

ATM망에서의 효율적인 UPC 알고리즘의 성능 개선

박 성 곤[†]

요 약

ATM망에서 트래픽 제어는 서비스 질을 향상시키기 위한 방법으로 폭주가 일어난 후에 처리하는 반응 제어와 폭주가 발생하기 전에 처리하는 예방 제어의 2가지 방법이 있다. 특히 예방 제어의 방법은 연결 수락 제어, 사용 매개변수 제어, 망 매개변수 제어, 우선 순위 제어가 있다.

본 논문에서는 리키 버킷 기법에 점핑 윈도우 기법을 복합한 이중구조로 설계된 효율적인 사용 매개변수 제어 알고리즘을 제안한다. 제안 알고리즘은 먼저 리키 버킷을 사용하여 최대비트율을 제어하고, 점핑 윈도우를 사용하여 평균비트율의 위반여부를 판단하여 트래픽 제어를 한다. 제안된 알고리즘의 성능평가를 위하여 On/Off 트래픽 소스 모델을 가정하고 시뮬레이션 한 결과 기존 리키 버킷 기법보다 셀 손실률의 저하를 보였고, 버킷 크기도 감소시킬 수 있었다.

The Performance Improvement of an Efficient Usage Parameter Control Algorithm in ATM Networks

Sung Kon Park[†]

ABSTRACT

In the ATM networks, there are two method in traffic control as schemes to improve the quality of service; one is the reactive control after congestion and the other is the preventive control before congestion. The preventive control include the CAC(Connection Admission Control), the UPC(Usage Parameter Control), the NPC(Network Parameter Control) and the PC(Priority Control).

In this paper, we propose an efficient UPC algorithm that has a complex structure using the Jumping window algorithm within the Leaky Bucket algorithm. The proposed algorithm controls peak bit rate by the Leaky Bucket algorithm, then it does the traffic control to evaluate by the Jumping Window whether violates mean bit rate or not. As we assume On/Off traffic source model, our simulation results showed cell loss rate less than the pre-existential Leaky Bucket algorithm method, and it could decrease the demanded Bucket size.

1. 서 론

ATM망에서는 다양한 특성의 트래픽 소스 및 연결 형태에 대하여 요구되는 QoS(Quality of Service)를

보장하기 위해서 폭주상태를 사전에 예방하거나 폭주상태를 신속하게 회복하기 위한 효율적인 트래픽 제어 기술이 요구된다.

ATM망에서의 트래픽 제어 방식은 크게 반응 제어와 예방 제어로 나눌 수 있다. 반응 제어는 폭주 발생을 인식하면 그에 대한 반응 조치를 취하는 방식으로서 ATM망은 전송 속도가 155Mbps 이상의 고속이기

[†] 정 회 원: 원주전문대학 전산정보처리학과
논문접수: 1997년 1월 8일, 심사완료: 1997년 6월 3일

때문에 이 방법은 큰 효율을 거둘 수 없으므로 가급적 망이 폭주 상태에 빠지는 것을 예방할 수 있는 예방 제어 방식을 권고하고 있다.

ITU-T(International Telecommunication Union)의 권고안 I.371은 폭주 예방 기능으로 연결 수락 제어, 사용 매개변수 제어, 자원 관리, 우선 순위제어 및 트래픽 셰이핑 기능을 정의하였으며, 폭주 회복 기능으로 선택적 셀 폐기 및 EFCI를 정의하고 있다.

본 논문에서는 ATM 망의 효율성을 높이기 위하여 트래픽 제어 기술중 대표적인 사용 매개변수 제어(UPC) 알고리즘인 리키 버킷 방법의 단점을 보완할 수 있는 알고리즘에 대해 검토하고자 한다. 리키 버킷의 방법에 의한 사용 매개변수 제어는 다른 기법에 비해 전반적으로 높은 효율성을 인정받고 있지만 셀 손실율의 확보를 위해서 비현실적일 정도의 버킷을 필요로 하며, 평균비트율의 제어가 곤란하다는 단점이 있다[3].

본 논문에서는 사용 매개 변수 방법의 하나인 리키 버킷 알고리즘에 Jumping Window 기법을 적용하여 평균비트율을 감시하면서 셀 손실률과 버킷의 크기를 줄이고 또한 위반 트래픽에 대한 제어 성능을 높일 수 있는 새로운 알고리즘을 제안하며 시뮬레이션을 실시한 후 결과를 분석하고자 한다. 2장에서는 ATM 망에서의 기존 트래픽 제어 기법에 대하여 검토하고, 3장에서는 제안한 사용 매개변수 알고리즘에 대하여 설명하고, 4장에서는 시뮬레이션 결과와 분석을, 5장에서는 결론 및 향후 과제를 기술한다.

