

BISDN ATM 환경에서의 트래픽 제어 방식에 관한 연구 - 가변 크기 창을 사용한 동적 사용 파라미터 제어 -

正會員 李 紀 奉*, 宋 周 錫**, 이 준 원***

A Study on Traffic Control in BISDN(ATM Networks)
- Dynamic Usage Parameter Control using Variable Size Windows -

Ki Bong Lee*, Ju Suck Soung**, Jun Won Lee*** Regular Members

이 논문은 1994년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음

要 約

본 논문은 움직이는 창을 사용하여 일정시간의 평균 비트율을 계산한 뒤 이 결과에 따라 Leaky Bucket의 Leaky율을 변화 시키는 새로운 사용 파라미터 제어 알고리즘을 제안한다. 기존의 알고리즘에서 제공하는 움직이는 창은 고정된 크기를 가지며 시간창이 허락하는 셀의 한계값을 넘어가는 셀은 폐기한다.

본 논문이 제안하는 제어 모델은 적은 수의 창을 가지고 평균 비트율을 위반하는 트래픽을 정확하게 제어하기 위해 창의 한계값을 동적으로 관리하는 특성을 가진다. 또한 기존의 모델과는 달리 창은 평균 비트율의 위반 여부만을 판단하는 역할을 하며 위반 트래픽에 대한 제어는 Leaky율의 조절을 통해 하게 된다.

버스티한 On/Off 트래픽 소스를 가정하고 시뮬레이션 한 결과 기존의 사용 파라미터 제어 알고리즘인 Leaky Bucket이 평균 비트율 제어를 위해 큰 수의 계수기 한계값을 필요로 하며 위반 트래픽에 대해 느린 반응시간을 보이는데 반해 제안하는 알고리즘은 작은 수의 계수기 한계값과 위반 트래픽에 대해 빠른 반응 시간을 보장한다.

ABSTRACT

In this paper, we propose UPC algorithm which uses moving windows to monitor the mean bit rate of input traffic. The proposed algorithm observes and computes the mean bit rate during some time period and then adjusts the leaky rate of leaky bucket by the computed result. The windows used in existing algorithms have fixed size windows and discard cells which violate the time window's limits. In order to monitor the traffic which violates the mean bit rate exactly, the proposed algorithm uses the small

* (주)데이콤

** 연세대학교 전산학과 부교수

*** 한국전자통신 연구소 초고속망 연구 실장

論文番號 : 94185

接受日字 : 1994年 7月 8日

number of windows and manages the windows dynamically.

This algorithm is different from existing algorithms because windows just check whether traffic violates the mean bit rate or not and the control of violating traffic is achieved by adjusting the leaky rate.

Assuming bursty On/Off traffic source, our simulation shows the following results. Small number of counter limits enough to control the mean bit rate in this input traffic, it can carry out control more generally than any other algorithm does.

1. 서 론

B-ISDN에서 제공하는 서비스는 B-ISDN의 광대역 능력으로인하여 기존망과는 다른 특징을 가진다. 즉 고대역, 항등 비트율 및 가변 비트율의 서비스를 제공하며 다중 매체 통신과 사용자와 망간의 서비스 품질 협상을 가능하도록 하고 있다. 이와 같이 다양한 서비스의 제공을 위해 B-ISDN은 비동기 전송 모드(ATM)를 전송방법으로 채택하고 있다. ATM은 정보를 셀(cell)이란 고정된 크기의 정보단위로 분할하여 전송하기 때문에 대역폭 할당에 있어 유연성(flexibility)을 제공하고, 통계적 다중화 방법을 이용하여 대역폭 사용에 있어 효율을 높이고 있다. 그러나 이러한 통계적 다중화 방법은 소스 트래픽 패턴을 매우 예측하기 어렵게 만들고, 망을 과잉밀집의 상태로 만들 위험을 안게 된다. 또한 각각의 트래픽 흐름은 셀 손실율, 전송 지연, 지연 변이등의 다양한 서비스 품질(QoS)를 요구하며, 망은 이러한 서비스 품질 기준을 만족 시켜 주어야 한다.[11]

이상과 같이 복잡한 트래픽환경을 가지는 ATM 망에서 망자원의 효율적 운용과 다양한 서비스 품질 기준을 만족시키기 위해서는 트래픽의 변화를 예측하고 과잉밀집을 미리 방지하는 등의 트래픽 제어 기술이 요구된다.

ATM망은 사용자와 망이 요구하는 성능을 만족시키기 위하여 호 수락 제어(Call Admission Control), 사용 파라미터 제어(Usage Parameter Control), 우선 순위 제어(Priority Control)등의 트래픽 제어를 제공하고 있다.

호 수락 제어는 ATM망의 호 설정(Call Setup)단계에서 사용자와 망사이의 협의에 의해 이루어지게 된다. 사용자가 신고하는 트래픽 특성값과 서비스 품질 기준 그리고 망의 상태를 바탕으로 새로이 연결을 요구하는 호와 이미 서비스중인 호 모두의 서비스 품질을 만족하고 동시에 망자원을 효율적으로 사용할 수 있도록 호의 수락 여부를 결정하는 것이다.

일단 호가 설정된 이후에는 호 설정시 협약된 트래픽 파라

미터를 사용자가 위반하지 않도록 사용자 접면(User Network Interface)에서 트래픽을 감시하는 기능이 필요하다. 이를 사용 파라미터 제어라고 한다.

사용 파라미터 제어는 트래픽 파라미터를 위반하는 셀을 가능한 한 빨리 감지해야 하며 파라미터를 위반하는 셀에 대해서는 폐기나 Tagging등의 반응을 취해야 한다. 또한 사용 파라미터 제어모델은 가상 채널(VC)이나 가상 경로(VP)상에 설치되기 때문에 간단한 구조를 가지고 있어야한다. 지금까지 제안된 대표적인 사용 파라미터 제어 모델은 Jumping Window[4]와 Leaky Bucket[4]등이 있다.

망내에서 과잉밀집이 발생하면 셀 손실율이 커지고 전송 지연도 커지게 된다. 그런데 B-ISDN에서는 각 서비스마다 요구되는 서비스 품질 수준이 다르므로 이를 만족시키기 위해 과잉밀집 발생시에 셀 손실율 요구치가 낮은 서비스에 속한 셀들을 우선적으로 폐기시키고 전송 지연 요구조건이 엄격한 서비스의 셀을 먼저 처리 할 필요가 있다. 그래서 우선 순위 제어가 필요하게 된다. 우선 순위 제어는 크게 두가지로 나눌 수가 있는데 스케줄링에 의한 시간 우선 제어와 선택적 셀 폐기에 의한 공간 우선 제어가 있다.

