논문 00-25-4B-10 한국통신학회논문지 '00-4 Vol.25 No.4B

ATM 망에서 다양한 트래픽을 지원하기 위한 동적 셀 스케줄링 알고리즘

정회원 심 재 정*, 이 원 호**, 변 재 영*, 고 성 제*

A New Implementable Scheduling Algorithm Supporting Various Traffics in ATM Networks

Jae-Jeong Shim*, Won-Ho Lee**, Jae-Young Pyun*, and Sung-Jea Ko* *Regular Members*

요 약

본 논문에서는 ATM 네트워크에서 다양한 멀티미디어 트래픽을 효율적으로 전송하기 위해 동적 우선순위 셀 전송 스케줄링 기법인 AWRR/DT(Adaptive Weighted Round Robin with Delay Tolerance)를 제안하였다. AWRR/DT는 멀티미디어 트래픽을 지연 특성에 따라 여러 개의 실시간 클래스와 하나의 비실시간 클래스로 분류하고, 각 클래스의 지연 특성과 입력 트래픽의 양을 고려하여 매 사이클마다 해당 클래스에 적절한 가중치(weight)를 할당하도록 설계되었다. 또한, 제안한 알고리즘은 셀 폐기 메커니즘을 가지고 있어서 순간적인 서비스 품질(QoS: Quality of Service) 열화가 계속 연속적으로 이어지는 현상을 줄여준다.

AWRR/DT의 성능을 평가하기 위하여, SLAM II를 이용한 컴퓨터 시뮬레이션을 통해 기존의 스케줄링 기법들과 평균 지연 측면에서 비교하였다. 그 결과, 제안한 기법이 실시간 트래픽 클래스의 QoS를 만족하면서도 비실시간 트래픽 클래스의 평균 지연을 감소시킬 수 있음을 확인하였다.

ABSTRACT

In this paper, we propose a new scheduling algorithm called the Adaptive Weighted Round Robin with Delay Tolerance (AWRR/DT). The proposed scheme is based on the per-class queueing mechanism in which a number of connections of similar characteristics are multiplexed into one class-queue. Traffic classes of the proposed method are classified into a single non-real-time traffic class and other real-time traffic classes. The proposed scheme determines the weights of classes according to the input traffic and delay characteristics of each class at the beginning of every cycle. Furthermore, this scheme incorporates a cell discarding method to reduce the QoS degradation that may be incurred by congestion of networks.

We have evaluated the proposed scheme through discrete-event simulation. Simulation results indicate that the proposed scheme can reduce the average delay of non-real-time class while maintaining the QoS of real-time classes. The proposed algorithm can be effectively applied to high-speed networks such as ATM networks.

I. 서 론

광대역 통신망에서의 트래픽은 크게 텍스트 데이터와 같은 손실 민감 트래픽과 지연에 민감한 음성, 영상 등의 지연 민감 트래픽으로 분류되는데, 각각의 트래픽 특성에 따라 서로 다른 서비스 품질 요구 사항을 가지고 있다. 지연 민감 트래픽의 경우는

* 고려대학교 전자공학과 (sjj@dali.korea.ac.kr)
** 데이콤 초고속인터넷사업팀
논문번호: 99429-1027, 접수일자:1999년 10월 27일
※ 본 연구는 한국과학재단 특정기초연구과제(과제번호:97-0100-1501-5)의 지원을 받은 연구 결과입니다.

종단간에서의 트래픽 손실에 의한 서비스 품질 저하보다는 전송 지연에 의한 품질 저하의 영향을 크게 받으며, 반면에 손실 민감 트래픽의 경우는 망에서의 트래픽 폭주 현상과 트래픽의 에러 손실에 매우 민감하다. 이와 같은 트래픽을 효과적으로 서비스하기 위해 HOL-PJ (Head-Of-the Line with Priority Jumps)[2], MLT (Minimum Laxity Threshold)[3], QLT (Queue Length Threshold)[3], WRR (Weighted Round Robin)[4]과 같은 스케쥴링 기법이 제안되었다.

먼저, HOL-PJ에서는 도착 트래픽을 실시간 서비스 우선순위에 따라 클래스별로 분류하며, 상위 우선순위 큐의 트래픽을 우선적으로 서비스한다. 그리고 큐에 입력되는 패킷마다 도착 시각을 관리하여, 패킷이 큐마다 설정된 지연 허용시간을 넘어서면 그보다 하나 높은 우선순위 큐로 옮겨지게 된다. 이 방식은 우선순위가 높은 클래스의 QoS는 보장해 줄 수 있으나 반면에 우선순위가 낮은 클래스의 심각한 품질 저하를 가져 올 수 있다.

MLT에서는 트래픽 클래스를 실시간 클래스와 비실시간 클래스로 분류하고, 패킷이 스케쥴러로부터 서비스를 받아야 하는 시점까지의 여유 타임슬롯(이완시간:laxity time)을 모든 패킷에 할당하여 관리한다. 그리고 실시간 클래스 패킷들의 최소 여유 타임슬롯이 정해진 임계치보다 작거나 같을 경우에 실시간 클래스를, 그렇지 않은 경우에는 비실시간 클래스를 서비스한다. 이렇게 함으로써, 실시간 트래픽의 지연 요구사항 보장하고 지연 변이(delay variance)도 상대적으로 작게 할 수 있는 장점이 있다.