2. 사용 매개변수 제어 알고리즘

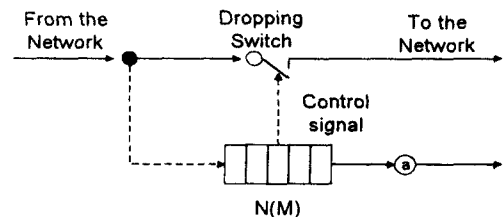
사용 매개변수 제어는 협약한 매개변수의 위반을 찾아내고 그에 따른 적절한 행동을 취하여 이미 설정된 다른 연결의 QoS에 영향을 주지 않도록 망 자원을 보호하는 것이다. 이러한 기능은 모든 트래픽이 다중화되기 전에 각 트래픽 호원에 대하여 수행되어야 하며, 이 기법의 대표적인 알고리즘으로는 LB(Leaky Bucket)[2][4][15], JW(Jumping Window)[5][14], SW(Sliding Window)[5], EWMA(Exponential Weighted Moving Average) [5] 등이 있다.

2.1 리키 버킷

사용 매개 변수 제어 기법중 트래픽의 최대비트율 제어에는 비교적 우수한 성능을 제공하지만 평균비트율 제어에는 성능이 둔감한 것으로 평가되고 있는 리키 버킷의 기본적인 개념은 셀 입력이 있을 때마다 계수기의 값을 하나씩 증가시켜 사용 매개 변수 값을 감시한다. 사용 파라미터를 감시한다. 또한 계수기는 입력셀과는 독립적으로 일정한 리키 율(leak rate)로 계수기 값을 감소시키고 리키 율(leak rate)은 제어 대상 트래픽의 평균비트율, 최대비트율 등을 고려하여 결정한다. 만일 입력 비트율이 계수기의 리키율을 넘어설 때에는 계수기 값은 계속 증가하게 되고 계수기 값이 한계값에 도달하게 되면 그 다음에 입력되는 셀들은 계수기 값이 다시 한계값 이하로 떨어질 때까지 모두 폐기되게 된다. (그림 3)에 리키 버킷 방법을 나타내었는데, 계수기 M은 일정한 리키율 (a)로 감소하게 되고, 버킷의 크기 N과 M이 같아지게 되면 이후에 입력되는 셀들은 폐기하게 된다.

한편, LB 방법에 의해 수행되는 사용 매개 변수 제어가 설립된 VP/VC에 대해 서비스 품질(QoS)을 보장하지 못한다는 연구 결과도 보고되고 있으며, CDV의 영향에 의해 policing accuracy가 떨어지는 문제점을 내포하고 있는 것으로 알려져 있다[6].

이를 보상하기 위한 방향으로 제시된 알고리즘이 VLB(Virtual Leaky Bucket) 알고리즘이다. 이것은 Policing과 Smoothing 기능을 제공하는 것으로 LB 뒷단에 Spacer를 두어 강제적으로 셀간격을 유지시키고 토큰 풀을 Red와 Green 두 종류로 나누었다. Green 토큰은 LB와 같이 최대비트율(PBR) 감시를 위하여 사용되고, Green 토큰이 없는 경우에는 Red 토큰이 있는 경우에 사용 매개변수를 위반한 셀임을 표시하고 망내로 유입시킨다. 이 방식에서는 Red 토큰 풀을



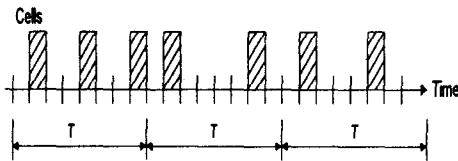
(그림 1) Leaky bucket
(Fig. 1) Leaky bucket

어떻게 생성하는가에 따라 실제 평균비트율을 보장하게 되며, 본 논문의 경우에는 이를 해결하기 위하여 JW 방식을 채용하였다.

2.2 윈도우 기법

윈도우 기법 중에서 간단한 것으로 JW 방법이 있으며, 이것은 일정시간(윈도우)안에 정보원으로 수용할 수 있는 셀의 최대 수를 N으로 하고, 그 수를 넘어서 도착하는 셀은 표시하거나 폐기되는 기법이다. 새로운 윈도우는 전의 윈도우가 끝나면 바로 시작된다. 이때 셀의 개수를 세는 계수가 값은 각 윈도우 시작 초기에 0으로 리셋된다.

JW 기법에서는 윈도우 구간이 정보원의 셀과 동기화되어 있지 않지만, 이러한 동기 문제를 해결한 TJM(Triggered Jumping Window) 기법에서는 윈도우가 비연속적이며 한 윈도우 구간이 끝난 후에 처음으로 도착하는 셀에 의해 새 윈도우가 시작된다.



(그림 2) Jumping window 기법
(Fig. 2) Jumping windows mechanism.

EWMA는 JW과 같이 연속적인 윈도우 구간을 가지나, 각 윈도우에서 수용할 수 있는 셀의 수를 전 윈도우에서 수용한 셀과 각 윈도우에 허용된 평균 셀 수 N의 합수로 표현된다. 이외에도 SW 기법 등이 있다.

