

碩士學位論文

ATM에서 공정한 QoS 제공을 위한
문턱값 기반 버퍼 관리 알고리즘



濟州大學校 大學院 서관
JEJU NATIONAL UNIVERSITY LIBRARY
電氣電子工學科

高 維 新

2003年 6月

ATM에서 공정한 QoS 제공을 위한 문턱값 기반 버퍼 관리 알고리즘

指導教授 高 誠 澤

高 維 新

이 論文을 工學 碩士學位 論文으로 提出함



高維新의 工學 碩士學位 論文을 認准함.

審査委員長 _____ 印

委 員 _____ 印

委 員 _____ 印

濟州大學校 大學院

2003年 6月

Threshold Based Buffer Management Algorithm for Fair QoS in ATM

You-Shin Koh

(Supervised by Professor Sung-Teak Ko)



A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Engineering

2003. 6.

Department of Electric and Electronic Engineering
GRADUATE SCHOOL
CHEJU NATIONAL UNIVERSITY

목 차

Summary	1
약어설명	2
I. 서 론	3
II. ATM 트래픽 및 TBBM 알고리즘	6
1. 트래픽 서술자	6
2. 망 자원 관리	9
3. 큐 관리	11
4. 등가 용량	13
5. TBBM 알고리즘	15
III. 컴퓨터 시뮬레이션 및 실험 결과	19
1. ATM 트래픽 생성	19
2. 버퍼 크기와 CLR 관계	21
3. 다중화 출력셀들과 CLR 관계	24
4. CLR과 CLR 편차 특성	26
5. 성능평가	35
IV. 결 론	39
참고문헌	40
부록	42

Summary

A primary role of traffic management is to protect the network and end-system from congestion and to promote the efficient use of network resources. Theoretical equivalent capacity may significantly overestimate the required bandwidth for aggregate traffic. Statistical multiplexing can share a link with a capacity less than the sum of peak bit rate requirement for several variable-bit-rate connections, but it should not guarantee fairness of the channels.

We proposed the TBBM(threshold based buffer management) algorithm to improve fairness and utilization of ATM network. The result shows that the required bandwidth of TBBM algorithm is 14.3% lower in audio traffic and 41.8% lower in video traffic than that of theoretically calculated equivalent capacity method. And also the result reveals that the CLR(cell loss ratio) variation between channels of TBBM algorithm is much lower than that of statistically multiplexed traffic.

약어 설명

ATM	asynchronous transfer mode
B-ISDN	broadband integrated services digital network
CAC	call admission control
CDF	cumulative distribution function
CDV	cell delay variation
CLR	cell loss ratio
GCRA	generic cell rate algorithm
maxCTD	maximum cell transfer delay
MBS	maximum burst size
MCR	minimum cell rate
NPC	network parameter control
OCR	output cell rate
PC	priority control
PCR	peak cell rate
CDV	cell delay variation
rt-VBR	real-time variable bit rate
QoS	quality of service
SCR	sustainable cell rate
TBBM	threshold based buffer management
UNI	user network interface
UPC	usage parameter control
VCC	virtual circuit connection
VPC	virtual path connection
WFQ	weight fair queuing

I. 서 론

음성, 데이터 및 멀티미디어 서비스까지 수용할 수 있는 초고속 통신망의 기본이 되는 ATM(asynchronous transfer mode)기술이 90년대 초반부터 활발히 연구되어 93년부터는 상용서비스가 시작되었다. ATM망에서는 모든 데이터를 초고속으로 처리하기 위하여 고정길이의 셀 53bytes를 사용하며, 프레임릴레이 기술에서 사용하는 이용자 데이터 오류검출 기능도 제거하였으며, 셀 교환 기능도 대부분 하드웨어 소자에 의하여 이루어진다. 즉, ATM 기술은 회선교환기술의 장점과 데이터 교환기술의 장점을 모두 수용할 수 있게 되었다. (Othmar Kyas, 1997)

ATM의 장점은 효율성 증대 및 다양한 서비스를 동시에 지원할 수 있는 유연성이다. 그러나, 이러한 장점이 충분히 발휘되려면 트래픽 제어의 문제가 해결되어야 한다. 이를 위해서 ATM망은 연결 수락 제어(CAC : call admission control), 사용자 변수 제어(UPC : usage parameter control), 망 변수 제어(NPC : network parameter control), 우선 순위 제어(PC : priority control) 등의 트래픽 제어 능력을 제공하고 있다. 트래픽 제어 문제는 기존의 패킷 통신망에서도 많이 연구되었다. 그러나 이 경우에는 초고속 정보통신망(B-ISDN : broadband integrated services digital network)에서 고속의 연결성 및 비연결성 실시간 서비스가 요구하는 까다로운 서비스 품질이나 지리적으로 광활한 서비스 지역 등을 고려하지 않았었다. 그러므로, 기존의 패킷 망은 대역에 의해서 제한되었는데 B-ISDN은 기본적으로 지연에 의해서 제한된다. B-ISDN에서는 체증에 대한 대책이 체증이 일어난 후에 처리하는 것보다, 체증이 일어나지 않게 미리 예방하는 측면에서 검토되어야 한다는 점이다. ITU-T에서 제시한 트래픽 제어의 기본 목표는 다음과 같다. 첫째, 망의 보호이다. 즉, 일시적으로 많은 자원을 이용할 수 있는 권리가 악용되는 것을 막는 것이다. 둘째, 망의 성능 유지이다. 즉, 망이 제고하는 셀 전달 지연이나 셀 손실율 등의 성능지수가 어느 범위 내에 유지되도록 하는 것이다. 셋째, 망 자원의 효율적인 사용이다. 즉, 요구되는 서비스 품질

을 만족하는데 필요한 최소한의 자원만을 사용하도록 하는 것이다. (ATM forum, 1999)

다양한 요구사항을 가진 응용들을 지원하기 위해서는 대역폭의 증대뿐만 아니라 전송 신뢰성, 실시간성 등 충족시켜야 될 많은 기술들이 필요하다. 이러한 기술들을 간단히 표현하면 서비스/응용의 QoS(quality of service) 관리 기술이라고 통칭할 수 있다. 즉, 최선형 서비스 형태에서 탈피해서 차세대 인터넷의 상업용 QoS 서비스를 제공하기 위해서는 QoS 관리 기술이 필수 불가결한 요소이다. 종단 사용자간의 서비스 QoS는 사용자 단말, 전달 링크, 스위치나 라우터 같은 망 장치에서의 지연 및 처리율, 손실 특성 등에 의해 복합적으로 영향을 받게 되므로, 종단간 QoS 보장형 서비스를 지원하기 위해서는 이들 QoS 요소 변수들을 제어할 수 있는 네트워크 서비스 측면의 제어 메커니즘이 필요하다. 예를 들면, 망에서 실시간 서비스를 지원하기 위해서는 요구되는 자원을 미리 예약하거나 망 내의 트래픽 부하를 반영하여 라우터의 큐잉 기능을 적절히 설정하거나 혹은 설정된 트래픽 특성이 제대로 운영되는지를 모니터링 하는 기능 등이 이에 해당한다. (최태상 외, 2000)

ATM은 연결과 대역할당의 독립성을 특징으로 하는 전송방식으로 호 연결에 대한 요청이 있을 때 가상 채널을 설정하여 정보를 전달하며, 가상채널이 설정될 때마다 사용자의 요구에 따라 전송용량을 할당하므로 기존의 동기식 전송방식보다 전송자원 활용 면에서 장점을 가지게 된다. 그리고, 특성이 다른 다양한 응용 서비스를 ATM 망에서는 통계적 다중화 방식으로 전송하므로 다른 전송 기법에 비해 전송효율이 높다. 반면에 효과적인 트래픽 제어를 수행하지 못할 경우 다중화기 버퍼에서의 셀 전송지연 및 오버플로우에 의한 셀 손실이 발생할 수 있는 문제점을 지니고 있어(Othmar Kyas, 1997), 스위치 내부 링크의 대역 사용량을 측정하여 폭주 및 트래픽 제어를 수행하고 링크 이용율을 높이기 위한 대역 재할당 알고리즘(Ricardo 2001), 신경회로망을 이용한 대역 관리(고성택 외 1998), 퍼지 이론을 이용한 대역 관리(Qiang, 2000)등 다양한 방법으로 제안되고 있으며, 대역폭 할당에 대한 이론적인 등가 용량에 대해서는 항상 오버에스티메이션 되어 있어 실제로 요구되는 대역폭보다 훨씬 높다.(Raif, 1995)

본 논문에서는 변동이 심한 실시간 음성과 영상 서비스들에 대해서 이론적으로 계산되어지는 등가 용량보다 높은 이용율을 제공하고, 통계적 다중화에서 제공하기 어려운 채널들의 공정한 QoS를 보장하기 위해 문턱값에 기반을 둔 버퍼 관리 알고리즘(TBBM : threshold based buffer management)을 제안하였다. 이 알고리즘의 목표는 각 채널에서 요구하는 다양한 QoS를 공정하게 제공하고, 동시에 적은 링크 대역폭을 사용하면서도 요구하는 QoS를 충분히 보장하는 것이다. 구현 방법은 문턱값을 기준으로 채널들의 버퍼 점유율에 따라 출력되는 셀을 차등하게 처리하며, 이때의 출력 셀율은 각 트래픽이 지나는 순간 최대 셀율(PCR : peak cell rate), 평균 셀율(ACR : average cell rate), 상수 α 에 의해 정하여 진다. 이렇게 함으로써, 예측이 어려운 입력 트래픽에 대하여 다중화 출력 셀율을 일정하게 유지 즉, 링크 대역폭을 일정하게 사용할 수 있다. 컴퓨터 모의 실험을 통하여 제안된 방법의 효율성을 평가하였다.

본 논문의 구성은 2장에서 TBBM 알고리즘을 제안하기에 앞서 ATM 망의 트래픽 서술자와 망 자원 관리, 큐 관리, 등가 용량에 대한 기본적인 사항을 알아보고, 3장에서는 실험에 사용될 ATM 트래픽 생성, 트래픽별 셀 손실율(CLR : cell loss ratio)과 버퍼 크기의 관계, 문턱값과 α 에 따른 CLR과 채널간 CLR 편차 등을 실험한 후 TBBM 알고리즘에 대해서 공정성과 이용율 측면에서 성능 평가를 하였다. 마지막 4장에서는 결론을 제시하였다.

II. ATM 트래픽 및 TBBM 알고리즘

1. 트래픽 서술자 (traffic descriptor)

ATM 네트워크에서 하나의 연결을 설정하는 작업 중 하나는 통신부하 계약을 체결하기 위하여 관련된 스테이션들과 협상하는 일이다. 이 계약에는 연결되어 있는 동안 유지되어야 할 통신부하 특성들이 명시된다. 특정 스테이션간의 연결을 설정하는 동안에 요구한 통신부하 파라미터들의 목록은 근원지 통신부하 서술자에 보관된다. 통신부하 파라미터들은 통신부하 특성(예를 들면 순간 최대 셀 전송률, 평균 셀 전송률, 최대 부하의 지속 기간과 전화 또는 비디오 등의 전송장치 타입)을 질량화하고 또한 정량화 한 것이다.

통신부하 계약은 요구된 QoS에 대해 고려되는 데 셀 손실율(CLR : cell loss ratio)와 셀 지연 변동(CDV : cell delay variation) 허용 한계 등과 같은 근원지 통신부하 설명자의 통신부하 파라미터들을 포함한다. (Othmar, 1997) 현재 ATM 포럼에서는 다양한 트래픽을 적절하게 수용하기 위해 트래픽의 특성과 요구되는 서비스 품질에 따라 ATM 서비스 등급을 고정 비트율(CBR : constant bit rate), 실시간 가변 비트율(rt-VBR : real-time variable bit rate), 비실시간 가변 비트율(nrt-VBR : non real-time variable bit rate), 가용 비트율(ABR : available bit rate), 비규정 비트율(UBR : unspecified bit rate), 보장 프레임율(GFR : guaranteed frame rate)로 구분하였다. (ATM forum, 1999)

이 논문에서 다루어질 실시간형 음성과 영상 서비스는 rt-VBR로 분류되며 대표적인 데이터형태로는 voice mail, telephone conversation, videotex, HDTV - quality TV 등이 있다. 데이터의 특성상 셀 손실뿐만 아니라 지연과 지연변동에 대한 허용오차가 매우 낮으며 이런 버스트한 트래픽 특성을 정형화하는 데에는 PCR 등 트래픽 서술자들이 사용된다.

1) 순간 최대 셀 전송률 (PCR : peak cell rate)

순간 최대 셀 전송률은 최소 시간 t 의 역수로 정의되는데 최소시간 t 는 ATM 셀을 전송하라는 명령이 물리계층 서비스 접속점에 도착한 상태에서 다음 번 명령이 도착할 때까지의 경과해야 할 최소 경과시간을 나타낸다.