그러나, HOL-PJ와 MLT에서와 같이 패킷마다 도착 시각이나 여유 타임슬롯을 태깅(tagging)하는 방식은 시스템 오버헤드를 일으키는 원인으로 작용하여 ATM망과 같은 고속 통신망에서는 적용에 어려움이 있다. 이러한 문제는 망의 트래픽 로드가 큰 혼잡 상황에서 더욱 중요한 문제점으로 부각된다.

반면 QLT는 트래픽 클래스를 MLT에서와 같이 실시간 클래스와 비실시간 클래스로 분류하고, 비실시간 트래픽 버퍼 내의 큐 사이즈가 특정 임계값을 초과할 때 비실시간 클래스가 서비스 받도록 한다. 이 방식에서는 서비스의 판단 기준으로 큐 사이즈만을 이용함으로써 셀 태깅에 의한 오버헤드가 줄어드는 장점이 있다. 그러나, QLT에서는 적절한 임계값의 설정이 또 다른 문제점으로 남게 되며 변화하는 망의 트래픽 상태를 반영할 수 있는 메커니즘도 가지고 있지 않다.

다음으로 ATM망에서 많이 사용되고 있는 스케쥴링 방식인 WRR은 round-robin 방식에 의한 비교적 단순한 서비스 루틴을 가지고 있지만, 큐마다의 이미 결정된 가중치에 의해서 서비스 순위와 서비스 양이 결정되므로 트래픽 상태의 변화에 효과적으로 적응할 수 없고 따라서 QoS 보장 메커니즘을 가지고 있지 않다.

본 논문에서는 ATM 네트워크에서 다양한 멀티미디어 트래픽을 효율적으로 전송하기 위해 동적 우선순위 셀 전송 스케쥴링 기법인 AWRR/DT (Adaptive Weighted Round Robin with Delay Tolerance)를 제안한다. 제안하는 알고리즘은 HOL-PJ나 MLT에서와 태깅 방식을 사용하지 않으면서도 각각의 트래픽 특성에 맞는 QoS를 보장해 줄 수 있고, 현재의 망 상태에 동적으로 대처할 수 있는 메커니즘도 가지고 있다.

본 논문의 구성은 다음과 같다. 이어지는 Ⅱ장에서는 본 논문에서 제안하는 동적 우선 순위 제어 기법인 AWRR/DT의 개념과 그 알고리즘을 설명한다. Ⅲ장에서는 시뮬레이션을 통하여 기존의 셀 스케쥴링 기법들과 제안된 기법을 비교하며, 마지막으로 Ⅳ장에서 본 연구의 결론을 맺는다.

Ⅱ. 제안한 스케쥴링 기법

제안하는 기법의 기본적인 개념은 트래픽을 지연 특성에 따라 클래스 단위로 나누어 서비스하는 시스템인 per-class queueing system에서, 각 클래스의 큐에 입력된 셀들이 해당 큐에서 허용될 수 있는 최대 지연값 내에서 서비스되도록 해 주는 것이다. 이것은 지연이 가장 중요한 서비스 품질 지표의 하나이고 이를 적정수준 이내로 제한해 주어야 안정된 서비스 품질을 유지할 수 있기 때문이다. 제안하는 알고리듬은 이러한 지연 요구사항을 만족시켜주기 위하여 각 클래스의 지연 특성과 그 클래스로 입력되는 셀들의 양을 기준으로 매 사이클마다 해당 클래스의 가중치를 적절히 조정해 준다.

그림 1은 본 논문에서 제안하는 AWRR/DT 알고리즘이 적용되는 시스템 모델을 나타내고 있다. 제안하는 알고리즘에서 각 큐에 할당된 가중치가 초기화되는 시간 단위를 사이클(cycle)이라고 정의하고, 한 사이클에서 처리되는 셀의 총 개수는 모든 클래스의 가중치 총합과 같다. 시스템의 입력이 되는 트래픽 소스들은 그림에서와 같이 N-1 개의 실

