

論文96-33A-2-4

# ATM망에서 가상 버퍼를 이용한 사용자 파라메타 제어 메카니즘

## (Usage Parameter Control Mechanism Using a Virtual Buffer for ATM Networks)

朴昌郁\*, 林寅七\*\*

(Chang-Wouk Park and In-Chil Lim)

### 요 약

본 논문은 가상 버퍼를 관측하여 토큰 발생률을 가변시킴으로써 리키 버킷의 성능을 향상시킨 새로운 UPC(Usage Parameter Control) 메카니즘을 제안한다. 광대역 종합정보통신망에서 사용자의 호는 망의 호 설정 제어에 의해서 설정된다. 이 때, UPC는 호가 협약된 트래픽 파라메타들을 준수하는지를 감시하고, 위반 시 적절한 제재 조치를 취함으로써 망내에서 일어나는 체증을 예방하는 역할을 수행한다. 협약을 위반한 호에 대한 기존 UPC의 셀손실률이 이상적인 UPC와는 차이를 보인다. 제안하는 방식은 협약을 위반하는 호에 대해서 뛰어난 감시 능력과 반응시간을 가진다. 제안한 방식은 온-오프 모델을 입력 트래픽으로 사용하며, 모의 실험을 통해 효율성을 입증한다.

### Abstract

This paper proposes a new UPC(Usage Parameter Control) mechanism that varies the token generation rate of Leaky Bucket by observing virtual buffer, thus achieving an improvement of performance. An acceptance of a call in B-ISDN is determined by Call Admission Control. A UPC that polices the agreed traffic parameters of a call and takes appropriate actions to prevent congestion in networks. The cell loss probability of conventional UPC is far from that of ideal UPC. The proposed mechanism has a better policing ability and response time for nonconforming call. The proposed mechanism uses ON-OFF model as input traffic. The results of simulation show the efficiency of the proposed mechanism.

### 1. 서 론

ATM망은 정보들을 53바이트 크기의 셀에 담아 전송하게 되며 통계적 다중화를 통해 대역폭을 효율적으로 이용할 수 있지만 망내에 혼잡(congestion)이 발생할 수 있다. ATM의 장점이 충분히 발휘되려면 트래픽 제어 및 자원 관리의 문제가 해결되어야 한다. 고속 광대역 종합정보통신망(BISDN : Broadband Integrated Services Digital Network)에서는 반응제어(reactive control)가 효과를 나타내는데까지 걸

리는 지연 시간이 치명적일 것으로 예상된다. 따라서 ATM망에서는 반응 제어보다는 예방 제어(preventive control)가 트래픽 제어의 주된 수단으로 대두되고 있다<sup>[1]</sup>. 예방 제어의 예로서 연결 수락 제어(CAC : Call Admission Control)와 사용자 파라메타 제어(UPC : Usage Parameter Control)를 들 수 있다. 사용자가 신호원의 통계적 특성을 나타내는 트래픽 파라메타를 망에 전달하면, CAC는 망의 현재 상태와 신호원의 특성을 토대로 연결 설정 여부를 결정한다. 연결이 설정되면 신호원은 망에 셀들을 전송하고, 망 관리부는 진입하는 셀들이 트래픽 협약을 준수하는지를 감시하기 위해 UPC를 수행해야 한다. 관측하고자 하는 트래픽 파라메타들은 최대셀률, 평균셀률 등 여러가지가 있으나 이 중 제어가 어려운 평균셀률

\* 正會員, 漢陽大學校 工科大学 電子工學科  
(Dept. of Electronics Eng., Hanyang Univ.)

接受日字: 1995年12月11日, 수정완료일: 1996年1月27日

에 관한 연구가 활발하다<sup>1,2,12</sup>.

ATM망에서 지원할 트래픽들은 대부분 군집성(Bursty)의 특성을 가지고 있기 때문에 정확한 평균셀률의 감시 및 제어를 수행하기 위해서는 긴 관측 시간을 필요로 한다. 그러나 긴 관측 시간은 트래픽 협약을 위반한 호에 대해 적절한 제재 조치를 취하지 못하기 때문에 망내에 혼잡을 발생시켜서 협약을 준수하는 다른 호의 서비스 품질을 나쁘게 할 수 있다. 기존의 제안된 UPC 메카니즘들은 LB(Leaky Bucket), JW(Jumping Window), TJW(Triggered Jumping Window), MW(Moving Window) 그리고 EWMA(Exponentially Weighted Moving Window Average)등이 있으나 협약을 위반한 호에 대한 셀손실률 및 반응시간이 이상적인 평균셀률 UPC 메카니즘과는 차이를 보인다. 따라서 보다 이상적인 제어에 근접하기 위해 보다 섬세한 제어가 요구된다.