2) 유지 가능 셀 전송률 (SCR : sustainable cell rate)

유지 가능 셀 전송률은 평균 셀 전송률의 상한 허용치이다. 이 경우에 모든 네트워크는 순간 최대 셀 전송률을 기준으로 하기보다 유지 가능 셀 전송률을 기준으로 예약한다. 셀 전송률이 유지 가능 셀 전송률보다 높지만 일반 셀 전송률 알고리즘(GCRA : generic cell rate algorithm)이 정하는 폭주 허용한계 범위 내에 있을 경우는 통신부하 계약을 따른다고 간주한다. 통신부하 계약을 따르지 않는 셀들은 UPC에 의해 표시되거나 제거된다. 유지 가능 셀 전송률과 체증 허용한계는 순간 최대 셀 전송률보다 좀더 정확하게 셀 흐름을 설명할 수 있다.

3) 최대 버스트 크기 (MBS : maximum burst size)

ATM 신호 처리 메시지에서 버스트 허용 오차는 많은 셀로 코딩되는 MBS를 통해 전달된다. SCR이나 GCRA와 함께 버스트 허용 오차는 GCRA의 규정에 일치하는 상태로 순간 최대 속도에서 전달할 수 있는 MBS를 결정한다.

4) 최소 셀 비율 (MCR : minimum cell rate)

ABR 전송을 위해서만 정의되며, ACR의 최소값으로 명기한다.

5) 최대 셀 전송 지연(maxCTD : maximum cell transfer delay)

ATM의 특정한 연결 방식에서 발신 사용자 망 인터페이스(UNI : user network interface)에서의 셀 탈출 이벤트와 수신 UNI에서의 그에 해당하는 셀 진입 이벤트 사이에서 경과된 최대 시간으로 두 지점 사이의 maxCTD는 총 ATM 노드 간 전송 지연 시간과 총 ATM 노드 프로세싱 지연 시간의 최대 합이다.

6) 피크-투-피크 셀 지연 변동(Peak-to-peak CDV : cell delay variation)

ATM의 타임슬롯 특성과 관련해서 한 라인에 다수의 ATM 셀 스트림을 다중화 하거나 셀 스트림에 동작 관리 유지 셀(OAM : operations administration and maintenance)들을 삽입하기 때문에 셀을 전송하는 과정에서 자주 지연을 초래한다. 셀 지연 변동은 주어진 연결에 대하여 최종 수신노드에서 연속된 두 셀의 도착사이의 경과한 시간 간격이 바뀌는 정도를 설명한다. 셀 지연 변동 허용 한계는 GCRA의 제한 인자의 함수이고, 순간 최대 셀 전송률의 역수 T는 GCRA의 증가 인자이다.

7) 셀 손실율(CLR : cell loss ratio)

특정 기간동안 손실된 셀들의 개수와 이 기간 중 총 전송된 셀들의 개수의 비율이다. 심한 에러를 가진 셀 블록 기간 중 전송되거나 손실된 셀들은 포함되지 않는다.

8) 셀 오류율(CER : cell error ratio)

전송 에러가 발생한 셀들을 전송에러가 발생한 셀들과 성공적으로 전송된 셀들을 합한 수로 나눈 값이다. 심한 에러가 집중적으로 발생하는 셀 블록은 여기에 포함시키지 않는다.



2. 망 자원 관리

ATM 네트워크 트래픽 관리는 네트워크 어플리케이션을 위한 서로 다른 QoS를 적절하게 제공하기 위한 네트워크의 능력과 근본적으로 관련이 되어 있다. ATM 포럼의 트래픽 관리에서는 트래픽을 관리하기 위한 과정과 트래픽과 QoS에 관련된 서술자들을 정의한다. 트래픽 관리의 주요한 목적은 혼잡으로부터 망과 중단 시스템을 보호하는 것이며, 부가적으로는 망 자원의 능률적인 사용을 촉진하는 것이다.

호 설정 단계에서 연결 요구(또는 재연결 요구)를 수용할 수 있는지 또는 거절해야 하는지를 결정하기 위해 네트워크가 취하는 행동을 CAC라 한다. ATM망에서 하나의 연결 요구는 CAC 결과를 근거로 하여 허용되는데, 호가 요구하는 QoS를 망 전체를 통해 지원 가능하고 이미 존재하는 호들의 협정된 QoS 또한 보장할 수 있는 충분한 자원들이 사용 가능할 때만 허용한다. 연결 후 네트워크 요소들의 상태에 따라 ATM 연결에 수용된 트래픽을 정규화하기 위해 네트워크와 중단 시스템간에 일련의 피드백을 취한다. 또한, 망 자원을 보호하기 위해 UPC와 NPC가 네트워크 노드 인터페이스에서 호 설정시 협정된 QoS의 준수 여부와 트래픽을 모니터링을 하는데, 주요한 목적은 이미 설정된 다른 연결의 QoS에 영향을 미칠 수 있는, 고의적이든 고의적이지 않든, 협상된 파라미터를 위반하는 행위로부터 네트워크 자원들을 보호하는 것이다. 좀더 효율적인 네트워크를 위해 트래픽 특성들을 요구하는 데로 수정하기 위해 사용하는 트래픽 셰이핑은 네트워크를 통신 폭주 상태로 만들 수 있는 통신량 급증을 제한하기 위해 대기열을 사용하는 것으로, 데이터는 버퍼 처리 후 조절된 크기로 네트워크로 보내지므로 트래픽이 특정한 연결 장치에 대해 약속된 트래픽 공간 내에 맞게 할 수 있다.

망 자원 관리 측면에서의 트래픽 관리는 이용 가능한 망 자원을 요구된 연결들에게 효율적으로 할당함으로써 수행되는데, 현재까지는 가상 경로(VP : virtual path)를 이용하여 트래픽을 제어하는 방법에 대해서만 표준화가 된 상태이며, 다

른 기법에 대해서는 계속 연구 중에 있다. 가상 경로 연결(VPC : virtual path connection)를 이용한 트래픽 제어는 연결 수락 제어과정을 단순화하며, 서로 다른 QoS를 요구하는 트래픽 형태들을 분리함으로써 일종의 우선 순위 제어를 구현할 수 있게 하는 데 있다. 아울러 VPC를 이용하면 트래픽 제어 안의 작동을 위한 메시지를 해당 VPC내의 모든 가상 채널 연결(VCC : virtual channel connection)로 동시에 보낼 수 있다. (ATM forum, 1999)

이처럼, ATM을 효율적으로 실현하기 위해서는 두 개의 서로 상반되는 요구사항을 적절히 만족시켜야 한다. 즉, 요구되는 QoS를 만족시키면서 사용 대역폭은 가능한 낮추어야 한다. 요구되는 대역폭을 효율적으로 낮추는 것은 통계적 다중화에 의하거나, 동적 대역폭 할당 등에 의해서 실현될 수 있다. 통계적 다중화란 N개의 소스를 서로 다중화 하여 한 개의 소스로 만드는 것이다. N개의 소스가 서로 합해진 경우의 효과적인 대역폭은 한 개의 소스에 의한 효과적인 대역폭을 N배한 것보다 일반적으로 적다.

요구되는 대역폭은 사용자와 네트워크간에 트래픽 계약을 서로 협상하면서 결정된다. 이러한 대역폭의 배정은 일정 주기마다 혹은 소스로부터의 요구에 의해서 변경된다. 시간에 따라 변하는 트래픽의 경우 시간에 따라 서로 다른 대역폭을 배정하는 것 즉, 대역폭을 동적으로 할당함으로써 회선의 이용율을 높이게 된다. 현재의 트래픽 양에 따라 동시에 현재의 요구되는 대역폭을 할당하는 것은 불가능하다. 왜냐하면, 자원의 예약은 최소한 프로토콜이 네트워크를 왕복하는데 필요한 시간만큼의 지연시간이 필요하기 때문이다. 또한, 이러한 동작은 네트워크의 프로세싱에 크게 부담을 주어서는 안 된다. 따라서, 동적 대역폭을 할당하기 위해서는 현재나 과거의 트래픽 정보들로부터 미래의 트래픽에 대한 정확한 예측을 필요로 한다. ATM 트래픽의 특성상 미래에 대한 트래픽을 정확히 예측하는 것은 대단히 어렵다. 그러나 일련의 ATM 트래픽이 갖고 있는 어떤 연관성들로부터 미래의 트래픽에 대한 정확한 예측은 할 수 없더라도 미래의 트래픽의 대역폭에 대한 상한선이나 특정한 트래픽 서술자에 대한 예측은 가능하며 이에 대한 연구가 활발히 되어 왔다.(고성택 외 1999)

3. 큐 관리(Queue management)

큐잉(queuing)은 라우터 구조에서 핵심 부분으로 프레임들이 입력 프로세스를 거쳐 패킷으로 합쳐지고 무결성을 검사 받은 뒤 포워딩 프로세스에 의해 출력될 인터페이스가 결정되고 출력 프로세스에 의해 다시 프레임화 되어 다음 라우터로 전달되는 과정을 포함한다. 큐 관리는 이 과정에서 프로세스간의 연결 역할을 하게 된다. 따라서 큐잉이 지금까지, 현재, 그리고 미래의 서비스 차별화를 위해 해오거나 할 역할을 이해하는 것이 매우 중요하다. (최태상, 1999)

먼저, FIFO(first in first out) 큐잉은 트래픽을 선저장 후전달 방식으로 처리하는 가장 대표적인 방법이다. 네트워크의 대역폭이 충분히 크고 스위칭/포워딩 성능이 뛰어난 버스트 트래픽만 처리하면 되는 경우가 대부분이므로 FIFO 큐잉 방식이 적절하지만, 실제 상황에서는 트래픽의 양이 FIFO 큐가 처리할 수 없을 만큼 발생할 경우가 많이 생기고 있다. FIFO 큐가 채워질 경우 서비스의 종류와 무관하게 패킷이 버려지는 상황이 발생하여, 차별적인 서비스를 제공해야 할 경우에 한계에 부딪히게 된다. 따라서 아래에서 설명될 다른 형태의 방법이 필요하게 된다.

우선 순위 큐잉은 FIFO 큐잉 방식을 변형한 것으로써 특정 유형의 트래픽을 구분하여 출력 큐의 앞부분으로 보내 먼저 처리될 수 있도록 한 방식이다. 이 방식은 가장 초보적인 서비스 차별화를 가능하게 하지만 여러 가지 단점을 가지고 있다. 서비스 차별화 단계를 많이 만들수록 처리 부담을 가중시키고 패킷 포워딩 성능을 저하시키게 된다. 그리고, 가장 높은 우선 순위 트래픽의 양이 많을 경우 순위가 낮은 트래픽은 버퍼 고갈로 인해 손실율이 높아지게 되고, 지연에 민감한 응용의 경우에는 제대로 작동을 하지 않을 수도 있게 된다. 점차 고속화되는 네트워크 환경에서 이 큐잉방식은 확장성을 지원하기가 힘들다. 우선 순위 큐잉 방식의 단점인 우선권이 있는 클래스를 제외한 타 클래스 트래픽의 자원을 완전히 거부하는 경우를 방지하기 위해서 최근에 제안된 방식으로 동류의 알고리즘이 이미 운영체제에서 많이 활용되고 있다.

클래스 기반 큐잉(CBQ :class based queuing)는 우선 순위 큐잉방식의 변형으로써 하나의 출력 큐 대신에 여러 개의 출력 큐를 클래스별로 두어서 우선 순위를 정하고 각 큐별로 서비스되는 트래픽의 양을 조절할 수 있는 방식이다. 이렇게 함으로써 어느 특정 클래스의 트래픽이 전 시스템 자원을 모두 독점하는 것을 방지한다. CBQ는 각 클래스별로 정해진 양의 대역폭을 보장할 수 있는 방식으로 알려져 있지만, 실제로는 완전 보장이라기 보다는 우선 순위 큐잉 방식보다는 약간 더 완화된, 클래스별로 자원이 완전 고갈되는 것을 막으면서도 각 클래스에 적절한 서비스를 제공할 수 있는 방식이다. 그러나, 이 방식도 여전히 복잡한 큐 관리에 소요되는 계산 부담 때문에 고속의 네트워크의 경우에는 확장성이 부족하게 된다.

가중치 공정 큐잉(WFQ : weight fair queuing)는 소량의 트래픽이 대량의 트래픽에 의해서 피해를 보지 않도록 플로우 별로 큐를 두어 트래픽을 조절하는 공정성 측면과 특정 기준에 따라 가중치를 정하고 이에 따라 같은 양의 트래픽을 가진 플로우 간에도 차별을 두는 가중치 측면을 복합적으로 적용한 큐잉 방식이다. WFQ 방식도 우선 순위 큐잉 혹은 CBQ와 비슷한 특성을 가지고 있는 관계로 고속의 네트워크 환경에서 확장성을 가지기가 어렵다. 또한, 트래픽 플로우 간을 차별 할 수 있는 메커니즘의 부재가 이 방식의 큰 단점이다.