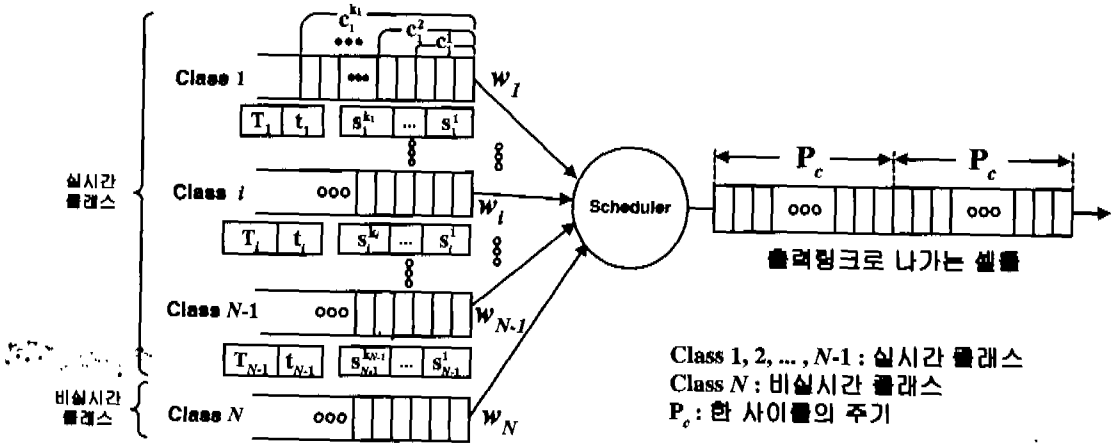


그림 1. AWRR/DT의 시스템 모델

시간 트래픽 클래스와 하나의 비실시간 트래픽 클래스로 구분되어 비슷한 특성을 갖는 트래픽들은 동일한 클래스 큐에 입력된다. 비실시간 트래픽 클래스의 경우에는 그 특성을 세부적으로 분류할 만한 특징값이 없기 때문에 여기서는 하나의 클래스만을 두었다.

실시간 클래스들은 각각의 지연 특성을 반영하는 지연 허용치(delay tolerance, T_i)를 가진다. 이 지연 허용치은 각 실시간 클래스 큐에 입력된 셀이 QoS를 만족하면서 큐 내에 체제할 수 있는 최대 지연 시간이다. 또한, 실시간 클래스 i는 $k_i = \lfloor T_i/P_c \rfloor$개의 카운터 변수들(예를 들어, 그림 1에서 클래스 1의 경우 s_1^1, s_1^2, ···, $s_1^{k_1}$)을 가진다. 여기서, k_i는 실시간 클래스 i에 입력된 셀이 그 클래스의 지연 허용 범위 내에서 서비스를 받기 위한 여유 사이클 횟수이다. 즉, 클래스 i에 입력된 셀들은 QoS를 만족하기 위해서 큐에 입력된 이후 k_i 사이클 내에 서비스를 받아야 하며, k_i 사이클 내에 서비스를 받지 못한 셀은 지연 허용 범위를 초과하여 주어진 QoS를 만족하지 못하게 된다. T_i/P_c 값이 정수값을 가지지 않을 때에도 QoS를 만족해야 하므로 플로어(floor) 함수를 사용하였다.

s_i^j는 클래스 i에서, 이어지는 j개의 사이클 내에 서비스 받아야 하는 셀의 수이다. 그리고, s_i^j개의 셀들의 묶음을 클러스터 c_i^j라고 정의한다(그림 1 참조). 따라서 c_i^j는 클래스 i에서 이어지는 j개의 사이클 내에 서비스 받아야 하는 셀들의 묶음이라고 말할 수 있다.

카운터 변수들은 $s_i^{j-1} = s_i^j - [$이전 사이클에서 서비스를 받거나 폐기된 셀의 수$]_i$의 관계를 가지며 이 때 j=2, 3, ···, k_i 이다. 각 클래스의 가중치 값은 매 사이클의 시작 시점에서 계산되며, s_i^j와 같은 변수값들은 매 사이클마다 가중치 값이 결정되기 바로 직전에 변경되어 다음 사이클에서 서비스될 가중치 값에 반영된다. 각 클래스의 카운터 변수 값들과 가중치는 한 사이클에 한 번씩 조정된다.

실시간 클래스 i의 경우, 가중치 값은 매 사이클이 시작되기 바로 직전에 식 (1)에 의해서 결정된다. 예를 들어 어떤 클래스의 지연 허용치가 스케줄러 한 사이클 주기의 n 배라 하면, 이 클래스에 입력된 셀들은 늦어도 입력된 시점에서 n 사이클이 경과하기 전까지는 서비스가 되어야 QoS를 만족한다고 볼 수 있는데, 이것이 가중치를 결정하는 기준이 된다. 식 (1)에서 각 항들은 각각에 해당하는 클러스터에 포함된 셀들이 이러한 기준을 만족하기 위한 가중치 값들이다. 여기서 해당 클래스의 가중치는 그 클래스 내의 모든 클러스터가 QoS를 만족할 수 있도록 하기 위해 클러스터마다 계산된 가중치들 중에서 가장 큰 값을 선택한다.

$$w_i = \max\left\{\left\lceil \frac{s_i^{k_i}\cdot P_c}{T_i - t_i}\right\rceil, \left\lceil \frac{s_i^{k_i-1}\cdot P_c}{T_i - t_i - P_c}\right\rceil, \cdots, \left\lceil \frac{s_i^1\cdot P_c}{T_i - t_i - (k_i-1)\cdot P_c}\right\rceil\right\} \quad \text{for } i=1, 2, \cdots, N\text{-}1 \qquad (1)$$