본 논문은 사용 파라미터 제어에 관한 기존 알고리즘을 소개하고 기존 사용 파라미터 제어 알고리즘의 문제를 해결 할 수 있는 새로운 알고리즘을 제안한다.

기존의 사용 파라미터 제어 알고리즘은 버스티한 특성을 가지는 트래픽의 최대 비트율제어에는 비교적 우수한 성능을 제공하지만 평균비트율의 제어에는 효과적이지 못한 것으로 알려져 있다.[4]

버스티한 트래픽의 평균비트율을 측정하기 위해서는 긴 측정시간을 제공해야 하지만 그 기간중 필요한 버퍼 크기나 계수기 한계값이 비현실적으로 커져 실질적 구현이 어려워지며, 한계값이 작으면 정상 트래픽에 대해 잘못된 셀 손실율을 일으켜 서비스 품질을 저하시킨다. 또한 긴 측정 시간은 느린 반응시간을 가지게 되며 협약된

트래픽 파라미터를 위반하는 트래픽에 빠른 동작을 취할 수 없게 된다.

본 논문에서 제안하는 평균 비트율 제어 모델은 움직이는 창을 이용하여 일정시간의 평균 비트율을 계산한 뒤 이 결과에 따라 Leaky Bucket의 Leaky율을 변화시키도록 한다. 기존의 사용 파라미터 제어 알고리즘이 평균 비트율 제어를 위해 큰 수의 계수기 한계값을 필요로 하고 늦은 반응 시간을가지는데 비해 제안하는 알고리즘은 작은 수의 계수기 한계값을 가지며 위반 트래픽에 대하여 빠른 반응 시간을 보장한다. 또한 화상, 음성 등 입력 트래픽의 종류에 관계없이 일정한 크기의 계수기 한계값과 일정한 수의 움직이는 창을 가지므로 보다 일반적인 제어를 행할 수 있다.

본 논문의 2장은 기존의 사용 파라미터 제어 알고리즘을 소개하고 3장은 제안 하는 평균 비트율 제어 방법을 설명한다. 4장은 시뮬레이션 결과와 분석을 5장은 결론을 기술하고 있다.

II. 기존의 사용파라미터 제어 알고리즘

사용 파라미터 제어는 협약한 파라미터의 준수여부를 감시하며 협약한 트래픽 파라미터를 위반하는 트래픽에 대해 적절한 조치를 취함으로써 망의 효율을 높이는 역할을 한다.

지금까지 제안된 사용 파라미터 제어알고리즘은 시간 창을 이용한 방법, Leaky Bucket과 그 변형, 간격 제어를 이용하는 Virtual Scheduling Algorithm(VSA)(6)등이 있다.

1. Jumping Window

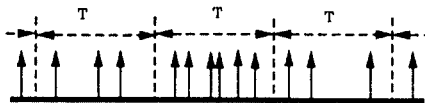


그림 1. Jumping Window
Fig. 1. Jumping Window

일정한 시간 간격(T)의 창을 설정하고 하나의 창안에 수용가능 셀의 갯수(N)를 정한 후, 창구간 동안의셀의 갯수를 세어 정해진 한계값을 넘으면 나머지 셀은 버린다. 새로운 시간 창은 앞의 창이 끝나면 바로 시작된다.

또한 셀의 갯수를 세는 계수기도 새로운 시간 창의 시작과 함께 0 이 된다.

2. Leaky Bucket

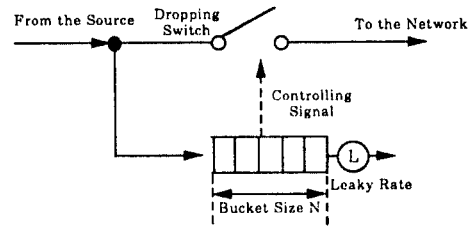


그림 2. Leaky Bucket
Fig. 2. Leaky Bucket

Leaky Bucket은 셀 입력이 있을 때마다 계수기 값을 하나씩 증가시켜 현재 사용되고 있는 사용 파라미터를 감시한다. 또한 계수기는 입력 셀과는 독립적으로 일정한 Leaky율로 계수기 값을 감소시키고, Leaky율은 대상 트래픽의 평균 비트율, 최대 비트율을 고려하여 결정된다. 만약 입력 비트율이 계수기의 Leaky율을 넘어설 때는 계수기 값은 계속 증가하게 되고, 계수기 값이 한계값에 도달하게 되면 그 다음에 입력되는 셀들은 계수기 값이 다시 한계값 이하로 떨어질 때까지 모두 폐기된다.

3. Virtual Scheduling Algorithm

ATM망에서 셀은 전송도중 다중화 단계물거치거나 OAM셀의 삽입등에 의하여 셀 지연 변이(cell delay variation)가 생기게 된다. 이는 사용자가 호 설정시 신고한 주기로 셀을 발생시킨다 하여도 셀 지연 변이가 일어나 셀이 전송주기가 변화 될 수 있다는 것을 의미한다.

셀 지연 변이로 연속되는 셀 사이의 간격이 좁아지는 경우 이는 협약된 최대 비트율을 위반하는 결과가 되어 협약된 파라미터에 순응하는 셀에 대한 폐기율이 높아지는 결과를 초래하게 되어 요구하는 서비스 질을 보장할 수 없게 된다. 따라서 입력 트래픽의 셀스트림 구조를 적절히 변화시킬 수 있는 셀 스페이싱 기능을 가진 사

용 파라미터 제어장치가 필요하다.

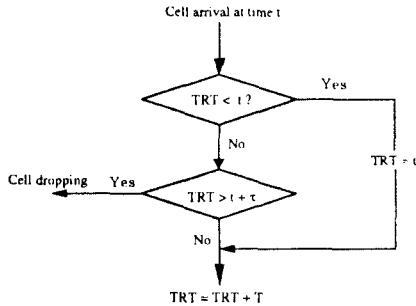


그림 3. Virtual Scheduling Algorithm
Fig. 3. Virtual Scheduling Algorithm

그림 3은 셀 지연 변이를 고려한 Virtual Scheduling Algorithm(VSA)을 보이고 있다. T 는 호 설정시 망과사용자 사이에 협의된 셀간 최소 전송간격을 의미하고 τ 는 셀 지연 변이의 영향을 고려하기 위해 도입된 허용 오차이다.

셀 전송 시간은 이론적인 셀 전송시간 TRT에 의해 관리된다. 즉 TRT는 VSA에서 셀 전송 시간이 스케줄링된 때 그 전에 셀이 전송되어서는 안되는 시간이다. 현재 시간 t 에서 셀이 도착 하였다고 할때 TRT가 t 보다 크면 셀이 예상 도착 시간보다 더 빨리 도착한 경우이므로 VSA는 이 셀의 폐기 여부를 판단 하여야 하는데 허용 가능한 셀 지연 변이를 고려하더라도 TRT가 더 크게 되면 협상된 최소 전송 간격을 위반한 경우이므로 이 셀은 폐기 하게 된다.

