

# 계층적 FIFO: 프레임 기반 패킷 전송 스케줄링 기법을 위한 지연 감축 방안

정희원 김 휘 용\*, 유 상 조\*\*, 김 성 대\*

## HFIFO(Hierarchical First-In First-Out) : A Delay Reduction Method for Frame-based Packet Transmit Scheduling Algorithm

Hui-Yong Kim\*, Sang-Jo Yoo\*\*, Seong-Dea Kim\* *Regular Members*

### 요 약

본 논문에서는 프레임 기반 패킷 전송 스케줄링을 위한 지연 감축 방안을 제안한다. ATM과 같은 초고속 통신망은 사용자에게 대역폭 과 패킷 지연과 같은 성능을 보장하여야 한다. 스케줄링 방법에 있어 프레임 기반 구조는 사용자에게 요구되는 대역폭의 지원과 간단한 레이트-제어 방법을 제공하지만 나쁜 지연 특성을 가지게 된다. 본 논문에서 제안한 지연 감축 방법은 HRR (Hierarchical Round-Robin)과 같은 계층적 프레임 구조를 사용하지만 지연 특성을 좋게 하기 위해 고정된 우선순위를 갖는 라운드 로빈 방식을 사용하지 않고, 동적으로 우선순위를 변화 시킴으로써 광대역과 협대역 연결간의 지연에 있어서의 불평등을 해소하였다. 해석적 비교 및 모의실험 결과 제안된 HFIFO 방법이 기존의 HRR과 같은 프레임 구조의 장점을 그대로 따르면서, 연결간 공정한 지연품질 제공하며 전체 지연 값을 줄일 수 있음을 알 수 있었다.

### ABSTRACT

In this paper, we propose a delay reduction method for frame-based packet transmit scheduling algorithm. A high-speed network such as ATM network has to provide some performance guarantees such as bandwidth and delay bound. Framing strategy naturally guarantees bandwidth and enables simple rate-control while having the inherently bad delay characteristics. The proposed delay reduction method uses the same hierarchical frame structure as HRR (Hierarchical Round-Robin) but does not use the static priority scheme such as round-robin. Instead, we use a dynamic priority change scheme so that the delay unfairness between wide bandwidth connection and narrow bandwidth connection can be eliminated. That is, we use FIFO (First-In First-Out) concept to effectively reduce the occurrence of worst-case delay and to enhance delay distribution. We compare the performance for the proposed algorithm with that of HRR. The analytic and simulation results show that HFIFO inherits almost all merits of HRR with fairly better delay characteristics.

### 1. 서 론

멀티미디어 서비스는 각 미디어별로 서로 다른 통신 대역폭과 QoS (quality of service : 서비스 품질)를 요구한다. 따라서 멀티미디어 서비스를 수용하려면 망은 각 응용 서비스 연결에 대해 대역폭과

QoS 등의 성능을 보장해 주는 능력을 갖춰야 한다. 각 연결에 대해 협상된 대역폭과 QoS를 제공하기 위해서는 효과적인 전송 스케줄링 알고리즘이 요구된다<sup>[1]</sup>. 전송 스케줄링 알고리즘은 하나의 출력 링크를 공유하는 여러 연결들에 대해 호 수락시 할당된 대역폭과 QoS에 따라 다음 전송할 패킷을 결정하는 규칙이다. 효과적인 전송 스케줄링 알고리즘은

\*한국과학기술원 전자전산학과,

논문번호 : K01107-0326, 접수일자 : 2001년 3월 26일

\*\*인하대학교 정보통신전문대학원

대역폭을 공정하게 배분하고, 할당된 범위를 벗어나 잘못 사용하는 연결로부터 다른 연결들을 보호할 수 있어야 하며, 협상된 QoS에 따라 지연과 지연변이, 그리고 패킷 손실율을 제한시킬 수 있어야 한다.

패킷 전송 스케줄링 방법 중 프레임 구조는 유효 대역폭을 공정하고 독립적으로 제공할 뿐 아니라 별도의 전송률 제어가 없이도 전송률을 제어할 수 있는 장점을 가진다. 그러나 HRR (hierarchical round robin)<sup>[2][13]</sup>과 같은 기존의 프레임 기반 스케줄링 구조는 소모 대역폭이 작은 연결의 경우 지연 특성이 매우 나빠지고 각 연결간 경계 되는 지연이 공정하지 않게 되는 단점이 있다. 본 논문에서는 일반적으로 프레임 구조의 전송률 제어 특성을 살펴보고 기존의 프레임 기법 스케줄링 알고리즘들의 한계를 지적하며, 이를 바탕으로 새로운 프레임 구조 스케줄링 기법인 HFIFO (hierarchical first-in first-out)를 제안하고자 한다.

본 논문의 구성은 다음과 같다. 먼저 제 II 장에서는 전송 스케줄러의 동작 환경과 기존의 패킷 전송 스케줄러에 대해 살펴본다. 제 III 장에서는 계층적 프레임 구조를 갖으며, 향상된 지연특성을 제공하는 새로운 알고리즘인 HFIFO를 제안한다. HFIFO는 프레임 구조를 사용하여 대역폭을 공정하고 독립적으로 제공하면서도 프레임 구조의 단점인 지연시간의 증가 문제를 FIFO의 개념을 도입해 해결한 알고리즘이다. 이 알고리즘은 낮은 작업 복잡도를 가지므로 고속 교환 환경에의 적용이 용이하며, 전송률 제어를 별도로 설치하지 않고도 전송률을 제어하므로 단말과 같이 하드웨어가 간단해야 되는 응용에 적합하다. 제 IV 장 성능평가에서는 제안한 알고리즘의 성능을 해석적 결과와 모의실험 결과로 나누어 검증하며, 제 V 장에서 본 논문의

결론을 맺는다.