위 식에서 $k_i = \lfloor T_i/P_c \rfloor$, i = 1, 2, ···, N-1 이고 t_i는 각각의 실시간 클래스의 타이머 값으로 $0 < t_i \le P_c$이다. 여기서 타이머 값은 한 사이클에서

각 클래스에 입력된 셀들이 다음 사이클의 가중치 결정에 반영되기 전에 이미 지연된 시간이다. 타이머는 입력된 셀이 큐 안에 체제할 수 있는 지연 허용치를 보정해 주기 위한 것으로, 각 실시간 클래스마다 하나씩 존재하며 해당 큐에 셀이 도착할 때마다 '0'으로 초기화된다. 따라서, 타이머는 큐에서 가장 최근에 입력된 셀이 도착한 시점에서부터 현재까지의 경과된 시간을 가지고 있게 되는데, 각 클래스마다의 가중치 값들이 결정되는 시점에는 '매 사이클에서 마지막으로 입력된 셀의 도착 시점과 다음 사이클의 시작 시점간의 시간차'를 갖게 된다. 이 값은 각 클래스의 가중치를 결정할 때 지연 허용치에서 상쇄되어 큐에 입력된 셀들이 해당 클래스의 지연 허용치 내에 서비스를 받을 수 있도록 한다.

식 (1)의 각 항에서 카운터 변수(s_i^j)에 곱해진 값은 해당 클러스터의 서비스 긴급도(urgency factor)를 나타내는 수치이다. 이 값은 지연 허용치(T_i)에서 가중치를 결정할 시점에 이미 지연된 시간(t_i)과 클러스터 c_i^j의 지연값인 $(k_i-j)\cdot P_c$ 만큼의 시간을 뺀 값으로 한 사이클 주기를 나눈 것이다. 클러스터 c_i^j의 서비스 긴급도를 u_i^j라 하면, $u_i^j = P_c/\{T_i - t_i - (k_i-j)\cdot P_c\}$로 표현된다. 따라서, 클래스 i의 경우 각 클러스터에 대한 긴급도는 $u_i^1 > u_i^2 > \cdots > u_i^{k_i}$의 관계를 가진다. 식 (1)을 서비스 긴급도에 대해서 다시 쓰면 $w_i = \max\{\lceil s_i^{k_i}\cdot u_i^{k_i}\rceil, \lceil s_i^{k_i-1}\cdot u_i^{k_i-1}\rceil, \cdots, \lceil s_i^1\cdot u_i^1\rceil\}$와 같이 나타낼 수 있다.

u_i^j의 역수를 취하면, 해당 클러스터 내의 모든 셀들을 서비스해야 할 여유 사이클 수를 나타내는데, $0 < (u_i^j)^{-1} < k_i$ 범위의 값을 가진다. 결국, 각 클러스터에 대한 가중치는, 해당 클러스터에 있는 셀 수를, 그 클러스터 내의 모든 셀들을 서비스해야 할 여유 사이클 수로 나눈 값이다. 가중치는 QoS를 만족하면서 정수값을 가져야 하므로 계산된 값보다 같거나 큰 최소 정수를 택하기 위해 실링(ceiling) 함수를 사용한다.

식 (1)에 의해서 실시간 클래스들의 가중치 값들이 결정되면, 한 사이클 주기에서 이들 실시간 클래스들을 서비스하고 남는 여분의 서비스 시간은 비실시간 클래스를 서비스하는데 사용된다. 따라서, 비실시간 클래스의 가중치 값은

$$w_N = P_c - \sum_{i=1}^{N-1} w_i \tag{2}$$

와 같이 결정된다. 만약 $w_N < 0$ 인 경우, 즉 실시간 클래스들의 가중치 총합이 한 사이클 내에 처리할 수 있는 셀 수를 초과하는 경우에는 클래스들의 가중치들을 다시 조정해야 한다. 이 경우, 실시간 클래스들은 그 가중치 총합이 처리되는 시간이 P_c가 되도록 각 가중치의 상대적인 크기에 해당하는 만큼의 가중치를 다시 할당받게 되고, 비실시간 클래스는 그 가중치가 '0'으로 초기화된다. 이를 식으로 표현하면,

$$w_i' = \min\left\{\left\lceil\left(w_i \Big/ \sum_{m=1}^{N-1} w_m\right)\cdot P_c\right\rceil, r_{i-1}\right\} \quad \text{for } i=1, 2, \cdots, N \tag{3}$$

과 같고, 여기서 $r_{i-1} = P_c - \sum_{m=1}^{i-1} w_m'$ 이다.