현재 셀 도착 시간 t 가 TRT보다 크면 이 셀은 예상 도착시간보다 늦게 도착한 경우이므로 VSA는 항상 이 셀을 받아 들인다. 이 경우 다음 셀이 최소 전송간격을 위반해 들어올 가능성이 높으므로 TRT를 현재시간 t 로 갱신된다.

Ⅲ. 제안 알고리즘

1. 기존 사용 파라미터 제어 알고리즘의 문제점 제시

Jumping Window와 같이 창을 가지는 제어 알고리즘은 창 안에서 허용되는 최대 셀의 수 N 과 창의 시간 간격 T 의 비로 트래픽을 감시한다. Leaky Bucket의 경우는 계수기 한계값과 Leaky율이 트래픽의 평균 비

트를 제어한다.

버스티 하면서 평균 비트율을 어기지 않는 정상 트래픽에 대해 창을 가지는 알고리즘은 높은 셀 손실율을 보인다. 버스티한 특성을 가지는 정상 트래픽에 대해 창을 가지는 알고리즘이 셀 손실율을 줄이고자 하는 경우 창이 허용하는 셀 갯수의 한계값을 높이거나 시간간격 T 를 늘려주면 된다. 그러나 이 경우에도 성능의 향상은 미미하며 측정시간을 크게 늘려 셀 손실율을 줄이게 되면 협약된 파라미터를 위반하는 트래픽에 대해 느리게 반응하게 된다.

Leaky Bucket에서 셀 손실율을 줄이기 위해서는 계수기 한계값을 올려 줌으로 셀 손실율을 줄일 수 있다. 그러나 이 경우 10^5 의 셀 손실율을 얻기위해서 10^5 이상의 셀 버퍼나 계수기 한계값이 필요하며 실제 ATM이 요구하는 10^9 의 셀 손실율을 얻기 위해서는 비현실적인 버퍼수나 계수기 한계값으로 인하여 구현이 어려워지게 된다. 또한 큰 계수기 한계값으로 인한 긴 측정시간은 협약한 파라미터를 위반하는 트래픽에 매우 느린 반응시간을 가지게 되어 망을 과잉밀집 상태에 빠뜨릴 우려가 있다.

Leaky Bucket은 창을 가지는 알고리즘에 비하여 우수한 성능을 보임에도 불구하고 위에 언급한문제에 인하여 평균 비트율 제어에는 적합하지 못한것으로 평가 되어지고 있다.

간격 제어를 가지는 VSA의 경우 도착하는 셀의 간격을 기반으로 사용 파라미터 제어를 행하기 때문에 최대 비트율 제어에는 적합하지만 일정시간 동안의 입력 비트율을 계산해야하는 평균 비트율 제어에는 적합하지 않다.

2. 트래픽 소스 모델

제안하는 알고리즘과 기존의 사용 파라미터 제어 알고리즘 모델을 비교하기 위하여 그림4와 같은 "Two phase On/Off model"을 사용한다. 이 모델은 "Packetized Voice", "Still Picture", "Interactive Data Service"등 실제 트래픽과 유사한 성질을 가지고 있다.

또한 이모델은 On-Off의 두가지 상태를 가지며 On상태(버스트 구간)에서는 협약한만큼의 최대 비트율로 트래픽을 발생하고 Off상태(침묵 구간)동안은 트래픽의 발생없이 침묵한다.

버스트 구간은 그림 4와 같이 지수분포(exponential)를 가

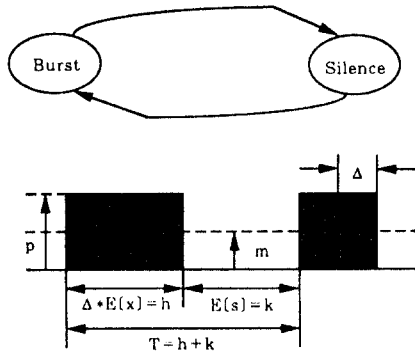


그림 4. ON/OFF 소스 모델
Fig. 4. On/Off Source Model

지고 평균 h 의 시간을 보내며 침묵기간은 지수분포(exponential)를 가지고 평균 k 의 시간을 보낸다. On/Off모델은 다음과 같은 파라미터를 가진다.

- 1) 최대 비트율 : $[p]$
- 2) 평균 버스트 구간 : $[h]$
- 3) 평균 침묵 구간 : $[k]$
- 4) Burstiness : $[\beta = (h+k)/h = p/m]$
- 5) 평균 비트율 : $[m = h * p / (h+k)]$

평균 비트율이나 Burstiness등의 파라미터는 최대비트율, 평균 버스트기간, 평균 침묵 기간등의 파라미터를 가지고 구할 수 있다.

그림 4에서 Δ 는셀간 도착 간격이며 $E[x]$ 는 평균 버스트 기간 동안의 평균 셀 수 이다.

3. 계수기 한계값과 Leaky율과의 관계

그림 5는 최대 비트율 32Kbps, 평균 비트율 11.2Kbps를 가지는 음성 데이터의 경우에 10^{-5} 의 셀 손실율을 얻기 위한 계수기 한계값과 Leaky율과의관계를 보이고 있다.

그림 5에서 사용한 C 값은 원하는 Leaky율을 얻기 위해 평균 비트율에 곱해지는 제어 인자값이다. 계수기 한계값이 작아질수록 낮은 셀손실율을 얻기위해 Leaky율을 올려주어야 함을 알 수 있다. Leaky율이 올라가면 작은 수의 계수기 한계값을 가지고도 낮은 셀 손실율을 보장할 수 있지만 평균 비트율을 위반하는 트래픽을

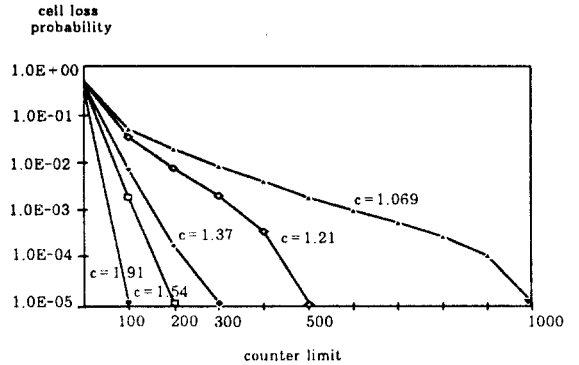


그림 5. 계수기 한계값과 Leaky율과의 관계
Fig. 5. The relation of counter limit and leaky rate

적절히 제어 할 수없게 된다.

