

ATM 스위치에서 셀 손실률 감소를 위한 개선된 동적 임계치 기법

김 양 섭[†] · 권 혁 인^{††} · 김 영 찬^{†††}

요 약

출력큐잉을 하는 공유 메모리 ATM 스위치에서 출력 큐들은 공용 셀 버퍼내의 공간을 유동적으로 공유한다. 따라서, 공유메모리 ATM 스위치에서는 서로 다른 출력 큐 사이에 메모리를 공평하게 사용할 수 있도록 버퍼관리 기법이 필요하다. 전통적인 버퍼관리 기법중에 정적 임계치 기법은 구현하기가 간단하지만 효율적이지 않고 변화하는 트래픽 조건에 적응성을 갖지 못한다. 푸쉬아웃 기법은 효율적이며 적응성을 가지고 있으나 구현하기가 매우 힘들다 동적 임계치 기법은 정적 임계치의 단순성과 푸쉬아웃의 적응성의 장점을 함께 가지고 있다.

VBR 트래픽의 경우에 소소는 높은 우선순위와 낮은 우선순위 셀을 생성한다 그러나, 동적 임계치 기법에서는 과부하 기간동안 높은 우선순위 셀을 보호하지 않는다 이는 공유 메모리 ATM 스위치의 성능의 저하를 야기한다. 본 논문에서는 동적 임계치 기법과 비선택적인 푸쉬아웃 기법을 혼합한 버퍼관리 기법을 제안한다. 이 기법은 개선된 동적 임계치 기법으로, 과부하 기간동안 높은 우선순위 셀을 보호한다. 또한, 비퍼 오버플로우가 발생하는 시점에서 비선택적인 푸쉬아웃 기법이 적용된다

Advanced Dynamic Threshold Buffer Management Scheme for Reducing Cell Loss Rate in ATM Switch

Yang-Seob Kim[†] · Hyuck-In Kwon^{††} · Young-Chan Kim^{†††}

ABSTRACT

In a shared memory ATM switch with output queueing, queues share space flexibly in the common cell buffer. Buffer management schemes are needed to fairly regulate the sharing of memory among different output port queues. Of the conventional schemes, Static Threshold is simple but not efficient and does not adapt to the changing of traffic conditions. Pushout is the most efficient and adaptive scheme but very difficult to implement. Dynamic Threshold scheme combines the simplicity of Static Threshold and the adaptability of Pushout.

In the VBR traffic, both high priority and low priority cells are generated. However, in Dynamic threshold, high priority cells are not protected when overload occurs. It consequently degrades the performance of the shared memory ATM switch. In this paper, we propose a buffer management scheme which combines the Dynamic Threshold and the Non-Selective Pushout. This scheme is called an Advanced Dynamic Threshold. In this scheme, high priority cells are protected from being discarded when overload condition happens. When overflow occurs in the buffer, Non-Selective Pushout is applied.

※ 이 연구는 1996년도 한국학술진흥재단의 자유공모제 연구비의 지원을 받음

† 정 회 원 : 중일대학교 대학원 컴퓨터공학과

†† 동 심 회 원 : 중일대학교 경영학부 교수

††† 정 회 원 : 중일대학교 컴퓨터공학과 교수

논문접수 : 1999년 3월 18일, 심사완료 : 2000년 1월 10일

1. 서론

현재까지 문헌상에서 제안되어 온 대부분의 ATM 스위치 구조는 스위치 내의 여러 자원에 대한 경쟁으로 인해 서비스가 지연되는 셀을 수용하기 위하여 몇 가지 버퍼링 기법을 사용하고 있다. 여기서 비퍼의 위치와 비퍼 관리 기법은 스위치의 전반적인 성능에 직접적으로 많은 영향을 끼치게 된다. 여러 논문에서 입력 큐잉, 출력 큐잉 및 완전하게 공유되는 버퍼링을 기반으로 한 스위치에 대해서 제안되고 연구[1]되었으며 출력 큐잉과 완전하게 공유되는 버퍼링이 지연에 민감한 트래픽의 경우에 최적의 성능을 보이는 것으로 결론지어졌다.

그러나, 아무런 비퍼 관리 기법도 사용하지 않은 출력 큐잉 공유 메모리 ATM 스위치는 파부하 조건하에서 좋은 성능을 나타내지 않는다. 그 이유는 하나의 출력 포트가 메모리의 대부분을 차지하기 때문이다. 이는 전반적인 스위치의 처리율을 저하시키고 스위치 내에서의 혼잡을 야기하며 결국 셀 손실을 증가시키게 된다. 이 문제에 대한 해결책은 비퍼 사용을 제한함으로써 셀 당 비퍼 점유시간을 줄이는 것이다.

여러 논문에서 여러 가지 종류의 제한 비퍼링 기법이 제안되었다[2, 4, 5, 6, 11] 첫 번째 유형은 개별 큐가 이용할 수 있는 비퍼의 최대 또는 최소량에 제한을 두는 방안이다[2, 11] 이 기법은 대개 정적 임계치 기법이라 불린다 정적 임계치 기법은 구현하기가 매우 간단하다. 그러나, 트래픽 조건에 따라 적응성이 없기 때문에 성능이 다소 떨어지는 면이 있다.

두 번째 유형은 푸쉬아웃[2, 4, 5]이라는 기법이다. 푸쉬아웃에는 선택적인 푸쉬아웃과 비선택적인 푸쉬아웃이 있다. 각각은 푸쉬아웃을 수행할 때 푸쉬아웃되는 셀을 선택하는 기준에 따라 구분된다. 푸쉬아웃 기법은 공평성과 적응성이라는 관점에서 매우 좋은 성능을 제공하지만, 다른 기법에 비해 상대적으로 구현하기가 복잡하다는 단점이 있다.

정적 임계치의 단순성과 푸쉬아웃의 적응성을 갖춘 비퍼 관리 기법은 동적 임계치라 부르며, 출력포트별 논리적인 큐의 길이를 제어 임계치를 사용해 제한하고 있다[6]. 동적 임계치 기법은 정적 임계치 기법에 비해 더 적응성이 뛰어나며 따라서 더욱 효율성이 뛰어나다 또한, 정적 임계치 기법에 비해 상대적으로 더 적은 비퍼 공간을 낭비하며 공평성 부분에서 더 좋은 성

능을 나타내고 있다 반면에 푸쉬아웃에 비해 상대적으로 적응성이 떨어지는 면이 있으나, 푸쉬아웃을 구현하기가 복잡하데 비해 동적 임계치는 구현이 간단하다는 장점이 있다.

이러한 여러 장점에도 불구하고 동적 임계치 기법은 가장 중요한 문제점을 내포하고 있다. 먼저 동적 임계치 기법은 높은 우선순위 셀에 대한 셀 손실률을 고려하지 않고 있다 일반적으로 VBR 트래픽은 높은 우선순위 셀과 낮은 우선순위 셀을 구분하여 생성함으로써 그 셀이 목적지까지 도달하는 중간에 혼잡이 발생했을 때, 낮은 우선순위 셀을 먼저 버려서 높은 우선순위 셀을 보호하도록 하고 있다. 이와 같은 높은 우선순위 셀은 최대한 보호되어야 한다.

그러나, 기존의 동적 임계치 기법에서 단순하게 높은 우선순위 셀을 보호하게 되면 다른 문제를 야기한다 기존의 동적 임계치 기법에서는 비퍼 오버플로우가 발생한 시점에 도착하는 모든 셀을 무조건 버린다. 그러므로, 단순하게 제어 임계치를 초과한 시점에 도착한 높은 우선순위 셀을 버리지 않고 큐에 수용하게 되면 사용가능한 비퍼 공간이 급격히 줄어들게 되고 오버플로우가 발생하게 된다 이와 같은 오버플로우가 발생한 시점에 휴면상태에 있던 큐에 갑자기 셀이 집중하게 되면 이 큐에 도착하는 셀은 모두 버리게 된다. 이는 특정 큐가 공유 메모리를 모두 독점하는 현상으로 전개될 수도 있다.

