

# 단일서버에서의 누적적 공정서비스 모델 (A Cumulative Fair Service Model in Single Server)

이 주 현<sup>†</sup>      박 경 호<sup>†</sup>      황 호 영<sup>\*\*</sup>      민 상 렬<sup>\*\*\*</sup>  
(Ju Hyun Lee)    (Kyeong Ho Park)    (Ho Young Hwang)    (Sang Lyul Min)

**요약** Generalized Processor Sharing(GPS) 모델은 세션에게 서비스를 제공시 순시적 공정서비스를 통해 서버 용량을 분배한다. 이 공정서비스는 단지 현재 서버 내에 대기 중인 세션만을 고려하여 서버 용량을 분배하기 때문에, 서비스를 제공하는 순간 서비스를 받는 세션 간에는 공정한 서비스가 제공된다고 볼 수 있다. 그러나 긴 시간의 관점에서 보면 같은 가중치를 가지는 한 세션이 다른 세션에 비해 평균적으로 다른 용량으로 서비스 받는 문제가 발생한다. 본 연구에서는 긴 시간의 관점에서 모든 세션이 항상 공평한 서비스용량을 제공받을 수 있는 누적 공정서비스 모델(CFS)을 제안한다. 이 CFS 모델은 각 세션에게 누적적 공정 서비스 용량을 제공하기 때문에 세션관점 공정 서비스를 제공한다. 그리고 이 모델과 이 모델을 참조하여 각 세션의 패킷을 스케줄링 하는 패킷 누적공정 서비스(P\_CFS)알고리즘의 특성과 성능을 분석하였다. 또한 성능평가를 통해 각 세션에게 긴 시간의 관점에서 세션의 가중치에 비례하는 서비스용량이 제공되는 것을 검증하였다.

**키워드** : 누적적 공정 서비스, 순시적 공정 서비스, GPS, 서비스커브, 공정큐잉알고리즘, 스케줄링

**Abstract** Generalized Processor Sharing(GPS) model provides instantaneous fair services to currently backlogged sessions. Since this fair service distributes server capacity to backlogged sessions in proportion to their weights, the fairness is only valid between the sessions serviced at the same time. From the long time view, however, this fair service provides different server capacity with one session or another, even if these sessions have the same weights. This paper proposes a cumulative fair service(CFS) model to provide fair server capacity to all sessions in the long time view. This model provides fair service in session viewpoint because it distributes server capacity in proportion to the weights of sessions. The model and an algorithm referencing that model are analyzed for their properties and performances. Performance evaluations verify that the proposed algorithm provides proportional service capacity to sessions in the long time view.

**Key words** : Cumulative fair service(CFS), Instantaneous fair service, Generalized Processor Sharing, Service Curve, Fair Queueing algorithm, Scheduling

## 1. 서론

최근 컴퓨터, 네트워크, 이동통신의 기술이 급격히 진보하고 있다. 이러한 진보는 인터넷전화, VoD(Video on Demand), DMB(Digital Media Broadcasting)등의 다양한 형태의 멀티미디어 서비스를 가능하게 하였다. 이와 같은 멀티미디어 서비스는 예측 가능한 성능을 위해

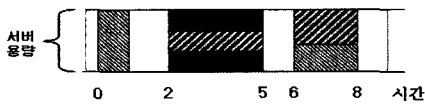
보장서비스를 제공하는 스케줄러를 요구한다. 이러한 보장서비스를 제공하기 위해 Packet GPS(PGPS)[1], Weighted Fair Queueing(WFQ)[2], Start-Time Fair Queueing(SFQ)[3], Worst-case Fair Weighted Fair Queueing(WF2Q)[4], Self-Clocked Fair Queueing(SCFQ)[5] 등의 여러 공정큐잉알고리즘이 제안되었다. 유선 뿐만 아니라 무선, 이동통신 환경에서 서버의 자원을 공평하게 이용하기 위한 알고리즘들[6,7]이 제안되었다. 이 알고리즘들은 모두 공통적으로 Generalized Processor Sharing(GPS) 모델을 기반으로 스케줄링한다. 한편 보장서비스 뿐만 아니라 트래픽 특성에 따라 다양한 서비스를 제공하고자 서비스커브(Service Curve)[8-10]가 제안되었다. 이 서비스커브는 세션에 제공하는 서비스 지연과 대역폭을 분리함으로써 VoIP, WWW, 멀티미디어