3. 제안 사용 매개변수 알고리즘

3.1 기존 사용 매개변수 알고리즘의 문제점

리키 버킷에서 최대 비트율 제어는 토큰 발생율의 조절로 가능하나 평균비트율은 일정한 추정 알고리즘을 필요로 하게 된다. 평균비트율을 측정하기 위해 요구되는 측정시간은 고속의 ATM 망에서는 측정기간 중에 접속이 종료되거나 이미 망이 폭주 상태에 빠지게 되는 등의 여러가지 문제점이 발생하게 된다.

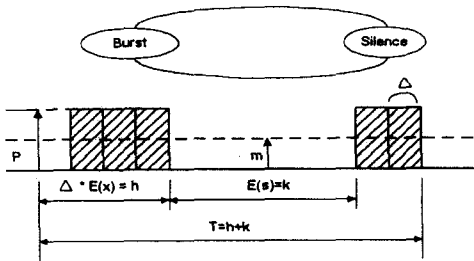
리키 버킷 알고리즘에서 셀 손실율을 줄이기 위해서는 계수기 한계치 N을 올려 주어야 한다. 그러나 이 경우 10^{-5} 의 셀 손실율을 얻기 위해서 10^5 이상의 셀 버퍼나 계수기 한계치가 필요하며 실제 ATM이 요구하는 10^{-9} 이상의 셀손실율을 얻기 위해서는 비현실적인 버퍼 수나 계수기 한계치로 인하여 실제적 구현이 어려워지게 된다. 또한 큰 계수기 한계치로 인한 긴 측정시간은 협약한 파라미터를 위반하는 트래픽에 매우 느린 반응을 보이게 되어 망을 폭주 상태에 빠트릴 수 있게 한다. 리키 버킷은 JW나 MW등과 같이 윈도우를 가지는 알고리즘에 비하여 우수한 성능을 보임에도 불구하고 위에서 언급한 문제로 인하여 평균비트율 제어에는 적합치 못한 것으로 평가되고 있다. 또한, 리키 버킷 알고리즘은 감시의 엄격성 때문에 유용한 정보를 폐기시키게 되는 단점이 있는 것으로 보고되고 있다. N. Yamanaka [13]등의 시뮬레이션 결과에 따르면 리키 버킷 알고리즘에 의해 허용되는 가장 버스티한 트래픽 형태를 사용한 전통적 대역폭 할당 방법은 설립된 VP/VC에 대해 QoS를 보장하지 못하는 것을 보여주고 있다.

K. Shimokoshi[2]가 지적한 바에 의하면 리키 버킷 알고리즘은 CDV에 의하여 셀 도착 과정이 왜곡되면 정확도가 떨어지게 된다. 이는 연속된 셀의 도착간격을 감시하는 리키 버킷 알고리즘의 원리에 기인한다. JW와 같이 윈도우를 사용하는 감시 매커니즘은 윈도우 간격보다 더 긴 fluctuation을 허용하지 못하는 단점을 가지고 있으며 간격제어기를 가지는 VSA의 경우 도착하는 셀의 간격을 기반으로 사용 파라미터 제어를 행하기 때문에 최대 비트율 제어에는 적합하지만 일정시간 동안의 입력 비트율을 계산해야 하는 평균비트율 제어에는 적합하지 않다.

3.2 트래픽 소스 모델

본 논문에서 제안하는 알고리즘의 성능 평가를 위해 사용된 트래픽 소스 모델은 On/Off 모델을 사용하였다. On 상태(버스트 구간)에서는 협약한 대로 최대 비트율로 트래픽을 발생시키고, Off 상태(침묵 구간)에서는 트래픽을 발생시키지 않는 On/Off 모델과 대용량 트래픽 소스의 다중화에 적용되는 First Order Markov Process 및 LAN 트래픽 소스에 적용되는 Poisson Process 모델이 있으나 정지 영상, packetized

voice, telephone source 등 실제 트래픽과 유사한 성질을 갖고 있는 On/Off 모델을 사용하였다. 버스트 구간 및 침묵 구간은 지수분포를 가지는 것으로 가정하였다. 평균비트율이나 버스트성 등의 매개변수는 최대비트율, 평균 버스트 기간, 평균 침묵 기간 등의 매개변수를 가지고 구할 수 있다. (그림 3)에서 Δ 는 셀 간 도착 간격이고 $E[x]$ 는 평균 버스트 기간동안의 평균 셀 수이다.



최대비트율 : p 버스트 구간 : h 평균 침묵 구간 : k
 평균비트율 : $m = h \cdot p / (h + k)$ 버스트성 : $b = (h + k) / h = p / m$

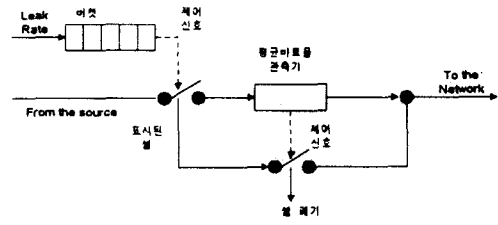
(그림 3) On/Off 소스 모델
 (Fig. 3) On/Off source model

3.3 제안 알고리즘

3.3.1 알고리즘의 개념

본 논문에서 제안한 알고리즘의 제어 모델은 (그림 4)와 같이 리키 버킷 기법과 JW 기법이 복합된 이중 구조로 되어 있다. 리키 버킷을 통과한 셀들은 평균 비트율 관측기를 통해 망으로 전송되고, 리키 버킷에 의해 표시된 셀들은 망으로 전송되는 셀율이 평균비트율을 초과하지 않는 한 망으로 전송된다. 평균비트율 관측기에서는 일정 윈도우 기간동안 지나간 셀 수를 계측하여 평균비트율을 위반하지 않고 있다면 리키 버킷 기법에 의해 표시된 셀들에 대해 accept 신호를 보낸다. 표시된 셀은 평균비트율 관측기에서 accept 신호가 있는 경우에만 다시 망으로 전송되고 그렇지 않으면 폐기하게 된다.