본 논문에서 제안하고 있는 개선된 동적 임계치 기법은 높은 우선순위 셀을 최대한으로 보호하는 데에 주안점을 두고 있다. 즉, 제어 임계치를 초과한 시점에 도착한 셀이 높은 우선순위를 가지면 비퍼에 수용하는 것이다. 이때, 발생할 수 있는 특정 큐의 공유 메모리 독점을 방지하기 위해 비퍼 오버플로우가 발생하는 시점부터 비선택적인 푸쉬아웃 기법을 적용한다. 즉, 비퍼가 꽉 차 있을 때, 높은 우선순위 셀이 도착하면 비퍼 내의 여러 큐 중 길이가 가장 긴 큐의 헤더에서 셀을 하나 버리고 목적지 큐에 셀을 수용하게 된다. 이와 같은 정책을 적용함으로써 개선된 동적 임계치 기법은 기존의 동적 임계치 기법의 장점을 그대로 유지하며 스위치의 성능을 더욱 향상시킬 수 있다.

본 논문은 다음과 같이 구성된다. 2절에서는 스위치의 비퍼 내에서 동적 임계치 비퍼 관리 기법을 포함하여 현재까지 논의되어 온 비퍼 관리 기법에 대해 다룬다. 3절에서는 기존의 비퍼 관리 기법에서 발생되었던

문제점과 그러한 문제점을 해결하기 위한 개선된 동적 임계치 버퍼 관리 기법에 대해 설명한다. 4절은 개선된 동적 임계치 버퍼 관리 기법의 성능 평가를 위해 실험하기 위한 스위치 모델과 트래픽 모델에 대해 다루며 5, 6절은 그 실험 결과 및 결론을 맺는다.

2. 기존의 버퍼 관리 기법

2.1 정적 임계치 기법

이 기법에서는 만약 낮은 우선순위의 셀이 도착할 때 현재의 큐 길이가 주어진 임계치 보다 작을 때에만 큐에 저장된다[2, 11]. 임계치 기법은 크게 두가지 방식으로 사용된다. 첫 번째로, 전역 임계치(Global Threshold)는 임계치를 공유 메모리 전체 크기에 대해 적용한다 즉, 공유 메모리 전체에 대해 하나의 임계치를 설정하고 공유 메모리 내의 전체 큐 길이가 전역 임계치 보다 적을 때에만 버퍼에 저장된다.

두 번째로, 개별 임계치(Individual Threshold)는 각각의 출력 버퍼 별 논리적인 큐에 개별 임계치를 설정하는 방법이다. 예를 들어, i 번째 출력 포트에 셀이 도착했을 때, i 번째 논리적인 큐의 길이가 임계치보다 작을 경우에만 큐에 저장된다.

이와 같은 임계치 기법은 여러 기법들 중에서 가장 간단하다는 장점이 있는 반면 적절한 임계치를 설정하기가 어렵고 트래픽 상황에 따라 적응성이 떨어진다는 단점이 있다. 즉, 임계치를 어떻게 설정하느냐에 따라 전체 스위치의 성능이 크게 좌우된다는 것이다.

2.2 푸쉬아웃 기법

푸쉬아웃은 여러 출력 큐가 전체 버퍼 공간을 모두 공유할 수 있도록 하며, 일단 버퍼가 완전히 차 있지 않으면 도착한 모든 셀(우선순위에 관계없이)은 큐에 저장될 수 있다[2, 4, 5]. 하지만, 버퍼가 꽉 차 있을 때, 도착하는 셀은 버퍼 내에 이미 저장되어 있는 셀을 밀어내고(Pushing Out) 그 여유 공간을 차지한다. 푸쉬아웃은 이와 같은 기본정책을 따르고 있으며 버퍼가 꽉 차 있을 때 그 대상을 정하는 방식에 따라 선택적인 푸쉬아웃(Selective Pushout)과 비선택적인 푸쉬아웃(NonSelective Pushout) 등의 두 가지 형태로 구분된다.

선택적인 푸쉬아웃의 경우, shb는 우선순위의 셀이 버퍼에 도착할 때 버퍼가 꽉 차 있으면 버퍼 내에 있

는 낮은 우선순위 셀을 밀어내고 그 셀의 자리를 차지한다. 이때, shb는 우선순위의 셀이 낮은 우선순위가 차지하고 있는 물리적인 메모리 위치를 빼앗을 수 있지만 논리적인 FIFO 큐 내에서의 자리를 차지할 수 있는 것은 아니다. 즉, 자리를 빼앗고 빼앗기는 셀은 목적지가 서로 다른 출력 포트에 향하고 서로 다른 논리적인 FIFO 큐에 속할 수도 있다는 것이다. 또한, 높은 우선순위의 셀은 자신의 목적지 출력 포트의 논리적인 FIFO 큐의 맨 끝에 추가된다.

노출한 높은 우선순위 셀이 낮은 우선순위 셀 중의 하나를 희생양으로 선택할 때 여러 조건을 적용할 수 있다. 첫 번째로, 높은 우선순위 셀이 낮은 우선순위 셀을 푸쉬아웃할 때 그 셀을 여러 큐 중 가장 길이가 긴 큐에서 셀을 선택한다. 이렇게 함으로써, 짧은 길이의 큐에게 더 많은 길이를 확보할 수 있는 기회를 줄 수 있고 출력 큐 사이의 버퍼 공간을 공유하는데 좀 더 높은 공평성을 유지할 수 있다. 또한, 더 많은 메모리 출력 포트를 활성화시킴으로써 스위치의 효율성을 증가시킬 수 있다.

만약 가장 긴 큐에 낮은 우선순위 셀이 하나 이상 있을 때, 큐의 헤더에서 가장 가까운 곳에 있는 셀이 푸쉬아웃 된다. 만약 가장 긴 큐에 낮은 우선순위 셀이 없을 경우, 큐의 헤더에서 가장 가까운 곳에 있는 높은 우선순위 셀을 푸쉬아웃한다. 이는 높은 우선순위 셀에 대한 손실률을 향상시키지는 않지만 전체적인 큐 길이의 균등화에 도움이 된다. 같은 목적으로 만약 낮은 우선순위 셀이 도착했을 때 버퍼가 꽉 찬 경우에도 가장 긴 큐에 있는 낮은 우선순위 셀을 푸쉬아웃한다. 이때, 낮은 우선순위 셀이 없을 경우 도착한 셀을 버린다.

선택적인 푸쉬아웃 기법을 구현하기 위해, 스위치는 공유 메모리 버퍼에 저장되어 있는 각 셀의 우선순위를 추적하고 있어야 한다. 또한 FIFO 큐의 중간에서 낮은 우선순위 셀이 푸쉬아웃 될 때 출력 포트에 대한 논리적인 FIFO 큐를 조정할 수 있도록 충분한 개수의 포인터를 유지해야 한다. 이와 같은 문제점(구현상 복잡성)으로 인해 비선택적인 푸쉬아웃 기법이 제안되었다.

