

토큰 버킷을 적용한 다이내믹 스페이서 UPC 알고리즘

박 용 근*, 한 헌 수*

Dynamic Spacer UPC Algorithm Adopting Token Bucket for Traffic Control in ATM Network

YongGoun Park *, HernSoo Hahn *

요 약

ATM망에서 트래픽의 버스트니스를 완화하기 위한 UPC(Usage Parameter Control) 알고리즘을 제안한다. 기존의 다이내믹 스페이서는 그린 토큰이 축적되어 있을 경우 도착하는 셀을 스페이서와 상관없이 네트워크로 셀을 유입시키는 동적인 스페이서 기능을 수행함으로써 CDV(Cell Delay Variation)에 의한 셀을 위반셀로 구별하지 못하고 그대로 통과시키는 단점이 있다. 즉 스페이서 기능을 사용하지 않음으로써 버스트니스해 질 수 있다. 따라서 본 논문에서는 버스트니스를 완화하기 위한 토큰 버킷을 다이내믹 스페이서 이전에 사용함으로써 다이내믹 스페이서의 버스트니스를 완화시키는 토큰 버킷을 적용한 다이내믹 스페이서 UPC 알고리즘을 제안한다.

▶ Keyword : UPC Algorithm(UPC 알고리즘), ATM, Dynamic Spacer(다이내믹 스페이서), CDV(Cell Delay Variation : 셀 지연 변이), burstiness(버스트니스)

• 제1저자 : 박용근
* 숭실대학교 전자공학과

I. 서론

ATM이란 B-ISDN을 구현하기 위한 통신방식으로써 사용자 정보를 53바이트의 셀 형태로 전달된 후 원래의 정보로 환원하는 방식이다. ATM 채널의 용량은 ATM 셀의 수효로써 계량화된다[3].

UPC/NPC는 협상된 트래픽 이상의 셀이 망으로 입력되는 것을 제한함으로써 망의 폭주를 예방하는 역할을 수행한다. ATM의 특성상 셀 전송시 발생하는 지연이 일정하지 않기 때문에 셀 지연 변이가 발생한다. 즉 CDV(Cell Delay Variation)에 의해서 셀이 한쪽으로 쏠리는 현상이 나타남으로써 일시적으로 규정된 Cell Rate를 초과할 수 있다. Guillemin등은 이러한 이유로 UPC/NPC에서 셀간 간격을 제어하는 간격 제어기(Spacer-controller)를 제안하였다[2].

Dynamic Spacer의 경우에 오랜 침묵이후의 도착한 셀이 토큰이 비어 있지 않을 경우 네트워크로 곧바로 셀을 유입시킴으로써 네트워크쪽에 일시적으로 피크 셀 레이트를 초과할 수 있다.

그러므로 본 논문에서는 토큰 버킷 방식을 적용하여 출력쪽에 놓고 있는 호스트가 있을 경우 토큰을 축적함으로써 도착하는 셀이 토큰을 받고서 리키버킷을 거쳐서 네트워크에 진입하게 함으로써 버스트니스를 개선하고자 한다.

II. ATM망에서의 트래픽 제어

사용자는 네트워크에 트래픽 사용 허가를 요청하고 CAC(연결 허용 제어)는 네트워크의 자원 사용 현황을 파악하여서 요청된 대역폭을 허가할 수 있을 경우에 사용자에게 승인하게 된다.

UPC는 사용자가 허가된 대역폭 이상을 사용하게 될 경우를 감시하는 기능을 한다. 또한 CDV(Cell Delay Variation)에 의한 셀은 UPC에서 위반 셀을 발견하지 못할 수 있다. 따라서 스페이서에 의해서 일정 간격을 유지하여 셀을 방출하게 된다.

Congestion Control은 CAC에 의해서 허가된 대역폭을 네트워크 자원의 급작스런 변동(장애)으로 유지시켜 줄 수 없을 때 발생한다. 또는 UPC를 거친 셀이라도 CDV에 의한 셀

의 예측할 수 없는 통계적 셀률 변동에 의한 위반 셀을 감시할 수 없을 때 발생할 수 있다.