다음 절에서 제안하는 사용 파라미터제어 알고리즘은 작은 수의 계수기 한계값을 가지고 정확하게 평균 비트율을 제어 하도록 Leaky Bucket 을 변형 시킨 모델 이다.

4. 제안하는 평균 비트율 제어 알고리즘

본 논문에서 제안하는 사용 파라미터 알고리즘은 그림 6과 같이 최대 비트율 제어부와 평균 비트율 제어부의 이중구조로 이루어진다.

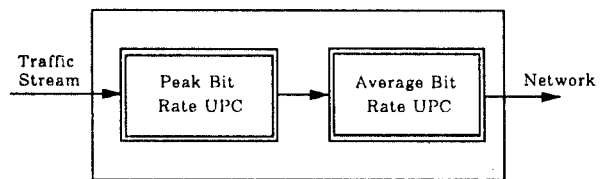


그림 6. 제안 사용 파라미터 제어 모델
Fig. 6. Proposed usage parameter control model

사용 파라미터 제어를 통과하는 트래픽은 먼저 최대 비트율 제어를 통과 하게 되며 최대 비트율을 지키도록 제어된 셀 스트림은 다시 평균 비트율 제어를 통과 한다.

최대 비트율 제어기는 최대 비트율에서의 셀간 전송

간격을 정의하며 전송간격을 위반하는 셀은 폐기하도록 한다.

현재까지 제안된 성능이 우수하고 실현이 용이한 알고리즘은 앞장에서 언급한 Leaky Bucket과 VSA가있다. 두 모델의 성능은 비슷한 것으로 알려져 있으며 실현에 필요한 하드웨어양도 거의 비슷하다. [9]

본 논문에서는 최대 비트율 제어기를 지나온 셀 스트림은 최대 비트율을 지키는 것으로 가정하고 평균 비트율 제어에 초점을 맞추어 제어모델을 제안하고자 한다. 제안하는 모델은 계수기를 가지는 Leaky Bucket의 변형이다.

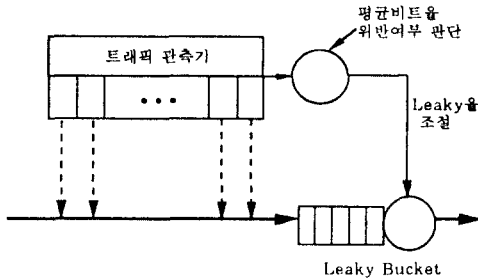


그림 7. 제안하는 평균 비트율 제어 모델
Fig. 7. Proposed mean bit rate control model

본 논문에 제안하는 평균 비트율 제어 모델은 그림 7와 같다.

4.1 평균 비트율 위반 여부

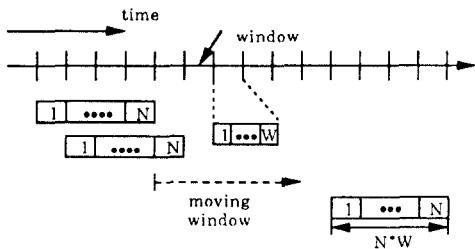


그림 8. 트래픽 관측기의 구조
Fig. 8. The structure of traffic monitor

최대 비트율 제어기를 지나친 셀들은 N개의 창으로 이루어진 트래픽 관측기를 지나간다. 각 창은 W개의 주

기동안 지나간 셀의 수를 기억하게 된다. 하나의 주기(T)는 평균 버스트 구간 h와 평균침묵 구간 k의 합이다. 즉 움직이는 창은 N*W 개의 주기동안 지나간 셀의 수를 기억 하게된다. W개의 새로운 주기가 지나갈 때마다 움직이는 창의 정보는 갱신되며 1개의 시간 창동안 지나간 평균 셀의 수(A)를 계산하게 된다.

1개의 시간 창동안 지나간 셀의평균 갯수 A가 설정된 창의 한계값 보다 크면 평균 비트율을 위반한 것으로 간주하며 작으면 평균비트율에 순응하는 것으로 인식한다.

- 평균 비트율 위반여부 판단 알고리즘
- A = N개의 창의 기억하는 셀의 수 / N
- N : 창의 갯수
- if (A > 창의 한계값)
- 평균 비트율 위반
- else
- 평균 비트율에 순응

제안하는 알고리즘은 W개의 주기마다 평균 비트율 위반여부를 판단하므로 기존의 Leaky Bucket이 반응시간이 느린 것으로 평가 되는데 반해 보다 빠른 반응시간을 가지게 된다.

4.2 창의 한계값 설정

협약한 트래픽파라미터에 순응하는 트래픽의 경우 1개의 창동안 지나가는 이론적인 셀의 갯수 E_x는 다음과 같다.

$$E_x = (PBR * h * C_A) * W / L$$

L : 셀의 비트수, PBR : 최대 비트율

W : 한개의 창의 가지는 주기의 갯수

C_A : 트래픽의 버스티한 성질을 흡수하기 위한 과부하 인자값

버스티한 트래픽의 경우 시간에 따른 트래픽 양이 변동적이기 때문에 짧은 기간동안의 평균 비트율 측정은 협약된 평균 비트율에 순응하는 트래픽을 위반 트래픽으로 잘못 판단할 우려가있다. 협약한 평균 비트율에 순응하지만 짧은 시간의 측정 구간 동안에는 평균 비트율을 넘어서는 정상 트래픽에 대해 1 보다 큰 값의 과부하 인자값 C_A가 필요하다.

평균 비트율의 위반여부를 판단하는 창의 한계값 Q의

초기값은 E_x 로 놓는다. 창 의 한계값을 초기값(E_x)으로 고정시키면 빠른 반응시간을 가지지만 순용 트래픽에 대해 낮은 셀 손실율을 보장하기 위한 허용 과부하 인자값(C_A)이 창 의 수에 따라 변하게 되어 위반 트래픽을 정확하게 제어해 내지 못한다.

창 의 한계값이 고정되면 위반 트래픽을 정확하게 제어해 내기 위해서는 많은 수의 창이 필요하게 되며 이 경우 구현에 많은 비용이 들게 된다. 또한 1개의 시간 창 동안 지나간 평균 셀의 수 A 를 계산하는 것도 복잡하게 된다. 적은 수의 창을 쓰면 구현에 드는비용은 적지만 허용 과부하 인자값이 커지게되며 위반 트래픽을 정상 트래픽으로 오인해 정확한 제어를 할 수가 없게 된다.

본 논문은 적은 수의 창을 사용하면서 위반 트래픽을 정확하게 제어하기 위해 창 의 한계값을 동적으로 관리한다.

