this paper we address the design of srRAS proposed in

RFC2963 using FPGA and its related tools.

This paper is organized as follows. Chapter 2 describe the architecture of srRAS with srTCM (single rate Three Color Marker). Chapter 3 addresses architecture and operation of srTCM. Chapter 4 addresses the implementation to design srRAS using FPGA. Chapter 5 includes conclusion of this paper.

II. Rate Adaptive Shaper

Figure 1 shows srRAS used in conjunction with srTCM. srRAS is set up by initially assigning values to four parameters.

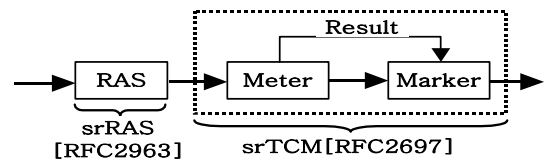


그림 1. srTCM 모델
Fig. 1. srTCM Model.

These parameters are two rates and two buffer thresholds. Two rates in bytes per second are Committed Information Rate (CIR) and Maximum Information Rate (MIR). Two buffer thresholds in bytes are CIR_th (CIR_threshold) and MIR_th (MIR_threshold). srTCM is the marker based on token bucket.

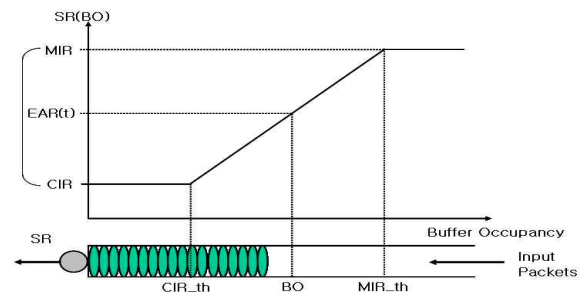


그림 2. RAS의 셰이핑 속도
Fig. 2. Shaping rate of RAS.

As shown in figure 2, the shaping rate of srRAS

is based on the average rate of incoming traffic and the instantaneous FIFO buffer occupancy.

The average rate can be computed by several means, but in this paper, we adapt the method used in [1]–[7]. The function of the arrival rate is as follows.

$$EAR(t)=[1-\exp(-T/K)*L/T]+\exp(-T/K)*EAR(t-1)$$

Where EAR(t) is the updated Estimated Arrival Rate, EAR(t-1) is the previous value of the Estimated Arrival Rate, T is the time passed since the previous packet arrives, L is the size of arrival packet, and K is constant for filtering out the estimation inaccuracies due to exponential smoothing. Another factor is the instantaneous FIFO buffer occupancy of srRAS. The relationship between the buffer occupancy and the shaping rate is as follows.

- 1) For buffer occupancy < CIR_th,

$$SR(BO)=\max(EAR(t),CIR),$$

$$BO=\text{Buffer Occupancy}$$
- 2) For CIR_th ≤ Buffer Occupancy < MIR_th,

$$SR(BO) =\max(EAR(t), F(BO)),$$

$$F(BO)=CIR+((MIR-CIR)/(MIR_th-CIR_th))*(BO-CIR_th)$$
- 3) Buffer Occupancy ≥ MIR_th,

$$SR(BO) = MIR$$

Figure 3 shows Time Schedule algorithm for srRASsr. In srRAS, a time schedule T1 is based on the output rate(=shaping rate) in srRAS. T1 is the time that the packet at the head of the queue of srRAS is to be released from the srRAS. For CIR EAR(t) MIR, $T1 = t + L_i / SR(BO)$, Where t is current time, L_i is the length of kth packet for flow i, and SR(BO) is the shaping rate(=output rate) of srRAS.

In srRAS, the shaper is not aware of the status of the meter in srTCM. This entails that shaper can unnecessarily delay the packet although there are enough tokens available to mark the packet green. To solves this problem, srRAS is coupled with the meter. The meter in srTCM informs srRAS of the green token status, and then srRAS can decide whether it sends the packet to srTCM immediately or not, according to the green token status. Therefore, the green packets can be released sooner than srRAS, and the delay in srRAS can be reduced. Another time schedule according to the green token state information from srTCM, T2, is calculated as the earliest time instant when the packet at the head of srRAS queue would be marked as green by srTCM.

$$T2=\max(t, t+(L - Bc(t))/CIR)$$

Where t is current time, L is the packet length in bytes at the head of srRAS queue, Bc(t) is the amount of green tokens in the token bucket of srTCM at t, and CIR in bytes per second is the Committed Information Rate of srTCM.