이러한 알고리즘은 버킷 크기의 한계로 인하여 무조건 폐기되는 순송 셀들에 대한 Type-I 에러와 최대 비트율은 위반하였지만 평균비트율을 위반하지 않은 셀들의 폐기로 인하여 발생하는 셀 손실률을 줄일



(그림 4) 제안 알고리즘의 제어 모델
 (Fig. 4) The control model of the proposed algorithm

수 있다.

3.3.2 버킷 크기

만일 버스트 기간에 제약이 없다면 아주 큰 수의 버킷을 사용할 수 있으나 일반적으로 버스트 기간에 제약이 따르게 된다. 리키율(Leak Rate)이 평균비트율과 같다면 필요한 버킷 크기는 다음과 같이 결정할 수 있다[13].

버킷 크기 = $p \cdot h \cdot (1 - 1/b)$ 최대비트율 : p

버스트 구간 : h 평균 침묵 구간 : k

평균비트율 : $m = h \cdot p / (h + k)$ 버스트성 : $b = (h + k) / h = p / m$

3.3.3 윈도우 한계값

평균비트율을 측정하는 평균비트율 관측기는 JW 알고리즘을 사용한다. 즉 일정한 윈도우 기간동안 지나간 셀의 평균수를 계산한다. 1개의 윈도우 기간동안 지나간 셀의 평균 개수가 설정된 윈도우의 셀 허용 한계값보다 크면 평균비트율을 위반한 것으로 간주한다. 하나의 윈도우 기간은 평균 버스트 기간 h 와 평균 침묵 기간 k 의 합이며 본 시뮬레이션에서는 적절한 윈도우 기간동안의 셀 수를 측정하여 평균비트율 위반 여부를 판단하도록 하고, 본 알고리즘에서는 1개의 윈도우 기간 내의 최대 셀 허용값을 평균비트율로 하였다.

<평균 비트율의 위반 여부 판단 알고리즘>

$A = N$ 개의 윈도우가 기억하는 셀의 수 / N

$N =$ 윈도우의 개수

if ($A >$ 윈도우의 한계값)

 평균비트율 위반

else

평균비트율 순용

4. 시뮬레이션 결과와 분석

4.1 시뮬레이션 환경

본 논문에서 제안한 알고리즘의 성능 평가를 위해 사용된 소스 모델의 트래픽 매개변수는 <표 1>과 같다. 리키 버킷, 토큰, 버퍼에 따른 셀 손실률과 동일 셀 손실률하에 따른 버킷 크기를 측정하기 위하여 'C' 언어와 SMPL의 시뮬레이션 서브시스템을 활용하여 SUN 워크스테이션에서 시뮬레이션을 실행하였다.

<표 1> 트래픽 매개변수
<Table 1> Traffic parameter

매개변수	패킷화된 음성	정지화상
최대비트율	64.0 Kbps	2Mbps
평균비트율	22.4 Kbps	1Mbps
평균 버스트 구간	0.352 sec	0.5 sec
평균 침묵 구간	0.650 sec	11.0 sec

ATM망에서는 셀 손실률을 10^{-9} 이하로 요구하고 있다. 그러나 이러한 셀 손실률을 얻기 위해서는 최소한 10^9 개 이상의 셀을 발생시켜야 하므로, 본 논문에서는 시뮬레이션의 제약으로 셀 손실률을 10^{-5} 이하로 하였다.

4.2 성능평가 기준

평균비트율 위반 트래픽에 대한 셀 위반율과 버킷 크기를 주요 평가 변수로 사용한다.

- 셀 위반율: 협약된 최대비트율을 준수하면서 평균 비트율 위반 트래픽에 대한 셀 위반율을 측정한다.
- 버킷 크기: 셀 도착율 1.0을 기준으로 하였을 경우의 셀 손실률을 저하시키기 위하여 추가로 필요한 버킷 크기를 측정한다.