---

† 학생회원 : 서울대학교 전기컴퓨터공학부  
jhlee@archi.snu.ac.kr  
kalynda@archi.snu.ac.kr  
\*\* 중신회원 : 안양대학교 디지털미디어학부 조교수  
hyhwang@anyang.ac.kr  
\*\*\* 중신회원 : 서울대학교 컴퓨터공학부 교수  
symin@dandelion.snu.ac.kr  
논문접수 : 2006년 5월 24일  
심사완료 : 2006년 8월 17일

등의 지연 민감 세션에게 빠른 서비스를 제공하는 특성을 지닌다. GPS와 서비스커브 모델 모두 세션에게 순시적 공정 개념의 서비스를 제공한다는 공통점을 지닌다.

순시적 공정서비스는 현 서버의 용량을 단지 현재 서버 내에서 대기 중인 세션만을 고려하여 분배한다. 따라서 서비스가 제공되는 순간 서비스를 받는 세션 간에는 공정한 서비스가 제공된다고 볼 수 있지만, 긴 시간의 관점에서 보면 같은 가중치를 가지는 두 세션 중 한 세션이 다른 세션에 비해 평균적으로 더 큰 용량으로 서비스 받는 문제가 발생할 수 있다. 그림 1(a)는 이와 같은 문제의 한 예를 보여주고 있다. 현재 서버에서 서비스 받는 모든 세션이 같은 가중치로 서비스 받고, 모두 크기 1인 패킷을 생성한다고 가정한다. 그림 1(a)에서 보듯이, 세션 1이 시각 0에 도착하여 혼자 서비스 받고 떠나고, 시각 2에 세션 2,3,4가 동시에 도착하여 각각 서버용량의 1/3용량으로 서비스 받고 떠난다. 이어 시각 6에 세션 1,2패킷이 도착하여 각각 서버용량의 1/2로 서비스를 받는다. 이 예에서 보듯이 순시적 공정서비스방식은 세션들이 같은 가중치를 갖더라도 도착 시 서버에 대기 중인 세션에 따라 다른 서버용량으로 서비스된다.



(a) 순시적 공정서비스의 예



(b) 누적적 공정서비스의 예  
그림 1

누적적 공정서비스에서는 같은 가중치를 가지는 각 세션이 일정한 크기의 패킷을 서비스 받을 때, 도착시각 혹은 대기 중인 세션과 관계없이 누적적으로 일정한 서버용량으로 서비스 된다. 이 공정서비스에서는 그림 1(b)에서 보는 것과 같이 각 세션의 가중치가 같고, 각 패킷의 크기가 일정하다면, 각 패킷들은 같은 서비스 지연, 즉 같은 서비스용량으로 서비스된다. 순시적 공정서비스에서는 현재 서버에 대기 중인 세션들에 의해 각 세션의 서비스 용량이 결정되지만, 누적적 공정서비스는 현재 서버에 대기 중인 세션에 관계없이, 긴 시간의 관점에서 각 세션에게 세션의 가중치에 비례하는 용량의 서비스를 제공한다. 긴 시간동안 이러한 서비스 제공을 관찰한다면 각 세션들이 자신의 가중치에 비례하는 용량의 서비스를 받게 되므로 누적적 공정서비스는 세션

관점에서의 공정서비스를 제공한다.