이 기법에선, 높은 우선순위 셀이 버퍼에 도착할 당시 버퍼가 꽉 차 있을 경우, 우선순위를 고려하지 않고 가장 긴 큐의 헤더에 있는 셀을 푸쉬아웃 시킨다. 또한, 낮은 우선순위 셀이 도착할 당시에 버퍼가 꽉 차 있을 경우, 단순히 그 셀을 버린다. 이 기법을 사용할 경

우, 버퍼 내에 낮은 우선순위 셀이 존재하더라도 높은 우선순위의 셀이 푸쉬아웃 될 수가 있다. 비록 비선형적인 푸쉬아웃이 선택적인 푸쉬아웃에 비해 높은 우선순위 셀에 대한 손실률 관점에서는 좋은 성능을 보여 주지 않지만, 출력 버퍼 큐의 균등화에는 도움을 주고 있으며 상대적으로 구현이 간단하다는 장점이 있다.

2.3 동적 임계치 기법

만약 스위치 내의 여러 출력 포트 중 오직 하나의 포트만이 활성 중일 때, 가능하면 많은 공유 메모리 공간을 할당해 주는 것이 좋다. 반면에, 다수의 출력 버퍼가 모두 활성일 경우에는 공유 메모리 공간을 모든 큐에 공평하게 할당해 주는 것이 좋다. 이와 같은 조건을 만족시키기 위해 제어 임계치(control threshold)라는 개념이 제시되었다 여기서의 제어 임계치란 사용되지 않고 있는 버퍼 공간의 전체크기라 할 수 있다. 이 때의 제어 임계치는 일종의 개별 임계치와 유사한 개념이지만 둘의 차이점은 동적으로 변화느냐 변하지 않느냐에 따라 구분이 된다. 즉, 제어 임계치는 때 타임 슬롯 마다 변하기 때문에 동적이다

각각의 출력 포트별 큐는 함수 f 에 의해 자신의 큐의 길이를 제한한다. 여기서 함수 f 는 사용되지 않고 있는 버퍼 공간의 전체크기인 제어 임계치를 구하는 함수이다. 다음은 매 시간 t 마다 제어 임계치 $T(t)$ 를 구하는 함수를 나타낸다[6].

$$T(t) = f(B - Q(t)) = f(B - \sum_i Q^i(t)) \quad (1)$$

여기서 $Q^i(t)$ 는 i 번째 출력 포트에 대한 큐의 길이, $Q(t)$ 는 현재 사용중인 전체 큐의 길이, B 는 전체 버퍼 공간의 크기를 의미한다.

만약 시간 t 시점에 i 번째 포트에 셀이 도착했을 때 $Q^i(t) \geq T(t)$ 이면 그 셀은 버려지고 반대일 경우 그 셀은 i 번째 큐에 저장된다. 이 때, 이 큐에 도착하는 모든 셀은 큐의 길이가 제어 임계치보다 짧아질 때까지 계속해서 버려진다.

제어 임계치를 구하기 위한 함수 f 로서 가장 간단한 방법은 단순히 α 를 비사용중인 버퍼 공간에 곱하는 것이다. 그 식은 다음과 같다.

$$T(t) = \alpha \cdot (B - Q(t)) = \alpha \cdot (B - \sum_i Q^i(t)) \quad (2)$$

여기서 α 가 2의 배수이면, 단지 쉬프터(shiftter)만으

로 임계치 계산을 할 수 있기 때문에 구현을 매우 간단하게 할 수 있다. 따라서, 대개의 경우 α 는 2의 배수로 사용하고 있다.

만약 버퍼에 꾸준히 트래픽이 발생하고 있는 상황 (steady-state)에서 S 개의 매우 활성중인 큐가 존재한다고 가정하면, 전체 버퍼 크기(Q)는 다음과 같을 것이다

$$Q = S \cdot T + \Omega \quad (3)$$

여기서 Ω 는 제어 임계치 보다 작은 크기의 버퍼를 차지하고 있는 큐의 길이의 합을 의미하며 T 는 steady-state 상황에서의 제어 임계치 값을 의미한다. 이 때, T 는 위 식을 (2)식에 대입함으로써 다음과 같이 구할 수 있다

$$Q' = T = \frac{\alpha \cdot (B - Q)}{1 + \alpha \cdot S} \quad (4)$$

또한, 예약된 버퍼 크기 θ 는 알고리즘에 의해 다음과 같이 구할 수 있다.

$$\theta = \frac{B - Q}{1 + \alpha \cdot S} \quad (5)$$

여기서 $\alpha=2$ 이면 제어 알고리즘에 의해 각 큐의 길이는 예약된 버퍼 크기보다 2배 정도를 차지할 수 있다. 따라서, 경쟁이 발생하지 않는 상황일 경우, 하나의 큐는 전체 버퍼의 2/3를 차지할 수 있으며 1/3는 사용되지 않은 채로 남아 있게 된다 만약, 2개의 큐가 매우 활성중인 경우에 각 큐는 2B/5의 크기를 차지할 수 있으며 B/5는 예약된 것이다. 이와 같은 식으로 8개의 큐가 매우 활성 중인 경우, 각 큐는 2B/17의 크기를 차지할 수 있으며 B/17는 예약된 채로 남아 있게 된다.

동적 임계치[6]는 트래픽 상황의 변화에 따라 계속해서 바뀌게 되어 트래픽 상황을 반영하게 된다. 예를 들어, 부하가 적은 출력 버퍼에 갑자기 트래픽이 몰리게 되면 그 버퍼의 길이가 증가하게 되고 전체 공유 메모리 점유량은 증가한다. 이 때, 제어 임계치의 값은 떨어지고 임계치를 초과한 큐에 도착한 셀은 일시적으로 버려지게 되며 버려진 공간만큼 활성중인 다른 큐에게 버퍼 공간을 제공하게 된다.

동적 임계치 기법은 적은 양의 버퍼 공간을 낭비하게 된다. 즉, 예약공간으로 남겨두게 되는데 이 공간은 실제로 두가지 기능을 위해 사용될 수 있다. 첫째로, 하나의 출력 포트가 짧은 기간동안 갑자기 활성화 될

때, 이 공간은 일종의 완충장치 역할을 하게 된다. 즉, 그 기간동안 발생할 수 있는 셀 손실을 줄일 수 있다. 둘째로, 하나의 출력 큐에 부하가 증가하고 여유 버퍼 공간을 점유하기 시작할 때, 버퍼 할당 기법에 현재 부하 조건이 변했으며 임계치 조절이 필요하다는 것을 신호를 통해 보낼 수 있다. 만약 이와 같이 여유 버퍼 공간이 없다면 부하 조건이 변경되었는 자의 여부를 결정하기 위해 각각의 출력 큐에 대한 셀 도착 속도와 손실률을 계속해서 감시해야 한다 하지만 이러한 모니터링 기법은 구현하기가 매우 복잡하다.

3. 개선된 동적 임계치 버퍼 관리 기법

3.1 기존 버퍼 관리 기법의 문제점

앞서 기술한 여러 버퍼 관리 기법은 각각의 장/단점을 가지고 있다. Choudhury는 경적 임계치 기법과 푸쉬아웃 기법의 단점을 보완하기 위해 제어 임계치를 이용한 버퍼 관리 기법을 제시하였다[6] 이 기법은 정적 임계치의 구현의 용이성과 푸쉬아웃의 높은 성능 및 적응성이라는 장점을 취합한 것으로서 망 내의 트래픽 상황에 따라 임계치를 재설정하는 정책을 채택했으나, 이 기법 역시 단점을 내포하고 있다.

VBR 트래픽의 경우, ATM 셀 내의 CLP(Cell Loss Priority) 비트를 0과 1로 우선순위를 할당해서 셀을 생성하게 된다. 이는 중간 스위치에서 혼잡이 발생했을 경우 높은 우선순위의 셀을 보호하고 낮은 우선순위의 셀을 버림으로써 높은 우선순위 셀에 대한 손실률을 줄이기 위한 방편이다. 그러나 이 논문에서는 이러한 높은 우선순위 셀에 대한 보호 정책을 아무 것도 제시하지 않고 있다.