1개의 시간창동안 지나간 셀이 이론적인 평균 셀의수 E_x 보다 작은 경우 다음 W 주기 동안 창 의 한계값은 다음과 같이 변한다.

$$\bullet Q(t+1) = Q(t) + (E_x - \text{지난 } W \text{ 주기 동안 지나간 셀의 수})$$

즉 창이 허용하는 셀의 한계값이 커지게 된다. 반면 W 주기 동안 지나간 셀의 수가 설정된 평균셀의 수 E_x 를 넘어가는 경우 다음 W 주기 동안의 창 의 한계값은 다음과 같다.

$$\bullet Q(t+1) = Q(t) - (\text{지난 } W \text{ 주기 동안 지나간 셀의 수} - E_x)$$

창의 허용 한계값을 위반하게되면 창 의 한계값은 작아지게 된다. 평균 비트율에 순용하는 정상트래픽의 경우 짧은 주기동안 측정된 버스티한 소스의 트래픽이 설정된 평균비트율보다 작으면 창 의 한계값이 커지게 되어 이후 측정된 평균 비트율이 설정된 평균 비트율보다 높은 경우 이를 흡수 할 수있도록 한다.

지속적으로 평균 비트율을 위반하는 트래픽의 경우 시간창에 허용되는 셀의 갯수는 위의 식에 의해 자꾸 줄어 들게 된다. 이 경우 소스가 정상적인 트래픽을 회복하여도 정상적인 크기의 창 의 허용 한계값을 가지게 되기까지는 많은 시간이 걸리며 그 기간 동안 대부분의 트래픽

은 평균 비트율을 위반한 것으로 간주한다. 전송되는 트래픽 소스의 셀 갯수가 설정된 평균 셀 갯수에 지속적으로 미달하는 경우 창이 허용하는셀 갯수의 한계값은 계속 커지게 되며 이는 앞 절에서 Leaky Bucket 의 문제점으로 지적한바가 있는 위반 트래픽에 대한 느린 반응 시간을 제안 모델도 보이게 될 우려가 있다. 그러므로 창이 허용하는 한계값은 일정한 값에서 더이상 증가하지 않도록 한계값을 가지도록 할 필요가 있다.

또한 지속적인 위반 트래픽은 창 의 한계값을 지속적으로 떨어지게만들며 이 또한 한계값을 가지도록 해 주어야 한다.

4.3 Leaky울 설정

호 설정시 협약된 평균 비트율에 순용하는 버스티한 특성을 가지는 트래픽의 대하여 낮은 셀 손실율을 가지기 위해서는 앞에서 언급한 바와 같이 매우 큰 수의 계수기 한계값을 가져 버스티한 트래픽 특성을 흡수하거나 Leaky울을 올려 보다 빠른 속도로 입력되는 셀을 처리하도록 하는 방법이 있다. Leaky울을 올려주는 경우 작은 수의 계수기 한계값으로 협약된 파라미터에 순용하는 트래픽에는 낮은 셀 손실율을 보장하지만 협약된 평균비트율을 위반하는 트래픽은 정확하게 검출할 수가 없게된다. 즉 협약된 평균 비트율을 위반하는 셀들을 폐기하지 못하고 망내로 전송하게 된다.

우수한 사용 파라미터 제어모델은 협약된 트래픽 파라미터에 순용하는 셀에 대해 최대한의 전송율을 보장하고 파라미터를 위반하는 트래픽을 정확하게 검출해서 폐기해야 한다. Leaky울을 올리는 경우 위반 트래픽을 정확하게 검출하지 못하게 된다.

본 논문이 제안하는 모델은 사용자의 소스 트래픽이 협약된 평균 비트율을 준수하는 경우 기존의 Leaky울을 지속시켜 낮은 셀 손실율을 보장한다. 하지만 움직이는 창을 이용한 트래픽 관측기가 사용자 트래픽이 평균 비트율을 위반하는 것으로 판단하는 경우 평균 비트율 이상으로 조절된 현재의 Leaky울을 평균 비트율로 낮춘다. Leaky울이 평균 비트율과 동일한 경우 낮은 셀 손실율을 보장하기 위해서는 매우 큰 수의 계수기 한계값이 필요하기 때문에 작은 계수기 한계값을 가진 상태에서 Leaky울을 떨어뜨리게 되면 폐기되는 셀의 양이 많아지게 된다. 일단 Leaky울이 평균 비트율로 떨어진 상태에서 트래픽 관측기가 사용자 트래픽이 다시 평균

비트율을 준수하는것으로 신호를 보내오는 경우 Leaky 울은 이전이 Leaky울로 올려 순용셀에 대해 투명성 (Transparency)을 보장하게 된다.

◦ Leaky 울 설정 알고리즘

Leaky 울 초기값 : $C_L * ABR$

if (평균 비트율 위반)

Leaky 울 = ABR

else

Leaky 울 = $C_L * ABR$

기존의 모델이 시간 창이 허락하는 셀의 한계값을 넘 어가는 셀을 폐기하는 것과는 달리 본 논문이 제안하는 모델에서 창은 평균 비트율의 위반여부를 판단하는 역할 을 하며 위반 트래픽에 대한 제어는 Leaky울의 조절을 통하여 하게된다.

Ⅳ. 시뮬레이션 결과와 분석

본 장에서는 제안하는 알고리즘의 성능분석을 위한 소 스 모델의 트래픽 파라미터를 정의하며, 기존의 사용 파 라미터 제어 알고리즘중 성능이 우수한 것으로 평가된 Leaky Bucket과 제안하는 알고리즘을 비교함으로 제 안하는 알고리즘의 성능을 분석하도록 한다.

1. 시뮬레이션 환경

본 논문에서 제안하는 알고리즘의 성능평가를 위해 사 용된 소스 모델의 트래픽 파라미터는 표1과 같다. 알고 리즘의 성능평가를 위한 시뮬레이션 언어는 C 언어와 SMPL이며 SUN Workstation상에서 실행하였다.

실제 ATM 네트워크가 요구하는 셀 손실율은 10^{-9} 이지만 시뮬레이션 상에서 이러한 셀 손실율을 얻기 위해서는 최소한

표1. 시뮬레이션에 사용한 소스모델의 트래픽 파라미터
Table 1. Traffic parameters which are used on simula- tion

paramrters	voice	Still Pictyre
Peak Rate	32 Kbps	2 Mbps
Mean Rate	11.2 Kbps	87 Kbps
Mean Buest Duration	0.352 sec	0.5 sec
Mean Silence Duration	0.650 sec	11.0 sec

10^6 개 이상의 셀을 발생 시켜야 하므로 본 논문에서는 10^5 의 셀 손실율을 얻을 때까지 시뮬레이션 하였다.

2. 제안 알고리즘의 성능평가 기준

본 논문은 평균 비트율 위반 트래픽에 대한 셀 위반율(vio- lation probability)과 반응시간(response time)을 주요 평가 변수로 사용한다.