본 연구에서는 그림 1(b)에서 보는 것과 같이 모든 세션이 긴 시간의 관점에서 공정한 용량으로 서비스 될 수 있도록 하는 틀을 제공한다. 이 틀을 통해 세 가지 형태의 서비스를 제공하려한다. 첫째로, 긴 시간의 관점에서 각 세션에게 그 세션의 가중치에 비례하는 용량의 서비스를 제공한다. 즉 현재 서버 내에 서비스 중인 세션에 관계없이 누적적으로 각 세션의 가중치에 비례하는 용량의 서비스를 제공한다. 이러한 서비스 제공을 통해 세션관점의 공정서비스를 실현한다. 둘째로, 대역폭 보장을 원하는 세션에게 보장서비스를 제공한다. 오디오나 비디오 세션 혹은 파일전송 세션의 경우, 세션 설정시에 일정 서버 대역폭 보장을 요구 할 수 있는데, 이 세션설정이 수락된 후, 실제 서비스가 진행될 때, 각 세션에게 약속했던 대역폭을 보장한다. 마지막으로, 간헐적으로 패킷을 생성하는 지연 민감 세션과 끊임없이 패킷을 생성하는 처리지향(throughput-oriented) 세션 간 서버용량을 경쟁 시, 지연 민감 세션이 우선적으로 처리될 수 있는 기반을 제공해준다. VoIP(Voice over IP)세션이나, 웹 세션들은 파일 전송세션에 비해 트래픽을 적게 생성하지만, 빠른 처리를 요하는 지연 민감 세션들인데, 기존의 GPS 기반의 알고리즘들은 지연과 대역폭이 결합되어 있기 때문에, 지연 민감한 세션을 서비스 시 큰 대역폭의 설정을 요구한다. 본 모델은 큰 대역폭 설정 없이 지연 민감한 세션에게 빠른 서비스를 제공할 수 있는 기반을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 누적적 공정서비스 모델을 제시한다. 3장에서는 이 모델을 실현하기 위한 구체적인 스케줄링 알고리즘을 설명한다. 4장에서는 모델과 스케줄링알고리즘의 특성 및 성능을 분석한다. 5장에서는 제안된 알고리즘을 시뮬레이션을 통해 성능분석을 수행하고 6장에서 결론을 맺는다.

## 2. 누적공정 서비스 모델

누적 공정서비스 모델(Cumulative Fair Service: CFS)은 긴 시간의 관점에서 각 세션들에게 세션의 가중치에 비례하는 서비스 용량을 제공하고자 하는 모델이다. 이 모델에는 각 세션에게 패킷서비스를 제공시 항상 일정한 서비스 용량을 제공하는 이상적인 누적 공정 서비스 모델(Ideal Cumulative Fair Service: I\_CFS)과 실제서버 용량을 고려하여 이상적인 누적 공정 서비스를 실현하는 실제적인 누적공정서비스 모델(Real Cumulative Fair Service: R\_CFS)이 있는데, 이장에서는 각각에 대해 설명한다.

### 2.1 이상적인 누적공정 서비스 모델

이상적인 누적공정 서비스 모델은 서버에 도착하는

각 세션의 패킷을 그 세션의 예약 용량이 비례하는 용량으로 서비스를 제공하는 가상적인 유체모델이다. 이 서비스 용량은 GPS에서와는 달리 그 패킷이 도착 시 서버에서 대기 중인 세션들과 관계없이 결정된다.  $S_i(\tau, t)$ 와  $SD_i(\tau, t)$ 를 각각 구간  $[\tau, t]$  동안 세션  $i$ 가 받은 서비스 양과 서비스지연의 합이라고 하고  $w_i$ 를 세션  $i$ 의 가중치라고 하면 L\_CFS 모델은 다음 관계식을 만족한다.

$$\frac{S_i(\tau, t)/SD_i(\tau, t)}{S_j(\tau, t)/SD_j(\tau, t)} = \frac{w_i}{w_j} \quad (1)$$

구간  $[\tau, t]$  동안 세션  $i$ 가 받은 서비스 양을 서비스 지연의 합으로 나누면, 이는 세션  $i$ 의 누적 서비스 용량이 된다. 그러므로 위 식은 구간  $[\tau, t]$  동안 세션  $i$ 와  $j$ 에게 자신의 가중치에 비례하는 누적 서비스 용량이 제공된다는 의미이다.