4. 등가용량

어떤 자원에 대한 대역폭의 할당은 대체로 효과적인 대역폭(effective bandwidth)에 의한 할당, 가상 대역폭(virtual bandwidth)에 의한 할당, 배치 포아송 방법(batch poisson approach)에 의한 할당, 정적 비트율(stationary bit rate approach)에 의한 할당 등의 방법에 의해서 이루어지고 있다.

효과적인 대역폭에 의한 대역폭의 할당은 어떤 ATM 신호원에 대한 등가 용량(equivalent capacity)에 근거하여 이루어진다. 2상태 모델을 갖는 신호원의 등가 용량의 산출은 정상상태(steady state)에서 버퍼가 가득 차고 소스가 활동상태에서 오버플로우에 의한 셀 손실이 발생할 확률을 계산함으로써 얻어진다. 정상 상태에서 오버플로우가 발생할 확률 p 는 식 (1)과 같이 주어진다. (Raif, 1995)

$$p = \frac{\rho(c - \rho R) \exp^{-X(c - \rho R)/[b(1 - \rho)(R - c)]}}{(1 - \rho)c - \rho(R - c) \exp^{-X(c - \rho R)/[b(1 - \rho)(R - c)]}} \quad (1)$$

여기서, R : 최대 셀율

ρ : 소스 이용율 (source utilization),

c : 링크 속도 (link speed) 혹은 등가용량(equivalent capacity),

b : 활동구간의 평균 길이이다.

위 식은 약간의 수식에 대한 계산과 c 에 대한 상한을 적용함에 따라서 식 (2)과 같이 간략화 된다.

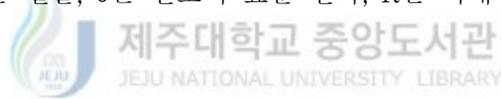
$$c = R \frac{y - B + \sqrt{(y - B)^2 + 4B\rho y}}{2y} \quad (2)$$

여기서, $y = (1 - \rho)R \ln\left(\frac{1}{\rho}\right)$, B 는 버퍼 크기이다.

위 식은 등가용량은 신호의 평균값, 피크값, 이용률, 버퍼의 크기, 그리고 CLR에 의존함을 보여주고 있다. 따라서, 동적 대역폭을 할당하기 위해서는 주어진 신호에 대한 서술자들인 평균값, 피크값, 이용률과 네트워크가 제공할 수 있는 버퍼의 크기 그리고 요구되는 QoS를 만족시키는 등가 용량을 계산하여 그 값으로 대역폭을 할당하게 된다. 그러나, 위의 등가 용량 값은 식을 간략화 하는 과정에서 상한을 사용함에 의해서 실제 요구되는 값보다 훨씬 높은 값을 가지므로 이용률의 관점에서 많은 손실을 보게 된다. 실제의 경우 인터폴레이션(interpolation)에 의해 얻어진 등가용량 계산이 더 편리하며 인터폴레이션된 등가용량은 식 (3)과 같다.

$$C = A_v + \frac{A_v \sigma}{R} \quad (3)$$

여기서, A_v 는 평균 셀율, σ 는 신호의 표준 편차, R 은 최대 셀율이다.



5. TBBM 알고리즘

네트워크 구조가 최상의 ATM 스위칭 유닛인가는 얼마나 많은 입/출력 포트의 수를 가질 수 있고, 얼마나 낮은 CLR에서 동작할 수 있는가로 판별된다. 버퍼의 크기는 셀의 손실이 얼마나 될 것인지를 결정하는 가장 큰 요소이다. 스위치 유닛의 성능은 일반적으로 셀 처리율, 셀 전송 지연, CLR의 세 가지 파라미터로 표시한다. 셀 처리율이란 스위칭 유닛이 처리할 수 있는 부하를 의미하며 한 타임슬롯동안 한 출력포트에서 셀이 출력될 확률로 표시된다. 최대 용량은 부하가 1인 경우이다. 셀 전송지연은 스위칭 유닛을 통과하는데 걸리는 지연 시간을 타임슬롯의 단위로 표시한다. CLR은 한 개의 입력포트로 셀이 입력된 상태에서 스위칭 유닛을 통과하는 동안 셀이 손실될 확률이다. (Othmar, 1997)

본 논문에서는 각 채널에 할당된 버퍼로 입력되는 트래픽에서 요구하는 QoS를 최대한 공정하게 보장할 뿐만 아니라 대역폭 사용율도 줄이기 위해 문턱값을 기반으로 하는 TBBM 알고리즘을 제안한다. TBBM 알고리즘의 기본 동작은 그림 1에 나타난 것처럼 입력 버퍼에 문턱값(TH : through)을 두어서 입력 버퍼로 들어오는 트래픽에 대해 버퍼 점유율에 따라 입력 버퍼에서 출력되는 셀율(OCR : output cell rate)을 조절하는 일종의 WFQ의 변형이다. 다중화 출력 셀율을 일정하게 유지하기 위해 버퍼 점유율이 문턱값을 초과하는 경우의 OCR은 문턱값을 초과하지 않는 경우의 OCR보다 더 많이 처리하며 반대로 버퍼 점유율이 문턱값을 초과하지 않는 경우의 OCR은 문턱값을 초과하는 경우의 OCR보다 더 적게 처리한다. 이렇게 함으로써, 채널들에서 문턱값이 없을 때 CLR 값보다 훨씬 적은 CLR을 얻을 뿐만 아니라 채널들간의 CLR 편차도 줄일 수 있다. 즉, 이 알고리즘의 목적은 다중화가 이루어지는 N개의 채널에 대해 QoS를 개별적으로 공정하게 제공함과 동시에 다중화 출력 셀율을 일정하게 유지하는 것으로, 시간에 대해 변동이 심한 특성을 지니는 rt-VBR 트래픽들의 대역폭 사용에 대한 예측이 가능하므로 간단한 CAC 구현 등 효과적인 트래픽 관리가 가능하게 될 것이다.

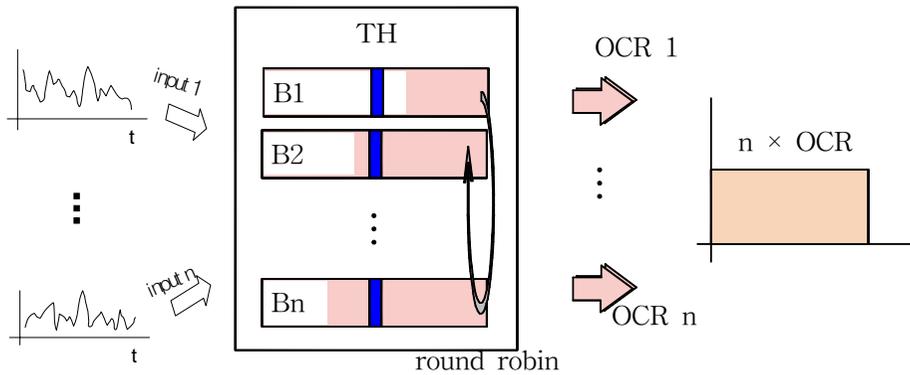


Fig. 1. TBBM algorithm.

예로, 그림 1에서 버퍼 B1은 버퍼 점유율이 문턱값 미만인 경우로 출력되는 OCR1은 버퍼 점유율이 문턱값을 초과하는 경우인 B2에서 출력되는 OCR2보다 작다. 버퍼 점유율이 문턱값을 초과하는 버퍼 수가 많을수록 처리되는 OCR1은 더 적어지고, 버퍼 점유율이 문턱값을 초과하는 버퍼 수가 작을수록 처리되는 OCR1은 더 많아진다. 상대적으로, 버퍼 점유율이 문턱값을 초과하는 버퍼 수가 많을수록 출력되는 OCR2는 더 적어지고, 버퍼 점유율이 문턱값을 초과하는 버퍼 수가 작을수록 출력되는 OCR2는 더 많아진다. 즉, 다중화 되는 다른 버퍼들의 상황에 따라 가감되는 출력 셀율은 다르며, 입력되는 트래픽들의 서술자인 PCR와 ACR의 차에 비례하도록 하였다. 식 (4)와 같이 버퍼 점유율이 문턱값을 초과하는 경우에는 OCR에 β 만큼을 더하여 처리하고, 버퍼 점유율이 문턱값을 초과하지 않는 경우에는 OCR에 γ 만큼을 빼어서 처리하며, 문턱값을 초과하지 않는 버퍼 수가 m 이라고 가정하면, $[(n-m) \times \beta]$ 와 $(m \times \gamma)$ 의 합이 "0"이 되도록 출력 셀율을 조절한다. 입력되는 트래픽들이 채널을 지나면 서는 다양한 출력 셀율을 보이지만, N개 소스에서 다중화 되는 출력 셀율은 일정하게 유지될 수 있다. 식 (4)은 n 개의 채널을 다중화 하는 경우 필요 대역폭 B를 구하는 식이며, β 와 γ 는 PCR과 ACR, a 로 정의된다.

$$\begin{aligned}
 OCR_{Total} &= \sum_{i=1}^n OCR_i \\
 &= (n-m) \times (OCR_{ch} + \beta) + m \times (OCR_{ch} - \gamma) = n \times OCR_{ch} \quad (4)
 \end{aligned}$$

- 버퍼 점유율이 문턱값 이상인 경우

$$OCR_i = OCR_{ch} + \beta$$

$$\beta = \frac{((PCR - ACR) \times \alpha)}{m}$$

- 버퍼 점유율이 문턱값 이하인 경우

$$OCR_i = OCR_{ch} - \gamma$$

$$\gamma = \frac{((PCR - ACR) \times \alpha)}{n - m}$$

여기서, OCR_{Total} 은 n개의 채널을 다중화하는 경우에 출력 셀율

OCR_{ch} 는 채널 출력 셀율,

OCR_i 는 버퍼 점유율에 따른 채널의 출력 셀율,

n은 입력 채널 수,

m은 문턱값을 이하인 버퍼 수,

α 는 0과 1 사이의 값을 갖는 상수이다.

만약, α 가 0이라면 문턱값에 상관없이 모든 채널에서의 출력 셀율은 OCR일 것이며, α 가 1이면 β , γ 값이 최대가 되는 경우로 버퍼 점유율에 따라 채널간 출력되는 셀율의 차가 가장 크다.

TBBM 알고리즘에 대한 일련의 절차는 그림 2에서 보여지고 있다. 이 논문에서는 실시간 오디오와 비디오 트래픽에 대한 모델(Lily Cheng, 1995)로부터 트래픽을 생성하여 실험하였다. 여기서, n_Buffer는 채널 수, GC는 생성된 트래픽, BC는 버퍼내의 셀 수, TH는 버퍼의 문턱값, BS는 버퍼 크기, LC는 손실되는 셀 수, overTH는 버퍼 점유율이 문턱값을 초과하는 경우의 버퍼 수, underTH는 버퍼 점유율이 문턱값을 초과하지 않는 경우의 버퍼 수이다.

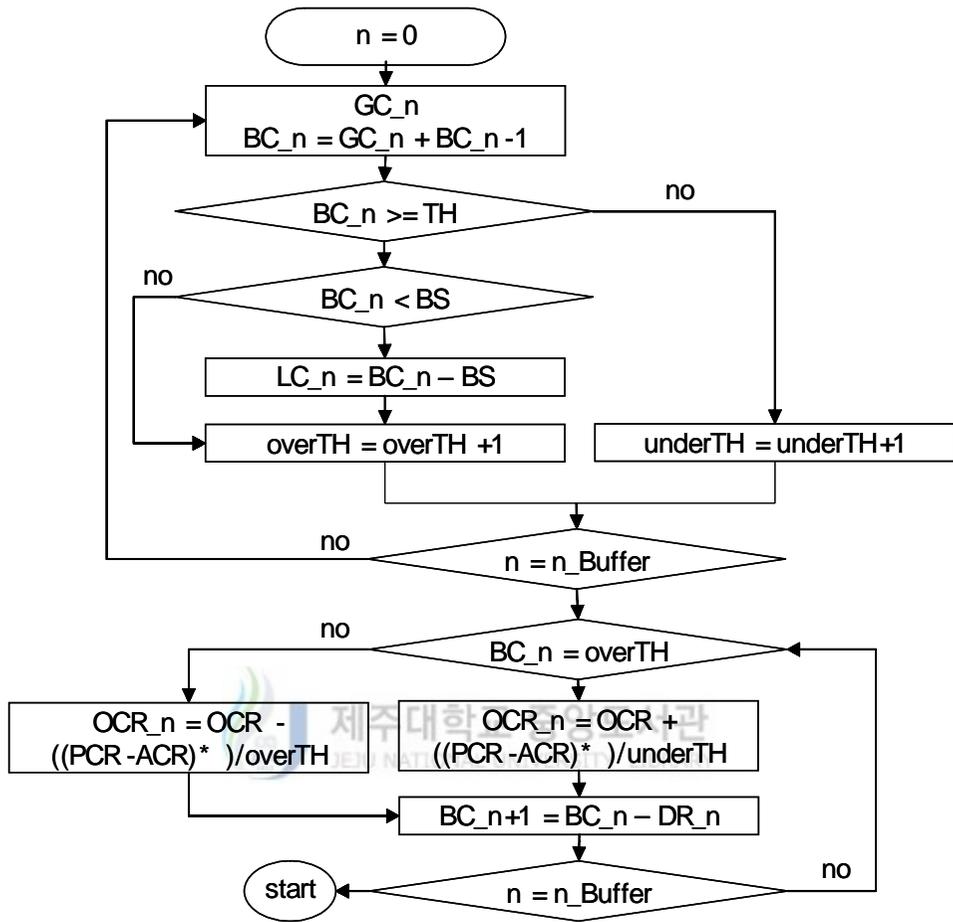


Fig. 2. Flowchart of TBBM algorithm.