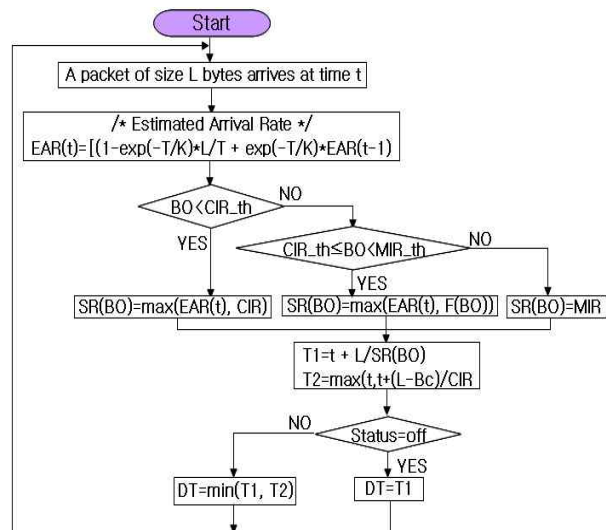


그림 3. RAS의 타임 스케줄 알고리즘
Fig. 3. Time Schedule algorithm for RAS.

III. Single Rate Three Color Marker

srTCM meters an IP packet stream, and marks its packets into green, yellow, or red before admitting them into a Differentiated service domain. Marking in srTCM is based on three parameters, committed information rate (CIR), committed burst size (CBS), and excess burst size (EBS). Figure 4 shows srTCM architecture described in RFC 2697.

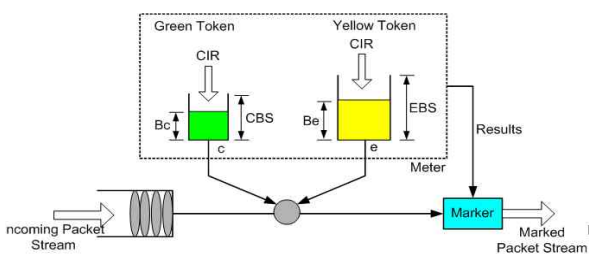


그림 4. srTCM 구조
Fig. 4. srTCM architecture.

srTCM consists of meter and marker. Meter consists of two token buckets, c and e. The function of meter is to meter each packet, and then pass the packet and the metering results to the marker. The function of marker is to set the DS field of packet to a particular codepoint according to the metering result. srTCM is configured by assigning values to the three parameters (CIR, CBS, and EBS) in initial time. Figure 5 shows the token counter update algorithm of the meter in srTCM.

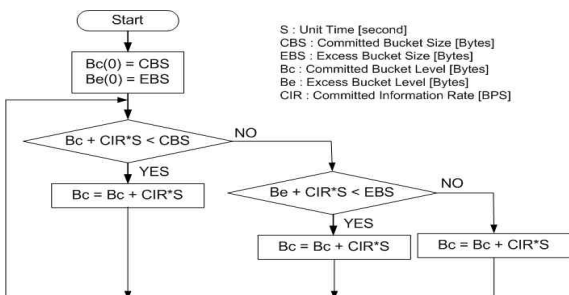


그림 5. srTCM에서 토큰 카운터 갱신 동작
Fig. 5. Token counter update operation in srTCM.

This algorithm is specified in terms of two token buckets, c and e, Both share the common token generation rate, CIR. Two burst sizes, CBS and EBS, are measured in bytes. At least one of them must be larger than zero. It is recommended that the burst size be equal to or larger than the size of the largest possible IP packet. Each burst size is related to each token bucket size. The maximum size of token bucket c is CBS and the maximum size of token bucket e is EBS. Initially, the levels of two buckets, Bc and Be , are assigned to $Bc(0)=CBS$ and $Be(0)=EBS$, respectively. The updating of Bc and Be is done by CIR as Figure 5 shows.