이 모델을 예를 들어 설명하면 그림 2와 같다. 그림 2에서는 세션 1과 세션 2가 각각 가중치  $w_1$ 과  $w_2$ 를 가지고 서비스 되는데, 세션의 패킷들의 도착시점과 관계없이 세션 1과 세션 2는 항상 일정한 서비스 용량인  $a * r_1 (r_1 = w_1 * r)$ ,  $a$ 는 비례상수)와  $a * r_2$ 로 서비스된다. 이는 마치 각 세션에게 항상 고정적인 서비스 용량이 제공되는 효과를 가지게 할 수 있다.

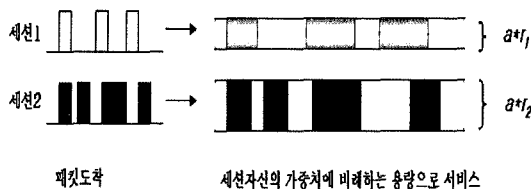


그림 2 L\_CFS 모델에서의 세션들의 패킷 서비스 용량

이 L\_CFS 모델은 각 세션의 관점에서 항상 자신의 가중치에 비례하는 일정한 서비스 용량이 제공되기 때문에 이상적인 공정 모델이지만, 이를 통해 실제 서비스 시에는 서버가 항상 일정한 서버용량  $r$ 로 서비스 되는 점이 고려되어야 한다. 실제적인 누적공정 서비스 모델은 이 사항을 고려하여 각 세션에게 서버용량을 할당해준다.

### 2.2 실제적인 누적공정 서비스 모델

실제적인 누적공정 서비스 모델(Real Cumulative Fair Service Model: R\_CFS)은 서버가 서버용량  $r$ 로 서비스되는 점을 고려하여, 긴 시간의 관점에서 각 세션들이 식 (1)과 같은 누적용량으로 서비스가 되도록 서버의 용량을 할당하는 유체모델이다. 식 (1)과 같은 서비스가 이루어지기 위해서 이 모델은 현재까지 각 세션이 받은 누적서비스 용량 값을 유지하는데, 이 값을 기준으로 현 시점에서 각 세션에게 실제 서버의 용량을 할당한다.

R\_CFS 모델은 누적적 공정서비스 뿐만 아니라, 필요

에 따라 각 세션에게 예약용량 보장서비스도 제공한다. 예를 들어 세션  $i$ 가 세션 설정 시에 최소한의 예약용량  $r_i$ 으로 서비스받기로 약속했다면, 이 모델은 약속된 예약용량으로 서비스를 수행한다. 그러나 만약 이 세션이 과거에 다른 세션에 비해 적은 용량으로 서비스했다면, 다른 세션들의 예약용량을 제외한 서버용량으로 서비스된다. 좀 더 자세히 설명하면, 이 모델에서는 각 세션이 과거 서비스를 어느 정도의 지연으로 받았는지를 나타내는 누적용량 값을 유지하는데, 현재 서버에서 서비스 중인 세션가운데 과거에 가장 적은 용량으로 서비스 받았던 세션에게는  $r - \sum_{i \in B} r_i$ 의 용량으로 서비스하며, 나머지 세션들은  $r_i$ 의 용량으로 서비스한다. 여기에서  $B$ 는 현재 서버에 대기 중인 세션들의 집합이다. 이 모델에서는 각 세션에 대해 누적용량 변수  $CC_i(t)$ 를 유지한다. 이 변수는 시각  $t$ 까지 세션  $i$ 가 받은 서비스의 양과 서비스지연 총합의 비로 결정된다. 만약 세션  $i$ 가 시각  $t$ 에 도착하여 시간  $\tau$  동안 서비스를 받고 있다면 세션의 누적용량  $CC_i(t+\tau)$ 은 다음과 같이 나타낼 수 있다.

$$CC_i(t+\tau) = \begin{cases} \frac{S_i(0, t) + (r - \sum_{i \in B} r_i) * \tau}{SD_i(0, t) + \tau} & [t, \tau] \text{ 동안 } r - \sum_{i \in B} r_i \text{ 용량으로 서비스 받은 세션} \\ \frac{S_i(0, t) + r_i * \tau}{SD_i(0, t) + \tau} & [t, \tau] \text{ 동안 } r_i \text{ 용량으로 서비스 받은 세션} \end{cases} \quad (2)$$

### 3. 알고리즘

R\_CFS 모델은 긴 시간의 관점에서 L\_CFS와 같은 철학을 가지고 동작하면서, 예약용량의 보장을 원하는 세션에게 보장서비스를 제공하는 유체모델이다. 이장에서는 이 유체모델을 참조하여 각 세션의 패킷을 스케줄링하는 패킷누적공정 서비스(Packet Cumulative Fair Service: P\_CFS) 알고리즘을 소개한다. 먼저 새 세션이 서버에 처음 도착 시 수행해야 하는 서비스 변수의 초기화를 설명한다.