Ⅲ. 컴퓨터 시뮬레이션과 실험 결과

1. ATM 트래픽 생성

트래픽 모델링, 혼잡 제어 알고리즘을 위한 디자인, 통신 프로토콜 실행, 그리고 스위치 구조 개발하는 것과 같은 많은 관점들 안에서 ATM의 특징을 조사하는 것은 매우 중요하며, 측정된 많은 데이터로부터 트래픽의 특징들을 조사할 수 있다. 측정하여 모아놓은 데이터로부터 트래픽이 가지고 있는 피크 비율과 평균 비율 같은 트래픽 파라미터들이 계산된다. 측정한 데이터의 누적 분포들은 ATM 작업량 모델을 입증할 길로써 실험된 데이터와 측정된 데이터 사이에 어떤 중요한 차이도 없으며, ATM 네트워크를 위한 혼잡 제어 알고리즘을 작성하는데 유용하게 사용된다. (Lily Cheng, 1995) 그림 3은 실시간 측정된 오디오(audio)와 비디오(video) 트래픽들의 누적분포함수(CDF : cumulative distribution function)로 본 실험에서 트래픽 생성에 사용되었으며, 그림 4는 트래픽 생성에 대한 과정을 보여주고 있다.

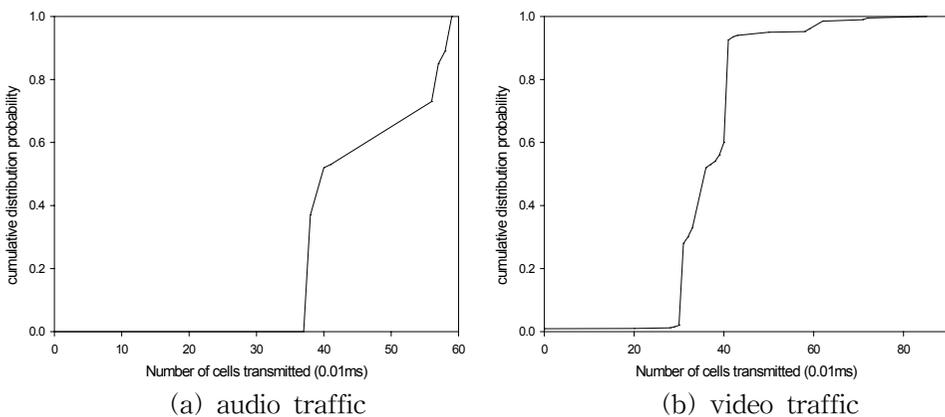


Fig. 3. CDFs of real-time audio and video traffic.

그림 5는 CDF로부터 생성된 비디오, 오디오 트래픽을 보여주고 있으며, 발
생된 트래픽으로부터 서비스별 트래픽이 지나는 트래픽 서술자들인 최대 셀율,
평균 셀율 등을 얻을 수 있다.

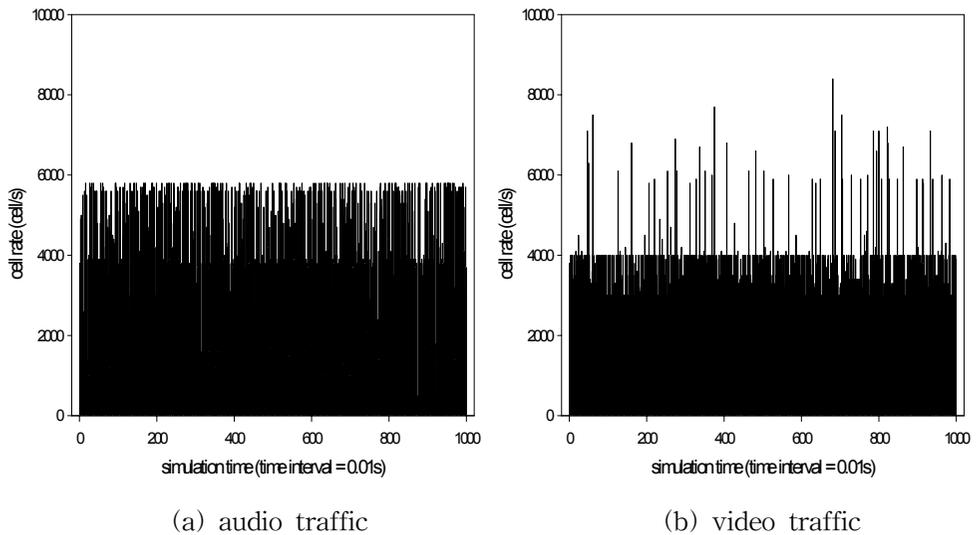
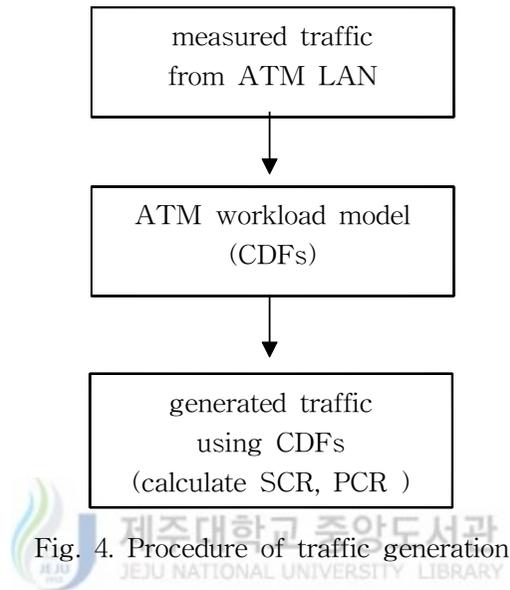


Fig. 5. Generated audio and video traffic using CDFs.

2. 버퍼 크기와 CLR 관계

일반적으로 CLR에 대해서 의미 있는 통계적 결론을 내리기 위해서는 많은 양의 셀에 대해 관찰하여야 하며 이는 비효과적인 네트워크 관리라는 반작용을 초래한다. 또한, 도착처리에 대한 다양성에도 불구하고 ATM 네트워크에서의 CLR이 버퍼 크기와 관계가 있다는 것이다. 이는 실제의 시스템의 행동을 구체화하기 위해 시스템내의 버퍼를 이용하는 대신 가상의 작은 버퍼를 이용하여 셀 손실에 대하여 관찰할 수 있으며, 이 방법은 CLR에 대한 모니터링시 짧은 관찰 기간이 요구된다. (Qiang, 2000)

시스템내의 버퍼에서 CLR 값은 아주 작지만, 가상의 작은 버퍼에서는 높은 CLR 값을 가진다. CLR은 가상의 작은 버퍼에서 넘치는 셀을 집계하고 관찰하는 것으로 구할 수 있으며, 선형 감퇴 알고리즘(Jan & Robert, 1995)은 시스템내의 큰 버퍼에서의 CLR 값을 판단하는데 응용된다. 가상의 작은 버퍼와 관련하여 통계적으로 의미 있는 CLR 정보를 얻기 위해서 시스템내의 버퍼에 사용된 입력 셀에 비교하여 많은 입력 셀들이 필요하나, 모니터링 기간은 상당히 줄어든다. 예로, 채널에서 10^{-9} cells/s인 CLR 값을 요구한다고 가정하면, 시스템내의 버퍼에서 요구하는 CLR을 모니터링하기 위해서는 시스템 내의 버퍼 크기에서 적어도 10^{10} sec정도 동안의 입력 셀들을 관찰해야만 통계적으로 의미 있는 모니터링이라고 할 수 있을 될 것이다. 그러나, 선형 감퇴 알고리즘을 이용하면 가상의 작은 버퍼 크기에서 구하여지는 CLR 값으로부터 구할 수 있으며, 이러한 내용들은 매우 낮은 데이터 링크 레벨의 비트 에러 가능성을 측정하기 위해서도 사용된다.

식 (5)은 일반적인 Markovian 소스를 지닌 시스템에서의 버퍼 크기와 CLR을 위한 모델로 제안되었으며, B는 버퍼 크기이고, α 와 δ 는 양의 상수이다. (Hongbo & Victor, 1996)

$$\log(CLR) \approx -\alpha - \delta B \quad (5)$$

네트워크 상에서 실시간으로 추적한 Ethernet 트래픽과 VBR 비디오 데이터 연구에 의해 버퍼 크기와 CLR의 관계가 기존의 Markovian 형태의 트래픽 모델과는 달리 식(6)으로 기술되었다. (Jan & Robert, 1995)

$$\log(CLR) \approx \alpha B^{-\beta} \quad (6)$$

여기서, B는 버퍼 크기, α 와 β 는 양의 상수이다.

그림 5와 같이 생성한 트래픽들에서도 버퍼 크기와 CLR의 관계가 식 (6)과 같이 선형적으로 감소하는지를 확인하기 위해 부하 변동에 따라 실험하였다. 버퍼에서의 출력 셀율은 부하가 0.9인 경우와 0.8인 경우로 하였으며, 부하식은 식 (7)과 같이 정의하였다. 여기서, L은 부하, T는 단위 시간, I는 입력 트래픽이며, O는 출력 트래픽이다.



$$L = \frac{\frac{1}{T} \int_0^T I dt}{\frac{1}{T} \int_0^T O dt} \quad (7)$$

다음은 실험 결과로 그림 6은 오디오, 그림 7은 비디오 트래픽의 버퍼 크기와 CLR에 대한 관계를 보여주고 있으며, 그 관계는 식 (6)과 같이 표현되었다. 오디오 트래픽인 경우 부하가 0.9에서는 $\alpha = 27.8861$, $\beta = 0.0262$, 부하가 0.8에 대해서는 $\alpha = 273.7177$, $\beta = 0.0753$ 로 나타났으며, 비디오 트래픽에서 부하가 0.9인 경우 $\alpha = 14.3890$, $\beta = 0.0567$, 부하가 0.8인 경우 $\alpha = 18.3628$, $\beta = 0.0759$ 로 나타났다. 예로, 비디오 트래픽에서 요구되는 CLR 값이 10^{-9} cells/s이라면 시스템 내의 큰 버퍼 크기에서 실험 시간을 10^{10} sec 이상으로 모니터링하지 않아도 이 관계식에 의해 버퍼 크기를 구할 수 있으며, 부하가 0.8인 경우에 구하여진 α 와 β 를 적용하면 필요로 하는 버퍼 크기는 3497cells이다.

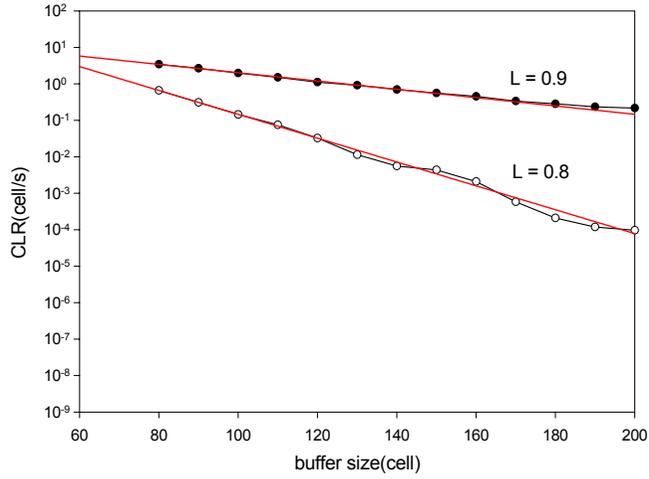


Fig. 6. The relationship between CLR and buffer size in the audio traffic.