일반적으로 VBR 트래픽은 셀이 목적지까지 도달하는 중간에 혼잡이 발생했을 때, 낮은 우선순위 셀을 먼저 버려서 높은 우선순위 셀을 보호하도록 하고 있다. 또한, 대부분의 높은 우선순위 셀은 동영상 자료인 경우에 한 장면을 표현하기 위해 반드시 필요로 하는 셀이다. 만약 이와 같은 높은 우선순위 셀 중 서로 인접해 있는 여러 개의 셀이 상실되면, 셀의 최종 도착지에서 원래 화면을 복구하기가 어려워진다. 따라서 이와같은 높은 우선순위 셀은 최대한 보호되어야 한다.

따라서, 본 논문에서 제시하고 있는 개선된 동적 임계치 버퍼 관리 기법에서는 동적 임계값을 초과한 시점에 도착한 셀의 CLP가 0이면 그 셀을 버리지 않고 큐

에 저장하는 정책을 취한다 이는 동적 임계치를 이용한 버퍼 관리 기법에서 혼잡을 위한 완충장치로서 사용하고 있는 어분의 공유 버퍼 공간을 이용함으로써 가능하다

그러나, 기존의 동적 임계치 기법에서 단순하게 높은 우선순위 셀을 보호하게 되면 다른 문제를 야기한다. 기존의 동적 임계치 기법에서는 비퍼 오버플로우가 발생한 시점에 도착하는 모든 셀을 무조건 버린다. 그러므로, 단순하게 제어 임계치를 초과한 시점에 도착한 높은 우선순위 셀을 버리지 않고 큐에 수용하게 되면 사용가능한 버퍼 공간이 급격히 줄어들게 되고 오버플로우가 발생하게 된다. 이와 같은 오버플로우가 발생한 시점에 휴먼상태에 있던 큐에 갑자기 셀이 침투하게 되면 이 큐에 도착하는 셀은 모두 버려지게 된다. 이는 특정 큐가 공유 메모리를 모두 독점하는 현상으로 발전할 수도 있다.

3.2 알고리즘

표기법	
제어 임계치	: CT
전체 버퍼 크기	: BS
i번째 큐 길이	: Q_i
사용중인 버퍼크기	: $\sum Q_i$


```

1. for each arriving cell, if (  $\sum Q_i < BS$  ) then goto 2
   else goto 5
2. if (  $Q_i < CT$  ) then goto 3
   else goto 4.
3. add this cell into i-th queue.
   goto 6
4. if ( CLP == 0 ) then add this cell into i-th queue
   else drop this cell
   goto 6.
5. if ( CLP == 0 ) then remove a cell from head in
   the longest queue and add this cell into
   i-th queue
   else drop this cell
6. end
    
```

(그림 1) 개선된 동적 임계치 알고리즘

본 논문에서 제시하고 있는 개선된 동적 임계치 기법의 알고리즘은 다음과 같다. 개선된 동적 임계치 기법은 기존의 동적 임계치 기법과 비선택적인 푸쉬아웃 기법을 혼합한 기법이다. 여기에 동적 임계치 기법에서 높은 우선순위 셀을 보호하지 않았던 점을 보완하는데에 중점을 두고 있다. 하지만 기존의 동적 임계치 기법에서 높은 우선순위 셀을 보호하는 정책을 쓰게 되면 높은 우선순위 셀을 보호하지 않는 것에 비해 버퍼가 빨리 꽂 차고 하나의 큐가 버퍼의 대부분을 차지하는 문제점이 발생한다. 이는 제어 임계치를 초과한 큐에서 버려야 할 셀이 줄이 들기 때문이다. 따라서, 높은 우선순위 셀을 보호하기 위해서는 이런 문제점에 대한 대안을 마련해야 한다.

앞에서 푸쉬아웃 기법 중 비선택적인 푸쉬아웃(NSP) 기법에 대해 설명하였다. 이 기법은 선택적인 푸쉬아웃(SP)에 비해 성능이 상대적으로 떨어지지만 SP 기법에 비해 구현이 간단하다는 장점과 다른 버퍼 관리 기법과 유사하거나 좀 더 좋은 성능을 보이고 있다.

개선된 동적 임계치 기법은 알고리즘에서처럼 높은 우선순위 셀을 보호할 경우에 발생하는 문제점을 어느 정도 해결할 수 있다. 즉, 제어 임계치를 초과한 시점부터 도착하는 셀 중 우선순위가 높은 셀을 기존 방법처럼 버리는 것이 아니라 모두 큐에 수용하며 낮은 우선순위의 셀은 기존 방법처럼 버린다.

또한, 제어 임계치를 초과하고 동시에 버퍼에서 오버플로우가 발생하는 시점부터 NSP의 알고리즘을 적용한다. 즉, 버퍼 오버플로우가 발생했을 때에 높은 우선순위 셀이 도착하면 버퍼 내의 여러 큐 중 길이가 가장 긴 큐의 헤더에서 셀을 하나 버리고 도착한 높은 우선순위 셀을 자신의 큐에 저장하게 된다. 그리고, 버퍼 오버플로우가 발생했을 때에 낮은 우선순위 셀이 도착하면 단지 그 셀을 버리면 된다.

만약 길이가 가장 긴 큐에서 버려지는(푸쉬아웃되는) 셀의 우선순위가 낮으면 스위치의 성능을 향상시킬 수 있지만, 버려지는 셀이 높은 우선순위를 갖는 셀이라면 실제 스위치의 성능에는 별다른 영향을 끼치지 못하게 된다. 하지만, 높은 우선순위 셀을 버림으로써 스위치 내의 큐들 사이에 공평성을 증대시킬 수 있다. 즉, 스위치의 버퍼를 가장 많이 사용하고 있는 큐의 길이를 줄이고 상대적으로 버퍼를 조금 사용하고 있는 큐에게 더 많은 버퍼를 할당하게 할 수 있다. 이외에, 버퍼 오버플로우가 발생했을 때, 휴면 상태에 있던 큐

에 갑자기 셀이 집중되는 경우 그 큐에 도착하는 셀이 버퍼를 차지할 수 있다는 효과를 얻을 수 있다.

이와 같은 정책을 채택함으로써 개선된 동적 임계치 기법은 기존의 기법에서 단순히 높은 우선순위 셀을 보호했을 경우에 발생할 수 있는 문제를 해결할 수가 있다. 아울러 기존의 동적 임계치 기법의 장점이었던 트래픽에의 적응성과 푸쉬아웃(SP) 기법에 비해 구현의 단순성의 장점을 그대로 유지하면서 버퍼 관리 기법의 성능을 향상시킬 수 있다.

4. 실험 모델

본 논문에서 실험을 위해 사용한 시뮬레이터는 NIST(National Institute of Standards and Technology)에서 개발한 NIST ATM Network Simulator[8]이다. 이 시뮬레이터는 ATM 망을 설계하는 도구로서 뿐만 아니라 ATM 망에서의 혼잡제어 기법과 버퍼관리 기법을 실험하기 위한 도구로서 개발되었다.

이 시뮬레이터는 시스템내에서의 전체 부하와 평균 버스트 길이 및 평균 침묵 기간 등을 페리미터로 받아서 동작하며 특정 입력 페리미터 값(부하, 평균 버스트 길이, 평균 침묵 기간 등)에 대해 매 시뮬레이션 마다 항상 동일한 결과를 보여 주게 된다.[8] 참조).