## II. 패킷전송 스케줄링 환경 및 기존 연구 현황

### 2.1 패킷전송 스케줄러 모형

전송 스케줄링이라 함은 하나의 출력 링크를 공유하는 연결들에 대해 각각의 트래픽 파라미터와 QoS 파라미터들을 참조하여 어떤 연결의 패킷을 어떤 순서로 출력 링크로 전송할 것인가를 결정하는 방법을 말한다. 이러한 전송 스케줄링을 담당하는 부분을 전송 스케줄러라 하며 전송 스케줄러는 단말 내부나 교환기의 출력 링크 앞 단 등에 설치가 요구된다. 전송 스케줄링 기법은 호 수락 제어 기법과 더불어 망에서 성능 보장 서비스를 수용하기 위한 중요한 열쇠라 할 수 있다<sup>[3]</sup>.

본 연구에서는 전송 스케줄러에서 사용할 트래픽 파라미터와 QoS 파라미터를 bandwidth와 delay bound라는 두 가지 파라미터로 모형화 하였다. bandwidth는 트래픽 파라미터들로 부터 특정한 계산식에 의해 구해지는 유효 대역폭을 출력 링크 전송률로 정규화 시킨 값을 말하며, delay bound는 QoS 파라미터 중 종단간 최대 전송지연(즉, maxCTD)과 라우팅 정보를 이용하여 하나의 전송 스케줄러에 할당된 국부적(local) 최대지연값을 지칭한다. 트래픽 파라미터로부터 bandwidth를 계산하는 방법과 종단간 최대 지연으로부터 국부적 delay bound를 계산하는 방법에 관하여 지금까지 여러 가지 알고리즘들이 제안되었으나 아직 표준이 이루어지지 않은 상태이다. 본 연구에서는 이미 bandwidth와 delay bound가 계산되어 전송 스케줄러가 그 정보를 가지고 있다고 가정한다. 본 연구에서 사용할 전송 스케줄러의 모형을 정리하여 그림 1에 보였다. 그림 1에서 전송 스케줄링 서버는 자신을 경유하는

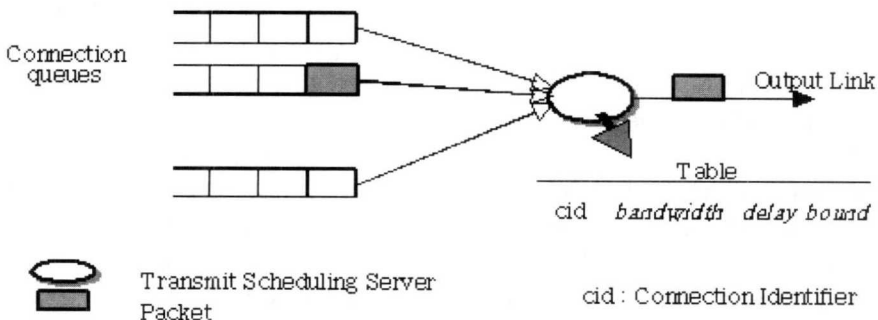


그림 1. 전송 스케줄러 모형

모든 연결들에 대해 bandwidth와 delay bound를 table로 저장하고 있으며, 이 table을 참조하여 bandwidth를 준수하여 도착한 패킷들에 대해 delay bound를 제공할 수 있도록 출력링크로 내보낼 패킷을 스케줄링 한다.

## 2.2 기존 연구 현황

본 절에서는 패킷을 대상으로 하는 기존의 스케줄링 알고리즘들을 작업보존 알고리즘과 비 작업보존 알고리즘으로 나누어 간략히 설명하고자 한다. 작업보존 알고리즘은 전송을 기다리는 패킷이 있으면 서버가 쉬지(idle) 않고 전송하지만, 비 작업보존 기법에서는 전송할 패킷이 있더라도 특정한 조건을 만족하는 패킷이 없으면 서비스를 하지 않고 서버를 쉬게 한다. 본 절에서 설명하는 알고리즘들은 모두 일반적인 패킷 교환망의 교환기를 그 응용 대상으로 하여 제안된 것들이다.

### 2.2.1 작업보존 알고리즘

WFQ (weighted fair queuing)[4]와 WF2Q[5], 그리고 SCFQ (self-clocked fair queuing)[6]는 모두 FFQ(fluid fair queuing, 혹은 GPS, generalized processor sharing)[7]를 패킷 환경에서 근사하는 방법들이다. WFQ는 FFQ 시스템에서 서비스가 완료되는 순서대로 패킷들을 정렬하여 순차적으로 전송한다. WF2Q는 FFQ 시스템에서 이미 서비스가 시작된 패킷들 중에서 먼저 서비스를 완료하는 순서로 전송함으로써 패킷 환경에서 FFQ를 가장 잘 근사화 한다. 가상시간<sup>[8]</sup>은 시분할 다중화(TDM) 시스템을 모방하는 것으로, 각 패킷마다 그 패킷이 속한 연결의 협상 정보(트래픽 및QoS 파라미터)와 도착 패턴을 참조하여 가상 전송 시각을 할당한 후 이 시각이 빠른 순서대로 전송하는 방법이다. 이상의 우선순위 정렬 메커니즘을 이용하는 스케줄링 기법들은 높은 복잡도를 갖으므로, ATM과 같은 고속 교환 환경에 적용하는 것은 적당치 않다. 또한 이 알고리즘들은 모두 최대지연과 대역폭이 서로 결합되어 있어 두 가지 파라미터를 독립적으로 할당할 수 없는 단점이 있다.

DRR (deficit round robin)[9]은 각 연결에 그 대역폭에 대응하는 quantum을 할당하고 매 라운드마다 이전 라운드에서 사용하지 못하고 남은 양과 새로운 quantum과의 합만큼의 서비스를 라운드 로빈으로 제공한다. DRR은 구현이 용이하고 작업 복잡도가  $O(1)$ 으로 유지되며 공정성 특성이 좋은 반

면 최대지연의 값이 라운드의 길이에 비례해서 커지는 단점이 있다.