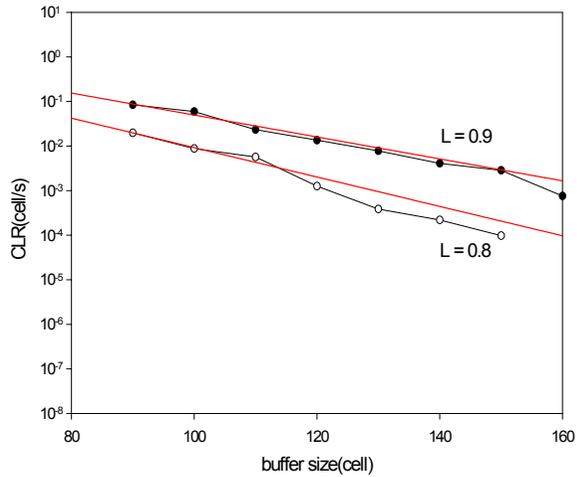


Fig. 7. The relationship between CLR and buffer size in the video traffic.

3. 다중화 출력 셀율과 CLR 관계

N개의 입력 버퍼에 대하여 통계적 다중화를 하여 N개의 입력 버퍼를 서로 합하여 하나의 소스로부터 출력되는 경우 대역폭은 한 개의 입력 버퍼에 대한 대역폭을 N배 한 것보다 일반적으로 적으며 또한, CLR 역시 단일 입력 버퍼에서 일정 셀율로 출력하는 것보다 일정 셀율에 N배를 하여 하나의 소스로부터 출력하는 경우가 더 적은 값을 가진다. 그림 8, 그림 9는 트래픽별로 N = 10개의 채널을 하나의 소스로 다중화한 것으로, OCR 변화에 대한 CLR의 관계를 최소자승 추정 방식(Marquardt, 1963)을 이용하여 식 (8)을 얻었다.

$$CLR \approx \frac{a}{1 + e^{-\frac{OCR-c}{b}}} \quad (8)$$

여기서, a, b, c는 상수이고, OCR은 출력 셀율이다.

10개의 오디오 트래픽 채널들에 대해 다중화 하는 경우 출력 셀율인 OCR 변화에 대한 CLR의 관계를 그림 8에서 보여주고 있으며, 식 (8)에서의 a = 1.0998, b = -0.8253, c = 362.3098로 나타났다. 또한, 채널 수를 10개로 비디오 트래픽을 다중화한 경우 OCR과 CLR의 관계는 그림 9와 같으며, 관계식에서의 상수 a = 0.2741, b = -2.0589, c = 384.2080 이다. 이 실험에서는 n개의 채널을 하나의 소스로 다중화한 후 요구하는 CLR을 충족시키긴 위해 OCR로 설정함은 링크에 필요로 하는 대역폭과 일치하는 것이다. 예로, 비디오 트래픽들을 다중화하는 경우 요구하는 CLR 값이 10^{-5} cells/s라면 식 (8)에 의해 OCR은 406cells/s이다. 이는, 10^{-5} cells/s인 다중화 CLR 값을 만족시키기 위해서는 406cells/s 정도의 링크 대역폭을 필요로 한다는 것이다. 또한, 앞서 논한 바와 같이 통계적 다중화에서의 링크 이용율은 높으나, 셀 손실 발생시 다중화가 이루어진 어느 채널의 셀인지를 확인하기가 어려워 다중화 CLR 값을 채널의 CLR 값에 적용할 수는 없을 것이다.

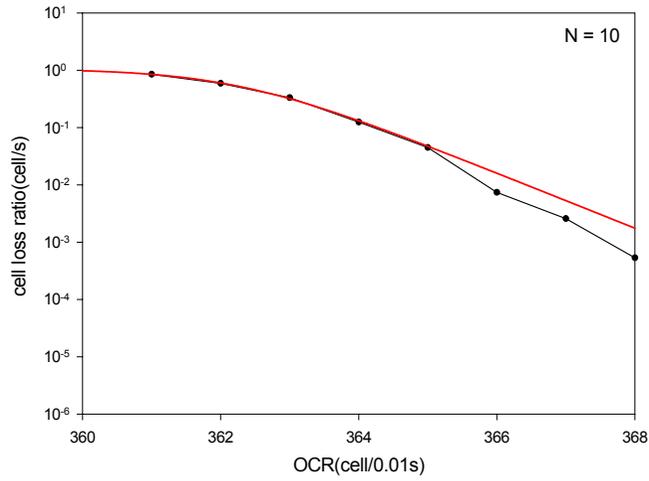


Fig. 8. The relationship between CLR and OCR in the multiplexed audio traffic.

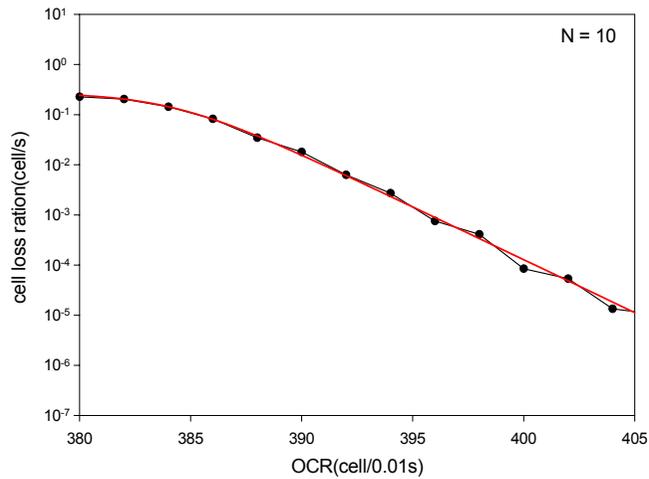


Fig. 9. The relationship between CLR and OCR in the multiplexed video traffic.

4. ATM 트래픽 CLR과 CLR 편차 특성

TBBM 알고리즘을 사용하였을 때 링크 대역폭에 대한 이용율을 높으면서 각 채널들에서 요구하는 QoS를 공정하게 제공하는지를 알아보기 위해 부하 변동에 대하여, 채널 수 변동에 대하여 서비스 트래픽별로 실험하였다. 부하는 0.9와 0.8인 경우, 채널 수는 10개와 5개인 경우로 하였다.

먼저, 일정한 대역폭을 부하 0.9로 설정하였을 때 문턱값 TH와 α 사이의 CLR 특성을 그림 10과 그림 11에서 보여주고 있다. 채널에서의 출력 셀율은 부하 L을 0.9로, 채널 수 ch를 10개로, 문턱값은 버퍼 크기의 0.2, 0.5, 0.8배로 하고, 버퍼 크기 B는 110cells로, α 은 0과 1사이의 값으로 설정하였다. 앞서 설명한 바와 같이 $\alpha = 0$ 인 경우는 버퍼 점유율과 문턱값에 관계없이 일정한 OCR로 처리하는 것으로 채널간 출력 셀율들의 편차가 없으나, $\alpha = 1$ 인 경우는 버퍼 점유율이 문턱값을 초과하는 경우는 출력 셀율 OCR에 (PCR - ACR) 만큼을 더 처리하고, 버퍼 점유율이 문턱값을 초과하지 않는 경우는 출력 셀율 OCR에 (PCR - ACR) 만큼을 덜 처리하여 채널간 출력 편차가 가장 크다.

오디오 트래픽인 경우 문턱값과 α 에 따른 CLR 특성을 그림 10에서 보여주고 있는데, 문턱값이 버퍼 크기의 0.8배인 경우에 가장 좋은 특성을 보이는 반면에 그림 11에서처럼 비디오 트래픽인 경우는 문턱값이 버퍼 크기의 0.5배인 경우에 CLR 특성이 가장 좋은 것으로 나타났다. 오디오 트래픽인 경우 문턱값이 버퍼 크기의 0.8배이고 $\alpha = 0$ 에서 채널간 CLR 편차는 0.1146, $\alpha > 0$ 에서 CLR 평균 편차는 0.0441로 나타났다. 비디오 트래픽인 경우는 문턱값이 버퍼 크기의 0.5배이고 $\alpha = 0$ 인 경우 채널간 CLR 편차는 0.0122이며, $\alpha > 0$ 인 경우 CLR 평균 편차는 0.0004로 나타나 일정한 셀율로 출력하는 경우보다 TBBM 알고리즘을 사용하였을 때가 채널간 공정성 면에서 나은 성향을 보이고 있다.

그림에서의 실선은 문턱값과 α 의 변동에 의한 CLR과 채널간 CLR 편차 특성을 최소 자승 추정 방식에 의해 구한 것이며, 수식과 수식에 따른 상수값을 표1에 정리하였다.

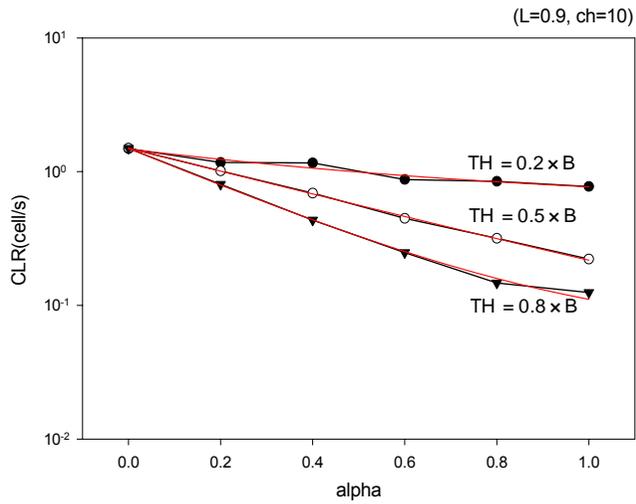


Fig. 10. The relationship between CLR and alpha in the multiplexed audio traffic.

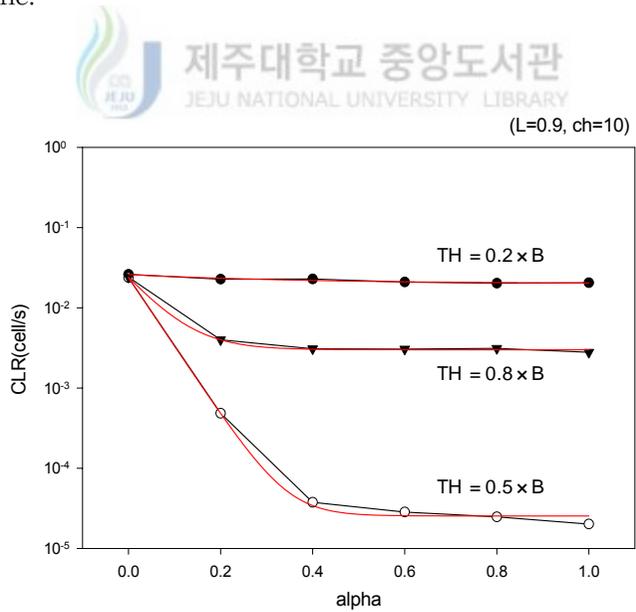


Fig. 11. The relationship between CLR and alpha in the multiplexed video traffic.

부하 변동에 대한 특성을 알아보기 위해 부하를 0.9에서 0.8로 설정하였다. 먼저, 그림 12, 그림 13은 오디오 트래픽에서 문턱값과 α 변동에 따른 CLR과 채널간 CLR 편차 특성을 보여주고 있다. 부하가 0.9일 때와 마찬가지로 부하가 0.8일 때도 문턱값이 버퍼 크기의 0.8배일 때 CLR과 채널간 CLR 편차 특성이 문턱값이 버퍼 크기의 0.2, 0.5배일 때 보다 좋은 특성을 보이고 있다. 문턱값이 버퍼 크기의 0.8배이고, $\alpha = 0$ 인 경우에 채널간 CLR 편차는 0.0215이며, $\alpha > 0$ 인 경우에는 채널간 CLR 평균 편차는 0.0006으로 TBBM 알고리즘을 적용하였을 때가 부하 변동에 무관하게 채널간 공정성 면에서 좋은 경향을 보여주고 있다. 다음으로는 비디오 트래픽인 경우의 문턱값과 α 변동에 따른 CLR과 채널간 CLR 편차 특성을 그림 14, 그림15에서 보여주고 있다. 비디오 트래픽에서도 부하가 0.8인 경우 부하가 0.9일 때와 마찬가지로 CLR과 채널간 CLR 편차 특성에서 문턱값이 버퍼 크기의 0.5배에서 가장 좋은 특성을 보이고 있다. 문턱값이 버퍼 크기의 0.5배이고, $\alpha = 0$ 인 경우에는 채널간 CLR 편차는 0.0023이며, $\alpha > 0$ 인 경우에는 채널간 CLR 평균 편차는 0.0001로 나타나 오디오 트래픽에서와 마찬가지로 채널들에게 QoS를 공정하게 제공하고 있음을 보여주고 있다. 여기서, L은 부하, ch는 채널 수, B는 버퍼 크기, TH는 문턱값이다.