하지만, NIST ATM Network Simulator는 아무런 버퍼관리 기법도 채택하지 않은 채로 구현이 되어 있다. 따라서, 본 논문에서 제안하고 있는 개선된 동적 임계치 버퍼관리 기법의 성능을 평가하기 위해 이 시뮬레이터의 소스를 수정하였다. 수정 사항은 스위치 내에서의 버퍼관리 기법에 관한 부분으로서 실험에서 비교하고 있는 기존의 동적 임계치 기법과 푸쉬아웃 및 개선된 동적 임계치 기법의 알고리즘을 추가한 것이다.

본 논문에서 사용하고 있는 제어 임계치를 구하기 위한 함수(수식1, 2 참조)에서 α 의 값에 따라 성능은 달라지게 되며 본 논문에서는 α 의 값을 1로 가정하고 있다. 이는 타 논문[6]에서 인용한 값으로서 간단한 시뮬레이션을 위해 적합한 값으로 알려져 있다.

4.1 트래픽 모델

본 논문에서는 두개의 공간 우선순위 클래스를 가진 셀들을 고려하고 있다. 이 때, 클래스 1은 클래스 2보다 낮은 우선순위를 갖는다. 클래스 2 셀은 손실에 민감한(loss-sensitive) 셀을 나타낸다. 즉, 스위치에서 혼

잡이 발생했을 경우에 먼저 보호되어야 하는 셀을 말한다. 클래스 2는 높은 셀 손실률을 수용할 수 있는 셀을 의미한다.

성능 비교 연구를 위해, 각각의 소스가 활성화(active)와 침묵(idle) 기간을 반복하는 미스트 트래픽 모델[2, 6]을 채택하고 있다. 활성화와 침묵 기간은 매리미터 α 와 β 에 따라 기하학적으로 분포된다. 셀은 활성화 기간 동안의 연속적인 타임 슬롯에 도착한다. 또한, 각 활성화 버스트에는 적어도 하나의 셀이 전송되지만 침묵 기간에는 셀이 하나도 전송되지 않아도 된다고 가정한다. 버스트 길이는 통계적으로 독립적인 것으로 간주한다. 이 때, 주어진 α 와 β 에 대해, 평균 버스트 길이(L_b), 평균 침묵 기간(L_{idle}), 부하 ρ 사이의 관계는 다음과 같다

$$L_b = \sum_{k=1}^{\infty} k \cdot \alpha \cdot (1-\alpha)^{k-1} = \frac{1}{\alpha}$$

$$L_{idle} = \sum_{k=0}^{\infty} k \cdot \beta \cdot (1-\beta)^k = \frac{1-\beta}{\beta}$$

$$\rho = \frac{L_b}{L_{idle} + L_b}$$

위 식에서, 주어진 평균 미스트 길이 L_b , 부하 ρ 는 버스트 사이의 평균 침묵 기간을 변경함으로써 바뀐다

각 소스는 평균 ρ_1 의 속도로 낮은 우선순위의 셀과 평균 ρ_2 의 속도로 높은 우선순위의 셀을 생성한다. 이 때, $\rho_1 + \rho_2 = \rho$ 이다. 셀 우선순위를 모델링하는 데는 두가지 방법이 있다.

첫 번째는 순수 버스트 모드(pure burst mode)인데, 이 경우에는 하나의 버스트 기간에 동일한 우선순위를 가진 셀만이 생성되며 낮은 우선순위는 $\frac{\rho_1}{\rho}$ 의 확률로 높은 우선순위는 $\frac{\rho_2}{\rho}$ 의 확률로 생성된다

두 번째는 혼합 버스트 모드(mixed burst mode)인데, 이 모드에서는 각 셀 단위로 독립적으로 무작위로 우선순위를 할당하게 된다. 이 때, 낮은 우선순위와 높은 우선순위의 발생 확률은 순수 버스트 모드와 동일하다. 이 두 소스 모델은 두 개의 서로 다른 종류의 부하에 대한 간소화된 표현 방법이다

이 중, 순수 버스트 모드는 최종 사용자가 높은 우선순위 패킷과 낮은 우선순위 패킷을 생성하며 이 패킷을 구성하는 ATM 셀이 ATM 스위치 멀티플렉서에 도착하기 전까지 멀티플렉싱 되지 않을 경우에 적합하다. 혼합 버스트 모드는 다수의 최종 사용자로부터 발생한 패킷들이 이미 ATM 셀로 변환되고 입력 라인이

ATM 스위치 멀티플렉서에 도착하기 전에 각 입력 라인 상에서 셀 단위로 멀티플렉싱 되는 경우에 적합하다.

또한, 혼합 미스트 모드는 최종 사용자가 계층화된 코딩(layered coding)을 처리한 음성이나 화상 자료와 같은 다중 우선순위 셀 스트림을 생성할 때도 적용된다. 따라서, 본 논문에서는 이 두가지의 트래픽 모드 중 혼합 버스트 모드를 채택하고 있다

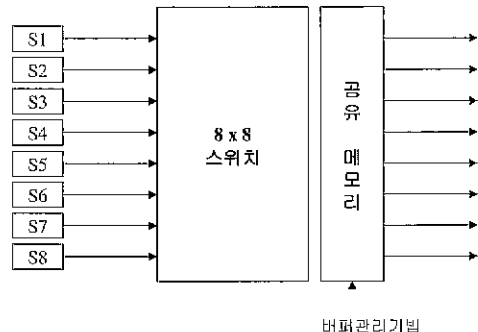
본 논문은 다양한 종류의 VBR 트래픽에 대한 셀 손실률을 최소화하는 것을 목적으로 하고 있다. 따라서, 실험에서 높은 우선순위의 트래픽과 낮은 우선순위의 트래픽 모두 매우 비스트하게 발생한다.

실제로 ATM 스위치에서 다수의 CBR 트래픽과 ABR 트래픽을 전송할 수 있다. 하지만, CBR과 같은 트래픽의 경우 적은 양의 예약 버퍼를 시용함으로써 QoS를 만족할 수 있다. 또한, ABR 트래픽의 경우 RM 셀을 이용한 피드백 정보를 이용하여 혼잡 제어를 하고 있다. 따라서, 본 논문에서는 CBR과 ABR 트래픽은 고려하지 않고 있다

5. 성능 평가

본 논문에서 제안하고 있는 개선된 동적 임계치 기법의 성능을 평가하기 위한 실험 모델은 (그림 2)와 같다. 즉, 스위치의 크기는 8x8이며 스위칭 패킷 내에는 버퍼가 존재하지 않고 출력 포트에 버퍼를 배치한다고 가정하고 있다

출력 버퍼는 공유 메모리 형태를 취한다. 즉, 여러 출력 포트가 하나의 물리적인 메모리를 공유하며 그 공유 메모리 내부에 각각의 출력 포트별로 논리적인 큐



(그림 2) 실험 모델(Simulation Model)

를 유지하게 된다. 이 때, 버퍼의 크기는 6500개의 셀을 수용할 수 있는 크기로 가정한다. 이 크기는 [6]에서 채택하고 있는 크기로서 아무런 버퍼관리 기법도 사용하지 않고 실험 모델내의 부하를 0.5라고 가정할 때, 평균 셀 손실률이 2%에서 10% 사이에 발생할 때의 크기이다.

각각의 입력력 포트의 속도는 155.520Mbps이다. 또한, 입력 포트에 셀을 보내는 소스에서의 셀 생성 속도는 평균 15Mbps(S1-S4)와 45Mbps(S5-S8)라고 가정한다[13]. 이는 성능 평가의 대상을 영상 자료로 가정한 결과이다. 일반적으로, 영상 자료의 평균 비트 속도는 1.5Mbps에서 시작하여 영상 자료를 코딩하는 기법에 따라 다양하다.