### 2.2.2 비 작업보존 알고리즘

종단간 최대지연과 버퍼 요구량을 해석적으로 계산하기 위해서는 망 전체에 걸쳐 각 연결별 트래픽 특성을 알아야만 한다. 작업보존 알고리즘에서는 망 내부의 부하 변동에 따라 트래픽 패턴이 왜곡되므로 망 전체에 걸쳐 트래픽 특성을 알아내는 데는 많은 어려움이 있다. 그러나, 비 작업보존 알고리즘은 각 교환기에서 트래픽 패턴의 왜곡을 제어하므로 여러 개의 교환기를 거치는 네트워크 환경에 있어서 작업보존 알고리즘보다 유리하다 할 수 있다<sup>[1]</sup>.

Stop-and-go<sup>[10]</sup> 알고리즘은 시간 축을 어떤 일정한 간격 T로 나누는 프레임 기법을 사용한다. 현재 프레임 기간 동안 도착한 패킷들은 다음 프레임이 되어야 서비스될 자격이 주어지게(eligible) 된다. 이렇게 함으로써 같은 프레임 내에 있는 패킷들을 망 전체를 통해 같은 프레임 내로 유지할 수 있으며, 따라서 각 노드에서 최대지연을 제한함으로써 임의의 망 구성에 대해 종단간 최대지연을 제한시키는 효과를 얻을 수 있다. HRR (hierarchical round robin)<sup>[2][13]</sup>은 프레임 기법을 사용한다는 점에서 Stop-and-go와 비슷하다. HRR은 여러 계층의 프레임들을 라운드 로빈 방식으로 서비스하는 기법으로 최대지연과 대역폭 할당을 어느 정도 동시에 향상시킬 수 있는 장점이 있다. 그러나 하위 계층으로 내려갈수록 최대지연이 상당히 증가하기 때문에 낮은 대역폭을 요구하는 연결에 대해서는 지연 특성이 나쁜 단점이 있다. HRR에 관해서는 III장의 계층적 FIFO기법에서 더 자세히 다룬다.

J-EDD (jitter-earliest-due-date)<sup>[11]</sup>와 RCSP (rate-controlled static priority)<sup>[12]</sup>는 스케줄링이 끝난 패킷 헤더에 여유 시간을 기록하고 다음 스케줄링 서버의 전송률 제어기에서 그 시간만큼을 지연시킴으로써 연결별로 지연변이의 bound를 제공하고 트래픽 패턴을 망 전체를 통해 일정하게 유지시키는 방법이다.

이 두 방법들은 스케줄링 서버를 대역폭과 지연변이를 제어하는 전송률 제어기와 최대지연 요구사항에 따라 실제 패킷을 전송하는 스케줄러로 구분함으로써 대역폭과 최대지연의 할당을 자유롭게 할 수 있지만, ATM과 같이 적은 크기의 패킷을 사용하는 경우 패킷 헤더에 시간 정보를 기록하는 것은 큰 overhead가 될 수 있다<sup>[16]</sup>.

### III. 계층적 FIFO 기법

본 장에서는 일반적인 프레임 구조의 전송률 제어 특성을 살펴보고, 기존의 프레임 기법 스케줄링 알고리즘들의 한계를 지적하며, 이를 바탕으로 새로운 프레임 구조 스케줄링 기법인 HFIFO를 제안한다.

#### 3.1 계층적 프레임 구조

기본적인 프레임 구조를 다음과 같이 정의한다. 먼저 시간 축을 어떤 일정한 간격  $T$ 의 프레임으로 나눈다. 각 연결은 bandwidth에 따라 quantum을 할당 받게 되며, 한 프레임당 한 번의 서비스 기회를 받고, 그때마다 한 quantum씩 서비스를 받는다. 현재 프레임을 서비스하는 동안 패킷이 도착하면 그 cid(connection identification number)를 저장해 두었다가 다음 프레임 시간이 되면 저장된 cid 순으로 서비스를 한다. 서비스할 패킷이 없는 경우에 전송 스케줄러는 best-effort 패킷을 서비스하거나 아무 패킷도 전송하지 않는다.  $i$ 번째 연결에 할당되는 quantum $_i$ 는 다음의 식 (1)에 의하여 구해진다.

$$quantum_i = \lceil bandwidth_i \times T_i \rceil \quad (1)$$

여기서, bandwidth $_i$ 는 연결  $i$ 에 할당된 대역폭을 출력 링크 대역폭으로 정규화 시킨 값이며,  $T_i$ 는 연결  $i$ 의 프레임 주기를 슬롯 단위로 표현한 값을 의미하며,  $\lceil x \rceil$ 는  $x$ 보다 크지 않은 최대의 정수를 의미한다. 기본적인 프레임 구조의 경우 모든 연결에 대해 가 성립한다.

그러나 프레임 구조는 최대지연과 대역폭 할당의 자유도가 결합되는 문제를 안고 있다. 기본적인 프레임 구조의 경우 협상한 bandwidth를 준수하여 도착한 패킷이 겪는 지연은  $2T$ 로 bound된다. 따라서 프레임 주기  $T$ 가 줄어들수록 최대지연이 감소하게 된다<sup>[2]</sup>. 반면 대역폭 할당의 최소 단위가  $1/T$ 이므로  $T$ 가 줄어들수록 작은 대역폭을 요구하는 연결을 수용하기가 어려워진다. 다시 말해, 기본적인 프레임 구조에서는 작은 대역폭 할당 단위와 작은 최대지연을 동시에 얻는 것이 불가능하다. 이러한 점은 기존에 제안된 Stop-and-go와 HRR 기법이 공통적으로 안고 있는 문제이다.