실시간 클래스의 경우, 식 (3)에 의해 조정된 가중치 값이 해당 사이클에서 계산된 최소 가중치 값(식 (4)에서 첫 번째 항)보다 작으면 그 차이 만큼에 해당하는 셀들을 큐에서 폐기시킨다. 이것은 조정된 가중치가 최소 가중치보다 작을 경우, 모든 클러스터의 QoS가 만족되지 못하게 되므로 과다한 지연으로 인한 셀손실이 불가피하기 때문이다. 따라서, 제안한 기법에서는 이러한 경우에 발생하는 실시간 트래픽의 QoS 열화를 줄이기 위해, 지연 요구사항을 만족하지 못할 셀들을 미리 폐기시킨다. 폐기되는 셀의 개수를 d_i라 하면,

$$d_i = \min\left\{\left\lceil\frac{s_i^{k_i}\cdot P_c}{T_i - t_i}\right\rceil, \left\lceil\frac{s_i^{k_i-1}\cdot P_c}{T_i - t_i - P_c}\right\rceil, \cdots, \left\lceil\frac{s_i^1\cdot P_c}{T_i - t_i - (k_i-1)\cdot P_c}\right\rceil\right\} - w_i' \tag{4}$$

와 같이 결정되고, 여기에서 $k_i = \lfloor T_i/P_c \rfloor$, $i = 1, 2, \cdots, N-1$ 이며, $d_i > 0$ 이다. d_i는 앞에서 언급한 바와 같이 최소 가중치와 조정된 가중치의 차에 해당하는 값이다. 이와 같은 셀의 폐기는 시스템에 과중한 트래픽이 입력되는 경우에 서비스 품질 열화가 누적되는 것을 막기 위해서 어차피 서비스 품질을 보장할 수 없다고 판단되는 셀들을 서비스하지 않고 버림으로써 이어서 입력되는 셀들에까지 서비스 품질 열화가 계속 이어지는 현상을 줄여준다. 실시간 서비스의 경우 지연 시간 내에 서비스가 되지 않으면 그 셀이 수신단에 도착하여도 재생(playout)되지 못하고 결국 폐기되기 때문에, 이러한

셀 폐기 메커니즘을 스케줄러에 적용함으로써 전체적인 QoS 향상을 가져올 수 있다. 표 1은 본 논문에서 사용되는 표기를 설명해 놓은 것이다.

표 1. AWRR/DT에서 사용되는 표기

표기 (Notation)	의 미
T_i	클래스 i의 지연 허용치
t_i	클래스 i의 타이머(값)
P_c	한 사이클의 주기
w_i	클래스 i의 가중치
w_i'	조정된 클래스 i의 가중치
s_i^j	클래스 i에서 이어지는 j사이클 내에 서비스 받아야 하는 셀들의 수
c_i^j	클래스 i의 j번째 클러스터, s_i^j개의 셀들을 포함한다.
u_i^j	클러스터 c_i^j의 서비스 긴급도
k_i	클래스 i에서 카운터 변수들의 개수
d_i	클래스 i에서 폐기될 셀들의 개수

스케줄러의 동작은 일반적인 라운드 로빈(round robin) 방법인 클래스를 돌아가면서 하나씩 처리해 주는 방법 대신에, 클래스를 돌아가면서 하나씩 서비스해 주되 실시간 클래스들을 먼저 이러한 방법으로 처리해 주고 비실시간 클래스에 할당된 가중치는 매 사이클의 마지막에 한꺼번에 처리되도록 하였다. 이렇게 함으로써, 매 사이클에서 실시간 클래스의 처리시점이 조금씩 앞당겨지게 된다.

그림 2는 제안한 알고리즘에서의 가중치 결정 과정을 간략하게 보인 것이다.

Step 1 : Initialization
Calculate k_i for $i=1,2,\cdots,N-1$.
Initialize t_i and s_i^j
for $i=1,2,\cdots,N-1$ and $j=1,2,\cdots,k_i$.
- **Do the following steps (Steps 2, 3, and 4) at the beginning of every cycle.**

Step 2 : Update s_i^j and determine w_i.
1) Update s_i^j for $i=1,2,\cdots,N-1$ and $j=2,3,\cdots,k_i$.
① $s_i^{j-1} = s_i^j - [$이전 사이클에서 서비스를 받거나 폐기된 셀의 수$]_i$
② $s_i^{k_i}$ = 현재 클래스 i에 입력되어 있는 셀의 총수
2) Determine w_i using (1) and (2).

Step 3 : If the sum of real-time weights is greater than P_c, redetermine the weights using (3).

Step 4 : If the redetermined weight of real-time class i is less than the minimum weight, determine the number of cells to be discarded from the class using (4).

그림 2. AWRR/DT 알고리즘에서의 가중치 결정

Ⅲ. 성능 분석 및 평가

이 장에서는 시뮬레이션을 통해 본 논문에서 제안한 AWRR/DT 기법과 기존 스케줄링 기법인 MLT, HOL-PJ를 비교한다.

