

## 비동기식망에서 windows를 이용한 손실 및 지연제어 기법 Cell Loss and Delay Control Scheme using Windows in ATM Networks

김남희, 김변곤, 조해성\*  
군산대학교, 건양대학교\*

Kim Nam-Hee, Kim Byun-Gon, Cho Hae-Seong\*  
Kunsan National Univ., Konyang Univ.\*

### 요약

광대역망에서 다양한 트래픽의 종류에 따른 셀 손실율과 셀 지연율의 요구사항을 만족시킬 수 있는 제어시스템의 설계는 매우 중요한 요소이다. 본 논문에서는 트래픽의 각 클래스 타입에 따라 셀 카운터와 윈도우 카운터를 이용한 우선순위 제어 기법을 제안하였다. 제안한 기법에서는 요구된 서비스의 질을 만족시키기 위해 셀 카운터와 윈도우 카운터의 비교에 의해 얻어진 셀 손실과 지연 요소를 획득하여 우선순위 제어를 수행하였다. 그리고 컴퓨터 시뮬레이션을 통하여 성능을 평가하였다.

### Abstract

Design of appropriate control schemes that can satisfy the cell loss, delay requirements with various traffic specification for B-ISDN is an extremely important challenging problem. In this paper, we propose a priority control scheme with a window counter and a cell counter per each type of class. The priority control for satisfying required service quality is performed with delay/loss factor obtained by comparing window counter with cell counter. The performance of proposed control scheme is estimated by computer simulation.

## I. 서론

ATM망에서 통계적 다중화를 이루기 위해서는 연결수락제어(connection admission control), 망자원관리(network resource management), 우선순위제어(priority control), 트래픽 셰이핑(traffic shaping) 및 선택적 셀 폐기 등과 같은 제어 기술의 적용은 필수적이다[1,2,8]. ATM망은 영상신호와 같은 버스티한 특성을 가진 트래픽, 연속적 데이터 트래픽, 시간지연에 민감한 트래픽 등의 다양한 QoS(quality of service)를 만족시키기 위해 우선순위 제어를 사용하는데 우선순위는 지연우선순위, 손실우선순위를 두어 제어하게 된다[3,4]. 그러나 대부분의 많은 연구들이 지연우선순위나 손실우선순위 중 하나에 치중되어 왔으며 손실에 민감한 트래픽과 지연에 민감한 트래픽을 모두 고려한 방법이 제안되었다[4,5]. 그러나 이들 기법의 경우 지연요소를 알기위해 셀 당 virtual clock을 추가해야 하는 등 오버헤드의 비중이 증가되었다. 본 논문에서는 다양한 종류의 트래픽 소스를 지연 및 손실에 민감한 정도에 따라 클래스로 분류하고 각 클래스의 지연 및 손실 요구사항을 만족시킬 수 있는 우선순위 제어 기법을 제안하였다. 제안된 우선순위제어에서는 각각의 클래스에 윈도우 카운터와 셀 카운터를 두었다. 각 클래스의 지연요소는 윈도우 카운터와 셀 카운터를 비교해 봄으로써 쉽게 구할 수 있고, 손실요소는 셀 카운터를 통해 쉽게

구할 수 있도록 하였다. 셀 서비스는 각 클래스의 지연 및 손실 요소에 따른 우선순위를 구해서 가장 높은 우선순위를 갖는 클래스의 셀을 서비스 하게 된다. 컴퓨터 시뮬레이션을 통해 기존의 우선순위 제어 기법과 본 논문에서 제안하는 기법을 셀 손실율과 셀 지연의 관점에서 비교분석하였다.

## 2. 윈도우를 이용한 효율적 우선순위 제어

다양한 서비스들의 QoS를 만족시키기 위해 본 연구에서는 지연 및 손실 우선순위를 동시에 고려한 우선순위 제어 기법을 제안하였다. 제안된 기법에서는 윈도우 기법을 이용한 윈도우 카운터, 버퍼에서의 셀 입출력에 따라 동작하는 셀 카운터를 가지고 지연 및 손실 요소를 구했다. 즉 지연요소는 윈도우 카운터와 셀 카운터의 값을 비교함으로써 쉽게 구할 수 있고 손실 요소는 셀 카운터의 값으로 구할 수 있다. 이렇게 구한 지연 및 손실 요소를 근거로 우선순위를 구하여 우선순위 제어를 수행하게 된다.

제안된 우선순위 제어는 ATM 스위치의 출력버퍼에서 이루어지는데 본 논문에서는 출력버퍼형의 N X N 스위치를 가정하였으며, 각 출력포트에 있는 출력버퍼에서 셀 전송 순서 결정 및 셀 폐기를 통한 우선순위 제어를 수행 할 수 있도록 하였다.