소스에서 셀을 생성할 때의 평균 버스트 기간은 20 셀로 설정하고 있다[6]. 이 때, 각 소스에서는 높은 우선순위 셀과 낮은 우선순위 셀을 85:15의 비율로 생성하고 있다[2]. 여기서의 셀 우선순위 모델링 기법은 혼합 버스트 모드를 기반으로 한다. 즉, 각 셀 단위로 무작위로 우선순위를 할당한다.

본 논문의 성능 평가에서 실험하는 대상은 기존의 동적 임계치 관리 기법(DT), 선택적인 푸쉬아웃 기법(SP), 개선된 동적 임계치 관리 기법이다. 성능 평가 요소는 (그림 2)의 공유 메모리에서 각각의 버퍼 관리 기법을 적용했을 때에 발생하는 전체 셀 손실률과 높은 우선순위 셀의 손실률이다. 이 때의 셀 손실률은 다음 식으로 계산된다.

$$\text{셀 손실률}(CLR) = \frac{\text{손실된 셀의 수}}{\text{수신된 셀의 수}}$$

이 때 높은 우선순위 셀의 손실률은 손실된 높은 우선순위 셀의 수를 수신된 높은 우선순위 셀의 수로 나타낸 결과이다

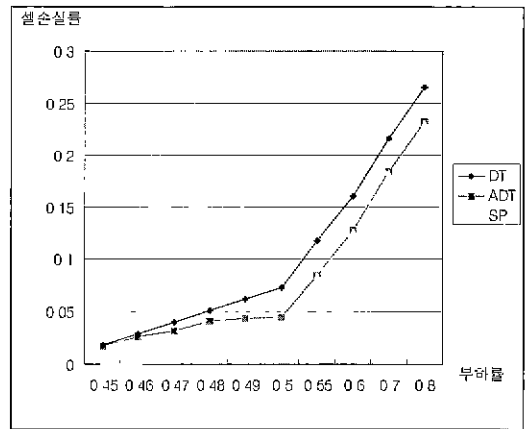
성능 평가를 위해 다음 두 가지 경우를 고려하고 있다.

- 단일 출력포트에 셀이 집중하는 경우
- 하나의 출력포트가 과부하이고, 다른 하나의 출력 포트에 갑자기 많은 양의 셀이 집중하는 경우

이 두 경우는 일반적으로 스위치 내에서 혼잡이 발생할 수 있는 가장 대표적인 경우이며, 따라서 본 논문에서 제안하고 있는 개선된 동적 임계치 버퍼관리 기법을 다른 버퍼관리 기법과 비교 평가하기에 적합하다고 할 수 있다.

5.1 단일 출력포트에 셀이 집중하는 경우

(그림 3)은 여러 부하에 따른 각 버퍼 관리 기법의 셀 손실률을 보여주고 있다. 여기서의 부하는 4.1절의 트래픽 모델에서 설명하고 있듯이 버스트 사이의 평균 간격을 변경함으로써 조절이 가능하다. 즉, 0.5의 부하률이라는 것은 평균 버스트 길이와 그 버스트 사이의 간격이 동일한 경우를 말하는 것이다. 따라서, 부하를 높이기 위해서는 버스트 사이의 간격을 줄임으로써 가능하다.



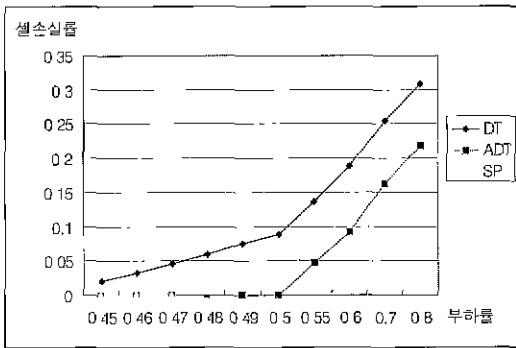
(그림 3) 단일 출력포트에 셀이 집중하는 경우의 셀손실률

(그림 3)에서 알 수 있듯이 ADT(개선된 동적 임계치 기법)가 모든 부하 구간에서 DT(동적 임계치 기법)에 비해 더 적은 수의 셀을 손실하는 것을 확인할 수 있다. 이는 기존의 동적 임계치 기법에서 사용하지 않고 있는 예비 공간(제어 임계치를 초과할 때 도착하는 셀이 차지하는 공간)을 높은 우선순위 셀이 차지하기 때문이다. 즉, 기존에는 제어 임계치를 초과한 시점에 도착한 높은 우선순위 셀을 버리는 정책을 썼지만 개선된 기법에서는 높은 우선순위 셀을 버려가 참 때까지 계속해서 버퍼에 수용하는 정책을 펴고 있기 때문이다

또한, ADT는 부하가 0.5보다 작은 경우(혼잡이 경미한 경우)에는 SP(선택적인 푸쉬아웃)에 비해 더 많은 셀손실률을 나타내지만, 부하가 0.5보다 큰 경우(혼잡이 심한 경우)에는 SP에 비해 약간 많지만 거의 유사한 정도의 셀손실률을 나타내는 것을 알 수 있다.

(그림 4)는 (그림 3)과 똑같은 조건에서 실험했을 때

에 발생하는 높은 우선순위 셀에 대한 손실률을 보여 주고 있다. 이 결과 역시 모든 부하 구간에서 ADT는 DT 기법에 비해 더 적은 셀손실률을 보여 주고 있다. 또한, ADT는 높은 우선순위 셀에 대한 셀손실률만을 고려할 때, 모든 부하 구간에서 SP에 비해 유사하거나 더 좋은 성능을 나타내는 것을 볼 수 있다. ADT와 SP 모두 높은 우선순위 셀을 보호하는 정책을 피고 있지만 SP는 버퍼가 꽉 찬 순간부터 높은 우선순위 셀을 보호하게 되지만 ADT는 제어 임계치를 초과하는 시점부터 높은 우선순위 셀을 보호하기 때문이다. 다시 말하면, SP에 비해 혼잡이 발생하는 시점보다 조금 더 앞선 상황부터 높은 우선순위 셀을 보호하기 때문에 더 좋은 성능을 보여 준다.



(그림 4) 단일 출력포트에 셀이 집중하는 경우의 높은 우선순위 셀의 손실률

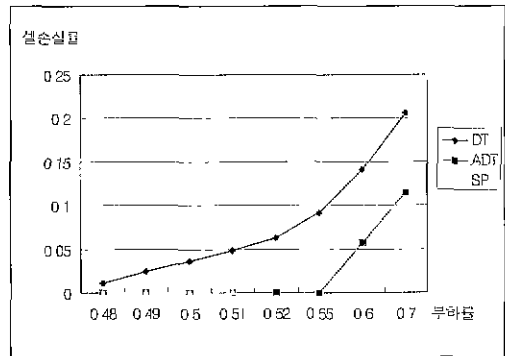
하지만, 높은 우선순위 셀에 대한 셀손실률만을 가지고 ADT가 SP에 비해 더 좋은 성능을 나타낸다고 말하기는 힘들다. (그림 4)와 (그림 3)을 조합해서 관찰해 보면, 부하가 0.5보다 적은 구간에서는 전체적인 셀손실률이 SP가 ADT에 비해 더 적으며 반대로 높은 우선순위 셀에 대한 셀손실률은 약간 높다는 것을 알 수 있다. 그러나, 그 구간에서는 높은 우선순위 셀에 대한 손실률 보다 전체 셀 손실률의 비율이 더 높기 때문에 SP의 성능이 더 좋다고 할 수 있다. 그렇지만, 부하가 0.5보다 높은 구간에서는 전체 셀손실률은 SP와 ADT가 유사하며 높은 우선순위 셀에 대한 손실률은 ADT가 더 작다는 것을 알 수 있다.