본 논문에서는 이와같은 차이를 최소화하기 위해 기존의 토큰 풀(Token Pool)을 갖는 LB를 이용하며 협약을 위반한 셀들을 감시하기 위해 가상 버퍼를 이용하는 새로운 UPC 메카니즘을 제안한다(이후 VB-LB: Virtual Buffer - Leaky Bucket이라 칭한다.). VB-LB에 도착하는 셀들 중 토큰풀내에 토큰이 없어서 손실되는 셀들은 가상 버퍼에 가상적으로 저장시키고, 저장된 셀들의 갯수는 토큰 발생률을 가변시키기 위한 관측대상이 된다. 관측되는 가상 버퍼내의 셀의 갯수가 임계치 이하일 경우는 협약을 지키는 셀들로 간주하여 정상적으로 토큰을 발생시키고, 임계치를 넘었을 경우는 협약을 위반한 셀들로 간주하여 토큰 발생률을 줄임으로써 위반 셀들의 망 진입을 억제하게 된다. 또한 흰 토큰과 검은 토큰의 개념을 도입하여 가상 버퍼내의 셀 감소를 토큰풀내의 토큰과 연관시킨다. 흰 토큰은 토큰 발생기에 의해 일정한 시간 간격으로 발생되어 토큰풀에 저장된다. 만약 토큰풀이 차게되면 이후 발생하는 토큰은 잃어버리게 된다. 검은 토큰은 가상 버퍼내의 셀이 감소될 때 흰 토큰을 검은 토큰으로 변환을 시킴으로써 생성된다. 즉 토큰풀내에 흰 토큰이 없다면 가상 버퍼내의 셀은 감소하지 않는다.

제안한 VB-LB는 망내의 혼잡을 유발하여 트래픽 협약을 준수하는 다른 호의 서비스 품질을 나쁘게 할 수 있는, 협약보다 더 많은 셀들을 전송하는 호에 대해 기존의 UPC 메카니즘보다 빨리 제재를 가할 수 있고

이상적인 평균셀률 UPC 메카니즘의 셀손실률에 근접함을 컴퓨터 시뮬레이션을 통해 입증한다. 제안한 알고리즘은 Sun Sparc Station상에서 C 언어로 구현하며 입력 트래픽은 온-오프 모델을 사용한다.

## II. 신호원과 UPC 메카니즘

UPC 메카니즘의 성능을 평가할 때 입력 신호원의 특성은 결과에 많은 영향을 미친다. 따라서 그림 1과 같이 버스트 상태와 침묵 상태를 반복하는 온-오프 모델을 사용한다. 이 모델은 음성이나 정지 화상을 모델링하는데 사용되고, 최대셀률, 평균셀률, 평균 피크 기간(peak duration) 등의 파라메타들이 상호 독립적으로 변화하는 것을 허용한다<sup>4,5</sup>.

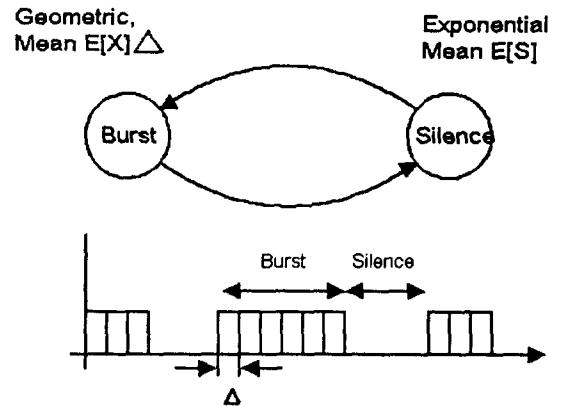


그림 1. 온-오프 모델

Fig. 1. On - Off Model.

온-오프 모델에서 셀은 침묵 상태에 있을 때 전혀 도착하지 않고, 버스트 상태에 있을 때는 일정한 간격을 두고 도착한다. 버스트당 셀의 수는 평균  $E[X]$ 를 갖는 기하(geometric)분포로 가정되고, 침묵(silence)기간은 평균  $E[S]$ 를 갖는 지수(exponential)분포로 가정된다. 그리고, 버스트당 셀 간격은  $\Delta$ 로 주어지고, 버스트 구간( $\alpha^{-1}$ )과 침묵 구간( $\beta^{-1}$ )은 식 (1)과 식 (2)로 표현될 수 있다.

$$\alpha^{-1} = E[X] \times \Delta \quad (1)$$

$$\beta^{-1} = E[S] \quad (2)$$

이 때, 평균셀률  $\lambda$ 는 다음과 같이 정의된다.

$$\lambda = \frac{E[X]}{\alpha^{-1} + \beta^{-1}} = \frac{1}{\Delta + \frac{\Delta}{\beta}} \quad (3)$$

식 (3)에서도 알 수 있듯이  $E[X]$ ,  $E[S]$  그리고  $\Delta$ 등이 평균셀수에 각각 영향을 미친다. 버스트당 셀의 수  $E[X]$ 가 증가하면 평균셀수가 증가하고, 침묵기간  $E[S]$ 가 증가하면 평균셀수가 감소한다. 또한, 셀 간격  $\Delta$ 가 증가하면 평균셀수는 감소한다.

UPC 메카니즘은 개별적인 연결들을 감시하여 실질적인 트래픽 흐름이 호 설정시 신고한 트래픽 파라미터를 준수하도록 보장해야 하며 또한 실시간적으로 동작해야 한다. 악의의 사용자나 장비의 고장으로 인해 호 설정시 신고된 트래픽 양을 초과하여 망을 과부하 상태로 만들 수 있기 때문에 연결이 수락된 후, UPC는 그 연결의 실질적인 트래픽 흐름이 호 설정시 신고된 비를 준수하는지를 감시하고 일단 위반이 발견되면, 위반 셀을 폐기(Discarding)시키거나 셀손실 우선순위(CLP : Cell Loss Priority) 비트를 이용하여 태깅(tagging)함으로써 혹은 지연을 시키는 등의 적절한 조치를 취해야만 한다<sup>[31]</sup>.