1. 클래스와 입력 소스 트래픽 모델링

AWRR/DT, MLT와 HOL-PJ 스케줄러는 모두 두 개의 실시간 트래픽 클래스(클래스 1, 클래스 2)와 하나의 비실시간 트래픽 클래스(클래스 3)를 갖는다. 클래스 1의 입력 트래픽은 10개의 ON-OFF 소스로부터 발생되어 다중화되며, 발생 초기에 5개의 활성화(On) 상태와 5개의 비활성화(Off) 상태를 갖는다. 이러한 형태의 전형적인 예로는 말하는 구간(On-state)과 묵음 구간(Off-state)을 가지는 음성통화를 들 수 있다. 이 때, 셀의 생성 간격은 지수분포(exponential distribution)를 따르고 활성화 지속시간과 비활성화 지속시간도 지수 분포를 이룬다[5]. 이는 그림 3에서와 같은 2-state Markov 모델에 해당하며, 실험에서 평균값은 각각 $1/\alpha=3.0$[s], $1/\beta=3.0$[s]로 가정하였다. 클래스 2와 3의 입력 트래픽은 셀 발생 간격이 지수 분포를 따르는 하나의 입력 소스로부터 각각 발생된다. 각 클래스의 트래픽은 생성되어 해당 클래스의 큐에 입력되는데, 이 때 큐의 크기(queue size) 제한은 없는 것으로 가정하였다.

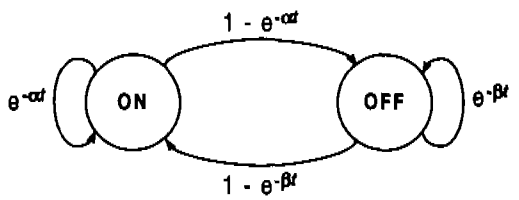


그림 3. 클래스 1 입력 트래픽을 위한 two-state Markov model

본 논문에서는 스케줄러에서 1 개의 ATM 셀을 서비스하기 위해 소요되는 시간을 셀 시간단위(ctu, cell time unit, 또는 time slot)로 정의하며 이후의 모의 실험에서 기본 단위 시간으로 사용한다. T_1은 60[ctu], T_2는 80[ctu], 그리고 P_c는 20[ctu]으로

각각 가정하였다. MLT의 경우에는 임계치를 10[ctu]으로 하였고, 클래스 1에 입력되는 셀의 초기 이완시간을 60[ctu], 클래스 2에 입력되는 셀의 초기 이완시간을 80[ctu]으로 가정하였다. HOL-PJ의 경우에는 클래스 2의 마감시간(time-out marker)을 20[ctu]으로 클래스 3의 마감시간을 50[ctu]으로 하였다. 이러한 패러미터들은 세 가지 기법들이 동일한 조건에서 비교될 수 있도록 결정되었다.

2. 시뮬레이션 환경

시뮬레이션은 앞에서 언급한 클래스와 입력 트래픽 소스 모델링을 바탕으로 Unix상에서 범용 시뮬레이션 언어인 SLAM Ⅱ를 이용하였다. 트래픽 부하(traffic load, 서버가 하나인 경우 traffic intensity: ρ)에 따른 각 클래스별 트래픽 비율은 1:1:2 (실시간:비실시간 = 1:1)로 가정하였다.

3. 성능 분석 지표

그림 4는 실시간 및 비실시간 클래스의 지연 특성(여기서는 확률밀도 함수)을 개념적으로 도시한 것이다. 그림 4에서, $f_r(t)$와 $f_n(t)$는 각각 실시간 클래스와 비실시간 클래스의 지연에 관한 확률밀도 함수를 나타낸다. 여기서, 실시간 클래스의 경우에는 각각의 셀들의 지연(정확히 말하면, 각 큐에 입력된 셀들이 서비스 받기 전까지 대기한 시간)이 해당 클래스의 지연 허용범위인 지연 허용치보다 크지 않으면 된다. 이는 실시간 서비스의 경우는 일반적으로 종단에서 사람이 듣거나 보는 것들이 대부분이어서 사람이 느끼지 못할 정도의 지연 안에서만 서비스가 이루어지면 서비스 품질에 아무런 문제가 없기 때문이다. 실시간 클래스의 경우 무조건 빨리 전송한다고 해서 서비스 품질이 향상되는 것은 아니다. 물론, 그림 4 (a)에서 지연 허용치를 넘어서 서비스 받는 셀들은 서비스 품질을 만족하지 못할 수 있으므로 최대한 억제되어야 한다. 반면에 그림 4 (b)와 같은 비실시간 클래스의 경우에는 가능한 빨리 전송해 줄수록 서비스 품질이 높아진다고 볼 수 있다.

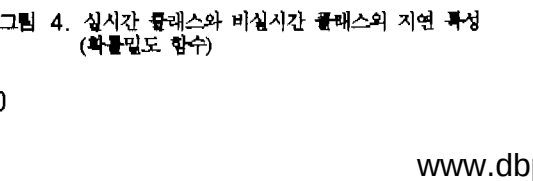


그림 4. 실시간 클래스와 비실시간 클래스의 지연 특성 (확률밀도 함수)