따라서 본 논문에서는 UPC에서의 감시 기능과 동적인 스페이서 기능을 가진 동적인 다이나믹 스페이서의 버스트니스 문제점을 개선하고자 한다.

III. 기존의 UPC 기법

3.1 Leaky Bucket

셀 발생시 1 증가하고 일정 기간 마다 1씩 감소시키는 계수기를 가지고 있다. 셀이 발생할 때 계수기의 최대값 Q보다 크면 그 셀을 폐기시키고 Q보다 작으면 통과시킨다[5].

3.2 토큰 풀(Pool)을 이용한 Leaky Bucket

셀은 대기 행렬로 들어가고, 도착한 셀은 토큰을 받는 경우에만 망으로 들어간다. 토큰이 없는 경우에는 입력 버퍼에서 대기한다. 도착한 셀은 토큰 풀에 토큰이 있을 경우만 토큰을 가지고 네트워크로 진입할 수 있고 토큰풀이 비어 있으면 버퍼에서 대기해야 되며 만약 버퍼가 다 차면 위반으로 판명된다[5].

3.3 스페이서와 표기를 이용한 Leaky Bucket

토큰풀을 이용한 리키버킷 기법에다가 스페이서와 위반 셀들에 대한 표기 기능을 추가한 모델이다[5].

3.4 Dynamic Spacer

스페이서와 표기를 이용한 리키버킷과 다르게 스페이서를 동적으로 이용한다. 즉 토큰을 받은 셀을 일정 간격으로 지연시키는 것을 개선하여 토큰풀에 그린 토큰이 비어 있을 경우에만 스페이서 기능을 사용하고 그렇지 않고서 토큰풀에 그린 토큰이 차 있을 경우는 스페이서 기능을 사용하지 않고서 그냥 통과시킨다. 이렇게 함으로써 셀 위반율과 셀 지연을 개선할 수 있지만 스페이서를 적용하지 않는 구간에서 셀이 버스트하게 전송될 수 있다. 즉 셀 지연 변이(CDV)에 의한 위반 셀을 구별할 수가 없다. 그러므로 위반셀을 그냥 통과시킬 수 있다[1].