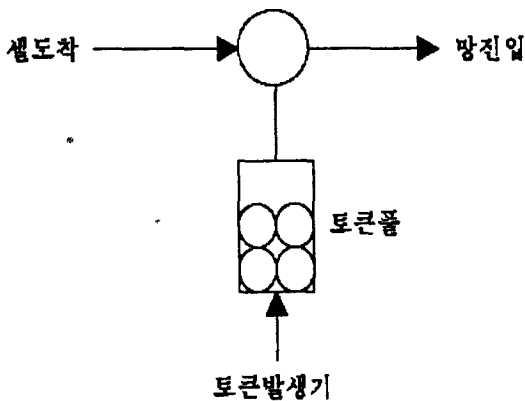


그림 2. 리키 버킷 메카니즘  
Fig. 2. The Leaky Bucket Mechanism.

그림 2의 토큰풀을 갖는 LB는 ATM망에서 고려되는 전형적인 UPC 메카니즘 중의 하나이다. 토큰풀의 크기는  $N$ 으로 고정되어 있고 토큰은 토큰 발생기에 의해 신호원의 평균셀수에 따라 고정된 발생 간격  $d$ 로 발생되어 토큰풀에 저장된다. 토큰풀내의 토큰의 수가  $N$ 이 되면 이후 발생되는 토큰은 손실된다. UPC에 도착한 셀은 토큰풀내에 토큰이 있을 경우 하나의 토큰을 제거하고 망으로 진입하게 되고 토큰풀이 비었을 경우 잃어버린 셀이 된다.

LB는  $G/D/1/s$  시스템으로 모델링될 수 있다<sup>[3,8,12]</sup>

1. 이산 시간(Discrete time)인 경우 이 모델은  $D$ 의

서비스 시간을 갖는 하나의 서버,  $s$ 개의 대기공간 그리고 일반 도착 프로세스  $G$  로서 구성된다. 모델의 서비스 시간  $D$ 는 토큰 발생간격  $d$ 와 같도록 선택되며 서비스를 받고 있는 셀을 포함한 시스템내의 셀 갯수는 토큰풀의 상태를 나타낸다. 따라서 토큰풀의 크기  $N$ 은  $s + 1$ 과 동일하다.

또 다른 평균셀률 UPC로서 윈도우 메카니즘을 들 수 있다. 이러한 메카니즘들은 모두 일정한 크기의 윈도우를 가지고 있다는 특징이 있다. 윈도우 메카니즘중에서 가장 간단한 방법으로는 JW를 들 수 있다. JW는 고정된 크기의 시간(윈도우)내에 받아들일 수 있는 셀의 최대 갯수를 제한한다. 새로운 윈도우는 바로 이전 윈도우의 끝에서 시작한다. JW는 다른 메카니즘보다 간단하며 구현하기가 쉬운 반면 LB보다 셀 손실 확률이 큰 단점이 있다.

JW가 셀의 흐름과 동기화되어 있지 않은 점을 개선한 방식이 TJW 메카니즘이다. TJW의 새로운 윈도우는 전 윈도우가 끝난 후 첫 셀이 도착할 때 시작된다. TJW도 또한 간단하지만 LB나 JW보다 셀 손실 확률이 크다.

JW와 TJW가 모두 한 개의 윈도우를 가지고 있는 반면 MW는 여러개의 윈도우를 가지고 있다. JW와 유사하게 주어진 윈도우 크기  $T$ 동안 받아들일 수 있는 최대 셀의 수는  $N$ 개로 제한된다. JW와 다른 점은 각 셀이 윈도우 주기  $T$ 동안 기억된다는 점이다. MW는 시간축을 따라 이동하는 윈도우로서 해석될 수 있으며 최대  $N$ 개 셀의 도착 시간이 윈도우 크기 동안 저장될 것을 필요로 한다. 그러므로  $N$ 이 커질수록 복잡해진다.

윈도우 메카니즘 중에서 가장 성능이 뛰어난 방식은 EWMA이다<sup>[5,81]</sup>. EWMA는 JW처럼 고정된 크기의 윈도우를 가지고 있다. JW와 다른 점은  $i$ 번째 윈도우내에 받아들일 수 있는 최대 셀의 수( $N_i$ )가 식 (4)처럼 윈도우 크기 동안 받아들일 수 있는 셀의 평균인  $N$ 과 전 윈도우에서 받아들인 셀의 지수 가중치화된 합(exponentially weighted sum)인  $S_{i-1}$ 의 함수라는 점이다.

$$N_i = \frac{N - \gamma S_{i-1}}{1 - \gamma} \quad 0 \leq \gamma < 1 \quad (4)$$

$$S_{i-1} = (1 - \gamma)X_{i-1} + \gamma S_{i-2} \quad (5)$$

따라서 식 (4)은 식 (6)과 같이 표현될 수 있다.

$$N_i = \frac{N - (1-\gamma)(\gamma X_{i-1} + \dots + \gamma^{i-1} X_1) - \gamma^{i+1} S_0}{1-\gamma} \quad (6)$$

여기서  $X_{i-1}$ 은  $i-1$ 번째 윈도우에서 받아들인 셀의 갯수이고,  $S_0$ 는 초기치이다. 이 식에서  $\gamma$ 는 EWMA의 유연성을 조절한다. 만약  $\gamma$ 가 0이면  $N_i$ 는 상수가 되고 이 알고리즘은 JW와 똑같다.  $\gamma$ 가 커질수록 위반 셀을 잘 검출하지만 반응 속도가 느려지는 단점이 있으며 다른 메카니즘에 비해 복잡하다.