Figure 6 shows the metering algorithm in srTCM. The algorithm is done in two different modes, and set at the initial time. In color-blind mode, the meter assumes that the incoming packet stream is uncolored. In this mode, all packets are processed in the same method. In the color-aware mode, the meter assumes that the incoming packet stream has been pre-marked as green, yellow, or red. In this mode, the packet is additionally processed through the different methods based on its color. In Figure 6, L is the packet size and is measured in bytes. Bc and Be are the levels of each bucket and are also measured in bytes. Marking is based on the CIR and two associated burst sizes, CBS and EBS. A packet is marked green if it doesn't exceed the CBS, yellow if it does exceed the CBS but not the EBS, and red otherwise. The srTCM is useful, for example, for ingress policing of a service. In such a policing scheme, only the length, not the peak rate, of burst determines service eligibility. The color is coded in the DS (Differentiated Service) field of the packet in the PHB-specified manner.

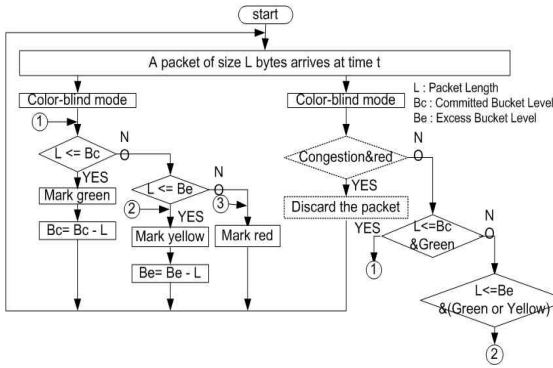


그림 6. srTCM에서 미터링 기능
Fig. 6. Metering function in srTCM

IV. Implementation of Rate Adaptive Shaper

Figure 7 shows the architecture of IP shaper consisting of srRAS and srTCM. This shaper is different from the shaper for ATM because of variable packet length. So the address management of memory is different from that of ATM. But another operation is similar to that of ATM. srRAS consists of input interface&IP header extractor block, searching function block, lookup table block, timing control block, queue control block, DT (Departure Time) calculator block, and token&metering block. These cooperate related to generate the appropriate address to read(write) packets from (to) the packet memory.

In this architecture, the virtual memories are divided into three major queues. First, packets that belong to the same flow are linked together in a logical queue, called the flow queue. Second, packets that have the same time stamp (DT) are linked together in the timing queue. Third, packets whose departure time is due or overdue are linked together in the departure queue.

The contents in the flow queue are the address of packets stored in packet memory. The contents of both the timing queue and the departure queue are FID (Flow Identifier) values. There is also an idle-address linked list (IALL) that keeps the

available space of the packet memory. timing&queue control block generates the necessary signals that are used to access all logical queues.

Token&metering block for srTCM also consists of token update function, bucket update function, metering function, and marker interface function. Token update function consists of two token buckets, green token bucket and yellow token bucket. The token generation rate of both buckets is CIR. The metering function the necessary signals to exchange DSCP (Differentiated Service Code Point) field to the new color value. This signal is sent to marking function&output interface block through marker interface function.

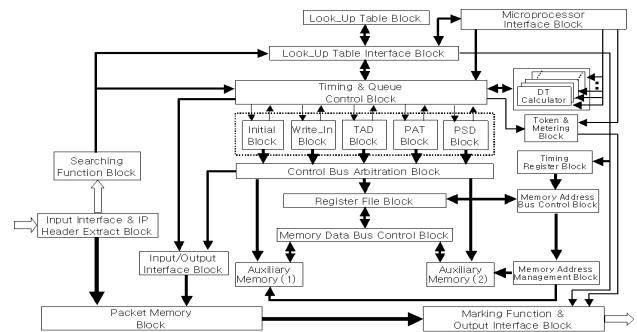


그림 7. FPGA을 위한 srRAS 블록도
Fig. 7. srRAS Block Diagram for FPGA.

Table 1 shows the virtual memory architecture of srRAS. It consists of address management RAM (AD RAM), flow queues(flow RAM), timing queues(timing RAM), departure queues (departure RAM), and their registers.

표 1. 메모리 구조
Table 1. Memory Architecture.

	AD RAM	Flow RAM	Timing RAM	Departure RAM			
RAM	AD RAM	FHP	FTP	THP	TTP	FID	NP
Reg.	ADB	FHTB	FHPB	THPB	TTPB	FID	NPB

AD RAM and flow RAM are related to flow queue architecture. AD RAM has a header pointer for idle starting address where the next arriving packet can use, and a trail pointer for specifying the end of idle address. flow RAM has a header pointer for the beginning of appropriate flow queue, and a trail pointer for the end of appropriate flow queue. timing RAM and departure RAM are related to timing and departure queue architecture. timing queue links the packets whose departure time are identical. because packets in timing queue are HOL packets of each flow, they can be uniquely identified by their FID in timing queue. the departure queue links the packets whose departure time are greater or equal to RT (Real Time) value.