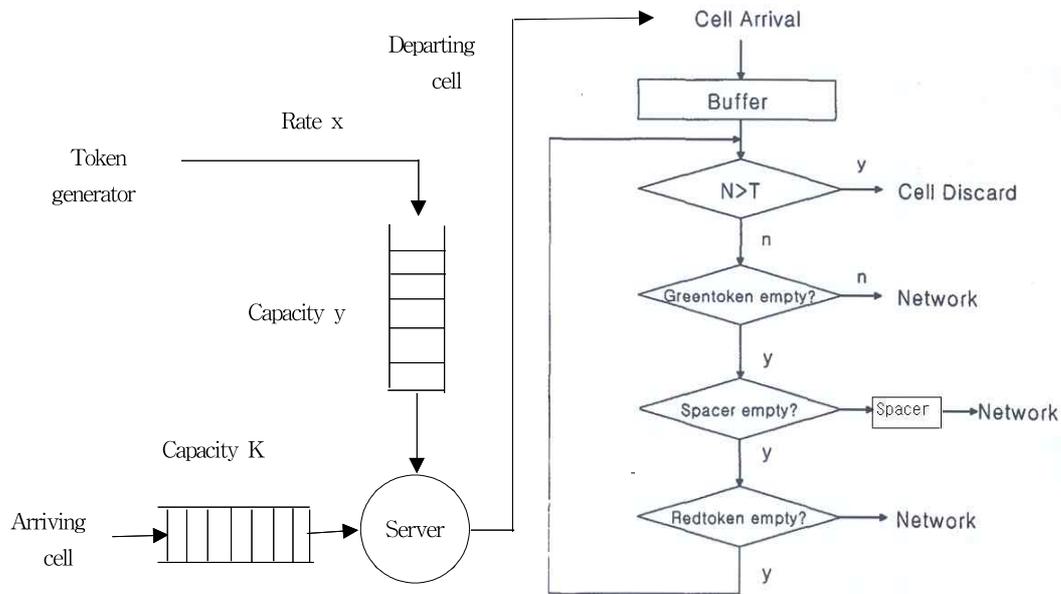


그림 1. 제한하는 토큰 버킷을 적용한 Dynamic Spacer의 UPC 알고리즘

IV. 토큰버킷을 적용한 동적 스페이스서의 UPC 알고리즘

동적 스페이스서를 거치지 않고서 곧바로 그린 토큰을 받아서 네트워크로 진입함으로써 버스트니스를 유발하는 단점을 개선하고자 토큰 버킷을 동적 스페이스서 이전에 적용하여 셀간 간격을 토큰 발생율과 토큰 축적량을 조절함으로써 버스트니스를 개선하고자 한다.

먼저 토큰 버킷의 기본적인 원리는 다음과 같다. 토큰 발생기는 초당 x 의 비율로 토큰을 생성하여 최대 용량이 y 인 토큰 버킷에 놓는다. 스스로부터 도착한 셀들은 최대 용량이 K 인 버퍼에 놓이게 된다. 이때 서버를 통해서 하나의 셀을 전송하기 위해서는 하나의 토큰이 버킷에서 제거되어야 한다. 만약 토큰 버킷이 비어 있다면, 그 셀은 다음 토큰을 기다리며 큐에 쌓이게 된다. 이 방법의 결과로 큐에 쌓여 있는 셀이 있으면서 버킷이 비어 있다면 이 셀들이 다 전송될 때까지 셀 지연 변동 없이 초당 x 셀의 유연한 흐름으로 전송된다. 따라서 토큰 버킷은 어느 순간 집중 생성되는(burst) 셀이 순탄하게 전송될 수 있도록 한다[8].

그러나 토큰 버킷의 문제점은 축적되는 토큰량 y 의 양이 너무 컸을 경우에는 최대 버스트 간격을 통제하는데도 불구하고 그것으로 인한 또 다른 버스트가 생성되거나 망에 폭주를 일으킨다는 것이다.

그러므로 본 논문에서는 축적되는 토큰량 y 의 값을 최소로 유지시켜 준다.

동적 스페이스서에서 버퍼는 선입선출의 큐로 이루어져 있으며 셀이 토큰 버킷을 거쳐서 도착할 때에는 이전 셀이 스페이스서를 통과할 때까지 버퍼에 대기한다. 버킷은 그린 토큰풀, 레드토큰풀로 이루어지고, 그린토큰풀은 유효 셀을 생성하고, 레드토큰풀은 위반 셀을 생성한다. 버퍼로 입력되는 셀의 수가 T 보다 큰 경우에 도착한 셀은 두가지의 형태로 나누어진다. 도착한 셀의 CLP가 1인 경우로 이것은 바로 폐기되고, 도착한 셀의 CLP가 0인 경우는 버퍼 내부에 우선 순위가 낮은 (CLP=1) 셀이 존재하는가를 검색하여, 존재하면 버퍼 내부에서 해당 셀을 제거하고 도착한 셀을 버퍼에 저장하며, 존재하지 않으면 도착한 셀을 폐기한다. 그림1을 보면, 그린 토큰이 차 있을 경우 스페이스서 기능없이 곧바로 네트워크로 보내고 그린토큰이 비어 있고 스페이스서가 차 있을 경우 스페이스서를 거쳐서 네트워크로 셀을 유입시킨다. 그렇지 않고 레드토큰이 차 있을 경우 레드토큰을 통해서 네트워크로 유입시킨다[1].

그림 1 우측 부분에서 보듯이 3.3 스페이서와 표기를 이용한 리키 버킷과의 차이점은 스페이서를 그린 토권이 차 있을 경우에는 스페이서 기능을 일률적으로 사용하는 것과

는 달리 다이내믹 스페이서는 스페이서 기능없이 곧바로 셀을 네트워크로 진입시킨다는 점이다.