HRR은 각 연결들을 대역폭에 따라 몇 개의 계층(level)으로 나누고, 계층에 따라 서로 다른 프레임 주기를 할당한 후, 각 계층을 라운드 로빈 방식

으로 서비스하는 기법이다. 그림 2에 HRR 스케줄링의 예를 보인다. 그림 2에서 level 1과 level 2는 모두 10 슬롯의 프레임으로 구성되지만, 실제로는 level 1의 패킷을 9슬롯 전송한 후 level 2의 패킷을 하나씩 전송하므로 level 1과 level 2의 프레임 주기는 각각 10 슬롯과 100 슬롯이 된다. 따라서 기본 프레임 구조와는 달리 각 level별로 서로 다른 최대지연을 가질 수 있으며, 최대지연의 값 또한 상위 level로 갈수록 더 작아짐을 알 수 있다. 그림 2의 예에서 level 1에 속하는 연결의 최대지연은 20 슬롯이고, level 2에 속하는 연결의 최대지연은 200 슬롯이 된다.

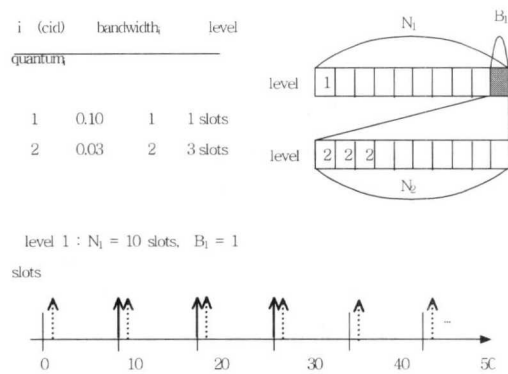


그림 2. HRR 스케줄링의 예

1.  $N$ 은 각 level 프레임의 슬롯 수를 의미하며,  $B$ 는  $N$  슬롯 중 하위 level에 할당된 슬롯 개수를 의미한다.
2. 서비스 순서는 다음과 같다: level 1을  $(N1-B1)$  슬롯만큼 서비스한 후, level 2를  $B1$  슬롯만큼 서비스하고, 다시 level 1 서비스부터 반복한다.
3. 화살표 점선은 연결 1의 패킷을, 실선은 연결 2의 패킷을 각각 의미한다.

3.2 제안된 도착순서 보존 계층적 프레임 구조  
HRR과 같이 계층적 프레임 구조를 사용하면, 대역폭 점유율이 큰 연결들은 상위 계층에 할당되므로 작은 최대지연을 제공받을 수 있다. 그러나 대역폭 점유율이 작은 연결들은 하위 계층에 할당되므로 최대지연이 증가하게 된다. 이러한 효과는 대역폭 할당을 미세하게 하면 할수록 심화되어 나타난다.

계층적 프레임 구조에서 하위 계층으로 갈수록 최대지연이 증가하는 것은 첫째, 프레임 구조의 특성상 프레임 주기가 증가할수록 최대지연이 증가하

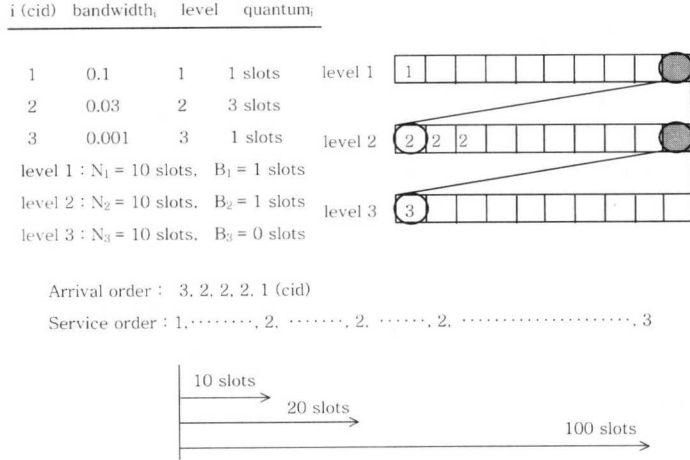


그림 3. HRR 기법의 계층간 불평등.

기 때문이며, 둘째, 라운드 로빈 시 상위 계층 프레임에 우선권이 주어지기 때문이다. 첫 번째 문제를 프레임 주기 증가에 따른 최대지연 증가문제, 두 번째 문제를 계층간 불평등 문제로 각각 부르기로 한다. 프레임 주기 증가에 따른 최대지연 증가문제는 프레임 구조의 근본적인 특성으로 바꿀 수 없는 부분이다. 그러나 최대지연의 값 자체를 줄이는 것과 최대지연이 발생할 확률을 줄여 지연 분포를 개선하는 것은 가능하다. 본 절에서는 계층간 불평등 문제를 해결하기 위해서는 도착순서대로 서비스를 수행하는 FIFO (first-in first-out)의 개념이 필요함을 보이고, 이 개념을 확장하여 HFIFO (hierarchical FIFO) 기법을 제안하며, 제안된 기법이 최대지연의 값을 줄이고 지연분포를 개선하는 효과가 있음을

보이도록 하겠다.

먼저 계층간 불평등 문제를 생각해 본다. 그림 2에 나타난 HRR의 예를 보면 level 2 패킷은 반드시 level 1의 한 프레임이 끝나야만 서비스 받을 자격이 주어지게 된다. 즉, level 2의 패킷이 가장 먼저 대기 행렬(queue)에 도착했다 하더라도 더 늦게 도착한 level 1의 패킷을 한 프레임 서비스 한 후에야 전송이 시작되는 것이다. 이러한 계층간 불평등은 대역폭 할당이 미세해져 level이 증가할수록 더욱 심화된다. 이러한 예를 그림 3에 보인다. 0.001의 대역폭을 점유하는 새로운 연결을 생각해 보자. 이 세 번째 연결은 그림 3에 나타난 대로 10 슬롯의 프레임으로 이루어진 level 3에 속하게 될 것이며 하나의 quantum을 할당 받을 것이다. 이 경우