### III. 가변 토큰발생 알고리즘

협약을 위반하고 많은 양의 셀들을 전송하는 경우 기존의 UPC 메카니즘들의 성능은 이상적인 평균셀률 UPC에 미치지 못한다. 왜냐하면, 윈도우 메카니즘은 윈도우가 커질수록, LB는 토큰풀의 크기가 커질수록 셀 손실 확률 면에서는 성능이 향상되지만, 반응시간 면에서는 성능이 떨어지는 단점을 가지고 있기 때문이다. LB가 입력 버퍼를 갖는 경우는 셀 지연 시간과 셀 손실 확률 사이의 트레이드 오프(trade-off)가 존재한다. 입력 버퍼가 없는 경우 들어오는 셀들은 버퍼 안에서 대기할 수 없고 폐기되기 때문에 셀들을 잃어버리게 된다. 무한 버퍼가 존재하는 경우 어떤 입력 셀도 잃어버리지 않지만 긴 지연 시간을 갖게 된다. 입력버퍼를 갖는 LB의 셀 손실 확률은 입력 버퍼의 크기나 토큰풀의 크기 중 어느 하나에 의존하지 않고, 두 크기의 합에 의존한다<sup>[11]</sup>. 그러나 입력 버퍼에 의한 지연은 실시간 전송을 요구하는 서비스에 대해 치명적일 수 있다. 따라서 본 논문에서는 지연없이 투명한(Transparent) 제어가 이루어질 수 있는 새로운 UPC 메카니즘을 제안한다.

그림 3은 제안하는 VB-LB로 토큰풀과 입계치를 갖는 가상 버퍼로 구성된다. 그림 3에서 토큰 발생기는 일정한 시간 간격으로 흰 토큰을 발생시키고 발생된 흰 토큰은 토큰풀에 저장된다. 이때 토큰 발생률( $\lambda_p$ )은 트래픽의 군집성때문에 평균셀률( $\lambda$ )보다는 약간의 여유를 두며 이를 overdimensioning factor(C)라고 한다. 그러므로 토큰 발생률( $\lambda_p$ )은 다음 식과 같다.

$$\lambda_p = C \cdot \lambda \quad (7)$$

토큰풀내의 흰 토큰은 가상 버퍼내에 저장되어 있는 셀을 감소시킬때 검은 토큰으로 변환된다.

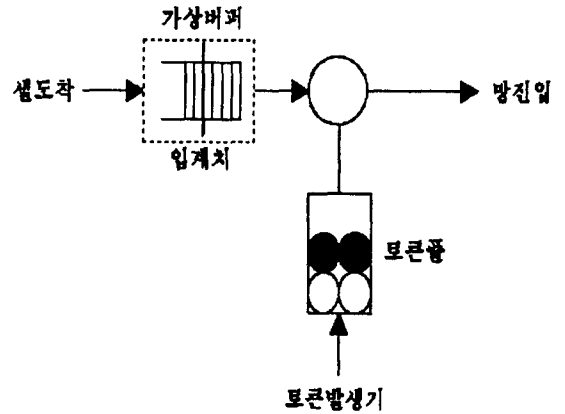


그림 3. 가상 버퍼를 가진 리키 버킷  
Fig. 3. The Leaky Bucket with Virtual Buffer.

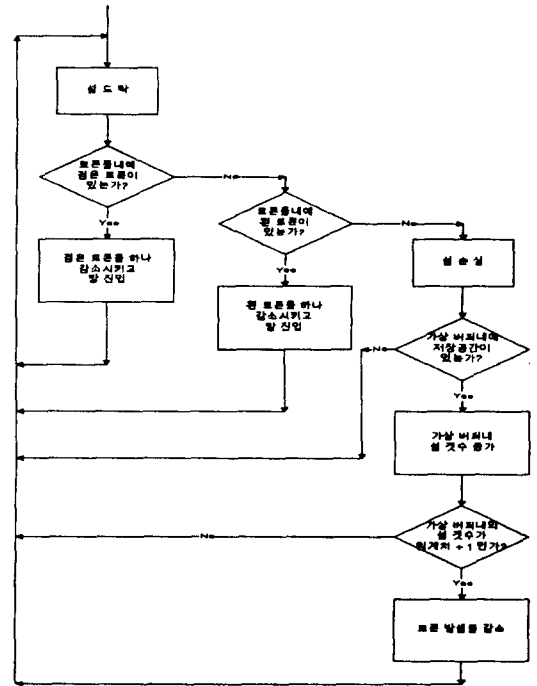


그림 4. 망 진입 및 가변 토큰 발생률 알고리즘  
Fig. 4. The network entrance and variable token generation rate algorithm.

그림 4는 도착 셀의 망 진입 및 가변 토큰 발생률 알고리즘을 보여준다. 도착하는 셀은 토큰풀내에 검은 토큰이 있다면 검은 토큰을 하나 제거하고 망으로 진입하게 되며 만약에 토큰 풀내에 검은 토큰이 없고 흰 토큰이 있다면 흰 토큰을 제거하고 망으로 진입하며

토큰 풀내에 토큰이 없으면 가상 버퍼에 저장된다. 이때 가상 버퍼에 저장된다는 의미는 사실상의 셀 손실을 의미하게 되며 토큰 발생률을 조정하기 위한 관측 대상으로 이용된다.