4.3 시뮬레이션 알고리즘

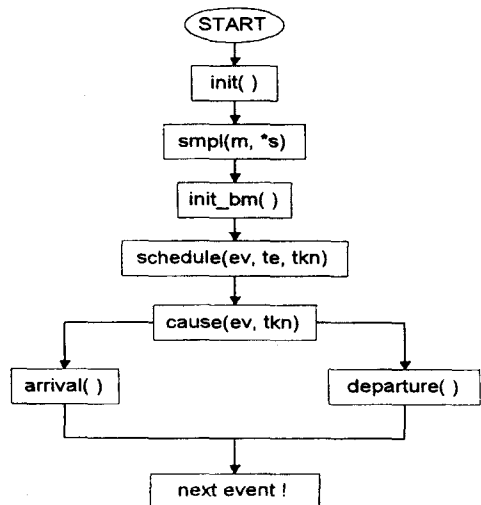
리키 버킷 방식 및 제안 알고리즘의 성능 비교 평가를 위한 제안 알고리즘 흐름도를 (그림 5)에 나타내

었다.

init()는 시뮬레이션을 위한 초기 트래픽 파라미터의 특성을 정의한다. 트래픽 파라미터로는 최대비트율, 평균 침묵구간, 평균 버스트구간이며 계수기 한계값에 따른 제어인자값이 설정되고, 매개변수에 따라 음성 트래픽원의 특성이 선택된다.

stream() 함수는 난수 발생기의 종류를 선택하게 되고, smpl(m, *s)는 시뮬레이션 sub-system을 초기화한다. 시뮬레이션 초기 상태를 안정상태(steady state)로 조정하기 위해서 init_bm() 함수가 사용된다. 하나의 event를 스케줄링하기 위해 schedule (ev, te, tkn) 함수가 사용된다. 이 함수에서 ev는 event 번호를, te는 inter-event time을 tkn은 event에 관한 token을 나타낸다.

te는 음수의 값을 갖지않고 event 발생 시간을 얻기 위해 현 시뮬레이션 시간값에 더해지게 된다. schedule() 함수는 event number (ev), event occurrence time(te), token(tkn)들의 내용으로 event list를 구성하게 된다. Event list는 event occurrence time 순서에 의해 오름차순으로 정렬되어 있으며 동일한 event occurrence time을 가지는 경우 도착순서(first-in, first-out)에 의해 정렬된다.



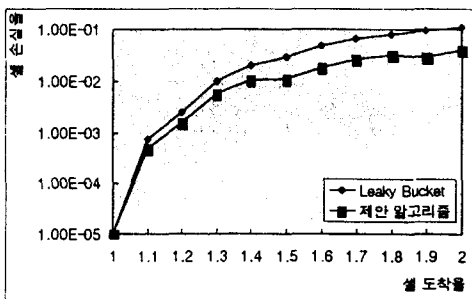
(그림 5) 제안 알고리즘의 흐름도
(Fig. 5) The flowchart of the proposed algorithm

cause(ev, tkn) 함수는 event list의 맨 위에 있는 entry 하나를 제거하고, 시뮬레이션 시간을 제거된 entry의 event occurrence time으로 나아가게 한다. 이후 ev의 값에 따라 arrival()과 departure() 프로시저로 구분되어 진행된다.

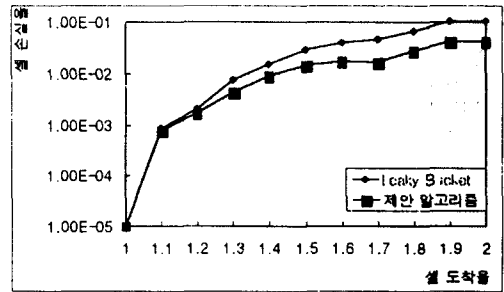
4.4 시뮬레이션 결과 및 분석

4.4.1 페킷화된 음성 셀 손실률

(그림 6)과 (그림 7)은 계수기의 한계값(버퍼 크기)을 300, 500으로 가정하고 시뮬레이션을 실행한 결과, 리키 버킷 알고리즘과 제안 알고리즘의 셀 손실률을 나타내었다. 제안 알고리즘에서 평균비트율을 측정하기 위해서는 일정한 시간 간격(윈도우) 동안 망으로 전송되는 셀 수를 측정하는 기법이 필요하게 된다. 윈도우는 한번의 측정이 완료되면 즉시 다음의 측정에 돌입하게 된다. 따라서 윈도우 간격 설정에 따라 반응시간이 결정되며, 평균비트율 관측기에서 현재의 망이 평균비트율을 위반하지 않고 있다면 최대비트율을 위반하여 표시된 셀을 망내로 전송하게 된다. <표 2>에는 셀 도착율에 대한 기존의 LB 알고리즘과 제안 알고리즘의 셀 손실률 비를 나타내었다. <표 2>에서 나타난 바와 같이 계수기 한계값이 300으로 가정하고 시뮬레이션을 실행한 결과, LB 알고리즘의 셀 도착율이 1.5일 때 셀 손실률은 0.028463이고, 제안 알고리즘의 셀 도착율이 1.5일 때 셀 손실률은 0.011079로 나타났다. 셀 도착율이 증가할수록 제안 알고리즘의 셀 손실률이 기존 LB 알고리즘 보다 감소됨을 알 수 있었다. 셀 손실률비는 계수기 한계값이 작을수록 제안 알고리즘이 더욱 큰 효과를 나타



(그림 6) 계수기 한계값 300일 때 셀 손실률 (음성)
(Fig. 6) Cell loss rate when count limits is 300 (voice).