◦ 반응시간 : 제안 알고리즘과 Leaky Bucket알고리 즘이 동일한 정도의 셀 손실율을 보장하는 경우 위 반 트래픽에 대한 첫번째 셀 폐기시간을 측정 한 후 이를 비교한다.

◦ 셀 위반율 : 사용자 트래픽으로 정규화된 평균 비트율 위 반 트래픽에 대한 셀 위반율을 측정한다.

3. 시뮬레이션 결과와 분석

3.1 패킷화된 음성 셀 위반율

그림 9는 계수기 한계값이 100개일 때 평균비트율을 위반하는 트래픽에 대하여 Leaky Bucket모델과 제안 알고리즘의 셀 위반율을 나타낸 것이다.협약된 최대 비 트율을 준수하면서 평균 비트율을 위반하는 트래픽은 평 균 버스트 시간을 늘려서 만든다.

본 논문의 시뮬레이션 환경에서 Leaky Bucket 모 델이 협약된 파라미터에 순용하는 트래픽에 대해 10^{-5} 의 셀 손실율을 얻기 위해서는 28500 이상의 계수기 한계값 이 필요하다.

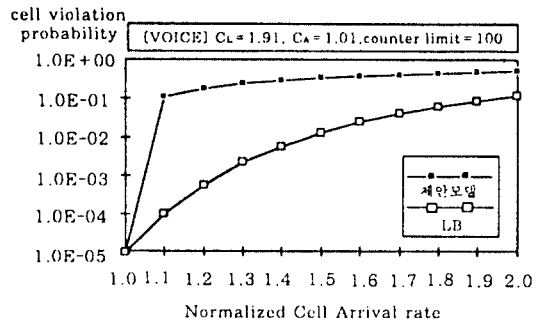


그림 9. 계수기 한계값이 100개일 때, 평균 비트율위반하는 음 성 트래픽에 대한 셀 위반율

Fig. 9. Cell violation probability of voice traffic violat- ing mean bit rate when the count limit is 100

계수기 한계값 100을 가지고 순응 트래픽에 10^{-5} 의 셀 손실율을 보장하기 위해서는 Leaky율이 상향조정되어야 하며 그림 9에서의 Leaky율은 평균 비트율을 $C_L(1.91)$ 배 해준값이다. 시뮬레이션에 사용된 가변크기를 가지는 창 의 갯수(N)는 1개 이며, 1개의 시간창은 10개의 주기(W)를 포함하고 있다. 시간 창에 포함되는 주기의 갯수를 줄이는 경우 정상 트래픽에 대해서도 셀 손실율이 높아지게 된다. 시뮬레이션 결과 버스트한 트래픽을 흡수하기 위한 과부하 인자값(C_A)은 1.01을 사용했다. 과부하 인자값은 트래픽의 버스트한 정도에 따라 달라지게 되며 음성 트래픽의 경우 버스트한 정도가 작기 때문에 작은 수의 과부하 인자값을 사용하여도 우수한 제어 결과를 얻을 수가 있다.

그림 9를 보면 기존의 Leaky Bucket 모델은 계수기 한계값 100을 가질때 10^{-5} 의 셀 손실율을 가지기 위해 앞에서 언급한 바와 같이 Leaky율을 올려 주었다. 이 경우 순응 셀에 대해서 뿐만 아니라 위반 트래픽에 대해서도 낮은 셀 위반율을 보여 정확한 제어를 하지 못한다.

이에 반해 제안 알고리즘은 위반 트래픽에 대해 거의 이상적인 제어를 해내는 것을 그림에서 볼 수가 있다. 한 가지 특이한 것은 위반 트래픽의 양이 많아져도 Leaky Bucket의 처리량은 일정한데 비해 제안 알고리즘은 오히려 더욱 낮은 처리량을 보이는 것을 셀 위반율을 통해서 알 수 있다. 이것은 위반 트래픽이 지속될 경우 창이 허용하는 셀의 한계값은 자꾸 떨어지게 되며 이로 인해

제안 알고리즘에서는 위반 트래픽의 양이 많아질수록 셀 위반율이 높아지게 된다.

그림 10과 그림 11은 계수기 한계값이 300 일때와 500 일때의 셀 위반율을 나타낸 것이다. 이 경우 위반 트래픽에 대해 Leaky Bucket의 셀 위반율이 계수기 한계값이 100 일때 보다 높아지는 것을 보게 되는데 계수기 한계값이 커지게 되면 호 수락 제어시 협의되는 Leaky율은 계수기 한계값이 100 일때 보다 낮아지게 된다. 계수기 한계값이 커지고 Leaky율이 낮아짐에 따라 계수기 한계값이 100 일때 보다는 위반 트래픽에 대해 우수한 제어를 보이지만 제안하는 알고리즘이 보다 정확한 제어를 하는 것을 그림에서 알 수 있다.

3.2 정지화상에 대한 셀 위반율

그림 12는 계수기 한계값이 100일때 평균 비트율을 위반하는 트래픽에 대하여 Leaky Bucket 모델과 제안 모델의 정지화상의 셀 위반율의 변화를 나타낸 것이다. 계수기 한계값이 100일때 Leaky율은 최대 비트율에 근접하게 되며 정지화상같이 버스트니스가 큰 데이터에서 이 정도의 계수기 한계값은 계수기 한계값의 역할로 큰 의미가 없게 된다.

정지화상은 표1과 같이 매우 큰 버스트니스 값을 가지며, 본 논문의 시뮬레이션 환경하에서 Leaky Bucket이 10^{-5} 의 셀 손실율을 가지기 위해서는 26만 정도의 계수기 한계값이 필요하다. 앞절의 음성 데이터의

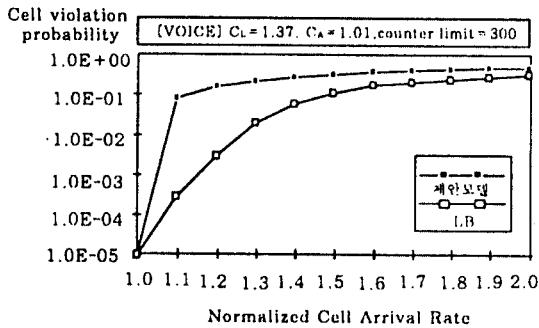


그림 10. 계수기 한계값이 300개일 때, 평균 비트율위반하는 음성 트래픽에 대한 셀 위반율
Fig. 10. Cell violation probability of voice traffic violating mean bit rate when the count limit is 300