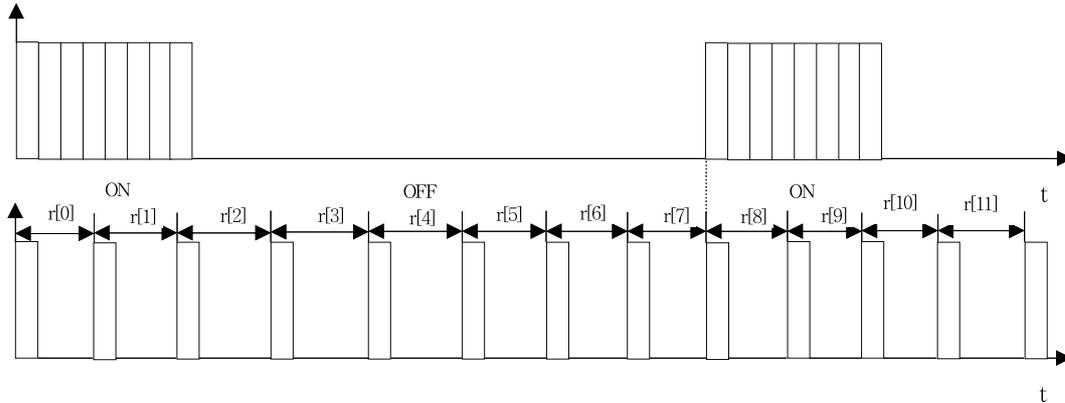


그림 2 셀 버스트니스의 원화

이렇게 할 경우 큰 버스트가 도착할 때 데이터 손실없이 더 빠르게 출력할 수 있다는 장점이 있다. 하지만 이 경우의 단점으로는 버스트한 트래픽을 곧바로 네트워크로 진입시킴으로써 폭주를 유발할 수 있다는 점이다.

그러므로 본 논문에서는 다이내믹 스페이서의 기능을 유지시키면서 버스트 트래픽의 문제를 해결하고자 다이내믹 스페이서 이전에 토큰 버킷을 적용하는 것이다.

$$r[n] = (n+2)\text{번째 셀 도착시간} - (n+1)\text{번째 셀 도착시간} \dots\dots (5.1)$$

이다. 그리고 피크 셀 레이트, 미니멈 셀 레이트, ON 구간의 평균 셀 레이트를 보면,

$$\text{Peak Cell Rate} = \frac{1}{\text{최소 셀 도착사이 간격}} \dots\dots (5.2)$$

ON구간의

$$\text{Minimum Cell Rate} = \frac{1}{\text{최대 셀 도착사이 간격}} \dots\dots (5.3)$$

ON구간의

$$\text{Mean Cell Rate} = \frac{1}{\text{평균 셀 도착사이 간격}} \dots\dots (5.4)$$

이다.

V. 시뮬레이션

5.1 트래픽 소스 모델

트래픽 소스 모델로 ON/OFF 소스 모델을 사용한다. 그림2를 보면 ON/OFF 모델의 트래픽 소스 모델의 ON구간의 버스트니스를 토큰 버킷을 적용해서 셀 간격을 넓혀줘서 OFF구간에 이르도록 함을 시뮬레이션을 통하여 보여주고 한다. C언어를 사용하여, ON기간의 셀 발생 사이 간격을 랜덤하게 발생시켜 주었다. ON구간의 셀 발생 사이 간격을 수식을 통하여 보면,

$$r[0] = 2\text{번째 셀 도착시간} - 1\text{번째 셀 도착시간}$$

$$r[1] = 3\text{번째 셀 도착시간} - 2\text{번째 셀 도착시간}$$

5.2 시뮬레이션 및 결과 분석

그림4에서는 r[19]까지는 제안하는 다이내믹 스페이서로만 표시되어 있지만 다이내믹 스페이서와 같이 표시된 것으로 보면 된다. 엑셀로 그래프를 그릴 당시 같이 표시되지 않

고서 제안하는 다이내믹 스페이서로만 표시되어서 그렇게 되었다.