그러므로, 혼잡이 경미한 경우에는 SP가 가장 좋은 성능을 보이며 혼잡이 매우 심한 경우에는 ADT가 가

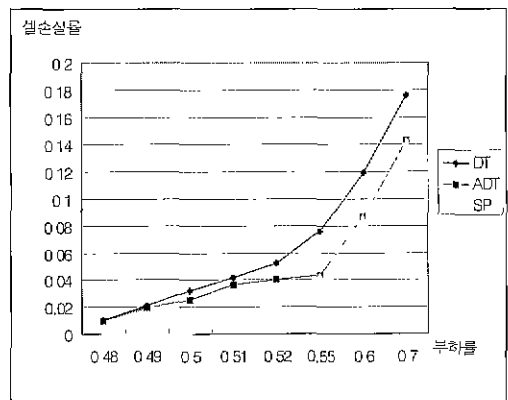
장 뛰어나다는 결론을 얻을 수 있다. 또한, 모든 부하 구간에서 DT에 비해 ADT의 성능이 뛰어나다는 것을 실험 결과에서 알 수가 있다.

5.2 하나의 큐가 과부하이고, 휴먼 큐에 갑자기 셀이 집중하는 경우

(그림 5)는 (그림 6)과 유사한 트래픽 조건이며 단지 하나의 소스에서 셀을 생성하는 시점을 지연시켜서 실험한 결과이다. 실험 결과는 그림에서 보는 바와 같이 모든 구간에서 ADT가 DT에 비해 더 좋은 성능을 나타내고 있다. 또한, 부하 조건이 0.55보다 작은 구간에서 ADT가 SP에 비해 더 나쁜 성능을 보이는 반면 0.55보다 큰 구간에서는 거의 유사한 성능을 보이고



(그림 5) 하나의 큐가 과부하이고, 휴먼 큐에 갑자기 셀이 집중하는 경우의 높은 우선순위 셀의 손실률



(그림 6) 하나의 큐가 과부하이고, 휴먼 큐에 갑자기 셀이 집중하는 경우의 셀손실률

있다. 이 결과는 단일 출력포트에 셀이 집중할 때와 유사한 결과를 나타내고 있다.

(그림 6)은 위 조건에서 실험했을 경우에 발생하는 높은 우선순위 셀에 대한 셀손실률을 보여 주고 있다. 이 결과 역시 (그림 4)의 결과와 유사하다. 즉, 모든 부하 구간에서 ADT가 DT에 비해 더 좋은 성능을 나타내고 있으며 혼잡이 경미한 구간에서는 DT와 유사하며 혼잡이 심한 구간에서는 오히려 SP보다 더 좋은 성능을 나타내고 있다.

(그림 5)와 (그림 6)을 통해 알 수 있듯이 개선된 동적 임계치 기법은 단일 출력포트에만 셀이 집중하는 경우와 하나의 출력포트가 과부하이고 어느 하나의 출력포트에 갑자기 많은 셀이 생성되는 경우 모두에서 기존의 동적 임계치 기법에 비해 더 좋은 성능을 보이고 있다. 또한, 혼잡이 경미할 때는 SP가 더 좋은 성능을 혼잡이 심할 경우에는 ADT가 더 좋은 성능을 보인다.

앞에서 언급했듯이 개선된 동적 임계치 기법은 기존의 동적 임계치 기법에 비해 제어 임계치를 초과한 시점에 도착한 높은 우선순위 셀을 버리지 않고 버퍼에 저장한다. 그리고, 기존의 동적 임계치 기법에서는 비퍼가 꽉 찬 시점에 도착하는 셀은 우선순위에 관계없이 버린다. 따라서, 개선된 동적 임계치 기법에서처럼 기존의 동적 임계치 기법에서 제어 임계치를 초과한 시점에 도착한 높은 우선순위 셀을 버리지 않는 정책을 적용하게 되면 버퍼 오버플로우가 빨리 발생하게 된다. 따라서, 이와 같은 원인으로 오버플로우가 발생한 시점에 휴면 상태에 있던 출력포트에 갑자기 셀이 집중하게 되면 그 큐에 도착하는 셀은 우선순위에 관계없이 버려지게 된다. 그 결과 특정 큐가 공유 메모리를 모두 독점하는 현상으로 전개될 수도 있다. 이는 공평성이라는 관점에서 위배된다.

즉, 공유 메모리를 거의 대부분 차지하고 있던 큐로 인해 버퍼 오버플로우가 발생한 경우에 제어 임계치 보다 작은 양의 트래픽을 보내고 있던 큐가 공평성을 제공받지 못하게 된다. 개선된 동적 임계치 기법에서는 이런 문제를 해결하기 위해 버퍼 오버플로우가 발생한 시점부터 비선형적인 푸쉬아웃을 적용하여 제어 임계치보다 적은 양의 셀이 도착하는 큐를 보호하고 있다.

이와 같이 버퍼 오버플로우가 발생한 시점부터 비선형적인 푸쉬아웃을 적용하는 정책이 원하는 결과를 나타내는 지를 보이기 위해 위 실험 중에 출력포트 별로 버려지는 셀의 수를 측정하였다. 측정된 결과 상대적

으로 적은 양의 셀이 도착하는 큐에서 아주 적은 양의 낮은 우선순위 셀만이 버려지고 높은 우선순위 셀은 하나도 버려지지 않았다. 이는 버퍼 오버플로우가 발생하는 시점에 낮은 우선순위 셀이 도착하면 그 셀을 버리고, 높은 우선순위 셀이 도착하면 공유 메모리를 가장 많이 사용하고 있는 큐의 헤더에서 셀을 하나 버리고 그 메모리 공간을 새로 도착하는 셀이 차지하도록 하는 정책을 펴고 있기 때문이다.

그러나, 이와 같은 정책을 펴더라도 큐의 공평성 문제를 완전히 해결했다고 볼 수는 없다. 즉, 개선된 동적 임계치 기법에서는 높은 우선순위 셀을 많이 밀생하는 큐가 버퍼를 많이 차지할 수 있게 되고 그로 인해 상대적으로 적은 양의 버퍼를 차지하고 있는 큐에 낮은 우선순위 셀이 도착하면 그 셀은 버려질 확률이 높아지게 된다. 따라서, 적은 양의 버퍼를 차지하고 있는 큐에 낮은 우선순위 셀이 많이 발생하게 되면 그 큐는 버퍼를 차지할 수 있는 기회가 상대적으로 적어지게 되어 공평성을 많이 제공받지 못하게 된다. 그러나, 버퍼가 꽉 차 있을 때, 길이가 작은 큐에 도착하는 높은 우선순위 셀을 우선적으로 보호하는 정책을 펴고 있기 때문에 높은 우선순위 셀에 대한 공평성은 기존의 동적 임계치 기법과 거의 유사한 공평성을 제공한다.

6. 결 론

공유 메모리 ATM 스위치에서의 비퍼관리 기법은 전체 스위치의 성능의 관점에서 가장 중요한 요인이다. 고속의 ATM 망에서 너무 시간이 많이 소요되는 복잡한 비퍼관리 기법을 사용하게 되면 셀의 지연이 심해지고 결국 사용자의 QoS를 만족시키지 못하게 된다.