넘을 경우는 협약을 위반한 것으로 간주하여 토큰 발생률을 줄임으로써 위반셀의 망진입을 억제하게 된다. 이때의 토큰 발생률을  $\lambda'$ 라 하면  $\lambda'$ 의 범위는 식 (8)과 같다.

$$\lambda \leq \lambda' \leq \lambda_p \tag{8}$$

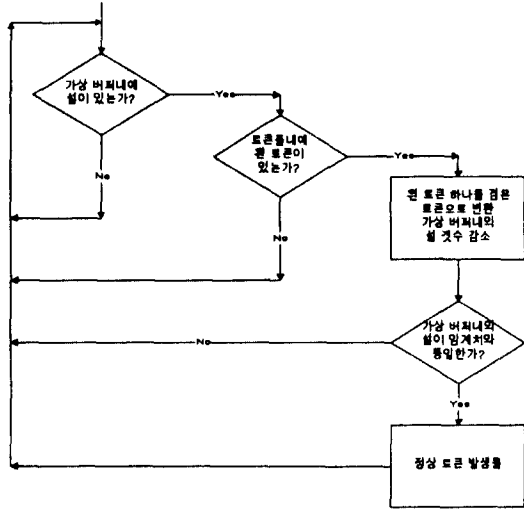


그림 5. 가상 버퍼 및 가변 토큰 발생률 알고리즘.  
Fig. 5. The virtual buffer and variable token generation rate algorithm.

그림 5는 가상 버퍼내의 셀 감소 및 가변 토큰 발생률 알고리즘을 보인다. 어느 순간 가상 버퍼 내에 셀이 있다는 것은 토큰풀내에 흰 토큰이 하나도 없다는 것을 의미한다. 토큰 발생기가 흰 토큰을 하나 발생시키자마자 가상 버퍼내의 셀은 흰 토큰을 검은 토큰으로 변환만 시키고 가상 버퍼를 벗어나게 된다. 이때 가상 버퍼내의 셀은 이미 잃어버린 셀이기 때문에 망으로 진입하는 것은 아니다. 또한 가상 버퍼 내의 셀 갯수가 임계치 이하가 될 경우는 단기간의 많은 군집셀들로 협약을 준수하는 것으로 간주하여 정상적인 토큰발생률( $\lambda_p$ )로 토큰을 발생시킨다.

#### IV. 실험 결과 및 분석

기존의 LB는 일정한 속도로 토큰을 발생시킴으로써 약속을 어기는 셀들을 정확히 검출하지 못하고 있다. 이와 같은 단점을 극복하기 위해 본 논문에서는 토큰 발생률을 가변시키는 방법을 제안한다. 토큰 발생률을 가변시키기 위해서는 두 가지 점을 고려해야 한다. 첫째는 어떤 정보를 기초로 해서 가변할 시점을 택할 것인가 하는 점이며, 둘째는 가변할 시점에서 얼마만큼의 변화를 주어야 할 것인가 하는 점이다. 첫번째 문제의 경우 본 논문은 가상 버퍼내의 셀 갯수를 근거로 하여 토큰 발생률을 가변시킨다. 토큰풀의 크기는 요구되는 서비스 품질과 군집성에 따라 결정된다. 따라서 토큰풀내에 토큰이 없는 경우에 도착하는 셀은 위반 셀이거나 협약을 지키지만 단기간의 큰 군집셀중의 하나일 수 있다. 따라서 토큰풀내에 토큰이 없어서 가상버퍼내에 저장되는 셀들은 협약 이상을 전송하는 위반셀 판단의 중요한 관측자료가 된다. 두번째 문제의 경우 가상 버퍼내의 임계치를 설정하여 가변시점을 결정한다. 가상 버퍼내의 임계치는 요구되어지는 셀손실률을 만족시키기 위해 시뮬레이션을 통해 결정된다. 임계치를

제안한 VB-LB의 성능을 평가하기 위하여 입력 트래픽은 온-오프 모델을 통하여 발생시켰으며 이 때, 입력 트래픽 특성은 버스트 상태에서 평균 셀  $E[X] = 22$ 개, 침묵 상태의 평균간격  $E[S] = 650\text{ms}$  그리고 버스트 상태내의 셀 간격  $\Delta = 16\text{ms}$ 로 하였다 [18].

제안하는 VB-LB는 입력 버퍼를 갖지 않는다. 입력 버퍼를 가진 LB의 평균 지연을 온-오프 입력 트래픽을 이용하여 시뮬레이션하였다. 그림 6은 버퍼의 크기와 토큰 풀의 크기를 183으로 일정하게 유지하고 버퍼의 크기를 증가시켰을 때의 평균 지연을 나타낸다. 버퍼의 크기가 커짐에 따라 지연이 크게 증가함을 알 수 있다. 따라서 본 논문은 이와같이 입력 버퍼에 의한 지연을 피하기 위해 버퍼가 없는 VB-LB를 제안한다. UPC의 성능은 약속을 준수한 셀을 손실 없이 통과시키는 능력, 약속을 어기는 셀을 잘 찾아내는 능력, 약속을 어기는 셀을 빠른 시간에 찾는 능력 그리고 복잡도 등으로 평가되어질 수 있다. 제안하는 VB-LB와 비교하고자 하는 UPC 메카니즘들이 약속을 준수하는 셀들을 손실 없이 통과시킬 수 있도록 동일하게 조정 한 후 약속을 어기는 셀을 찾아내는 능력 및 약속을 어기는 셀을 빠른 시간에 찾는 능력을 상호 비교한다.