(그림 7) 계수기 한계값 500일 때 셀 손실률 (음성)
(Fig. 7) Cell loss rate when count limits is 500 (voice).

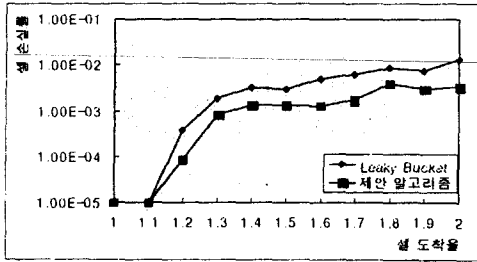
<표 2> 셀 도착에 대한 셀 손실률 비 (음성)
<Table 2> The cell loss ratio of departed cell (voice)

계수기 한계값	셀 도착율	LB 셀 손실률	제안 알고리즘 셀 손실률	셀 손실률 비 (LB = 1.0)
300	1.1	0.000727	0.000486	0.668
	1.3	0.009868	0.005626	0.570
	1.5	0.028463	0.011079	0.389
	1.7	0.067159	0.025684	0.382
	1.9	0.095540	0.028808	0.302
500	1.1	0.000851	0.000761	0.894
	1.3	0.007428	0.004214	0.567
	1.5	0.028029	0.014295	0.510
	1.7	0.046253	0.016121	0.349
	1.9	0.102633	0.042426	0.413

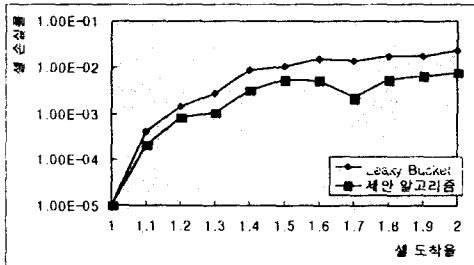
내었으며, 셀 도착율이 높을수록 셀 손실률비가 낮음을 알 수 있다.

4.4.2 정지화상에 대한 셀 손실률

(그림 7)과 (그림 8)에 계수기 한계값이 300일 때와 500으로 가정했을 경우 정지화상에 대한 리키 버킷 알고리즘과 제안 알고리즘의 셀 손실률의 시뮬레이션을 실행한 결과를 나타내었다. 계수기 한계값이 500으로 상향 조정되었을 경우에는 계수기 한계값이 작을 때 보다 초기 셀 도착율에서는 셀 손실률이 증가하나 셀 도착율이 증가함에 따라 셀 손실률이 낮아지고 있음을 보이고 있다.



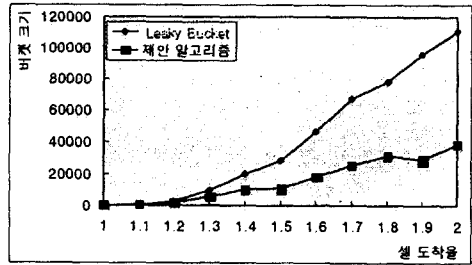
(그림 7) 계수기값 300일 때 셀 손실률 (정지화상).
(Fig. 7) Cell loss rate when a count limits is 300 (still picture).



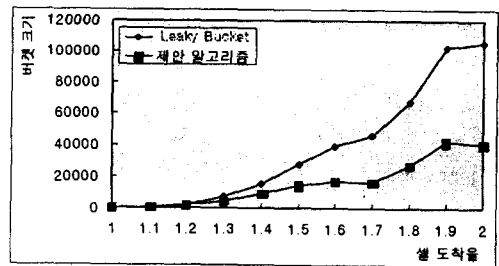
(그림 8) 계수기값 500일 때 셀 손실률 (정지화상).
(Fig. 8) Cell loss rate when a count limits is 500 (still picture).

4.4.3 버킷의 크기 감소(음성)

셀 손실률을 줄이기 위하여 추가로 필요한 버킷 크기를 리키 버킷 방법과 비교하여, 계수기 한계값을 300, 500으로 설정하여 시뮬레이션한 결과를 (그림 9)와 (그림 10)에 나타내었다. 계수기 한계값을 300으로 하였을 경우, 셀 도착율 1.9에서 추가로 요구되는 버킷을 2배 이하로 줄일 수 있었으며, 계수기 한계값이 500인 경우는 기존의 알고리즘이 95,540개의 버킷을 추가로 요구하는 반면에 제안 알고리즘은 28,808개의 버킷만 추가로 필요로 하며, 66,732개의 버킷을 줄였음을 보여주고 있다.



(그림 9) 계수기값 300일 때 버킷 크기(정지화상).
(Fig. 9) Bucket size when a counter limits is 300(still picture).



(그림 10) 계수기값 500일 때 버킷 크기(정지화상).
(Fig. 10) Bucket size when a count limits is 500 (still picture).