비퍼관리 기법에서 큐의 길이를 제어하는 것은 스위치의 공평성과 효율성에 지대한 영향을 미치게 된다. 현재까지 여러 연구결과를 통해 알려진 대표적인 비퍼관리 기법에는 경적 임계치 기법, 동적 임계치 기법과 푸쉬아웃 기법이 있다.

이 기법들 중 동적 임계치 기법은 정적 임계치 기법의 구현의 단순성의 장점과 푸쉬아웃 기법의 높은 효율성의 장점을 취합한 기법이다. 또한, 동적 임계치 기법은 트래픽 상황에 따라 적응성이 뛰어나다는 장점이 있다. 즉, 트래픽 조건에 따라 제어 임계치를 탄력적으로 조절함으로써 큐의 길이를 조절한다. 그러나 동적 임계치 기법은 적응성이라는 장점에도 불구하고 높은

우선순위 셀에 대한 보호를 전혀 고려하지 않고 있다.

이는 기존의 동적 임계치 기법에서 단순하게 높은 우선순위 셀을 보호하는 정책을 피게 되면 다른 큐를 위해 남지 둔 예비 버퍼 공간이 빠르게 소진되고 버퍼 오버플로우가 쉽게 발생하기 때문이다. 그러나, 이런 문제점에도 불구하고 VBR 트래픽의 경우, 높은 우선순위 셀을 보호하지 않으면 많은 문제를 야기하게 된다. 일반적으로 VBR 트래픽(음성이나 영상 자료)에서의 높은 우선순위 셀은 어느 한 장면의 영상이나 한 음절의 음성 중 상대편에서 복원 시에 가장 기본적으로 필요한 정보를 포함하게 된다. 따라서, 다수의 높은 우선순위 셀이 중간에서 상실되면 한 장면을 표현하기 위한 다른 셀(낮은 우선순위의 셀)들이 성공적으로 수신되었다고 하더라도 원래 영상이나 음성을 복원하기가 힘든 경우가 발생한다.

본 논문에서는 이런 문제를 해결하기 위해 개선된 동적 임계치 기법을 제안하였다. 이 기법은 기존의 동적 임계치 기법의 장점인 트래픽에의 적응성이라는 장점을 유지하면서 높은 우선순위 셀을 보호하는 정책을 채택하였다. 즉, 제어 임계치를 넘어서더라도 높은 우선순위 셀을 버리지 않고 큐에 수용하도록 한 것이다.

개선된 동적 임계치 기법의 성능을 평가하기 위해 본 논문에서는 두 가지 경우를 고려하여 실험하였다. 단일 출력포트에 셀이 집중하는 경우와 하나의 출력포트가 과부하이므로 휴면 상태에 있던 큐에 갑자기 셀이 집중하는 경우에 대해 실험하였다. 실험 결과 개선된 동적 임계치 기법은 모든 부하 조건에서 기존의 동적 임계치 기법에 비해 더 좋은 성능을 보였다. 또한, 혼잡이 경미한 경우에는 푸쉬아웃 기법에 비해 약간 떨어지는 성능을 보였지만 혼잡이 심한 상황에는 푸쉬아웃 기법에 비해 유사하거나 오히려 더 좋은 성능을 보였다.

따라서, 개선된 동적 임계치 기법은 기존의 동적 임계치 기법의 장점을 그대로 유지하면서 더욱 향상된 성능을 보이고 있으며 고속의 공유 메모리 ATM 스위치에서의 버퍼관리 기법으로 적용할 수 있다.

참 고 문 헌

- [1] T. Kozaki, Y. Sakurai, O. M. M. Mizukami, M. Uchida, Y. Sato and K. Asano, "32x32 Shared Buffer Type ATM Switch VLSIs for B-ISDN," in Proc. IEEE ICC '91, (Denver, Colorado), pp.23.5.1-23.5.5, June 1991.
- [2] Abhijit K. Choudhury and Ellen L. Hahne, "Space Priority Management in a Shared Memory ATM Switch," in Proc. IEEE GLOBECOM '93, Vol.3, pp.1375-1383, Dec. 1993.
- [3] Mark J. Karol, Michael G. Hluchyj and Samuel P. Morgan, "Input Versus Output Queueing On a Space-Division Packet Switch," IEEE Transactions on Communications, COM-35(12):1347-1356, December 1987.
- [4] G. Hebuterne and A. Gravey, "A space priority queueing mechanism for multiplexing ATM channels," in Proc. ITC Special Sem. (Adelaide, Australia), September 1990.
- [5] H. Kroner, G. Hebuterne, P. Boyer and A. Gravey, "Priority Management in ATM Switching Nodes," IEEE J. Select. Areas Commun. Vol.9, pp.418-427, April 1991.
- [6] Abhijit K. Choudhury and Ellen L. Hahne, "Dynamic Queue Length Thresholds in a Shared Memory ATM Switch," Proc. of the IEEE infocom'96 The Conference on Computer Communications - Volume 2, 4/24/96.
- [7] Hamid Ahmadi and Wolfgang E. Denzel, "A survey of modern high-performance switching techniques," IEEE Journal on Selected Areas in Communications, 7(7):1091-1102, September 1989.
- [8] "NIST ATM Network Simulator," <http://isdn.ncsl.nist.gov/>.
- [9] M. I. Irland, "Buffer Management in a Packet Switch," IEEE Trans. Commun., Vol.26, pp.328-337, March 1978.
- [10] G. J. Foschini and B. Gopinath, "Sharing Memory Optimally," IEEE Trans. Commun., Vol.3, pp.352-360, March. 1993.
- [11] L. Georgiadis, I. Cidon, R. Guerin and A. Khamisy, "Optimal Buffer Sharing," in Proc. IEEE INFOCOM'95, Vol.1. (Boston, Massachusetts), pp.24-31, April 1995.
- [12] T. E. Elazov, V. Ramaswami, W. Willinger and G. Latouche, "Performance of an ATM switch," Proceedings IEEE INFOCOM'90, pp.644-659, 1990.
- [13] K. L. Reid, "Priority Queueing in Asynchronous Transfer Mode Switches," Master's thesis, Department of Computational Science, University of Saskatchewan, 1992.

[1] T. Kozaki, Y. Sakurai, O. M. M. Mizukami, M. Uchida, Y. Sato and K. Asano, "32x32 Shared Buffer Type ATM Switch VLSIs for B-ISDN," in Proc. IEEE ICC '91, (Denver, Colorado), pp.23.5.1-23.5.5,



김 양 섭

e-mail : likant@sslslab.cse.cau.ac.kr
1997년 중앙대학교 컴퓨터공학과 (학사)
1999년 중앙대학교 대학원 컴퓨터 공학과(석사)
2002년~현재 중앙대학교 대학원 컴퓨터공학과(박사)

관심분야 : 망관리, 분산시스템



권 혁 인

e-mail : hikwon9@unitel.co.kr
1983년 중앙대학교 전자계산학과 (학사)
1985년 중앙대학교 전자계산학과 (석사)
1994년 프랑스 파리 6대학(컴퓨터 공학박사)

1985년~1990년 : 한국국방연구원 연구원
1995년~현재 : 중앙대학교 경영학과 부교수
관심분야 : 망관리, 네트워크 트래픽 제어, 분산시스템



김 영 찬

e-mail : yckim@sslslab.cse.cau.ac.kr
1965년 연세대학교 전기공학과 (공학사)
1968년 연세대학교 대학원 전자 공학과(공학석사)
1983년 연세대학교 대학원 전자 공학과(공학박사)

1982년~1983년 프랑스 Grenoble대학 연구교수
1996년~1996년 한국정보과학회장
1973년~현재 중앙대학교 컴퓨터공학과 교수
관심분야 : 운영체제, 분산시스템, 망관리