Figure 8 shows an srTCM block diagram for FPGA. The srTCM has a CPU interface, input interface, marker interface, bucket update block, metering block, token update block.

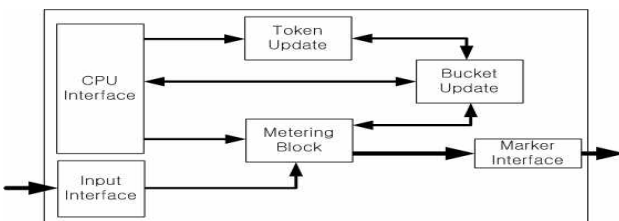


그림 8. FPGA를 위한 srTCM 블록도
Fig. 8. srTCM Block diagram for FPGA.

The CPU interface assigns the initial values of srTCM, such as CIR, CBS, EBS, unit time, and clock speed. The metering block performs the metering algorithm of srTCM, and then passes the results to marker interface. This block performs the calculation based on the received packet information from timing&queue control block and the current bucket value, and then determines the output color value and whether or not to update one of the two buckets according to calculation results. The token update block performs the calculation periodically using the current values of

Bc and Be and the parameters (CIR, S, CBS, EBS, etc.). It then determines whether or not to update Bc and Be values. The bucket update block manages the Bc and Be values between the metering block and the token update block because the bucket update block can receive the same Bc and Be values from different blocks (metering block or token update block). If this block receives the update request of Bc or Be from the metering block or the token update block, it updates the current value into the received value.

Figure 9 shows the bucket update operation. This operation has nine states, that is S0~S8. In S0, this operation initializes both token buckets, Bc and Be, at their maximum bucket size. After initialization of both token buckets, this block checks the update request signal for Bc or Be from the metering block or the token update block. In S1, this block checks the update request signal for Bc from the token update block. If the signal is there, this block updates the current Bc value to the requested value in S2. Otherwise, this block checks the update request signal for Bc from the metering block in S3. If the signal is there, this block updates the current Bc value to the requested value in S4. This operation is the same for the token bucket Be in S5~S8.

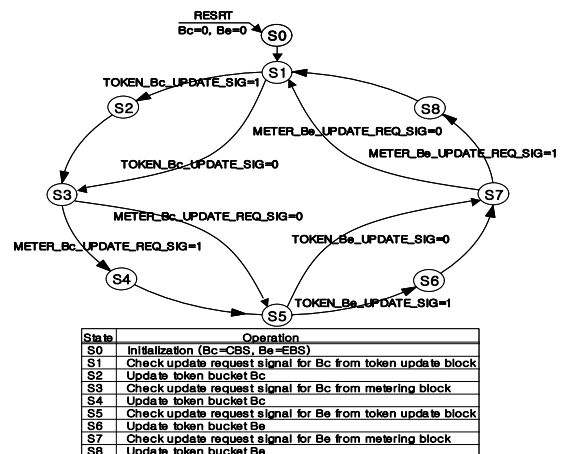


그림 9. srTCM에서 버킷 갱신 동작
Fig. 9. Bucket update operation in srTCM.

Figure 10 shows the operation of the token update block. There are four states in this operation.

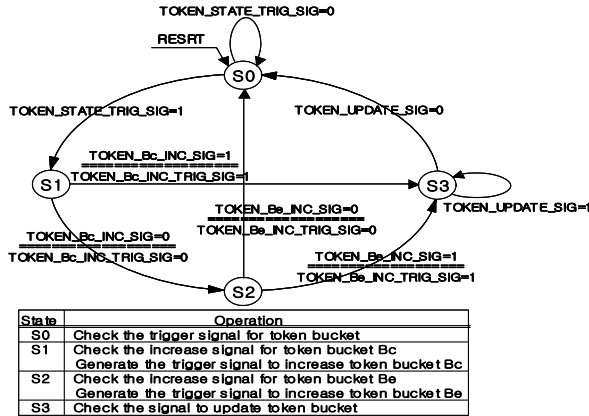


그림 10. srTCM의 토큰 갱신 동작
Fig. 10. Token update operation of srTCM.