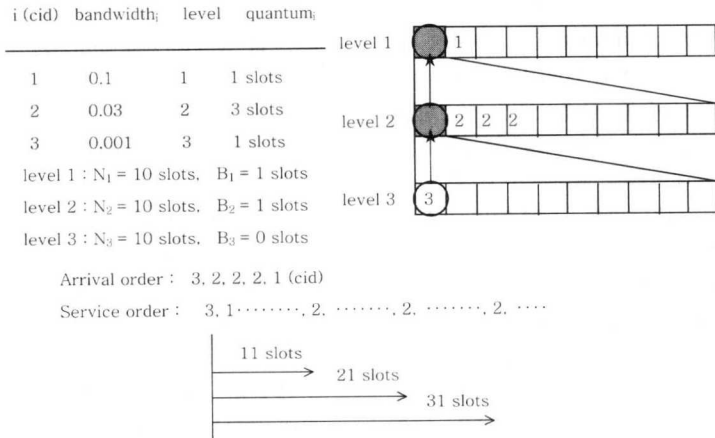


그림 4. FIFO 개념을 이용한 계층간 불평등의 해소

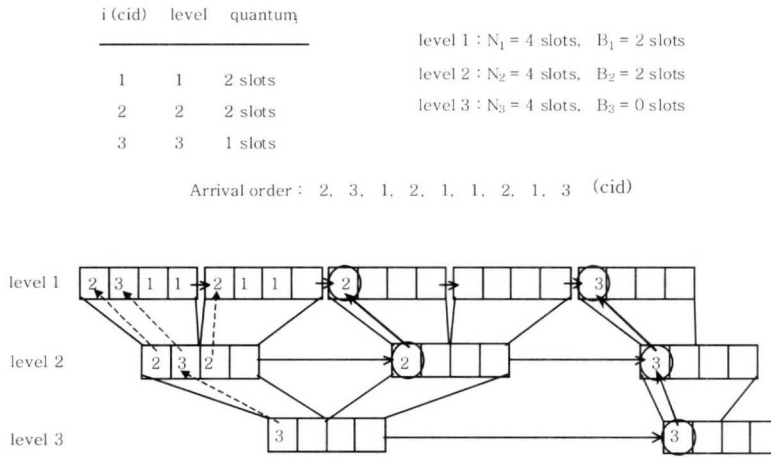


그림 5. 도착순서 저장 방법의 한 예

level 3에 속하는 패킷이 가장 먼저 도착하였다라고 level 2프레임이 끝나는 100 슬롯째에 전송이 이루어지게 된다. 만약 대역폭 할당의 미세함을 더욱 증가 시켜 level을 10개 정도 할당한다고 가정하면 최하위 계층의 패킷은 가장 먼저 도착했다 하더라도 109슬롯째에 전송이 이루어지게 되는 것이다.

이러한 계층간 불평등의 원인은 HRR 기법이 계층들 사이를 상위 계층에 정적인 우선순위를 둔 라운드 로빈 방식으로 서비스하기 때문이다. 따라서, 우선순위 자체를 도착 순서에 따라 동적으로 부여한다면, 다시 말해 우선순위에 근거한 라운드 로빈을 사용하지 않고 도착한 순서대로 서비스를 수행하는 FIFO의 개념을 사용한다면, 계층간 불평등은 사라질 것이며 하위 계층으로 갈수록 지연이 두드러지게 감소할 것임을 예측할 수 있다. 이러한 구조를 사용함으로써 최대지연이 프레임 주기 이하로 bound되며 (HRR의 1/2), 도착순서에 따라 서비스함으로써 최대지연이 발생하는 확률이 줄어들게 되어 하위 계층으로 갈수록 지연 분포가 상당히 개선되는 결과를 얻게 된다. 그림 4는 FIFO의 개념이 추가된 계층적 프레임 구조의 스케줄링 예를 보인다. 이 예를 통해 level과 quantum의 할당이 그림 3의 경우와 같을 때 level 3에 속하는 연결의 지연이 99 슬롯만큼 감소함을 알 수 있다.

### 3.3 HFIFO 알고리즘

3.2절에서와 같이 계층적 프레임 구조에 FIFO의 개념을 추가하는 것은 하위 계층에 할당된 슬롯의 위치를 고정시키지 않고 도착한 패킷의 순서에 따

라 변화시킴으로써 구현할 수 있다. 도착순서로 서비스를 하기 위해서는 각 계층별로 도착한 패킷의 cid를 모두 저장하고 있어야 하며 다른 계층의 패킷과 비교하여 도착 순서의 선후를 가릴 수 있는 메커니즘 또한 있어야 한다. 그러한 방법 중 하나를 그림 5에 보인다.

그림 5에 보인 방법은 셀이 도착할 때마다 전송될 위치에 그 cid를 기록한 후 level 1의 프레임에 기록된 cid의 순서대로 서비스를 하는 단순한 방법이다. 이와 같은 방법을 사용하면 할당된 quantum보다 많은 하위 계층의 셀이 도착했을 때 그것을 기록하기 위한 상위 계층의 프레임을 만들기 위해 메모리를 미리 잡아야 하는 문제가 발생한다. 그림 5는 연결 2와 연결3의 셀이 할당된 quantum보다 1슬롯씩 많이 도착하였을 때 level 1프레임이 미리 만들어지는 과정을 보여준다. 이러한 메모리 증가 문제는 level의 개수가 증가할수록 심각해진다. 따라서 메모리를 미리 잡지 않으면서도 계층간 도착 순서의 선후를 판별할 수 있는 방법이 요구된다. 이를 위해 제안된 방법은 그림 6과 같다.