#### 3.1 서비스변수 초기화

새로운 세션이 서버에 처음 도착 시 이 세션에 대한 누적서비스와 누적지연 파라미터의 초기화가 필요하다. 초기화는 기존의 세션들이 현재시점까지 받은 서비스 양에 관계없이 현재 새로 도착한 세션의 서비스가 도착시점 기준으로 기존의 세션들과 공정한 경쟁을 하기 위해 필요하다. 단순히 0으로 초기화한다면, 새 세션의 누적용량이 기존 세션의 누적용량과 비슷해질 때까지, 새 세션에 대한 지속적인 서비스가 진행될 수 있다. 이는 새 세션이 도착하기 이전에 다른 세션이 받은 서비스에 대해 새 세션이 보상받기 때문에 생긴다. 이러한 현상을

해결하기 위해 새로운 세션에 대한 누적서비스 및 누적 지연 파라메타의 초기화는 도착시점 모든 세션들의 대응되는 파라메타의 기중평균값으로 한다.  $CS_i(t)$ ,  $CD_i(t)$ 를 각각 기존세션  $i$ 가 시작  $t$ 까지 받은 서비스 및 서비스지연으로 놓으면, 새 세션의 누적서비스 및 누적지연은 다음과 같이 초기화할 수 있다.

$$CS_{new}(t) = \frac{\sum_{j=1}^J CS_j(t) * r_j}{\sum_{i=1}^I r_i} \quad (3)$$

$$CD_{new}(t) = \frac{\sum_{j=1}^J CD_j(t) * r_j}{\sum_{i=1}^I r_i} \quad (4)$$

**3.2 알고리즘**

I\_CFS 모델은 서버의 용량을 각 세션의 관점에서 세션의 가치에 비례하게 서비스해주는 가상적인 모델이고, R\_CFS모델은 I\_CFS철학을 유지한 채, 서버용량을 고려하여 각 세션에게 서비스를 제공하는 유체모델이었다. 이장에서는 R\_CFS모델을 참조하여 실제 스케줄링을 수행하는 패킷 누적 공정 서비스 알고리즘(P\_CFS)을 소개하고자 한다.

P\_CFS알고리즘은 각 세션의 패킷을 처리하기 위해 세션 당 2개의 변수, 패킷 당 두 개의 변수를 유지한다. 즉 세션 당 세션의 현재까지 받은 누적서비스의 양과, 각 패킷서비스를 받을 때 서비스지연의 합 값을 유지한다. 각 패킷 당 패킷의 도착시간, 패킷의 R\_CFS 참조모델에서의 종료시간을 유지한다.

$A_i(P(k))$ ,  $F_i(P(k))$ 를 각각 세션  $i$ 의  $k$ 번째 패킷의 도착시간과 종료시간이라 하면, 알고리즘의 구체적인 동작은 다음과 같다.

1) 세션  $i$ 의  $k$ 번째 패킷이 도착 시 도착시간이 다음과 같이 정해진다.

$$A_i(P(k)) = \max(\text{real\_time}, F_i(P(k-1))) \quad (5)$$

2) 알고리즘은 R\_CFS참조모델을 이용하여 현 패킷의 종료시간을 참조모델에서의 종료시간으로 정한다.

3) 스케줄링 시 각 패킷의 종료시간 기준하여 오름차순으로 서비스 되도록 스케줄링 한다.