In state S0, the bucket block checks the triggering signal for the token bucket. In state S1, this block checks the increase signal for token bucket Bc, and then generates the trigger signal to increase Bc. In state S2, this block checks the increase signal for token bucket Be, and then generates the trigger signal to increase Be. In state S3, this block checks the signal to update the token buckets, Bc or Be. This block generates the trigger signal to update token bucket Bc or Be, and then sends the signal and the token value for Bc and Be to the bucket update block.

Figure 11 shows the central processor state flow of timing&queue control block in srRAS. It consists srRAS and srTCM whose operations are parallel. the operation of each state in timing&queue control block according to figure 7 is shown in table 2. Because the bottleneck of srRAS operation is the time for calculating the DT from DT calculator block, another many operations can be performed during DT calculation.

In figure 11, timing&queue control block arbitrates write_in function, TAD function, PAT function, and PSD function, of queue control block,

and then generate the necessary to control the virtual memories and the packet memory.

In write_in function, newly arriving packets are appended to corresponding flow queue according to their FID (Flow IDentifier) values. first address and packet length are used as an address to store the packet in packet memory with first address increasing one-by-one by packet length, and set extension bit to one except of last location.

In TAD function, when a packet becomes the HOL (Head of Line) packet of flow queue, it will be assigned a departure time from DT calculator block and join a timing queue. In PAT function, As the real time ticks, the timing queue whose DT equal to RT (Real Time) will become the departure queue or be appended to the tail of the departure queue, dependent on whether or not the departure queue is constructed. In PSD function, The HOL packet of the departure queue is read out. Its content, FID is then used to access the HOL packet of the corresponding flow queue, where the packet address is obtained to transmit the packet in the packet memory.

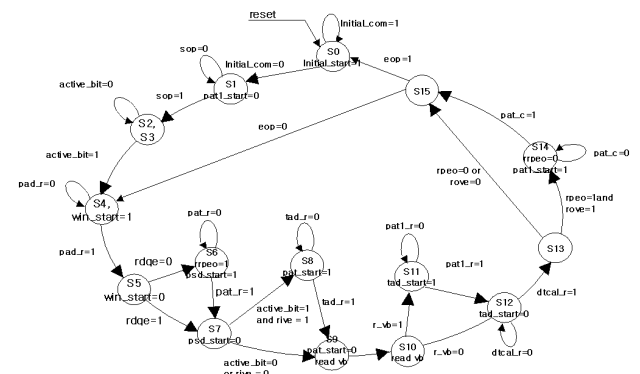


그림 11. Central processor 동작 흐름도
Fig. 11. Central processor state flow diagram.

Table 2 shows the operation flow of central processor according to central processor state flow shown in figure 11. the operation of central processor consists of 16 states (S0~S15). As the bottleneck of srRAS operation is the time for

calculating the DT from DT calculator block, another many operations can be performed during DT calculation. the quantitative analyses for RAS design must be considered, which can evaluate the

performance of srRAS. Four parameters for design are memory size and it's structure, the number of clock cycles, clock rate, and DT calculation time.

표 2. Central processor의 동작 흐름

Table 2. Operation flow of central processor.

Current State	Next State	Description	Remark
S0	S1	Reset Start the Initialization of auxiliary memory	
S1	S2	End the Initialization of auxiliary memory	
S2	S3	Check SOP(Start of Packet) bit	pat1_start disable
S3	S4	Check Active_bit	
S4	S5	Start the operation to store the packet into packet memory Wait psd_r = 1(padfsm block ready)	winfsm block enable
S5	S6(Rdqe=0), S7(Rdqe=1),	Check rdqe bit (0=>S6, 1=>S7)	winfsm block disable
S6		Rrpeo=1(reading out packet), psd_start = 1	Departure Queue present padfsm start
S7	S8(both=1) S9(ether=0 or tad_r=1)	psd_start = 0 Check active_bit and rive bit	Departure Queue absent padfsm disable
S8	S9	pat_start =1 Wait tad_r(tadfsm block ready)	Patfsm block enable
S9	S10	pat_start=0 Read vb(Validity bit for timing queue)	Patfsm block disable
S10	S11(vb=1) S12(vb=0)	Read vb(Validity bit for timing queue) Check vb bit	
S11	S12(pat1_r=1)	tad_srst = 1 Wait pat1_r bit(patfsm block ready)	tadfsm block enable
S12	S13(dtcal_r=1)	tad_srst = 0 Wait dtcal_r(DT calculation ready)	tadfsm block disable
S13	S14(both=1) S15(ether=1)	Check rpeo(reading out packet from packet memory) bit and rove(flowing queue empty after output) bit	
S14	S15(pat_c=1)	Set rrpeo=1 and pat1_start = 1 Wait pat_c(patfsm operation complete) bit	Notify the output and patfsm block enable
S15	S0(EOP=1) S4(EOP=0)	Check eop bit	