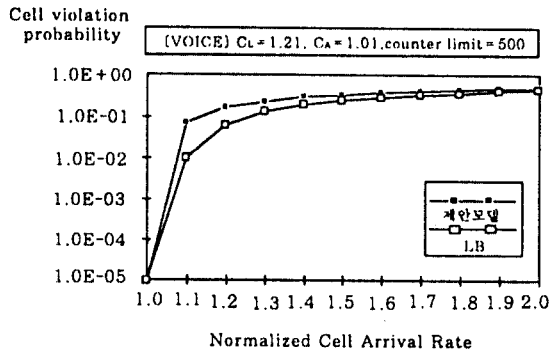


그림 11. 계수기 한계값이 500개일 때, 평균 비트율위반하는 음성 트래픽에 대한 셀 위반율
Fig. 11. Cell violation probability of voice traffic violating mean bit rate when the count limit is 500

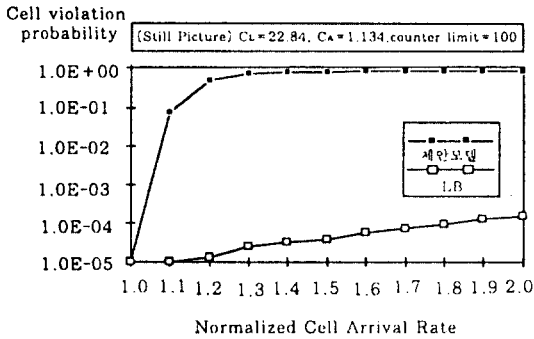


그림 12. 계수기 한계값이 100개일 때, 평균 비트율을위반하는 정지화상 트래픽에 대한 셀 위반율
 Fig. 12. Cell violation probability of still picture traf ficviolating mean bit rate when the count limit is 100

경우 1개의 가변 크기 창이면 위반 트래픽에대한 정확한 제어가 가능 했지만 정지화상의 경우는 시뮬레이션 결과 3개 가량의 창을 사용해야 원하는 정도의 셀 위반율을 얻을 수 있었다. 각 시간 창은 10개의 주기를 포함하고 있으며, 버스트한 트래픽을 흡수하기 위한 과부하 인자 값(C_A)은 1.134를 사용하였다.

그림 12는 정지화상에 대해 기존의 Leaky Bucket 과 창의 갯수가 3개 일때 제안모델의 셀 위반율을 비교 한것이다. 시뮬레이션 결과 창의 갯수가 많아 질수록 위반 트래픽에 대해 셀 위반율이 높아지게 된다.

정지화상과 음성을 비교해 보면 보다 버스티한 특성을 가지는 트래픽에 대하여 더 큰 값의 허용 과부하 인자값 이 필요함을 알수 있다. 또한 제어를 위해 필요한 창의 수도 음성보다 정지화상이 더 필요함을 알수 있다.

그림 13와 그림 14은 계수기 한계값이 500일 때와 1000일때 정지화상에 대한 셀 위반율을 보이고 있다.이 경우들도 앞의 음성 데이터와 마찬가지로 제안 알고리즘 이 Leaky Bucket보다 우수한 제어 곡선을 보이고 있 다.

3.3 반응시간에 대한 시뮬레이션 결과

계수기를 가지는 Leaky Bucket의 경우 셀 버퍼를 가지는 Leaky Bucket보다 구현비용은 적게 들지만 입력 트래픽을 투명하게 네트워크로 전송시키기 때문에 네트워크의 성능의 저하를 가져오거나 대역폭 사용의 낭비를 가져 온다. 낮은 셀 손실율을 보장하기 위한 계수기 한계값은 매우 크게되며 매우 큰 계수기 한계값은 위반 트래픽의 셀이 계수기 한계값을 넘어갈때 까지 트래픽을 아무런 제어없이 투명하게 네트워크로 전송하게 되어 할 당된 것 이상으로 대역폭을 사용하는 것을 허락하게 되 어 결과적으로 망을 과잉밀집의 상태에 빠질 위험이 있다. 이와 같이 큰 계수기 한계값은 위반 트래픽에 느리게 반응하게 되며 느린 반응시간은 지체한 바와 같은 단점을 가지게 된다.

그림 15와 16은 음성과 정지화상의 위반 트래픽에 대

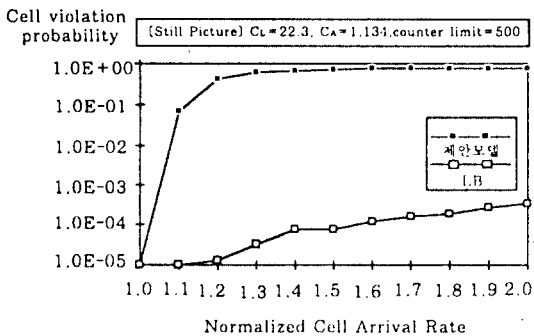


그림 13. 계수기 한계값이 500개일 때, 평균 비트율을위반하는 정지화상 트래픽에 대한 셀 위반율
 Fig. 13. Cell violation probability of still picture traf ficviolating mean bit rate when the count limit is 500

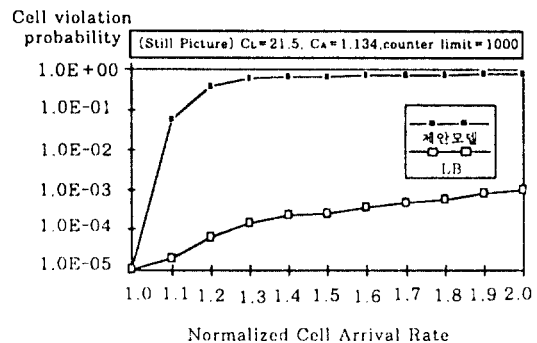


그림 14. 계수기 한계값이 1000개일 때, 평균 비트율을위반하는 정지화상 트래픽에 대한 셀 위반율
 Figure 14. Cell violation probability of still picture traf ficviolating mean bit rate when the count limit is 1000

그림 15. 위반 트래픽에 대한 반응시간 (1)
Fig. 15. Response time of violating traffic(1)

(음성, 단위 초)

정규화된 입력셀율	Leaky Bucket	제안모델
1.1	14918.97	46.28
1.2	6753.87	40.48
1.3	4256.64	40.84
1.4	3338.27	19.00
1.5	2751.48	19.67
1.6	2281.76	20.55
1.7	1945.01	21.41
1.8	1717.19	14.78
1.9	1510.62	13.58
2.0	1337.87	13.93

그림 16. 위반 트래픽에 대한 반응시간 (2)
Fig. 16. Response time of violating traffic(2)

(정지화상, 단위 초)

정규화된 입력셀율	Leaky Bucket	제안모델
1.1	23399.49	702.82
1.2	9865.63	483.21
1.3	4863.43	485.16
1.4	4018.95	487.23
1.5	3480.37	345.02
1.6	3021.78	346.05
1.7	2542.60	346.05
1.8	2416.72	346.05
1.9	2303.88	346.05
2.0	2225.80	346.05

해 첫번째 기각이 일어나는 시뮬레이션 시간을 비교한 것이다. 즉 위반 트래픽에 대해 첫 반응이 일어나는 시간을 보인 것이다. Leaky Bucket과 제안 알고리즘 모두 순응 트래픽에 대해 10^{-5} 의 셀 손실율을 보장하며 위반 트래픽에 대해 동일한 정도의 셀 위반율을 보인다.