제한된 구조에서 입력주소 제어부는 공유버퍼의 입력주소 관리부와 축출주소 관리부로 나누어지는데 입력주소 관리부는 입력되는 셀들이 저장될 공유버퍼의 빈 주소공간을 관리하고 축출주소 관리부는 공유버퍼에 빈 공간의 주소가 없을 때 축출될 손실우선순위가 낮은 클래스 셀의 주소이다. 또한 입력되는 셀들의 헤더를 입력받아 클래스의 정보를 얻어 공유버퍼에 저장하고 그 주소를 해당 클래스 버퍼에 입력한다. 만일 공유버퍼에 빈 공간이 없으면 손실우선순위가 낮은 셀은 폐기하고, 손실우선순위가 높은 셀은 축출주소관리부에서 주소를 얻어 손실우선순위가 낮은 셀을 축출할 수 있도록 하였다.

축출기법이 적용되기 위해서는 손실 우선순위가 높은 클래스 주소 버퍼의 공간이 많아야 하므로 손실 우선순위가 높은 클래스의 주소버퍼를 크게 할 필요가 있다. 각 클래스 주소버퍼는 해당 클래스 셀들이 저장된 공유버퍼의 주소를 저장하고 있다. 각 클래스 제어부는 윈도우 카운터와 셀 카운터로 구성되어 있으며, 각 클래스의 버퍼에서 서비스 받을 셀의 주소를 우선순위 제어부로 전달한다. 윈도우 카운터는 해당 클래스의 셀이 입력되면 증가하고 일정지연시간(window size)이 경과되면 감소된다. 지연시간은 클래스의 지연우선순위와 공유버퍼의 크기에 따라 결정된다. 즉 지연우선순위가 높은 빠른 서비스를 요구하므로 지연시간을 작게 설정하고 지연우선순위가 낮은 클래스는 지연시간을 크게 설정한다. 또한 공유버퍼의 크기가 작으면 모든 클래스의 셀들이 지연을 겪을 가능성이 적으므로 적은 지연시간을 설정한다. 윈도우 카운터는 클래스 버퍼에 셀이 없으면 초기화 시키고 셀 카운터는 해당 클래스의 셀이 입력되면 증가하고, 서비스를 받으면 감소한다. 즉 셀 카운터는 현재상태의 클래스 버퍼에 있는 셀들의 숫자를 카운트한다. 윈도우 우선순위 제어부는 각 클래스의 제어부로부터 윈도우 카운터와 셀 카운터의 값을 입력받아 각 클래스의 지연 및 손실요소를 구하고, 지연 및 손실요소에 따른 클래스의 지연 및 손실 우선순위를 구하여 가장 높은 우선순위를 갖는 클래스를 결정한다.

또한 각 클래스의 제어부로부터 각 클래스에서 서비스 받을 셀의 주소를 넘겨받아 가장 높은 우선순위를 갖는 클래스에서 서비스 받을 셀의 주소를 출력주소 제어부에 입력한다. 손실 및 지연 우선순위를 구하는 방법은 다음과 같다.

각 클래스의 지연요소를 알기 위해서 셀이 버퍼에 도착하여 얼마나 많은 시간이 경과 되었는지를 알기 위해서는 모든 셀에 시간 스탬프(time stamp)를 붙여야 된다. 그러나 모든 셀에 타임 스탬프를 붙이게 되면 오버헤드가 증가하게 되는 단점이 있다. 따라서 본 논문에서는 셀 당 타임 스탬프를 붙이지 않고 클래스별로 얼마나 많은 지연을 겪고 있는지를 검사함으로써 오버헤드를 줄이고 각 클래스 셀들의 지연요소를 쉽게 알 수 있도록 하였다. 즉, 윈도우 카운터와 셀 카운터의 값을 비교함으로써 각 셀 당 타임스탬프를 붙이지 않고도 쉽게 지연정도를 파악

할 수 있도록 하였다. 같은 수의 셀들이 같은 시간에 입력된다 고 가정하면 지연우선순위가 높은 클래스의 윈도우 크기가 작으므로 윈도우 카운터 값은 지연우선순위가 낮은 클래스의 윈도우 카운터 값보다 작은 값을 가지게 된다. 예를 들면 윈도우 카운터 값이 10이고 셀 카운터 값이 20이면 10개의 셀들은 각 클래스의 일정시간 지연시간이 지나도록 서비스 받지 못하고 있다는 것을 의미하고 반대로 카운터 값이 20이고 셀 카운터 값이 10이면 해당 클래스의 셀들은 일정 지연 시간 안에 서비스 받고 있다는 것을 의미한다. 따라서 셀 카운터의 값에서 윈도우 카운터의 값을 빼면 각 클래스의 지연요소 값을 알 수 있다. 본 논문에서는 각 클래스의 지연요소를 구해서 지연요소에 따른 지연우선순위 값을 구하여 값이 양수이면 해당 클래스의 셀은 지연을 겪고 있으므로 지연우선순위를 높여준다. 이때 선형 및 비선형 함수를 이용하여 지연우선순위 값을 결정하게 된다. 지연요소가 음수이면 해당 클래스의 셀은 지연을 겪지 않고 정상적으로 서비스 받고 있다는 것을 의미하므로 지연우선순위 값은 각 클래스에 따라 일정하게 유지하게 된다. 손실요소는 각 클래스의 주소버퍼에 임계치를 두고 셀 카운터 값으로부터 임계치에 따라 구한다. 또한 축출기법을 적용하여 손실우선순위가 높은 클래스의 셀 손실율을 향상시킬 수 있다. 그리고 지연요소와 손실요소에 따른 지연 및 손실 우선순위가 결정하여 가장 우선순위가 높은 우선순위 값을 갖는 클래스의 셀을 서비스 함으로서 모든 클래스의 셀이 공평하게 서비스 받을 수 있도록 하였다.