그림4는 토큰 버킷의 토큰 축적량이 2이므로 그림2를 참조함으로써 셀 간격으로는 $r[19]$ 까지이므로 다이내믹 스페이서와 제안하는(토큰 버킷을 적용한) 다이내믹 스페이서의 $r[19]$ 까지는 셀 간격이 동일하다. 하지만 이유로는 셀 침묵기간인 OFF 이후 셀 발생기간인 ON기간동안 토큰 축적량까지는 축적된 토큰을 받아서 셀은 도착하는 셀 간격으로 곧바로 네트워크로 진입하기 때문이다.

그림5는 토큰 축적량을 0으로 제한함으로써 토큰 발생률에 따라서 제안하는 다이내믹 스페이서가 토큰 발생률에 따라서 셀 간격 시간이 조절됨을 알 수 있다. 여기에서 최대 셀 발생 사이 간격 시간과 토큰 발생 사이 간격 시간이 같음을 볼 수 있다.

결과적으로 그림4와 그림5를 통해서 토큰 버킷을 적용한 다이내믹 스페이서는 버스트니스한 셀 사이 간격을 넓혀 주어서 스무딩 기능을 수행할 수 있음을 알 수 있다.

토큰 버킷에서 토큰이 축적되지 않고 적절하게 발생할 경우 그것을 받고서 다이내믹 스페이서에 도착할 경우 그린토큰이 비어 있지 않을 경우 스페이서를 거치지 않고서도 네트워크로 진입할 경우에도(이미 토큰 버킷을 통해 버스트를 완화했기 때문에) 버스트니스가 완화됐음을 그림 5를 통하

여 볼 수 있다.

토큰 버킷이 셀의 도착없이 계속해서 크레딧(토큰)을 축적할 경우 그 이후 도착한 셀이 다이내믹 스페이서에 도착할 경우 $N>T$ 일 경우 셀을 폐기해야 한다. 그러므로 토큰 버킷의 경우 Credit(토큰)을 축적할 경우 축적하는 양을 제한해야 된다. 이렇게 할 경우 $N>T$ 가 아닐 경우, 미리 도착한 셀이 없을 경우이므로 당연히 그린토큰이 비어 있지 않으므로 스페이서를 거치지 않고 네트워크로 진입시킨다. 토큰 버킷의 축적된 토큰량 y 의 범위내에서의 셀의 전송의 경우 일시적인 버스트를 완화할 수는 없다. 그러므로 축적되는 y 의 범위를 0으로 한 경우의 그림 5는 그림 4와 비교했을 경우 초기 버스트니스 완화의 결과로 볼 수 있다. 다른 관점에서 볼 경우 셀의 버스트니스는 다이내믹 스페이스의 경우 지연(Delay)을 감소시킬 수 있다고도 볼 수 있다. 적절한 조화가 필요한 부분이다.

$N>T$ 가 아닐 경우 토큰 버킷이 Credit(토큰)을 축적할 경우가 다이내믹 스페이서의 그린토큰의 축적된 양보다 클 경우 그린토큰을 모두 써버린 이후에 토큰버킷으로부터 다이내믹 스페이스로 도착하는 셀은 그린토큰이 비어 있으므로 스페이서가 있을 경우에 스페이서에 의해 네트워크로 진입한다.