또한 복잡도는 시뮬레이션을 통해 비교 평가한다.

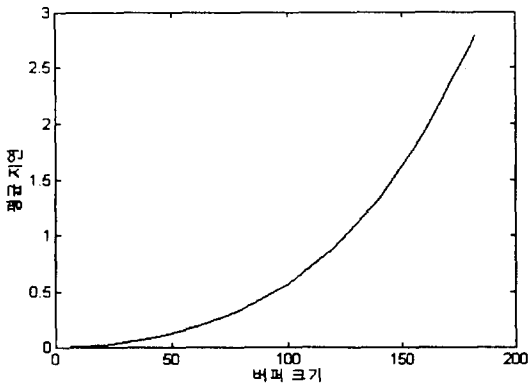


그림 6. 버퍼 크기에 따른 평균 지연  
Fig. 6. Influence of the buffer size on the average delay.

본 논문에서는 제안한 VB-LB를 LB, EWMA 그리고 JW와 비교를 하였다. 이러한 비교를 하기 위해서는 우선 동일한 조건을 만들어야 한다.

EWMA와 JW의 경우 조정할 수 있는 파라메타로 윈도우 크기 T는 식 (9)처럼 LB의 토큰풀 크기 N과 관측 속도  $\lambda_p$ 에 의해 결정될 수 있다. 윈도우내 허용할 수 있는 셀의 갯수는 토큰풀의 크기 N과 동일하게 정의한다. LB와 동일한 조건을 구하기 위해 식 (9)를 사용한다<sup>[8]</sup>.

$$T = \frac{N}{\lambda C} = \frac{N}{\lambda_p} \quad (9)$$

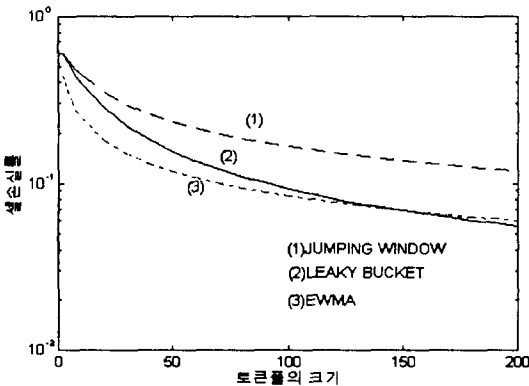


그림 7. 토큰풀 크기에 따른 셀손실률  
Fig. 7. Influence of the token pool size on the cell loss rate(C = 1.0).

그림 7은 토큰풀 크기가 셀손실률에 미치는 영향을 보여준다. 토큰 발생률은 식 (3)에 의해 계산된 평균

셀률인  $\lambda$ 로 한다. 즉 overdimension factor C는 1.0이다. 또한 EWMA의  $\gamma$ 의 값은 0.8로 선택하였다. 토큰풀의 크기가 150이상 부터는 LB가 가장 낮은 셀손실률을 보이고 있다. 토큰풀의 크기가 커짐에 따라서 셀손실률은 감소한다. 그러나 셀손실에 민감한 서비스의 셀손실률인  $10^{-9}$  을 만족시키기 위해서는 아주 큰 토큰풀 크기를 요구함을 알 수 있다.

그림 8은 토큰 발생률을 최대셀률에 근접한 값으로 발생시켰을때의 셀손실률을 보여준다. overdimension factor C를 2.28로 하였을 때 EWMA와 LB는 셀손실률이 급격하게 작아지고 JW의 경우도 C가 1.0일때보다 셀손실률이 많이 감소함을 알 수 있다. 그러나 JW는 동일한 셀손실률을 얻기 위해서 다른 메카니즘보다 큰 계수기를 사용해야 하며 이는 구현의 복잡도를 증가시키는 것이다. 그림 7과 8에서 토큰 발생률( $\lambda_p$ )이 셀손실률에 미치는 효과가 크다는 것을 알 수 있다.

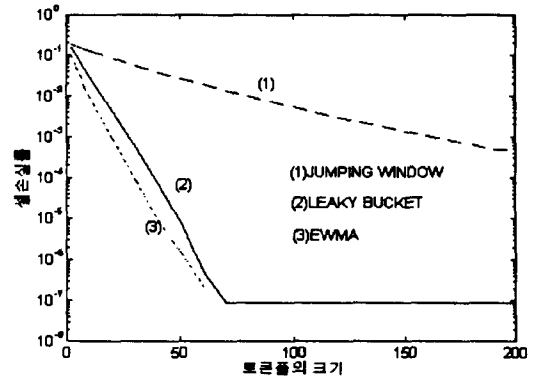


그림 8. 토큰풀 크기에 따른 셀손실률  
Fig. 8. Influence of the token pool size on the cell loss rate(C = 2.28).