이 그림에 나타난 방법에서는 하위 계층의 패킷이 할당된 quantum보다 많이 도착했을 경우에도 상위 계층 프레임을 미리 만들지 않고, 상위 계층의 패킷이 도착하였을 때 그 패킷보다 먼저 도착한 하위 계층의 패킷을 찾아 프레임에 기록한다. 서비스는 앞의 그림 5의 방법과 같이 level 1프레임에 기록된 cid의 순으로 이루어진다. 단, 이런 방법을 사용하면 상위 계층의 패킷이 전송되어 오지 않는 경우에는 하위 계층 패킷의 도착 여부를 확인할 수

i (cid)	level	quantum <sub>i</sub>
1	1	2 slots
2	2	2 slots
3	3	1 slots

level 1 :  $N_1 = 4$  slots,  $B_1 = 2$  slots  
 level 2 :  $N_2 = 4$  slots,  $B_2 = 2$  slots  
 level 3 :  $N_3 = 4$  slots,  $B_3 = 0$  slots

Arrival order : 2, 3, 1, 2, 1, 1, 2, 1, 3 (cid)

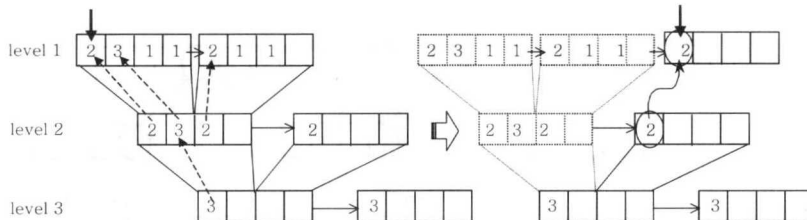


그림 6. 메모리를 미리 점유하지 않는 도착순서 판별 방법

좌측 그림은 각 계층별 프레임의 상태와 각 프레임에 연결된 하위 계층의 슬롯을 나타내고 있으며, 우측 그림은 level 1 프레임 두 개 만큼의 시간이 지난 후 각 프레임의 상태를 나타낸다. 그림에서 점선으로 표시된 프레임은 이미 서비스를 받고 메모리 영역에서 삭제된 프레임은 나타내며, 굵은 화살표는 현재 전송할 슬롯을 가리키는 read\_index를 나타낸다.

없으므로, 서비스 시 read\_index가 가리키는 슬롯이 비어 있으면 하위 계층을 검색하여 이전에 도착한 패킷이 없는가 확인하는 메커니즘을 전송 처리부에 추가하여야 한다.

#### IV. 모의실험 및 성능 평가

본 장에서는 지연 관련 몇 가지 평가 지수들을 이용해 제안한 알고리즘의 성능을 평가하고자 한다. HRR의 경우는 quantum이 남아 있더라도 일단 한번 서비스를 받으면 다음 프레임까지 기다려야만 서비스 기회가 주어진다. 따라서 프레임 주기가 FT인 연결의 패킷이 bandwidth를 준수하여 입력되었을 때 delay bound는  $2xFT$ 가 되었다(단, 여기서 bandwidth를 준수하여 입력된다 함은 한 프레임 주기 동안 quantum 이하의 개수로 패킷이 입력되는 것을 의미한다.) 이러한 경우는 현재 프레임의 처음에 한 번의 서비스 기회를 받은 후 바로 입력된 패킷이 다음 프레임의 마지막에 전송 되는 경우에 해당한다. 그러나 HFIFO의 경우는 quantum이 남아 있으면 언제나 그 프레임 기간 안에 서비스를 받게 되므로 해석적 delay bound는 FT가 된다.

$$delay\_bound = FT \tag{2}$$

지연 감소 관련 모의실험을 위해 다음의 표 1에

보인 대로 고정길이 패킷의 Poisson 트래픽을 발생시킨 후 그림 7과 같이 HRR과 HFIFO 서버를 통과시켜 A지점에서 모니터링 하였다. 트래픽은 평균적으로 협상한 대역폭(표 1에서의 bandwidth)만큼 생성되나, 순간적으로는 협상한 값보다 많이 생성될 수 있다. 연결은 모두 4개 이고, 계층은 모두 3개이며, 각 연결에 할당되는 quantum과 각 계층에 할당되는 프레임 수는 표 1과 같다. 표 1에서 bandwidth는 각 연결에 할당된 대역폭을 출력 링크 전송 대역폭으로 정규화 한 값이며, 프레임 당 슬롯 수는 100슬롯으로 하였고, 이 100슬롯 중 하위 계층에 할당된 슬롯은 10슬롯으로 하였다.

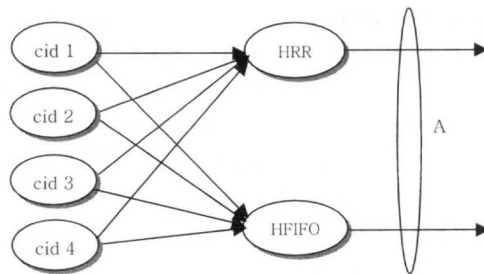


그림 7. 모의실험 환경

실험은 4번에 걸쳐 이루어 졌는데, 연결 1 부터 3까지는 그대로 두고 연결 4의 트래픽을 협상한 것보다 적게, 혹은 많이 생성 시킨 후 그 영향을 관찰 하였다. 실험 회수에 따른 연결 4 트래픽의 파라미

표 1. 연결1 ~ 연결4의 파라미터

cid	bandwidth	mean inter-arrival time	level	각 level에서의 quantum
1	0.5	2 slots	1	50 slots
2	0.05	20 slots	2	50 slots
3	0.005	200 slots	3	50 slots
4	0.02	50 slots	2	20 slots

level 1: N1 = 100 slots, B1 = 10 slots  
 level 2 : N2 = 100 slots, B2 = 10 slots  
 level 3 : N3 = 100 slots, B3 = 0 slots

표 2. 연결4의 트래픽 로드 변화 파라미터

trial	mean inter-arrival time	traffic load
1	500 slots	0.2 %
2	50 slots	2 %
3	5 slots	20 %
4	1 slot	100 %

터 변화를 표 2에 보였다. 표 2에서 알 수 있듯이 연결 4는 2%의 대역폭을 점유하도록 협상되었지만

실제로는 0.2%, 2%, 20%, 그리고 100%의 대역폭 점유율에 해당하는 트래픽을 생성시킨다. 실험은 10,000,000 슬롯 동안 수행하였다.