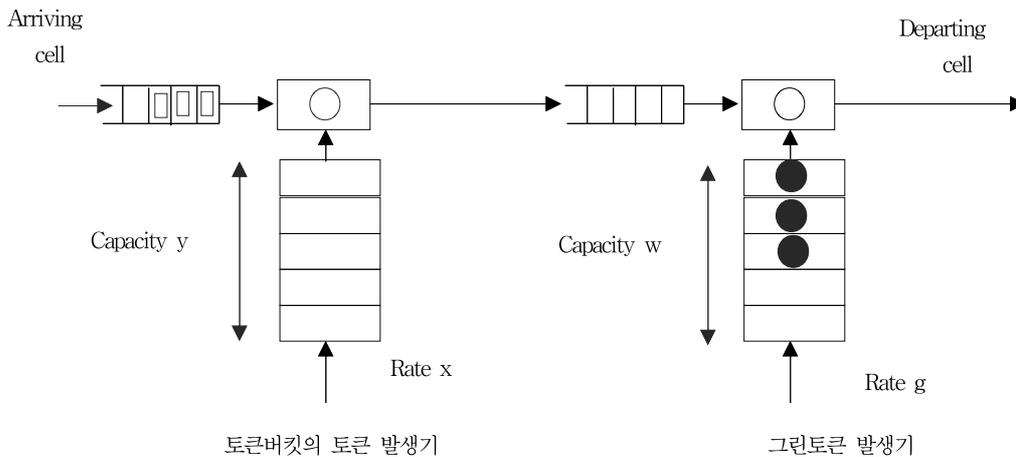


그림 3 토큰 버킷의 토큰발생률이 그린토큰의 발생률보다 작거나 같을 경우

그림 1을 참조하면 토큰 버킷의 경우, 전송률을 낮출 경우 (토큰 발생률을 낮출 경우) 토큰 버킷이 21개 축적할 경우 $r[0]$ 에서 $r[19]$ 까지는 랜덤하게 도착한 셀들이 이미 축적되어 있는 토큰을 받아서 곧바로 다이내믹 스페이스로 출발하고 그 이후 도착하는 $r[20]$ 에서 $r[40]$ 까지는 토큰 버킷의 토큰 발생률에 따라서 일정하게 다이내믹 스페이스로 출발한다. 그림 5는 토큰 버킷이 하나도 축적되지 않을 경우이고 $r[0]$ 에서 $r[40]$ 까지는 모두 토큰 버킷의 토큰 발생률에 따라 다이내믹 스페이스의 그린토큰을 받아서 네트워크로 진입하게 된다.

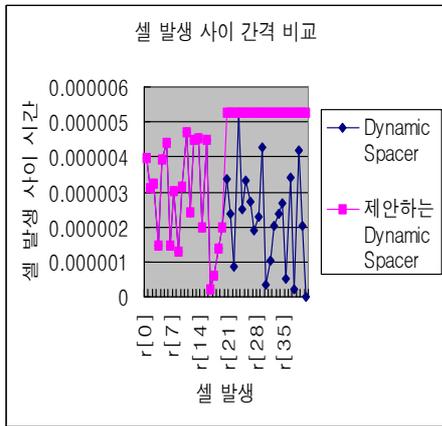


그림 4. 오랜 셀 발생 침묵이후의 다이내믹 스페이스와 토큰 버킷을 이용한 다이내믹 스페이스의 셀 발생 사이 간격 비교

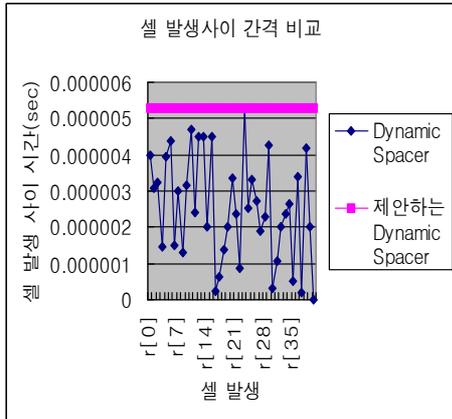


그림 5. 오랜 셀 발생 침묵이후의 다이내믹 스페이스와 축적되는 토큰량을 0으로 제한한 토큰 버킷을 이용한 다이내믹 스페이스의 셀 발생 사이 간격 비교

토큰 버킷을 거친 이후 다이내믹 스페이스의 그린토큰의 경우도 토큰 버킷의 발생률보다 크거나 같다고 보았을 경우 그린토큰이 차 있으므로 그린토큰을 받아서 토큰 버킷을 거친 셀을 토큰 버킷의 출력단과 같은 비율로 셀을 네트워크로 진입시킨다. 토큰 버킷을 거치지 않았을 경우의 다이내믹 스페이스의 그린토큰의 경우 그린 토큰이 42개 이상 차 있을 경우 CDV(Cell Delay Variation)에 의해 랜덤하게 발생한 셀들이 네트워크로 곧바로 진입하게 된다.