다중화 출력 셀율과 CLR 특성에서 설명된 바와 같이 N개의 채널을 하나의 소스로 다중화 하는 경우의 CLR 특성은 채널마다 개별적으로 출력 셀율로 처리하는 경우보다 훨씬 낮은 값을 보이지만, 다중화 과정에서 손실되는 셀에 대하여 어느 채널의 셀 인지를 확인하기가 어려우며 다중화 CLR 값을 채널이 제공하는 CLR 값이라고 하기에는 곤란할 것이다. 여기서는 채널마다 일정한 출력 셀율로 처리하는 α 가 0인 경우를 다중화 CLR 값으로 고려하였다. α 가 0인 경우의 CLR 값이나 채널간 CLR 편차 특성이 α 가 0이상인 경우 즉, TBBM 알고리즘을 적용한 경우에서의 CLR 값이나 채널간 CLR 편차 특성에 비하여 채널들에게 제공되는 QoS에 대한 공정성이 낮은 것으로 나타났다.

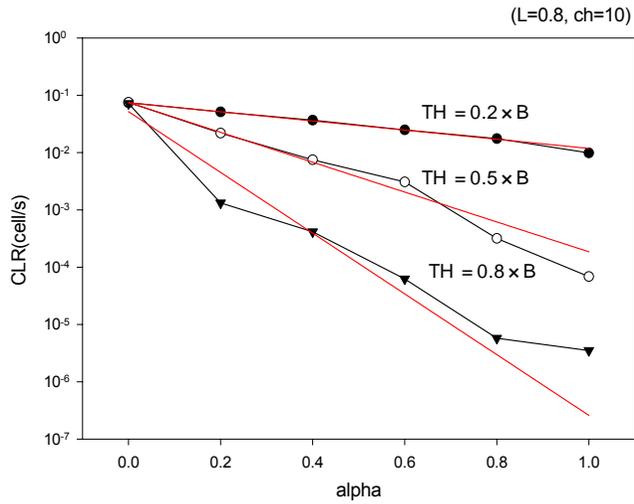


Fig. 12. The relationship between CLR and alpha in the multiplexed audio traffic.

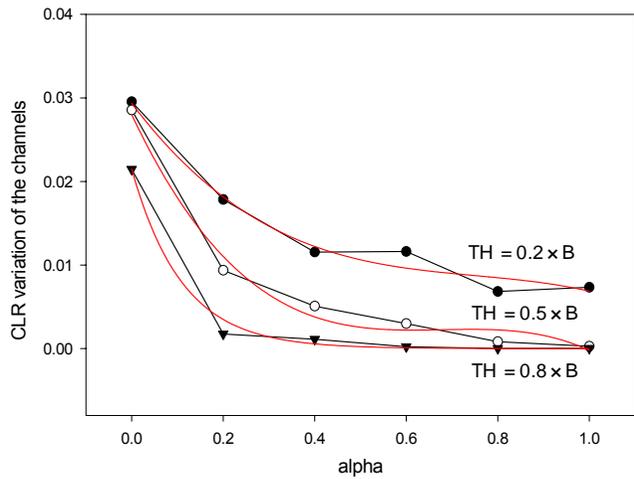


Fig. 13. The relationship between CLR variation of the channels and alpha in the multiplexed audio traffic.

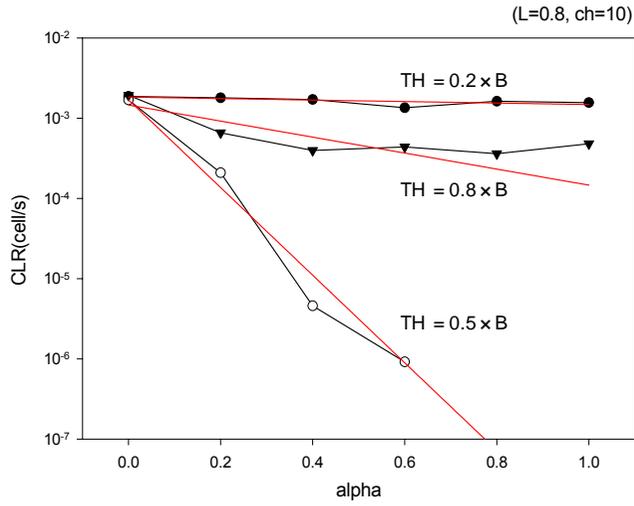


Fig. 14. The relationship between CLR and alpha in the multiplexed video traffic.

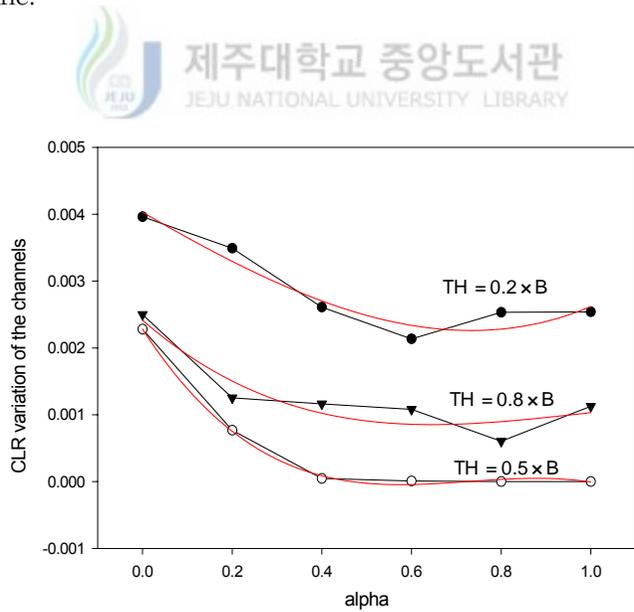


Fig. 15. The relationship between CLR variation of the channels and alpha in the multiplexed video traffic.

부하 변동에 관계없이 서비스 트래픽별로 문턱값과 α 변동에 따라 동일한 CLR과 채널간 CLR 편차 특성을 보였다. 이번에는 채널 수 변동과 관련하여 CLR과 채널간 CLR 편차 특성을 알아보기 위해 채널 수를 10개에서 5개로 설정하였다. 오디오 트래픽인 경우인 그림16, 그림 17에서와 같이 채널 수가 변경되더라도 동일한 CLR과 채널간 CLR 편차 특성을 채널 수가 10개인 경우와 마찬가지로 보여주고 있다. 여기서, L은 부하, ch는 채널 수, B는 버퍼 크기, TH는 문턱값이다. 오디오 트래픽인 경우 문턱값이 버퍼 크기가 0.8배일 때가 가장 좋은 특성임을 그림 18와 그림 19에서 보여주고 있다. 오디오 트래픽인 경우에 문턱값이 버퍼 크기의 0.8배이고, $\alpha = 0$ 인 경우에서의 채널간 CLR 편차는 0.0181이며, $\alpha > 0$ 인 경우의 채널간 CLR 평균 편차는 0.0006으로 보여지고 있다. 비디오 트래픽에서도 채널 수 변경에 대하여 CLR과 채널간 CLR 편차 특성이 문턱값에 대하여 동일하게 나타남을 그림18, 그림 19에서 보여주고 있다. 채널 수가 10개인 경우와 마찬가지로 문턱값이 버퍼 크기의 0.5배일 때가 좋은 결과를 보여주고 있다. 문턱값이 버퍼 크기의 0.5배이고, $\alpha = 0$ 인 경우에서의 채널간 CLR 편차는 0.0014이며, $\alpha > 0$ 인 경우에는 채널간 CLR 평균 편차는 0.00009로 나타났다.

서비스 트래픽별로 부하에 대하여, 채널 수에 대하여, 문턱값과 α 변동에 대하여 동일한 CLR과 채널간 CLR 편차 특성을 보여주고 있음을 알 수 있었다. 오디오 트래픽에서는 문턱값이 버퍼 크기의 0.8배일 때, 비디오 트래픽에서는 문턱값이 버퍼 크기의 0.5배일 때가 가장 좋은 CLR과 채널간 CLR 편차 특성을 보이며, 이는 채널들에게 QoS를 공정하게 제공해주고 있음을 알 수 있다.

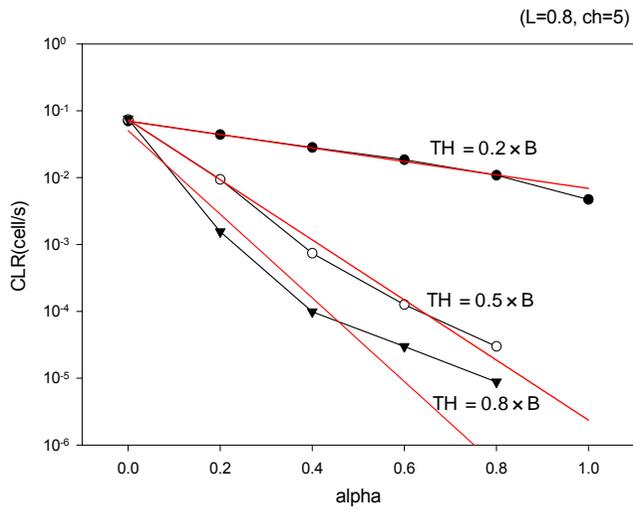


Fig. 16. The relationship between CLR and alpha in the multiplexed audio traffic.

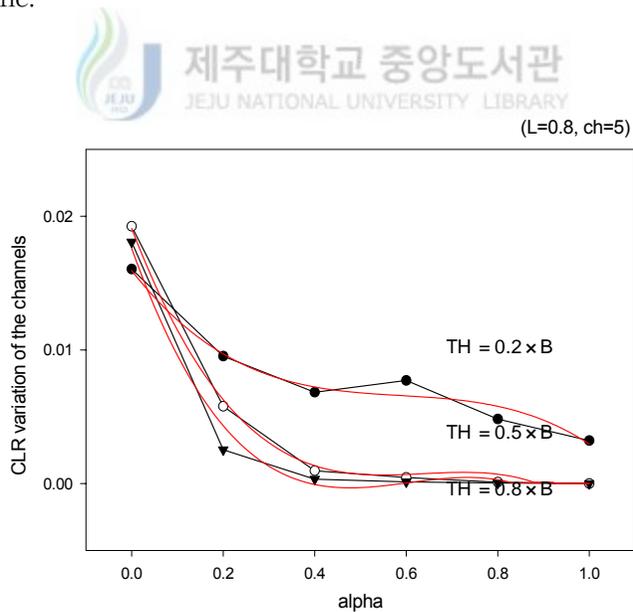


Fig. 17. The relationship between CLR variation of the channels and alpha in the multiplexed audio traffic.

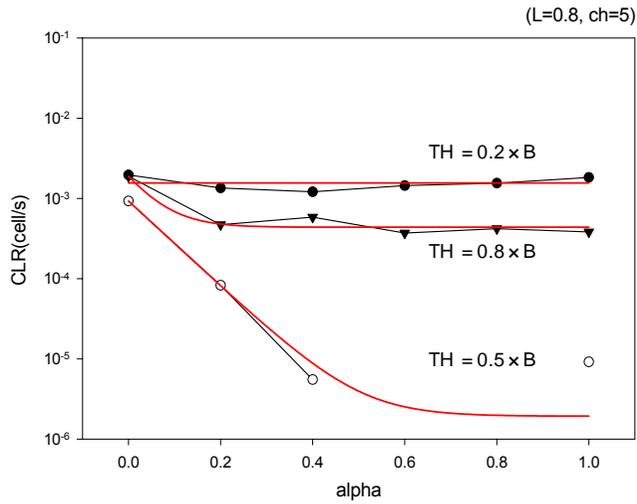


Fig. 18. The relationship between CLR and alpha in the multiplexed video traffic.

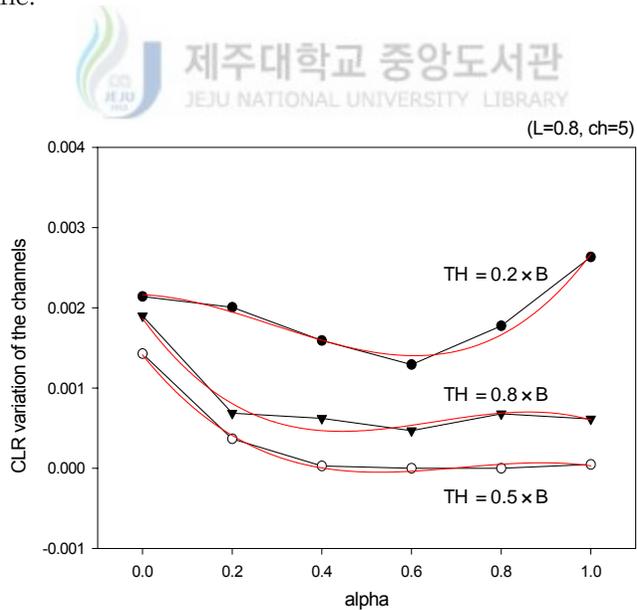


Fig. 19. The relationship between CLR variation of the channels and alpha in the multiplexed video traffic.

Table 1. Function and constant value from least square measurement.