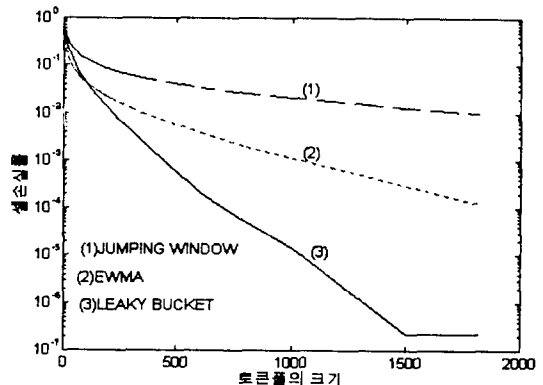


그림 9. 토큰 풀 크기에 따른 셀손실률  
Fig. 9. Influence of the token pool size on the cell loss rate(C = 1.1)

그림 9는 각 메카니즘을 동일한 조건에서 비교하기 위해 토크폴 크기에 따른 셀손실률을 구하였다. 본 논문은 요구되는 셀손실률을  $10^{-2}$ 으로 overdimension factor C는 1.1로 가정하고 각 메카니즘들을 비교하였다. 그림 9에서 요구되는 셀손실률을 만족시키기 위해서 LB는 216, EWMA는 346 그리고 JW는 1820의 토크폴 크기가 필요하다.

표 1. 가상 버퍼의 크기와 임계치에 의한 셀 손실률.

Table 1. Influence of the virtual buffer size and threshold on cell loss rate.

가상버퍼 크기 \ 임계치	30	50	100
1	0.0108318	0.0108429	0.0108435
5	0.0103992	0.0104086	0.0104093
9	0.0101773	0.0101860	0.0101862
13	0.0100649	0.0100709	0.0100717
17	0.0100038	0.0100107	0.0100114
18	0.0099932	0.0099989	0.0099995
25	0.0099571	0.0099602	0.0099611

표 1은 제안하는 가상버퍼를 이용하여 토크 발생률을 가변시키는 VB-LB의 셀손실률이다. 가상 버퍼내의 셀 갯수가 임계치이하일때는 토크 발생률  $\lambda_p$ 를  $1.1 \times \lambda$ 로, 임계치를 넘었을 때는 토크 발생률  $\lambda_p$ 를  $1.0 \times \lambda$ 로 가변시킨다. 가상 버퍼의 크기가 30, 50 그리고 100으로 변화할때, 또한 가상버퍼내의 임계치가 변화할때의 셀손실률로  $10^{-2}$ 을 만족하는 임계치는 모두 18임을 알 수 있다. VB-LB는 가상 버퍼의 크기를 30 그리고 임계치를 18로 가정한다. 그림 9와 표 1에서 각 메카니즘이 협약을 준수하는 호의 셀손실률을 만족하도록 하는 각 파라메타들을 결정하였다.

그림 10은 과부하(overload)상태일 때 각 UPC 메카니즘들의 위반 셀을 검출하는 능력을 나타낸다. 그림에서 x축은 overload factor이다. 이 값은 식 (10)과 같이 구할 수 있다.

$$overload\ factor = \frac{\text{입력 트래픽의 실질적인 평균셀률}}{\text{입력 트래픽의 약속된 평균셀률}} \quad (10)$$

이와 같이 약속을 어기는 셀들을 발생시키기 위해서 E

[X]를 증가시키는 방법과 E[S]를 감소시키는 방법이 있다. 본 논문에는 E[X]를 증가시키는 방법을 사용한다. 이상적인 평균셀률 UPC 메카니즘은 약속을 어긴 셀들을 모두 제거해야 한다. 즉 약속보다 2배의 셀들을 전송하는 호에 대해서 0.5의 셀손실률을 갖는 것이다. 제한한 VB-LB는 이상적인 평균셀률 UPC 메카니즘 성능에 수렴해감을 알 수 있다.

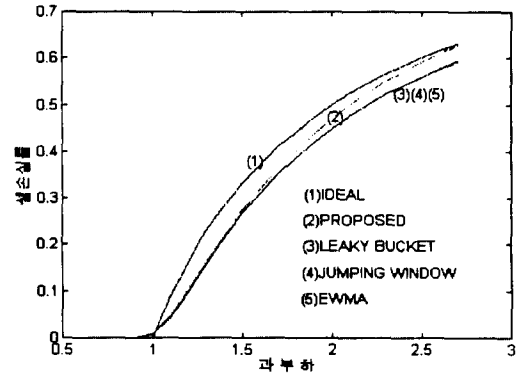


그림 10. 과부하에 대한 셀손실률  
Fig. 10. Influence of the overload on cell loss rate.

그림 11은 각 UPC 메카니즘의 반응시간을 알아보기 위한 실험이다. 이 때 overload factor는 1.5이다. X축은 발생하는 셀의 갯수를 Y축은 셀손실률을 의미한다. 약속보다 많은 셀들을 전송하는 경우 가능한 한 빨리 제재를 가하여 기존의 다른 호에 대해 악 영향을 최소화해야 한다. 그림 11에서 윈도우 방식인 EWMA는 반응시간이 다른 메카니즘에 비해 느림을 알 수 있으며 제한한 VB-LB가 기존의 메카니즘보다 반응시간이 빠름을 알 수 있다.