그림 4, 5에서 보듯이, 토큰 버킷의 축적량을 작게 할수록 랜덤한 버스트니스 셀 발생 사이 시간 간격을 최대 셀 간격으로 넓혀 줄 수 있음을 볼 수 있다. 즉 토큰 버킷의 토큰 발생률을 최대 셀 발생 사이 간격으로 맞추어 줌으로써 토큰 발생 간격으로 넓혀 줄 수 있음을 볼 수 있다.

토큰 발생기를 그린토큰 발생기보다 토큰 발생률을 낮추었을 경우 또한 OFF구간 동안 축적되는 토큰량 y 를 0으로 했을 경우 토큰 발생기의 토큰 발생률에 따라서 네트워크로 직접 유입되더라도 일정한 간격을 유지시켜 줄 수 있음을 그림 5에서 볼 수 있다.

VI. 결론

다이내믹 스페이스 이전에 토큰버킷을 추가로 적용함으로써 다이내믹 스페이스에서 셀 발생 침묵이후의 그린토큰이 축적되어 있을 경우의 네트워크의 셀의 버스트니스 전송을 완화시켜 줄 수 있음을 시뮬레이션 그림 4, 5에서 볼 수 있었다.

그림 4, 5의 결과의 차이는 토큰 버킷을 21개 축적되었을 경우와 토큰 버킷의 축적량이 0개이었을 경우의 차이점이다.

본 논문에서는 시뮬레이션을 ON-OFF 소스 모델을 이용하여 셀 발생 ON 구간에서 버스트니스를 완화하여 OFF구간에 이르도록 했다. 결과적으로 버스트니스를 완화시켜 일정하게 셀 간격을 넓혀 줬음을 알 수 있다. 또한 필요로 하는 구간에서의 값을 얻기 위해서 몇 가지 조건을 두었다.

앞으로의 연구에 있어서 전 구간에 걸쳐서 시뮬레이션을 할 필요가 있을 것 같다.

참고문헌

- [1] 김우완, "ATM망에서 트래픽 제어용 동적 지연기를 적용한 개선된 UPC 알고리즘" 한국 멀티미디어 학회 논문지, 8권 2호, 2005.(pp. 192-200)
- [2] 윤석현, "ATM Network에서 트래픽 특성에 의한 셀 셰이핑" 한국 컴퓨터정보학회 논문지, 15권 1호, pp17-23, 2007.6.
- [3] 송헌춘·김선형, "초고속정보통신망" 1판, 양서각, 1999.
- [4] 김영호, "ATM망에서 VBR에 관한 PCR의 차원 감시 알고리즘에 관한 연구" 홍익대학교 대학원 전자공학과 전자 전공 석사학위 청구 논문, 1995.6.30.
- [5] 이동호, "ATM 멀티플렉서의 통합 서비스에 대한 트래픽 제어에 관한 연구" 광운대학교 '94 통신학술 연구과제 1995.3.
- [6] 성단근, "ATM망에서의 트래픽 제어에 관한 연구" 한국 과학 기술원 '한국전자통신연구소 1992.5.31.
- [7] 이병기·강민호, "광대역 네트워크" 1판, 교학사, 2003.
- [8] 김종상·전화숙 공역, "데이터 통신및 컴퓨터 통신" 6판, 사이텍 미디어, 2001
- [9] 김한규·박동선·이재광·옴김, "데이터 통신과 네트워킹" 1판, (주)교보문고, 2005