- The relation between CLR and α in audio traffic : $CLR \approx a e^{-b \alpha}$
- The relation between CLR and α in video traffic : $CLR \approx y_0 + a e^{-b \alpha}$
- The relation between CLR variation of the channels and α in the audio, video traffic : $CLR \approx y_0 + a y + b y^2 + c y^3$

Channel	Item	Load	Traffic	Threshold	constant value			
					y_0	a	b	c
10	CLR	0.9	audio	0.2×B	-	-1.43850	0.67960	-
				0.5×B	-	-1.49670	1.95510	-
				0.8×B	-	-1.48490	2.98830	-
			video	0.2×B	0.02	0.00594	2.84850	-
		0.5×B		0.00299	0.0211	15.27600	-	
		0.8×B		0.00002	0.0238	19.74690	-	
		0.8	audio	0.2×B	-	-0.0742	1.83710	-
				0.5×B	-	-0.0748	6.00040	-
	0.8×B			-	-0.0517	12.20330	-	
	video		0.2×B	0.00149	0.00040	2.61800	-	
		0.5×B	0.00410	0.00151	9.38950	-		
		0.8×B	0.00000	-0.00145	2.30000	-		
	CLR var.	0.8	audio	0.2×B	0.02800	-0.11600	0.17250	-0.08470
				0.5×B	0.02070	-0.1191	0.2038	-0.10600
				0.8×B	0.02150	-0.05300	0.03550	-0.10019
			video	0.2×B	-0.01030	0.01480	-0.00679	-0.00679
0.5×B				0.00403	-0.00388	0.00073	0.00174	
0.8×B				0.00241	-0.00579	0.00672	0.00231	
5	CLR	0.8	audio	0.2×B	-	-0.07360	10.34850	-
				0.5×B	-	-0.07030	2.32120	-
				0.8×B	-	-0.05054	14.40300	-
			video	0.2×B	0.00146	0.00009	0.000000	-
				0.5×B	0.00000	0.00092	12.24470	-
				0.8×B	0.00043	0.00144	17.62240	-
	CLR var.	0.8	audio	0.2×B	0.01590	-0.04370	0.07100	-0.04020
				0.5×B	0.01910	-0.08980	0.14350	-0.07550
				0.8×B	0.01760	-0.09580	0.16550	-0.09100
			video	0.2×B	0.00216	-0.00025	-0.005356	0.00611
				0.5×B	0.00141	-0.00691	0.01040	-0.00490
				0.8×B	0.00186	-0.00764	0.01300	-0.00665

• B is buffer size, CLR var. is CLR variation of the input channels.

5. 성능평가

성능평가는 QoS 공정성과 링크 이용률 측면에서 살펴보았다. N개의 채널을 다중화 하는 과정에서 발생할 수 있는 셀 손실에 대하여 어떤 채널에서 다중화된 셀 인지를 확인하기가 어려우므로, $\alpha = 0$ 일 때, 채널들마다 일정한 출력 셀율로 하나의 소스에 다중화 되는 경우로 설정하였다. TBBM 알고리즘을 설정한 경우 앞서 살펴본 내용과 같이 오디오 트래픽에서는 문턱값이 버퍼 크기의 0.8배인 경우에, 비디오 트래픽에서는 문턱값이 버퍼 크기의 0.5배인 경우에 가장 좋은 결과가 나타남을 확인하였다.

채널간의 공정성을 확인하기 위해 $\alpha = 0$ 일 때와 $\alpha = 1$ 일 때 채널에서 일어나는 현상을 확인하였다. 오디오 트래픽에서는 CLR과 채널간 CLR 편차 특성이 좋았던 문턱값이 버퍼 크기의 0.8배인 경우를 그림 20에서 보여주고 있다. $\alpha = 0$ 으로 설정되었을 때 채널에서 출력되는 셀율은 입력 트래픽의 변동에 무관하게 출력 셀율이 일정하게 유지되는 것을 보여주고 있다. $\alpha = 1$ 로 설정되었을 때 채널에서 출력되는 셀율은 버퍼 점유율이 문턱값에 따라, 입력 트래픽의 변동에 따라, 다중화 채널들간의 버퍼 상황에 따라 동적으로 변동됨을 보여주고 있다. 즉, $\alpha = 1$ 인 경우에 출력 셀율이 동적으로 설정됨에 따라 $\alpha = 0$ 인 경우에 비하여 채널에서 적은 CLR 값과 CLR 편차가 보장되는 것이며, 이는 앞서 논한 바와 같다. 비디오 트래픽에 대해서는 그림 21에서 보여주고 있다. 오디오 트래픽인 경우에 부하가 0.8, 문턱값이 버퍼 크기가 0.8배, 채널 수가 10개인 설정하였을 때, $\alpha = 0$ 인 경우 CLR = 0.0717, 채널간 CLR 편차 = 0.0215, $\alpha = 1$ 인 경우의 CLR = 3.5×10^{-6} , 채널간 CLR 편차 = 0.5×10^{-6} 으로 나타났다. 비디오 트래픽인 경우에는 부하가 0.8, 문턱값이 버퍼 크기의 0.5배, 채널 수가 10개인 설정하였을 때, $\alpha = 0$ 인 경우의 CLR = 0.0017, 채널간 CLR 편차 = 0.0023, $\alpha = 1$ 인 경우의 CLR = 0, 채널간 CLR 편차 = 0으로 서비스 트래픽별로 채널간 공정성이 개선되고 있음을 보여주고 있다.

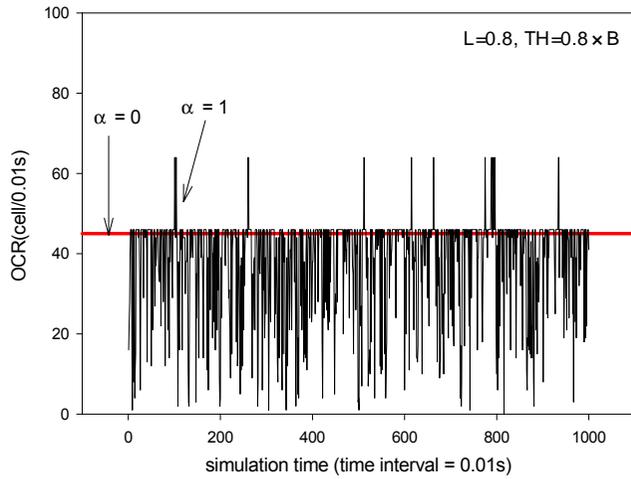


Fig. 20. OCR of a channel in audio traffic.

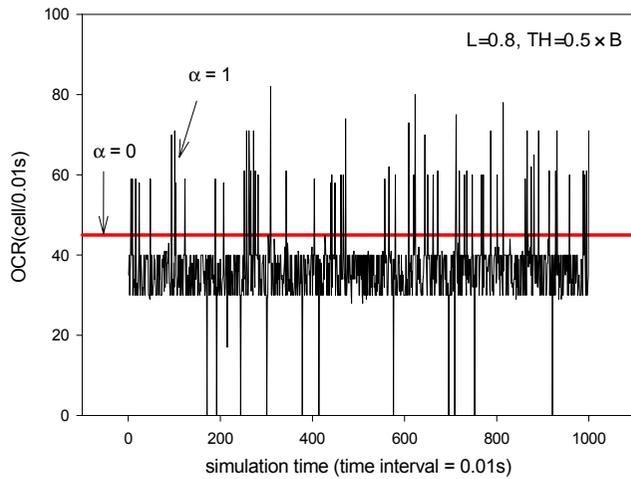


Fig. 21. OCR of a channel in video traffic.

다음으로 링크 이용률 측면에서 성능 평가하기 위해 식 (2)을 이용하여 이론적으로 계산된 증가용량, 식 (8)을 사용하여 구한 다중화 경우에 필요한 대역폭, TBBM 알고리즘을 적용했을 때의 요구 대역폭을 비교하였으며, 링크 대역폭에 대한 사용률은 서비스 트래픽별로 채널 수만큼의 PCR을 100%로 설정하였다. 요구 CLR = 10^{-5} , 출력 셀율은 부하가 0.8, 버퍼 크기는 110cells로 하였다. 위 설정을 만족시키기 위한 오디오 트래픽은 이론적으로 구한 증가용량의 사용률은 91%, 다중화에서 필요로 하는 대역폭 사용률은 69%, TBBM 알고리즘을 사용했을 때 필요한 대역폭 사용률은 78%로 그림 22에서 보여주고 있다. 비디오 트래픽인 경우에는 이론적으로 구한 증가용량의 사용률은 91%, 다중화에서 필요로 하는 대역폭 사용률은 44%, TBBM 알고리즘을 사용했을 때 필요한 대역폭 사용률은 53%로 그림 23에 나타내었다. 여기서 PCR은 채널 수만큼의 최고 셀율로 링크 대역 사용률을 설정한 경우, EQC는 이론적인 계산된 증가용량, TBBM은 제안된 알고리즘에 대한 대역폭 사용률, MUX은 다중화에서 필요로 하는 대역폭 사용률, L은 부하, B는 버퍼크기이다.

이론적으로 구하여 지는 증가용량은 오버에스티메이션되어 있어 필요 이상의 대역폭을 요구한다. TBBM 알고리즘에서는 사용 대역폭에 대하여 이론적으로 구하여지는 증가용량보다는 링크 사용률이 훨씬 낮다. 오디오 트래픽보다 변동이 심한 비디오 트래픽에서 더 많이 이용률이 좋아졌음을 알 수 있다. 다중화에서 요구되는 링크 사용률은 TBBM 알고리즘을 적용했을 때보다 좋으나, 다중화 CLR을 채널들의 CLR과 같다고는 할 수 없다는 단점이 있다. TBBM 알고리즘이 적용된 a에서는 다중화에 대한 채널간 CLR 편차보다 훨씬 적어 채널간 공정성이 좋아 졌음을 앞서 평가하였다.

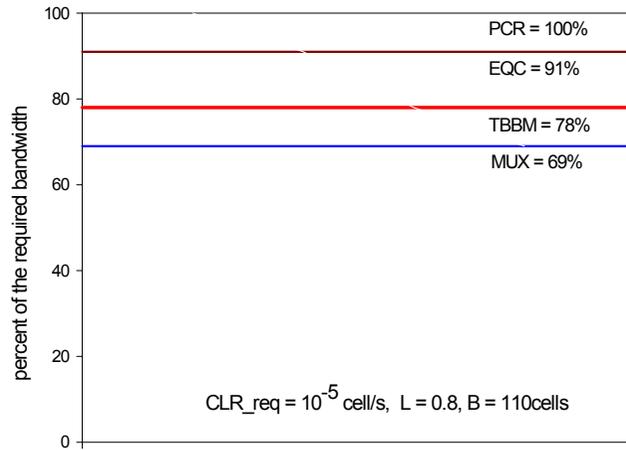


Fig. 22. Percent of the required bandwidth in audio traffic.

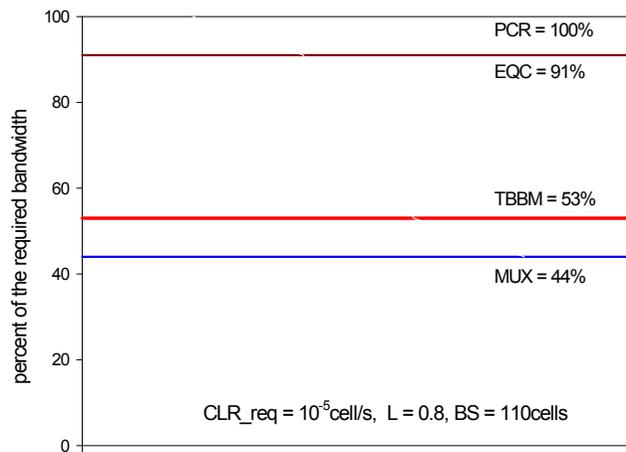


Fig. 23. Percent of the required bandwidth in audio traffic.

IV. 결 론

본 논문에서 제안한 문턱값에 기반을 둔 버퍼 관리 알고리즘인 TBBM 알고리즘은 요구하는 QoS에 대하여 최대한 공정하게 보장함과 동시에 변동이 많은 트래픽에 대하여 효과적인 대역폭 배정으로 링크 이용율을 높일 수 있는 방법을 제시하였다. TBBM 알고리즘은 입력 채널마다 문턱값을 둔 버퍼를 할당하고, 버퍼 점유율에 따라 동일 서비스내의 다른 채널과 비교하여 출력 셀율을 상대적으로 조정함으로써 각 채널마다 출력 셀율이 동적으로 처리되어 요구되는 QoS를 최대한 보장하고, 다중화 출력 셀율이 일정함으로써 대역폭 사용율을 줄일 수 있다. N개의 입력을 하나의 소스로 다중화하여 출력 셀율을 정하는 경우가 N개의 입력 버퍼들에 TBBM 알고리즘을 적용하여 출력 셀율을 조정하는 경우보다 링크 이용율은 낮으나, 후자인 경우가 입력 채널들간의 QoS를 공정하게 보장할 수 있으며, α 에 따라 입력 채널들간의 낮은 CLR 편차를 보였다.