P\_CFS는 R\_CFS를 참조하여 스케줄링 하므로, 패킷 알고리즘인 P\_CFS와 유체모델인 R\_CFS간 제공되는 서비스 용량, 즉 서비스 지연의 차이가 스케줄링시점에는 어느 정도 발생할 수 있지만, 긴 시간의 관점에서 볼 때 P\_CFS에서는 세션들에게 R\_CFS에서 제공받는 용량만큼의 서비스를 제공한다.

**4. 모델 및 알고리즘 분석**

이 장에서는 R\_CFS모델과 P\_CFS알고리즘에서의 패킷 지연, 모델과 알고리즘 간 진행되는 서비스 양의 차

이 및, 대기 중인 패킷양의 차이에 대해 분석한다.

$d_i^{R\_CFS}(k)$ 를 세션  $i$ 의  $k$ 번째 패킷의 R\_CFS에서 서비스 지연,  $Q_i(k)$ 를 이 패킷이 도착 시 보는 세션  $i$  서비스 큐의 길이, 그리고  $P_i(k)$ 를 이 패킷의 길이라고 하자.

**정리 1.** R\_CFS에서 세션  $i$ 의 패킷지연은 다음 부등식을 만족한다.

$$d_i^{R\_CFS}(k) \leq \frac{Q_i(k) + P_i(k)}{w_i * r} \quad (6)$$

**증명.** 세션  $i$ 의  $k$ 번째 패킷의 서비스 용량  $SC_i(k)$ 의 범위는  $w_i * r \leq SC_i(k) \leq r - \sum_{i \in B} w_i * r$ 이다. 최악의 지연이 발생하는 경우는 현재 도착한 패킷과, 이전에 도착한 후 대기 큐에서 대기 중인 패킷들 모두 가장 적은 용량인  $w_i * r$ 로 서비스 되는 경우이다. 이 때 지연은  $(Q_i(k) + P_i(k)) / (w_i * r)$ 가 되며 세션  $i$ 의 서비스는 용량  $w_i * r$ 보다 큰 값으로 서비스 받으므로 부등식(6)을 만족한다. □

$d_i^{P\_CFS}(k)$ 를 세션  $i$ 의  $k$ 번째 패킷의 P\_CFS에서 서비스 지연,  $P_{max}$ 는 최대패킷길이이라 하자.

**정리 2.** P\_CFS에서 세션  $i$ 의 패킷지연은 다음 부등식을 만족한다.

$$d_i^{P\_CFS}(k) \leq \frac{Q_i(k) + P_i(k) + w_i P_{max}}{w_i * r} \quad (7)$$

**증명.** 한 바쁜 구간에서 세션  $i$ 의 현재 도착한 패킷과 P\_CFS에 의해 현재 서비스 중인 패킷의 R\_CFS모델에서의 종료시간 비교 시 현재 도착한 패킷의 종료시간이 늦은 경우와 빠른 경우로 나누어 증명한다.

**경우 1.** 현재 도착한 세션  $i$ 패킷의 R\_CFS에서의 종료시간이 현재 P\_CFS에서 서비스 중인 패킷의 R\_CFS에서의 종료시간보다 늦은 경우: 한 바쁜 구간 내에서  $k$ 번째 서비스된 패킷의 P\_CFS에서의 종료시간은  $\sum_{j=1}^k P_j$ 가 된다. P\_CFS는 R\_CFS에서의 종료시간을 참조하여 스케줄링 하므로, R\_CFS에서  $k$ 번째 패킷이 끝날 시점에는,  $k$ 번째 패킷의 이전패킷들은 R\_CFS에서 이미 서비스가 종료됐고, 이 패킷과 이후에 종료될 패킷들이 자신의 예약용량 이상으로 동시에 서비스를 받는다. 그러므로

$$d_i^{P\_CFS}(k) \leq d_i^{R\_CFS}(k) \leq \frac{Q_i(k) + P_i(k)}{w_i * r}$$

를 만족한다.