다음의 그림 8-11은 각각 연결 4의 트래픽 부하를 0.2%, 2%, 20%, 100%로 변화시킬 때 연결 1부터 연결 4에 대한 지연 분포를 나타낸다. 이 그림들에서 알 수 있듯이 연결 1부터 연결 3까지의 지연 특성은 연결 4의 부하 변동에 크게 영향을 받지 않는다. 연결 4는 본래 2%의 대역폭 점유율을 할당 받았다. 따라서 트래픽 부하를 20%로 생성시킨 경우는 그 만큼의 불이익을 받아야 한다. 이러한 것은

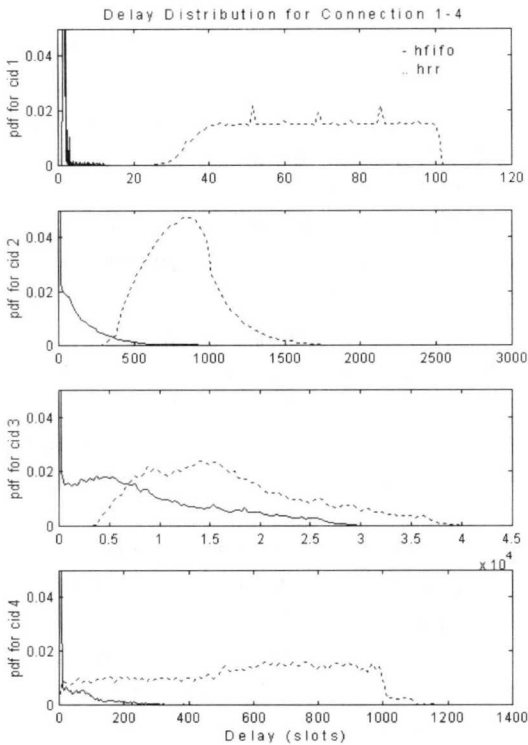


그림 8. HRR과 HFIFO의 지연 분포 비교 (Traffic Load is 0.2%)

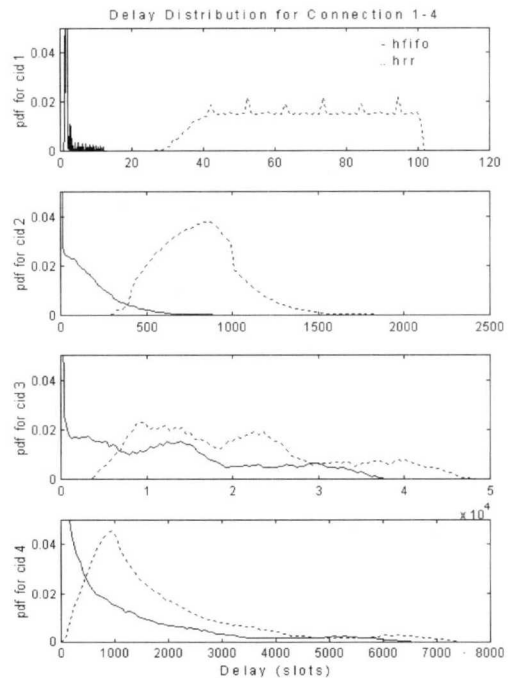


그림 9. HRR과 HFIFO의 지연 분포 비교 (Traffic Load is 2%)



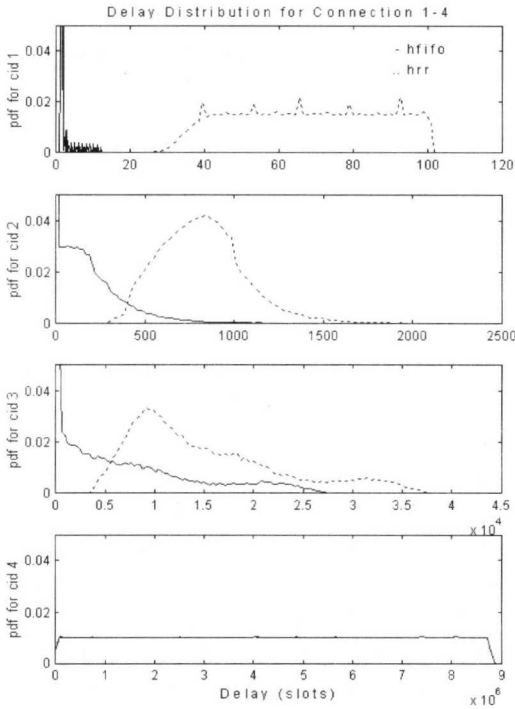


그림 10. HRR과 HFIFO의 지연 분포 비교 (Traffic Load is 20%)

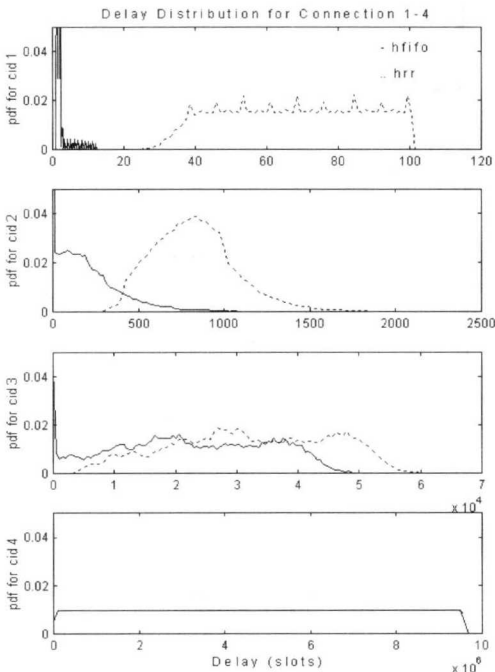


그림 11. HRR과 HFIFO의 지연 분포 비교 (Traffic Load is 100%)

그림 12에 잘 나타나 있다. 즉, 협상된 대역폭 점유율 이하로 트래픽이 생성되는 경우에는 지연 특성이 좋게 나타나지만 그것을 넘어서는 경우에는 지연 특성이 상당히 나빠졌다.