4. 성능 평가

그림 5, 6, 7은 입력 트래픽의 부하를 출력링크 대역폭 대비 0.1에서 1.2까지 증가시키면서 각 기법들을 적용했을 때의 평균 지연값을 비교한 결과이다. 그림 5는 가장 우선순위가 높은 클래스 1에 대한 전송지연을 비교한 결과로서, 세 기법 모두 최대 지연 허용범위인 60[ctu]보다 낮은 지연 시간을 보였다. 그림 6은 클래스 2에 대한 각 기법들의 평균 지연을 나타낸 것으로서, 모두 최대 지연 허용범위인 80[ctu]보다 낮은 지연 시간을 보이고 있다. 일반적으로 지연은 작을수록 좋으나, 실시간 클래스의 경우 지연 허용치 내에서만 서비스가 되면 QoS에 문제가 없다. 그림에서 HOL-PJ와 같은 기법은 실시간 클래스에 절대적인 서비스 우선권을 주어 지나치게 실시간 클래스의 지연이 작게 나타나는데, 이는 다른 트래픽 클래스의 지연이 그만큼 커지게 된다는 것을 의미한다. 제안한 기법의 경우 지연값이 MLT에 비해 다소 적은 수준에서 유지되고 있음을 알 수 있다 (그림 5, 그림 6 참조).

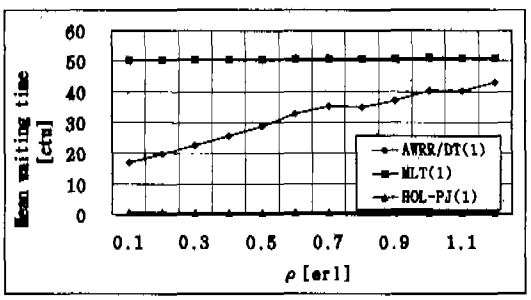


그림 5. 트래픽 부하에 따른 클래스 1의 평균 waiting time

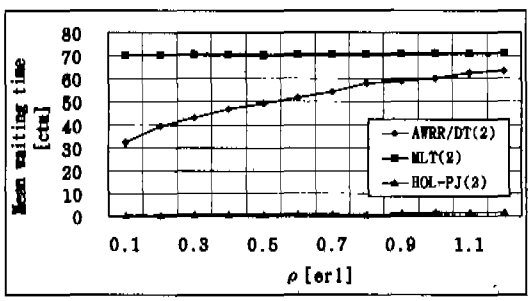


그림 6. 트래픽 부하에 따른 클래스 2의 평균 waiting time

그림 7은 비실시간 클래스(클래스 3)의 평균 지연을 나타낸 그래프로서 제안한 기법이 HOL-PJ보다 평균 지연이 작음을 알 수 있다. MLT의 경우에

는 제안한 AWRR/DT에 비해 비실시간 클래스의 평균 지연 시간이 더 짧은데, 이것은 MLT가 이완 시간이 작은 셀을 먼저 서비스 해 주는 메커니즘을 갖고 있어서 실시간 클래스들의 평균 지연이 지연 허용치에 거의 근접하여 나타나고 이의 결과로 비실시간 클래스의 평균 지연이 작아지는 것이다. 하지만, 기존의 MLT나 HOL-PJ와 같은 기법들은 패킷 하나 하나에 대해서 이완 시간이나 도착 시각을 처리하는데 필요한 프로세싱 오버헤드로 인해 실제로는 추가적인 지연이 발생할 수 있고, 이러한 문제는 망의 트래픽 로드가 크면 클수록 더욱 가중된다.

표 2는 제안한 AWRR/DT에서 셀 폐기 기법을 적용했을 때와 이를 적용하지 않았을 경우의 평균 지연 시간과 셀손실율을 비교한 결과이다. 클래스별 트래픽 비율이 1:1:2인 경우는 셀손실이 거의 발생하지 않아서 실시간 트래픽이 좀 더 많은 경우(1:1:1)를 비교하였다. 표에서 나타난 바와 같이 주어진 지연 허용치 내에 서비스받지 못할 셀들을 미리 폐기함으로써 나머지 셀들의 평균지연은 물론 전체적인 셀손실율(셀 폐기 메커니즘에 의해 폐기된 셀들을 포함)도 셀 폐기 메커니즘을 사용하지 않는 경우에 비해 감소하게 된다.

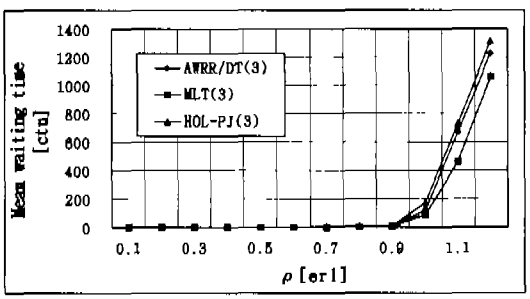


그림 7. 트래픽 부하에 따른 클래스 3의 평균 waiting time

표 2. 셀 폐기 메커니즘의 사용에 따른 평균 지연 시간과 셀 손실률 비교

AWRR/DT	셀 폐기 메커니즘 사용하지 않음				셀 폐기 메커니즘 사용			
Class 구분	Class I		Class II		Class I		Class II	
Input traffic ratio (1:1:1)	Mean waiting time [ctu]	Cell loss ratio (%)	Mean waiting time [ctu]	Cell loss ratio (%)	Mean waiting time [ctu]	Cell loss ratio (%)	Mean waiting time [ctu]	Cell loss ratio (%)
ρ=1.0	44.23	0.49	63.53	0.49	44.12	0.42	63.30	0.52
ρ=1.1	47.88	5.31	66.08	2.47	47.05	2.79	63.40	0.60
ρ=1.2	48.72	6.08	67.97	4.28	48.00	3.02	67.11	1.73