<표 3> 정지화상에 대한 도착 셀에 대한 셀 손실비

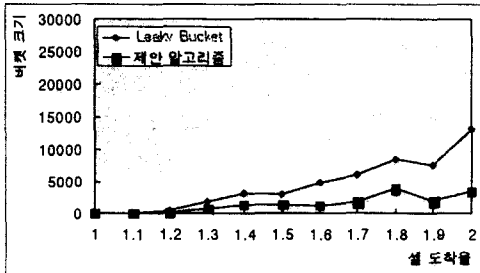
(Table 3) The cell loss ratio of departed cell (still picture)

계수기 한계값	셀 도착율	LB 셀 손실률	제안 알고리즘 셀 손실률	셀 손실률 비 (LB=1.0)
300	1.1	0.000001	0.000001	1.000
	1.3	0.001884	0.000816	0.433
	1.5	0.002977	0.001374	0.461
	1.7	0.006121	0.001750	0.286
	1.9	0.007614	0.002040	0.268
500	1.1	0.000399	0.000198	0.496
	1.3	0.002711	0.001044	0.385
	1.5	0.010577	0.005298	0.501
	1.7	0.013485	0.002250	0.167
	1.9	0.017411	0.006437	0.370

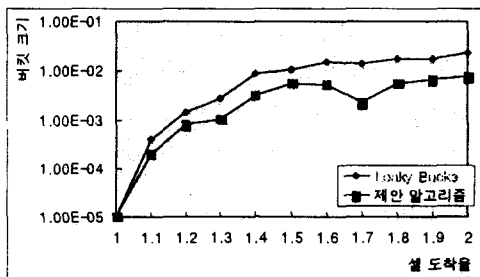
4.4.4 버킷 크기 감소 (정지화상)

(그림 11)과 (그림 12)는 정지화상에 대한 계수기 한계값을 300, 500으로 하였을 경우의 시뮬레이션 결과를 보여주었다. 계수기 한계값이 300일 경우에는 도착셀을 1.9에서 추가로 요구되는 버킷을 약 3배 정

도로 줄일 수 있었으며, 계수기 한계값이 500인 경우는 기존의 리키 버킷 알고리즘이 7,614개의 버킷을 추가로 요구하는 반면에 제안 알고리즘은 2,040개의 버킷만 추가로 필요로 하며, 5,574개의 버킷이 줄어들음을 보여 주었다.



(그림 11) 계수기값 300일 때 버킷 크기 (정치화상).
(Fig. 11) Bucket size when a count limits is 300 (still picture).



(그림 12) 계수기값 500일 때 버킷 크기 (정치화상).
(Fig. 12) Bucket size when a count limits is 500 (still picture).

이상의 시뮬레이션 결과에 의하면 제안 알고리즘이 기존의 LB 알고리즘 보다 동일한 셀 손실률 하에서 요구되는 버퍼의 크기를 줄일 수 있음을 알 수 있었다. 위반 트래픽에 반응시간도 사용 매개변수 제어 알고리즘의 성능을 평가하는데 있어서 중요한 인자로 알려져 있으나 평균비트를 측정기간의 설정에 따라 반응시간이 달라질 수 있는 제안 알고리즘의 특성상 구체적인 검토가 이루어지지 못하였다.

<표 4>에 제안 알고리즘과 LB 알고리즘을 비교하였다. 제안 알고리즘과의 중요한 차이점은 평균비트

<표 4> LB 알고리즘과 제안 알고리즘 시뮬레이션 비교
<Table 4> Comparison of leaky bucket vs the proposed algorithm

알고리즘 항목	Leaky Bucket	제안 알고리즘
셀 폐기 알고리즘	계수기 한계값을 초과하여 도착한 셀을 무조건적으로 폐기	계수기 한계값을 초과하여 도착한 셀을 표시한 후 망의 평균비트를 위반여부에 따라 최종 폐기
평균비트율 측정	큰 셀 버퍼나 계수기 한계값이 필요하며 이로 인해 위반 트래픽에 대한 느린 반응시간을 보임	윈도우 기반 평균비트율을 측정 알고리즘 추가하여 작은 셀 버퍼나 계수기 한계값으로도 위반 트래픽에 대하여 빠른 반응시간을 보임
Type I-Error	발생	발생하지 않음
반응 시간	계수기 한계값에 의해 결정	평균비트율 측정 윈도우 간격 설정에 따라 조절 가능
평균비트율 제어	평균비트율 제어 곤란	평균비트율 제어용이

율의 감시여부와 셀 폐기 알고리즘에 있다. 이러한 알고리즘 차이는 Type-I 에러의 발생을 줄일 수 있으며 이는 요구 버킷 크기의 감소로 이어지게 된다.

5. 결 론

ITU-T의 권고안 I.371은 폭주 예방 기능으로 연결 수락 제어, 사용 매개변수 제어, 자원 관리, 우선 순위 제어 및 트래픽 셰이핑 기능을 정의하였으며, 폭주 회복 기능으로 선택적 셀 폐기 및 EFCI를 정의하고 있다.