모의실험 결과, 제안된 HFIFO 프레임 기반 스케줄링 방법은 기존의 계층적 프레임 기반 스케줄링 방법에서 문제가 되었던 최대 지연이 크고, 계층간 불공정한 지연 분포를 크게 개선시킬 수 있음을 알 수 있다.

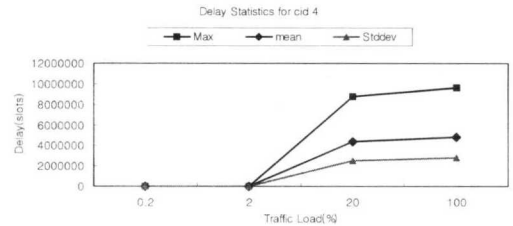


그림 12. 트래픽 부하 변동에 따른 연결 4의 지연 통계치 변화

## V. 결 론

본 연구에서는 초고속 통신망에서의 프레임 기반 전송 스케줄링 알고리즘의 지연 감축 방안을 제안하였다. 프레임 기반 알고리즘은 유효 대역폭을 공정하고 독립적으로 제공할 뿐 아니라 별도의 전송률 제어가 없이도 전송률을 제어할 수 있는 장점이 있으나, 반면 대역폭 할당을 미세하게 할수록 최대지연이 증가하는 단점이 있다. HFIFO 기법은 이러한 문제를 도착 순서 대로 서비스를 수행하는 FIFO의 개념을 도입해 해결하였다. 몇 가지 해석과 모의 실험을 통해 제안된 기법이 프레임 구조의 장점인 대역폭 제공의 공정성과 독립성을 도장하면서, 기존의 프레임 기반 스케줄링 알고리즘인 HRR에 비해 최대 지연을 비롯한 전반적인 지연 특성이 크게 향상됨을 알 수 있었다.

본 연구에서는 트래픽 파라미터와 QoS 파라미터를 bandwidth와 delay bound로 각각 모형화하여 전송 스케줄링에 응용하였다. 앞으로 좀 더 세분화된 트래픽 모형을 수용하기 위한 연구를 계속할 것이며, 최대지연 외에도 최대 지연변이나 패킷 손실율과 같은 QoS 파라미터들을 고려하여 제안된 기법을 개선할 예정이다.

참 고 문 헌

[1] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks", Proceedings of the IEEE, vol.83, no.10, pp.1373-1396, 1995

[2] C. Kalmanek, H. Kanakia, and S. Keshav, "Rate Controlled Servers for Very High-Speed Networks", Proceedings of IEEE Global Telecommunication Conference, pp.300.3.1~300.3.9, Dec. 1990

[3] D. Ferrari and D. Verma, "A Scheme for Real-time Channel Establishment in Wide-area Networks", IEEE Journal on Selected Areas in Communications, vol.8, pp.368-379, Apr. 1990

[4] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm", Proceedings of ACM SIGCOMM '89, 1989

[5] J. Bennett and H. Zhang, "WF2Q: Worst-case Fair Weighted Fair Queuing", Proceedings of IEEE INFOCOM '96, pp.1d.3.1~1d.3.9, 1996

[6] S. J. Golestani, "A Self-clocked Fair Queuing Scheme for Broadband Applications", Proceedings of IEEE INFOCOM '94, pp.5c.1.1~5c.1.11, 1994

[7] A. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrate Services Networks: The Single-Node Case", IEEE /ACM Transactions on Networking, vol.1, no.3, pp.344~357

[8] L. Zhang, "Virtual clock: A New Traffic Control Algorithm for Packet Switching Networks", ACM Transactions on Computer Systems, vol.9, no.2, pp.101~124, 1991

[9] M. Shreedhar, "Efficient Fair Queuing using Deficit Round Robin", IEEE/ACM Transactions on Networking, vol.4, no.3, 1996

[10] S. J. Golestani, "A start-and-go Queuing Framework for Congestion Management", Proceedings of ACM SIGCOMM '90, pp.8~18, 1990

[11] D. Verma, H. Zhang, and D. Ferrari, "Guaranteeing Delay Jitter Bounds in Packet

Switching Networks", Proceedings of Tricomm '91, 1991

[12] H.Zhang and Ferrari, "Rate-controlled Static Priority Queuing", Proceedings of IEEE INFOCOM '93, pp.2c.3.1~2c.3.10, 1993

[13] A. Hung and G. Kesidis, "Performance evaluation of hierarchical round-robin bandwidth scheduling for ATM", Proc. ITC-15, pp. 1247-1256, 1997

김 휘 용(Hui-Yong Kim)

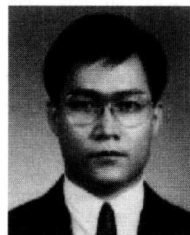
정회원



1994년 8월: 한국과학기술대학  
전기및전자공학과 (공학사)  
1998년 2월: 한국과학기술원 전  
기및전자공학과 (공학석사)  
1998년 3월 - 현재: 한국과학기술원  
전자전산학과 전기및전자공학 전공 박사과정  
<주관심 분야> 인터넷 QoS, 영상 통신, 컴퓨터 비전

유 상 조(Sang-Jo Yoo)

정회원



1988년 2월: 한양대학교 전자통신공학과 (공학사)  
1990년 2월: 한국과학기술원 전기 및 전자공학과 (공학석사)  
2000년 8월: 한국과학기술원 전자전산학과 (공학박사)

1990년 3월 - 2001년 2월: KT 연구개발본부  
2001년 3월 - 현재: 인하대학교 정보통신대학원 조교수  
<주관심 분야> 인터넷 QoS, 초고속 통신망 구조, 멀티미디어 네트워킹, 트래픽 엔지니어링