음성의 경우 Leaky Bucket의 Leaky율은 평균 비트율과 같으며 계수기 한계값의 크기는 28500이다. 정지화상의 경우도 Leaky율은 평균 비트율과 같고 계수기 한계값의 크기는 257600이다. 비교를 위한 제안 알고리즘의 계수기 한계값은 음성 100, 정지화상도 100이다.

그림에서 보듯이 제안 알고리즘이 Leaky Bucket보다 위반 트래픽에 대해 보다 매우 빨리 반응하는 것을 알 수 있다.

V. 결 론

ATM망과 같은 고속 통신망에서는 망내에 과잉밀집이 일어나지 않도록 미리 트래픽을 제어하는 예방제어가 피드백 정보에 의존하여 트래픽을 제어하는 반응제어보다 트래픽 제어방법으로 더 효율적이다. 예방제어 기법은 호 수락 제어(Call Admission Control), 사용 파라미터 제어(Usage Parameter Control), 우선 순위 제어(Priority Control)가 있다.

본 논문은 움직이는 창을 사용하여 일정시간의 평균

비트율을 계산한 뒤 이 결과에 따라 Leaky Bucket의 Leaky율을 조절하는 새로운 사용 파라미터 제어 알고리즘을 제안한다. 기존의 사용 파라미터 제어 알고리즘들은 최대 비트율 제어에는 우수한 성능을 보이지만 비교적 긴 시간동안 트래픽을 감시하는 평균 비트율 제어에는 적합하지 못한 것으로 알려져 있다. 비교적 우수한 성능을 보이는 Leaky Bucket은 평균 비트율 제어를 위해 매우 큰 수의 셀 버퍼나 계수기 한계값을 필요로 하게 되며 이로 인해 위반 트래픽에 대해 느린 반응시간을 보이게 된다.

제안하는 새로운 알고리즘은 기존의 Leaky Bucket 알고리즘에 일정한 수의 움직이는 창을 더하여 작은 수의 계수기 한계값과 셀 버퍼를 가지고도 정확한 평균 비트율 제어를 하도록 하고 있다.

SUN WorkStation 상에서 제안 알고리즘과 Leaky Bucket 알고리즘을 비교한 결과 제안 알고리즘이 Leaky Bucket에 비해 작은 수의 계수기 한계값을 가지고도 우수한 성능을 보이며 위반 트래픽에 대해서도 매우 빠른 반응시간을 보이는 것을 알 수 있다.

향후 연구방향으로 정확한 트래픽 제어에 필요한 창의 갯수, 각 창을 이루는 주기의 갯수, 협약한 트래픽 파라미터에 순응하는 소스 트래픽의 버스티한성질을 흡수하기 위한 과부하 인자값등에 대한 연구가 앞으로도 계속 이루어져야 할 것이다.

참고문헌

1. CCITT Recommendation I.371, "Traffic control and Congestion Control in BISDN", Study Group XVIII, Geneva, 1992
2. Irfan Khan, Victor O.K.Li, "A Traffic Control Mechanism for ATM Networks" GLOBECOM, pp. 1122-1126 1993
3. H.S. Park, W.S. Rhee, D.Y. Kwak, J.K. Kim, "The Policing Mechanism with State-Dependent control", ICC, pp. 1042-1046, 1993
4. Erwin P. Rathgeb, "Modeling and Performance Comparison of Policing Mechanisms for ATM Networks", IEEE JSAC, vol 9, pp.325-332, 1991
5. M. Butto, E. Cavallero, A. Tonietti, "Effectiveness of the 'Leaky Bucket' Policing Mechanism in ATM Networks", IEEE JSAC, vol 9, pp.335-342, 1991
6. F. Guillemin, P. Boyer, A. Dupuis, L. Romoëuf, "Peak Rate Enforcement in ATM Networks", INFOCOM, pp.753-758, 1992
7. 광귀일, "ATM 네트워크의 트래픽 제어를 위한 GRACE-LB 알고리즘의 설계", 연세대학교 대학원 전산학과, 1992
8. 강상혁, 변성혁, 정민영, 성단근, "ATM 트래픽 제어 기술", 전자공학회지 제19권 제8호, pp.11-23, 1992
9. 박홍식, "ATM 트래픽 제어 기술", 전자공학회지 제20권 제12호, pp.1304-1310, 1993
10. Duke Hong, Tatsuya Suda, "Congestion Control and Prevention in ATM Networks", IEEE Network Magazine, pp.10-16, 1991
11. Gillian M. Woodruff and Rungroj Kositpaiboon, "Multimedia Traffic Management Principles for Guaranteed ATM Network Performance", IEEE JSAC, vol8, pp.437-446, 1990
12. Adrian E. Eckberg, Bharat T. Doshi and Richard Z. Occolillo, "Controlling Congestion in BISDN/ATM: Issues and Strategies", IEEE Communication Magazine, pp.64-70, 1991



李紀奉 (Ki Bong Lee)
 1967년 6월 30일생
 1991년 2월 : 연세대학교 전산학과 졸업(이학사)
 1994년 8월 : 연세대학교 대학원 전산학과 졸업(이학석사)

현재 : (주)데이콤

※ 주관심분야 : ATM네트워크, 디지털 서명



宋周鑄 (Joo Seok Song) 정회원
 1953년 3월 2일생
 1976년 2월 : 서울대학교 전기공학과 졸업(학사)

1979년 2월 : 한국과학원 전기 및 전자공학과 졸업(석사)

1988년 8월 : Univ. of California at Berkeley 전산학과 졸업(학사)

1979년 2월~1982년 2월 : 한국전자통신연구소 연구원

1988년 9월~1989년 2월 : Naval Postgraduate School Information System Dept. 조교수

1989년 3월~현재 : 연세대학교 전산학과 부교수

※ 주관심분야 : ATM Network, Protocol Engineering Network Security 등

이준원 (Lee Jun Won)

1953년 8월 15일생

1976년 : 서울대 전자공학과 졸업

1977년~1979년 : 삼성전자부품 근무

1987년~1988년 : AT&T Bell Lab 연구소 방문 연구원

현재 : 한국전자통신 연구소 초고속망 연구실장