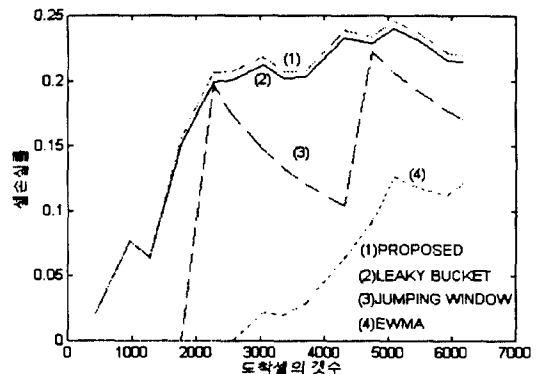


그림 11. 반응시간 비교  
Fig. 11. The comparison of reaction time.

## V. 결 론

본 논문에서는 가변적인 토큰 발생률을 갖는 VB-LB를 제안하였다. 협약을 위반하고 많은 셀들을 전송하는 호에 대한 기존의 UPC 메카니즘들의 셀손실률 및 반응시간은 이상적인 평균셀률 UPC 메카니즘의 성능에 미치지 못한다. 이와 같은 차이를 최소화하기 위해 제안된 VB-LB는 가상 버퍼내의 셀 갯수를 관측하여 토큰 발생률을 가변시킨다. 가상 버퍼내의 셀 갯수가 임계치를 넘었을 때 토큰 발생률( $\lambda_p$ )을 평균셀률( $\lambda$ )로 줄임으로써 위반 셀의 망 진입을 억제할 수 있었다. 시뮬레이션 결과, 제안한 VB-LB는 약속을 어기는 셀들에 대해 기존의 UPC 메카니즘들 보다 많은 셀들을 제거할 수 있었으며 이상적인 UPC 메카니즘의 성능에 수렴하는 셀손실률을 보였고, 반응시간 또한 기존의 UPC 메카니즘 보다 빠름을 알 수 있었다.

## 참 고 문 헌

- [1] J.J. Bae and T. Suda, "Survey of Traffic Control Schemes and Protocols in ATM Networks," Proceedings of the IEEE, vol. 79, no. 2, pp. 170-189, Feb. 1991.
- [2] X. Wu, I. Lambadaris, H. Lee and A.R. Kaye, "A Comparative Study of Some Leaky Bucket Network Access Schemes," IEEE ICC, pp. 1586-1591, 1994.
- [3] A.W. Berger and A.E. Eckberg, "A B-ISDN/ATM Traffic Descriptor, and Its Use in Traffic and Congestion Controls," IEEE GLOBECOM, pp. 266-270, 1991.
- [4] R. Krishnan and J.A. Silvester, "The Effect of Variance Reduction on the Performance of the Leaky Bucket," IEEE ICC, pp. 1974-1980, 1995.
- [5] T.F. Ibrahim and C.V. Chakravarthy, "The Enhanced Exponentially Weighted Moving Average Method: a Modified Flow Enforcement Technique for ATM Networks," IEEE ICC, pp. 1379-1383, 1994.
- [6] N. Yamanaka, Y. Sato and k. Sato, "Performance Limitation of Leaky Bucket Algorithm for Usage Parameter Control and Bandwidth Allocation Methods," IEICE trans. on commun., vol. E75-B, no. 2, pp. 82-86, Feb. 1992.
- [7] J.S. Turner, "New Directions in Communications(or Which Way to the Information Age)," IEEE Commun. Mag., vol. 24, no. 10, pp. 8-15, Oct. 1986.
- [8] E.P. Rathgeb, "Modeling and Performance Comparison of Policing Mechanisms for ATM Networks," IEEE J. Select. Areas Commun., vol. 9, no. 3, pp. 325-334, April 1991.
- [9] A.W. Berger, "Performance Analysis of a Rate-Control Throttle Where Tokens and Jobs Queue," IEEE J. Select. Areas Commun., vol. 9, no. 2, pp. 165-170, Feb. 1991.
- [10] M. Butto, E. Cavallero and A. Tonietti, "Effectiveness of the "Leaky Bucket" Policing Mechanism in ATM Networks," IEEE J. Select. Areas Commun., vol. 9, no. 3, pp. 335-342, April 1991.
- [11] A.W. Berger and W. Whitt, "The Pros and Cons of a Job Buffer in a Token-Bank Rate-Control Throttle," IEEE trans. on commun., vol. 42, no. 2/3/4, pp. 857-861, Feb. 1994.
- [12] I. Khan and V.O.K. Li, "A Traffic Control Mechanism for ATM Networks," IEEE GLOBECOM, pp. 1122-1126, 1993.
- [13] 조영주, 이동섭, 허영희, 심택수, 이주영, 박창욱, 김영우, 임인철, "ATM 망에서 가변적인 토큰 발생률을 갖는 사용 파라메타 제어 메카니즘," Proceedings of KITE, vol. 18, no. 1, pp. 194-197, June 1995



— 저 자 소 개 —



朴 昌 郁(正會員)

1964년 2월 10일생. 1982 년  
3월 - 1989년 2월 한양대학교  
전자공학과 공학사. 1990 년 3  
월 - 1992년 2월 한양대학교  
전자공학과 대학원 공학석사.

1992 년 3월 - 현재 한양대학교 전자공학과 대학  
원 박사과정. 주관심분야는 ATM traffic control  
및 ATM switch design 등임.

林 寅 七(正會員) 第 30卷 B編 第 2號 參照

현재 한양대학교 전자공학과 교수