Ⅳ. 결 론

본 논문에서는 ATM 네트워크에서 다양한 형태의 트래픽들을 동적으로 다중화하는 셀 스케줄링 기법인 AWRR/DT 알고리즘을 제안하였다. 제안된 알고리즘은 기존의 MLT나 HOL-PJ등의 기법처럼 패킷 하나 하나에 대해 이완 시간이나 도착 시각을 관리할 필요가 없어서 ATM 네트워크와 같은 고속 교환망에서도 응용될 수 있는 장점을 가지면서도, 망의 현재 트래픽 발생 상황과 지연특성을 고려하여 셀을 스케줄링함으로써 각 트래픽의 서비스 요구 조건을 충분히 만족시켜 줄 수 있다. 시뮬레이션을 통해 제안된 알고리즘이 실시간 클래스의 셀 손실률을 QoS에 영향을 주지 않는 정도에서 제한하면서, 비실시간 클래스의 평균 지연을 줄여줄 수 있음을 확인하였다. 또한, 제안한 기법은 망 트래픽 상태에 동적으로 대처할 수 있으며, 과중한 트래픽 입력에 대한 셀 폐기 기법을 포함하여 서비스 품질 열화가 연속적으로 일어나는 것을 막을 수 있다.

참 고 문 헌

[1] D. Hong and T. Suda, "Congestion Control and Prevention in ATM Networks," *IEEE Network Magazine*, Vol. 5, pp. 10-16. Jul. 1991.

[2] Y. Lim and J. E. Kobza, "Analysis of a Delay-Dependent priority Discipline in an Integrated Muticlass traffic Fast packet Switch," *IEEE Trans. on Communications*, Vol. 38, No. 5, pp. 659-665, May 1990.

[3] R. Chipalkatti and J. F. Kurose, "Scheduling Policies for Real-Time and Non-Real-Time Traffic in a Statistical Multiplexer," *Proceedings of IEEE Infocom '89*, pp. 774-783, April 1989.

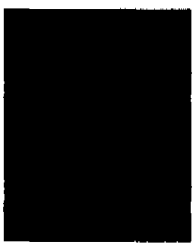
[4] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," *IEEE JSAC*, Vol. 9, No. 8, pp. 1265-1279, Oct. 1991.

[5] J. N. Daigle and J. D. Langford, "Models for Analysis of Packet Voice Communication systems," *IEEE JSAC*, Vol. 4, No 6, pp. 847-855, 1986.

심 재 정(Jae-Jeong Shim) 정회원

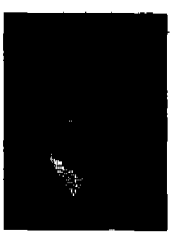
1995년 : 고려대학교 전자공학과 졸업(공학사)
1997년 : 고려대학교 대학원 전자공학과 졸업 (공학석사)
1997년 9월~현재 : 고려대학교 대학원 전자공학과 박사 과정, 고려대학교 부설 정보·통신기술공동연구소 연구원
<주관심 분야> B-ISDN/ATM, 유무선 통신망, 이동통신, 멀티미디어

이 원 호(Won-Ho Lee) 준회원



1998년 : 고려대학교 전자공학과 (공학사)
2000년 2월 : 고려대학교 대학원 메카트로닉스학과 (공학석사)
2000년 2월~현재 : 데이콤 초고속인터넷사업팀
<주관심 분야> 멀티미디어 통신, 초고속인터넷

변 재 영(Jae-Young Pyun) 정회원



1997년 : 조선대학교 전자공학과 (공학사)
1999년 : 전남대학교 대학원 전자공학과 (공학석사)
1999년 3월~현재 : 고려대학교 대학원 전자공학과 박사과정
<주관심 분야> ATM, 멀티미디어 통신

고 성 제(Sung-Jea Ko) 정회원

1980년 2월 : 고려대학교 전자공학과(공학사)
1986년 5월 : State Univ. of New York at Buffalo, 전기 및 컴퓨터공학과(공학석사)
1988년 8월 : State Univ. of New York at Buffalo, 전기 및 컴퓨터공학과(공학박사)
1981년 8월~1983년 12월 : 대한전선 중앙 연구소 연구원
1988년 8월~1992년 5월 : The Univ. of Michigan Dearbon, 전기 및 컴퓨터 공학과 조교수
1992년 3월~현재 : 고려대학교 전자공학과 교수
1996년 11월 : IEEE APCCAS best paper award
1997년 12월 : 대한전자공학회 해동논문상 수상
1999년 11월 : 한국통신학회 LG 학술상 IEEE Senior member, IEE member
<주관심 분야> 신호 및 영상 처리, 영상 압축 및 통신, 멀티미디어 통신