본 논문에서는 리키 버킷을 사용하여 최대비트율을 위반한 셀을 표시한 후, 만약 망으로 전송된 셀들의 평균비트율을 측정하여 협약을 이하라면 표시된 셀들을 망으로 재전송하는 새로운 알고리즘을 제안한다. 기존의 사용 매개변수 제어 기법중 가장 효율적인 것으로 보고되고 있는 LB 기법은 최대비트율 제어에는 우수한 성능을 보이지만 비교적 긴 시간 동안 트래픽을 감시하는 평균비트율 제어에는 적합하지 못한 것으로 평가되고 있다. 제안한 알고리즘은 윈도우에 기반한 JW 알고리즘을 추가한 이중구조로 설계함으로써, 동시에 최대비트율과 평균비트율을

제어하였다. 따라서 기존의 사용 매개변수 제어 알고리즘에서 평균비트율 제어가 곤란한 단점을 보완할 수 있는 새로운 알고리즘을 제안하고 시뮬레이션을 통해 성능을 분석하였다. 제안 알고리즘은 먼저 LB를 사용하여 최대비트율을 제어하고 다음에 JW를 사용하여 평균비트율을 제어한다. 현재의 망이 평균비트율을 위반하지 않는 한 LB 기법에 의해 위반셀로 판단된 셀들을 망으로 전송함으로써 셀 손실률을 줄였으며 동시에 셀 버퍼의 크기를 줄일 수 있게 하였다. LB 알고리즘은 평균비트율 제어를 위하여 큰 계수기 한계값을 필요로 하는데 비해 제안 알고리즘은 작은 계수기 한계값과 셀 버퍼를 가지고도 평균비트율을 제어할 수 있다.

제안 알고리즘의 시뮬레이션은 셀 도착 및 서비스 과정을 모델링하기 위하여 SMPL 라이브러리를 활용하여 SUN 워크스테이션 상에서 시뮬레이션 결과, 기존 리킷 버킷 방법보다 상당한 셀 손실률의 저하를 나타내었으며 요구되는 버킷 크기도 감소시킬 수 있었다.

참 고 문 헌

[1] 강상혁 외, "ATM 트래픽 제어기술", 전자공학회지, Vol. 19, No. 8, pp. 659-671, Aug. 1992.
 [2] Kiyoshi Shimokoshi, "Evaluation of policing mechanism for ATM networks", IEICE Trans. Commun., Vol. E76-B, No. 10, pp. 1341-1351, Nov. 1993.
 [3] M. Butto, E. Cavallero, A. Tonietti, "Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks", IEEE Journal of Selected Areas in Communications, Vol. 9, No. 3, pp. 335-342, Apr. 1991.
 [4] F. Guillemin, A. Dupuis, "A basic requirement for the policing function in ATM networks", Computer Networks and ISDN Systems, No. 24, pp. 311-320, 1992.
 [5] ITU-TS Recommendation I.371, White book, 1992.
 [6] J. S. Turner, "Managing Bandwidth in ATM Networks with Bursty Traffic," IEEE Network

Magazine, pp. 50-58, 1992.

[7] B. Lague, C. Rosenberg, F. Guillemin, "A Generation of Some Policing Mechanism", INFOCOM '92, IEEE, 1992.
 [8] J. W. Robert, "Traffic Control in the B-ISDN", Computer Networks and ISDN System, No. 25, pp. 1005-1064, 1993.
 [9] K. Tutufor, "On Admission Control and Policing in and ATM Based Networks", in Proceedings of ITC, 1990.
 [10] P. Castelli, E. Cavallero, A. Tonietti, "Policing and Call Admission Problems in ATM networks", Teletraffic and Datatraffic in a Period of Change. ITC-13, pp. 847-852, 1991.
 [11] N. Yamanaka, Y. Sato, K. Sato, "Usage Parameter Control and Bandwidth Method Considering Cell Delay Variation in ATM Networks", IEICE Trans. Commun., Vol. E76-B, No. 3, pp. 270-278, Mar. 1993.
 [12] Roberts, J. and Guillemin, F., "Jitter in ATM Networks and its Impact on Peak Rate Enforcement", Performance Evaluation, Vol. 16, No. 1-3, pp. 36-48, North-Holland, Nov. 1992.
 [13] 이기봉 외, "A Study on Traffic Control in B-ISDN(ATM Networks)", 한국통신학회지, Vol. 20, No. 3, pp. 569-580, May. 1995.
 [14] 이우섭 외, "ATM 트래픽에 대한 사용 파라미터 제어 방법들의 고찰", 전자통신연구소 주간기술동향, Vol. 25, pp. 29, 1992.



박 성 곤

1981년 광운대학교 전자공학과 졸업 (공학사)
 1984년 한양대학교 대학원 전자계산전공 졸업 (공학석사)
 1997년 충북대학교 대학원 컴퓨터공학과 졸업 (공학박사)

1981년~1986년 쌍용정보통신 연구소 교육센터
 1986년~1991년 경북전문대학 전자계산과 조교수
 1991년~현재 원주전문대학 전산정보처리학과 부교수
 관심분야: ATM 망 트래픽 제어, 멀티미디어 통신 서비스