실험 결과 이용율 측면에서 링크 대역폭 사용율에 대해 서비스 트래픽별로 채널 수만큼의 PCR을 100%로 설정한 경우 오디오 트래픽은 이론적인 증가용량 사용율이 91%, TBBM 알고리즘에서 대역폭 사용율이 78%, 다중화에서는 69%의 대역폭 사용율을 보였다. 비디오 트래픽은 이론적인 증가용량이 91%, TBBM 알고리즘인 경우 53%, 다중화시 44%의 대역폭 사용율을 보였다. 서비스 트래픽별로 이론적인 증가용량보다 높은 링크 이용율을 보였다. 공정성 면에서는 오디오 트래픽은 문턱값이 버퍼 크기의 0.8배, 비디오 트래픽은 문턱값이 버퍼 크기의 0.5배가 될 때 CLR 값과 채널간 CLR 편차들이 최소가 됨을 알 수 있었다. 따라서, 링크의 이용율을 높이고 채널간 CLR 편차를 줄여 QoS를 공정하게 제공하기 위해서는 출력 셀율을 정적으로 설정하는 것보다 문턱값을 지닌 버퍼를 이용하여 문턱값과 α 을 중심으로 채널마다 출력 셀율을 동적으로 처리함으로써 가능하리라 사료된다.

참 고 문 헌

ATM Forum technical committee, 1999, Traffic management specification version 4.1, AF-TM-0121.000, ATM Forum.

Hongbo Zhu, Victor S. Frost, 1996, "In-Service Monitoring for Cell Loss Quality of Service Violations in ATM Networks", IEEE/ACM transaction on networking, vol4, no2, pp240-248.

Jan Beran, Robert Sherman, Murad S. Taqqu, 1995, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", IEEE transactions on communications, vol43, no.2/3/4, pp1566-1579.

Lily Cheng, Herman, D.Hughes, 2000, "An ATM Traffic Model Based on Empirical Traffic Measurements", Int. Jol. of Modeling & Simulation, Vol. 20, No. 3, pp. 200-212.

Marquardt, D.W., 1963, "An Algorithm for Least Squares Estimation of Parameters", Journal of the Society of Industrial and Applied Mathematics, 11, pp.431-441.

Qiang Ren, G. Ramamurthy, 2000, "A Real-Time Dynamic Connection Admission Controller Based on Traffic Modeling, Measurement, and Fuzzy Logic Control", pp.184-196.

Othmar Kyas, 1997, ATM networks second edition, 흥릉과학출판사.

Raif O. Onvural, 1995, Asynchronous transfer mode networks : Performance Issues second edition, Artech House.

Ricardo Citro, Seong-Soon Joo, Sumit Ghosh, 2001, "An Adaptive, Dynamic Buffer Management(ADBM) Approach for Input Buffers in ATM Networks", IEEE, pp2860-2864.

고성택, 김경연, 김경식, 최영복, 1998, "신경회로망을 이용한 ATM 신호의 동적 대역폭 할당", 정보통신분야 우수학교 지원사업 연차보고서 1차

고성택, 김경연, 이용학, 안기중, 1999, "ATM망에서 트래픽 제어에 관한 연구", 정보통신우수시범학교지원사업, 우수98-89.

권택근, <http://www.ce.cnu.ac.kr/~tgkwon/>



김관웅, 김병곤, 전병실, 2002, "ATM-GFR에서 최소 전송율 보장 및 공평성 향상을 위한 Leaky Bucket 기반의 버퍼관리 알고리즘", 전자공학회논문지 제39권 제12호, pp8-16.

김지관, 1996, 표준 ATM, 교보문고.

유영일, 강병탁, 김관웅, 조해성, 정경택, 전병실, "셀 지연과 버퍼 이용률을 개선한 WRR 셀 스케줄링 알고리즘", 한국통신 통신망연구소.

김상철, 고성택, 2000, "ATM 망에서 효율적인 문턱 값 기반 호 수락 제어", 한국신호처리·시스템학회 학회, Vol. 1, pp. 33-36.

최태상, 정윤희, <http://kidbs.itfind.or.kr:8888/WZIN/jugidong/918/91801.html>

부 록

1. 변수 설정 다이얼로그

```
////////////////////////////////////  
// 설정된 변수를 받아들이는 함수 (적용단추)  
void CSetParameterDlg::OnServiceParameterApply()  
{  
    //서비스별 CDF 파일로부터 누적확률별 셀 수를 InDataFromPDF[][]에 넣음  
    if((m_File_CDF = fopen("PDF\\"+strDataFileName,"r+")) == NULL)  
    {  
        MessageBox("The file for PDF data is not in this directory.");  
    }  
    else  
    {  
        fscanf(m_File_CDF, "%d", &pDynamicCACDlg->m_sElement.NoOfDataFromPDF);  
        for(i = 0; i < pDynamicCACDlg->m_sElement.NoOfDataFromPDF; i++)  
        {  
            fscanf(m_File_CDF, "%f, %f",  
                &pDynamicCACDlg->m_sElement.InDataFromPDF[0][i],  
                &pDynamicCACDlg->m_sElement.InDataFromPDF[1][i]);  
        }  
    }  
    fclose(m_File_CDF);  
}
```

2. 시뮬레이션 다이얼로그

```
////////////////////////////////////  
// 문턱값과  $\alpha$ 변경에 따른 시뮬레이션 시작  
void CDynamicCACDlg::StartSimulation(int Threshold, double Alpha)  
{  
    m_Index = 0;  
    m_Threshold = Threshold;  
    m_Alpha = alpha;  
    // 문턱값과  $\alpha$ 값 변경시 변수 초기화  
    InitialMemberVariable();  
    SetTimer(0, m_TimeDiv, NULL);  
}
```

```

#include <stdlib.h>
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CDF를 이용하여 랜덤하게 셀 생성
void CDynamicCACDlg::CellGenerator()
{
    int i, j, n_index;
    double rnum, rn, temp;
    for(j = 0; j < m_NoOfChannel; j++)
    {
        rn = (float)rand();
        rnum = rn/RAND_MAX;
        for(i = 0; i < m_sElement.NoOfDataFromPDF; i++)
        {
            if(rnum < m_sElement.InDataFromPDF[1][i])
            {
                n_index = i-1;
                break;
            }
            else if(rnum < 0.000000001) //random number가 너무 적으면 처음 데이터로
            {
                n_index = 0;
                break;
            }
            else if(rnum > 0.999999999) //random number가 너무 크면 마지막 데이터로
            {
                n_index = m_sElement.NoOfDataFromPDF-1;
                break;
            }
        }
        temp = m_sElement.InDataFromPDF[1][n_index+1] -
            m_sElement.InDataFromPDF[1][n_index];
        if(temp < 1e-30)
        {
            temp = 1e-30;
        }
        m_sElement.Element[j].GeneratedCell =
            (UINT)(m_sElement.InDataFromPDF[0][n_index] +
                (m_sElement.InDataFromPDF[0][n_index+1]
                - m_sElement.InDataFromPDF[0][n_index]) * (rnum -
                m_sElement.InDataFromPDF[1][n_index])/temp);
    }
    CalculatorQueue();
}

```

```

////////////////////////////////////
// 문턱값에 따른 출력 셀을 계산
void CDynamicCACDlg::CalculatorQueue()
{
    int Buffer, DrainRate, overTH = 0, underTH = 0;
    for(int Channel = 0; Channel < m_NoOfChannel; Channel++)
    {
        Buffer = m_sElement.Element[Channel].GeneratedCell +
                m_sElement.Element[Channel].UsedBufferSize;
        if(Buffer >= m_Threshold)
        {
            overTH += 1;
        }
        underTH = m_NoOfChannel - overTH;
    }
    if(overTH == m_NoOfChannel)
    {
        for(Channel = 0; Channel < m_NoOfChannel; Channel++)
        {
            m_sElement.Element[Channel].LostCell = 0;
            Buffer = m_sElement.Element[Channel].GeneratedCell +
                    m_sElement.Element[Channel].UsedBufferSize;
            DrainRate = m_Drain;
            if(Buffer >= m_BufferSize)
            {
                m_sElement.Element[Channel].LostCell = Buffer - m_BufferSize;
                m_sElement.Element[Channel].GeneratedCell_Out = DrainRate;
                Buffer -= m_sElement.Element[Channel].LostCell;
                Buffer -= DrainRate;
            }
            else if(Buffer < m_BufferSize)
            {
                if(Buffer <= DrainRate)
                {
                    DrainRate = Buffer;
                }
                m_sElement.Element[Channel].GeneratedCell_Out = DrainRate;
                Buffer -= DrainRate;
            }
            RateCalculatorPerChannel(Buffer, Channel, DrainRate);
        }
    }
    else if(underTH == m_NoOfChannel)
    {
        for(Channel = 0; Channel < m_NoOfChannel; Channel++)
        {
            m_sElement.Element[Channel].LostCell = 0;

```



```

    }
    m_sElement.Element[Channel].GeneratedCell_Out = DrainRate;
    Buffer -= DrainRate;
}
else if(DrainRate <= 0)
{
    m_sElement.Element[Channel].GeneratedCell_Out= 0;
}
}
RateCalculatorPerChannel(Buffer, Channel, DrainRate);
}
}
RateCalculatorPerService(DrainRate);
}

////////////////////////////////////
// 각 채널들의 PCR, ACR, CLR 등을 계산
void CDynamicCACDlg::RateCalculatorPerChannel(int Buffer, int Channel, int Drain)
{
    m_sElement.Element[Channel].UsedBufferSize = Buffer;
    m_sElement.Element[Channel].TotalGeneratedCell +=
        m_sElement.Element[Channel].GeneratedCell;
    m_sElement.Element[Channel].TotalGeneratedCell_Out +=
        m_sElement.Element[Channel].GeneratedCell_Out;
    m_sElement.Element[Channel].TotalLostCell += m_sElement.Element[Channel].LostCell;
    if(m_sElement.Element[Channel].TotalGeneratedCell_Out == 0)
    {
        m_sElement.Element[Channel].CLR = 0;
    }
    else
    {
        m_sElement.Element[Channel].CLR =
            ((float)m_sElement.Element[Channel].TotalLostCell /
            (float)m_sElement.Element[Channel].TotalGeneratedCell_Out) / ((float)m_TimeDiv / 1000);
    }
    m_sElement.Element[Channel].ChannelCount ++;
    m_sElement.Element[Channel].ACR =
        m_sElement.Element[Channel].TotalGeneratedCell_Out /
        m_sElement.Element[Channel].ChannelCount;
    if(m_sElement.Element[Channel].PCR <=
        m_sElement.Element[Channel].GeneratedCell_Out)
    {
        m_sElement.Element[Channel].PCR =
            m_sElement.Element[Channel].GeneratedCell_Out;
    }
}
}

```

감사의 글

두 해를 보낸 대학원 생활이 제에게는 남다른 인생의 한 페이지가 되었으며, 고마우신 분들이 계시기에 제가 살아가고 있다는 평범한 진리를 새삼 느끼게 되었습니다.

이 작은 논문이 나오기까지 많은 분들의 도움이 있었습니다. 매 순간 어두운 앞길을 밝게 비추어 주신 고성택 교수님과 어설피기 짝이 없는 논문을 세세하게 검토해 주시고 애정 어린 충고를 잊지 않으신 김경식 교수님, 도양희 교수님, 격려의 시선으로 끝까지 지켜봐 주신 김경연 교수님, 이광만 교수님, 강민제 교수님께 특별한 감사의 마음을 전합니다. 그리고 무엇보다도 교수님들께서 보여주신 진심 어린 모습이 저에게는 큰 감사이면서 기쁨이었습니다.

대학원에 들어온 첫 학기부터 여러모로 도움을 준 동기이며 후배인 강은성, 김성운, 강숙인에게 고마운 마음을 전하며, 많은 생각과 고민을 같이 정리해 준 고경희에게 큰 감사를 전합니다. 논문에 아낌없는 조언과 충고를 해준 오테레사 선배님, 고보승, 김봉석, 오상현, 현미나에게도 감사의 마음을 전합니다. 항상 용기를 북돋아준 현소영, 이수현, 김미정, 교회선, 문선아에게도 고마움을 표합니다. 미숙한 사회 생활과 학문의 길에 많은 도움을 주신 문창은 선생님, 문상호 선생님, 김혜진 선생님, 이봉길 선생님, 이정재 선생님께도 감사드립니다.

끝으로, 이 논문은 묵묵히 지켜봐 주신 부모님과 걱정 어린 시선을 거두지 않았던 두 언니와 두 형부, 오빠가 계셨기에 가능했습니다. 소중한 가족의 일원으로 열심히 살기 위해 더욱 분발하리라는 소박한 다짐으로 고마운 마음을 대신하려 합니다.