### 3. 시뮬레이션 환경 및 성능

본 논문에서는 세 개의 클래스를 가정하였으며, class 1은 비디오 트래픽과 같은 지연과 손실에 민감한 트래픽이므로 비디오 소스를 모델링하여 시뮬레이션을 수행하였다. 비디오 소스는 두 가지 상태로 모델링 되는데 source-change 상태와 inter-scene-change 상태로 나뉜다. scene-change 상태는 화면의 변화가 일어나는 경우의 셀의 발생빈도가 증가하는 상태이고 inter-scene-change 상태는 급격한 화면의 변화가 없는 상태이므로 셀 발생 빈도는 scene-change 상태보다는 적은 상태이다. 먼저 inter-scene-change 상태에서는 독립적인  $M(10)$ 개의 on/off minisource의 합의로 모델링 되었고  $i$  개의 셀이 발생될 확률은 식 1과 같다.

$$a_m(i) = \binom{m}{i} A^m (1-A)^{m-i} \quad 0 \leq i \leq M \quad (\text{식1})$$

scene 상태에서의 셀 발생 확률은 표 2와 같고  $N$  개의 비디오 소스가 동시에 scene-change 상태가 될 확률은 식 2와 같다.

$$b_n(i) = \binom{N}{i} \lambda_{sc}^i (1 - \lambda_{sc})^{N-i} \quad (식2)$$

[표 2] 비디오 소스 입력 파라미터

mean cell arrival rate during scene change	$\lambda_{sc}$	$2.35 \times 10^{-1}$
mean ON length of minisource	$\gamma^{-1}$	$1.36 \times 10^5$
mean OFF length of minisource	$\omega^{-1}$	$2.71 \times 10^5$
mean cell arrival rate during minisource's ON state	A	$8.39 \times 10^{-3}$

비디오 소스는 식 1에 의해 발생된 셀 수와 N개의 scene-change 상태의 발생셀수의 합으로 구해진다. 우선순위 제어는 셀 발생 확률이 높아 셀의 폭주가 발생할 가능성이 높은 상황일 때 적절한 대응을 해야 하므로 컴퓨터 시뮬레이션은 비디오 소스가 항상 scene-change 상태여서 셀 발생 확률이 높은 상황을 가정하고 수행하였다. 음성 및 컴퓨터데이터는 poisson process로 가정하였으며 셀 발생 확률은 식 3과 같다.

$$v(i) = \frac{\lambda_v^i e^{-\lambda_v}}{i!}, \quad d(j) = \frac{\lambda_d^j e^{-\lambda_d}}{j!} \quad (식3)$$

$\lambda_v, \lambda_d$  는 음성, 컴퓨터 데이터의 평균도착율이다.

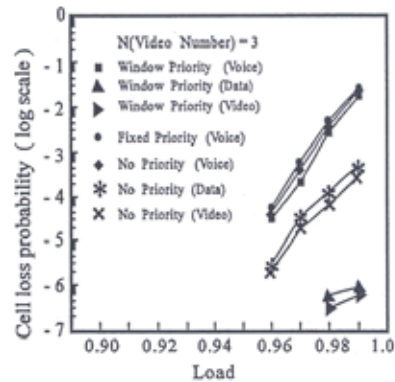
평균 비디오 셀 발생 확률은 N=1, 2일 때 각각 0.2630, 0.5261 이고, 음성과 컴퓨터 데이터 셀 발생 확률은 같은 값을 가지는데 전체로드와 비디오 셀 발생 확률의 차를 둘로 나눈 값이 된다.