V. Conclusion

This paper addresses the scheme to design srRAS proposed RFC2963 using FPGA. It is different from shaper of ATM because of variable packer length. This shaper performs the shaping function, and then the marking function for each packet based on RFC2963. This shaper has been proposed to use at the ingress of differentiated services networks providing AF PHB. And then srRAS can reduce the burstiness of the upstream

traffic of srTCM. By reducing the burstiness of the traffic, srRAS increases the percentage of packets marked as green by the downstream srTCM.

In this architecture, IP shaper mainly consists of a packet memory, a register, a flow identifier searching function, a DT calculator, a look-up table, virtual memories and Timing&queue control block. Timing&queue control block of these components is key elements in this architecture. Virtual memory consist of three parts, flow part, timing part, and departure part. Each memory part

may contain a number of logical queues, and that logical queue is operated by the concept of linked queue. Timing&queue control block generates the necessary signal that are used to access all logical queues, and the appropriate address to read/write a packet from/to the packet memory.

Microprocessor does not play a many role in this architecture because of processing and accessing time. It sets the initial values of parameters. The calculation of the departure time of arriving packet, and the decision of packet's color are done in another block. Packet memory uses commercial memory to store the arriving packet. Using the necessary information of IP header in each packet makes flow identifier, and then this flow identifier is used internally in conjunction with the concept of linked queue.

In the near future, srRAS will be implemented in FPGA using VHDL. To verify function and algorithm of srRAS, test platform will be manufactured. And traffic generator and monitoring function will also be developed.

Reference

- [1] O. Bonaventure and S. De Cnodder, "A Rate Adaptive Shaper for Differentiated Services," RFC2963, October 2000.
- [2] Heenanen J. and R. Guerin, "A Single Rate Three Color Marker," RFC2697, September 1999.
- [3] Heenanen J. and R. Guerin, "A Two Rate Three Color Marker," RFC2698, September 1999.
- [4] P.-C. Chen, "The design of a timing processor for ATM traffic shaper," Masters thesis, Polytechnic University, Brooklyn, NY, Jul. 1995.
- [5] J. S. Hong, "Design of an ATM shaping multiplexer algorithm and architecture," Ph. D. dissertation,

Electrical Engineering Department, Polytechnic University, Brooklyn, NY, Jan. 1997.

- [6] H. Jonathan Chao and Xiaolei Guo, *Quality of Service Control in High-Speed Networks*, John Wiley&Sons, 2002.
- [7] Stocia I., Shenker S. and H. Zhang, "Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high-speed networks", ACM SIGCOMM98, pp118~130, Sept. 1998.
- [8] Greville Armitage, "Quality of Service in IP Networks : foundation for a Multi-Service Internet," MTP, April 2000.
- [9] J. Heenanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC2597, June 1999.
- [10] Srinvas Vegesna, *IP Quality of Service*, Cisco Press, 2001.
- [11] Chuck Semeria and John W. Stewart III, "Supporting Differentiated Service Classes in Large IP Networks," White Paper, Juniper Network, Inc. 2001.
- [12] Douglas J. Smith "HDL Chip Design -A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs using VHDL or Verilog," Doone Publications, 1996.

박 천 관 (朴天寬)



- 1987년 2월 : 건국대학교 전자 공학과 졸업 (공학사)
 1991년 8월 : 충남대학교 대학원 전자공학과 졸업 (공학석사)
 1996년 8월 : 건국대학교 대학원 전자공학과 졸업(공학박사)
 1997년 3월 : 목포해양대학교 전임강사
 1997년 3월 ~1998년 2월 : ETRI 초빙연구원
 2002년 2월~2003년 2월 : PolyTechnic University 방문연구원
 2000년 4월~현재 : (주)넷비전텔레콤 기술이사
 2006년 현재 : 목포해양대학교 부교수