### 3.1 시뮬레이션 환경

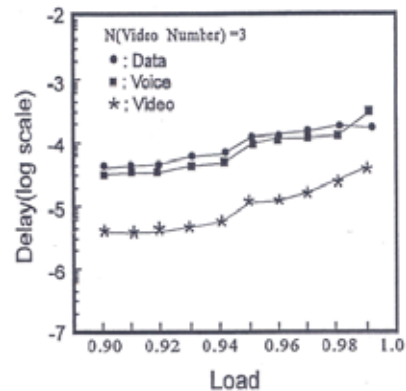
출력링크의 용량은 155Mbps를 가정하여 수행하였으며, 비디오 소스의 수(N)가 3일 때 부하 0.91에서 0.99까지 수행하였다. 시스템의 공유버퍼의 수는 120개이고 각 클래스 버퍼의 임계치는 11개를 가지고 수행하였다. 윈도우 제어기법은 선형 및 비선형 함수를 사용하여 수행하였으며, 시뮬레이션 결과는 no priority, fixed priority 제어기법과 제안된 기법을 셀 손실률 및 지연의 관점에서 비교분석하였다. no priority 법은 셀의 도착순서에 따라 서비스를 해주는 기법이다. 그러나 본 논문에서는 공유버퍼를 사용하여 추출기법이 적용된 no priority 기법을 적용하였다. Fixed priority control 기법은 비디오, 컴퓨터 데이터, 음성순으로 서비스 하는데 우선순위가 낮은 셀은 우선순위가 높은 셀이 없을 때 서비스 받게 된다. 본 논문에서는 fixed priority control 기법 역시 공유버퍼를 사용하였고 추출기법이 적용되었다. 클래스 버퍼의 임계치는 선형 및 비선형 함수에 관계없이 같은 값을 가진다.

### 3.2 성능평가

그림 1, 2는 각각 선형 함수를 이용한 윈도우 우선순위 제어 기법, 선형 함수를 이용한 윈도우 우선순위 제어기법의 평균 셀 지연 시간을 나타낸다. N이 3일 경우 윈도우 우선순위 제어기법은 셀 지연 시간의 관점에서는 N이 작을 경우와 같은 결과를 보이고 있다. 그리고 셀 손실률은 부하가 큰 경우에는 no priority 기법에서 발생한 비디오, 컴퓨터데이터, 즉 손시에 민감한 트래픽의 셀 손실률을 현저하게 줄이고 있음을 알 수 있다. 또한 fixed priority 제어기법은 손실에 민감하지 않은 트래픽 셀의 손실률이 지나치게 높게 나타나고 있다. 따라서 본 논문에서 제안한 윈도우 우선순위 제어 기법은 셀의 지연 측면에서는 다른 우선순위 제어기법보다 우수한 성능을 보이고 있고, 셀 손실 측면에서는 버스티한 트래픽의 증가와 부하의 증가로 인해 셀의 폭주상황이 발생할 가능성이 증가할 때 우수한 성능을 보이고 있음을 확인할 수 있다.



▶▶ 그림 1. N=3일 때 선형함수를 이용한 우선순위 기법의 셀 손실률



▶▶ 그림 2. N=3일 때 선형함수를 사용한 셀 지연시간

### 4. 결론

제안된 우선순위 제어 기법은 실시간 트래픽인 비디오 및 음성 트래픽의 지연시간을 비실시간 트래픽인 컴퓨터 데이터 트

래픽의 지연시간보다 줄여 주었다. 또한 셀 손실을 면에서는 버스티한 특성이 강한 비디오 트래픽의 수가 크고, 부하가 증가해서 셀의 폭주상황이 발생할 가능성이 높을 때 손실에 민감한 트래픽의 셀 손실을 줄여 줌으로서 적절한 제어를 수행하고 있음을 확인할 수 있었다.

또한 지연요소를 구하기 위한 하드웨어의 오버헤드를 줄임으로서 고속의 우선순위 제어가 가능하다. 그리고 윈도우 크기, 임계치, 우선순위 결정함수를 변화시켜 줌으로서 광대역망에서 다양한 신호원의 품질요구를 만족시킬 수 있으며 체증을 완화시키는 효과적인 트래픽 제어 기술에 적용될 수 있을 것이다.

#### ■ 참고 문헌 ■

- [1] T.Y.Huang and J. L. C. Wu, "Performance analysis of ATM switches using priority schemes," IEEE Proc. Vol.141, Aug. 1994.
- [2] H.Jonathan Chao, Daen Jeong, "Generalized priority queue manager design for ATM switches," IEEE Proc. Infocom, 1996.
- [3] Mutlu Arpacı, John A. Copeland, "Buffer management for shared-memory ATM switch," <http://www.comsoc.org/pubs/surveys>
- [4] H. Jonathan Chao, I. H. Pekcan, "Queue management with multiple delay and loss priority for ATM switches," IEEE Proc. ICC, pp.1181-1189 1994.
- [5] Alexander Kesselman, Yishay Mansour, "Harmonic Buffer Management Policy for Shared Memory Switches," IEEE INFOCOM, 2002.