**경우 2.** 현재 도착한 세션  $i$ 패킷의 R\_CFS에서의 종료시간이 현재 P\_CFS에서 서비스 중인 패킷의 R\_CFS에서의 종료시간보다 빠른 경우: P\_CFS는 R\_CFS에서의 패킷종료시간을 참조하여 스케줄링 한다. 이 경우는 R\_CFS상에서는 현재 도착한 패킷이 먼저 종료되는데, P\_CFS상에서는 현재 서비스 중인 패킷에 의해 서비스

종료시간이 늦추어져, 현재 서비스 중인 패킷과 종료순서가 바뀐 경우이다. 만약 종료순서가 바뀌지 않고 서비스 되었다면 경우 1에 해당하므로 부등식 (7)을 만족한다. 그러나 현재 서비스 중인 패킷에 의해 최악의 경우  $P_{max}/r$ 만큼 더 지연이 발생하므로 이 지연과 경우 1을 이용하면 다음 부등식을 얻는다.

$$d_i^{P-CFS}(k) \leq \frac{Q_i(k) + P_i(k) + w_i P_{max}}{w_i * r}$$

경우 1과 2의 결과에 의해 부등식(7)은 만족된다. □ 세션 i가 구간 $[\tau, t]$ 내에서 R\_CFS와 P\_CFS로 받은 서비스 양을 각각  $W_i^{R-CFS}(\tau, t)$  와  $W_i^{P-CFS}(\tau, t)$  라 하자.

**정리 3.** R\_CFS로 서비스 시 세션 i의 구간 $[\tau, t]$ 내에서 받은 서비스 양과 P\_CFS로 서비스 시 세션 i의 구간 $[\tau, t]$ 내에서 받은 서비스 양은 다음 부등식을 만족한다.

$$W_i^{R-CFS}(\tau, t) - W_i^{P-CFS}(\tau, t) \leq P_{max} \quad (8)$$

**증명.** R\_CFS에서는 세션 i의 패킷이 도착 즉시 최소한  $w_i r$ 의 용량으로 서비스가 시작되는 반면, P\_CFS는 도착이후에 스케줄링 시점에 선택이 되어야 서버 용량 r로 서비스가 시작된다. 따라서 부등식(8)의 좌변이 최대가 되는 시점은, P\_CFS에서 세션 i의 서비스가 시작되기 바로 직전 시간이다. 이 시간들 중 하나를  $t_k$ 로 표시하자. 그러면 다음의 부등식을 증명함으로써 부등식(8)의 성립을 증명할 수 있다.

$$W_i^{R-CFS}(\tau, t_k) - W_i^{P-CFS}(\tau, t_k) \leq P_{max} \quad (9)$$

여기에서 도착한 패킷의 R\_CFS와 P\_CFS에서의 종료시간을 각각  $F_i^{R-CFS}$ ,  $F_i^{P-CFS}$ 로 놓으면 다음이 성립한다.

$$W_i^{R-CFS}(\tau, t_k) = W_i^{R-CFS}(\tau, F_i^{R-CFS}) + W_i^{R-CFS}(F_i^{R-CFS}, t_k) \quad (10)$$

여기에서  $W_i^{R-CFS}(F_i^{R-CFS}, t_k)$ 의 최대값은  $F_i^{R-CFS}$ 이 최소값을 가질 때인데, 이 값은 정리 1과 정리 2를 이용하면  $F_i^{R-CFS} - P_{max}$ 가 될 때임을 알 수 있다. 그리고  $P_i$ 를 세션 i의 현재 서비스중인 패킷길이라 할 때  $W_i^{R-CFS}(F_i^{R-CFS} - P_{max}, t_k)$ 값의 최대값은  $P_{max} - P_i$ 이므로 다음 부등식이 성립 한다.

$$W_i^{R-CFS}(\tau, t_k) \leq W_i^{R-CFS}(\tau, F_i^{R-CFS}) + P_{max} - P_i \quad (11)$$

한편

$$W_i^{P-CFS}(\tau, t_k) = W_i^{P-CFS}(\tau, F_i^{P-CFS}) - P_i \quad (12)$$

이므로 식 (11)과 (12)를 식 (9)에 적용하면

$$W_i^{R-CFS}(\tau, t_k) - W_i^{P-CFS}(\tau, t_k) \leq P_{max} \quad (13)$$

를 얻는다. 그런데

$$W_i^{R-CFS}(\tau, t) - W_i^{P-CFS}(\tau, t) \leq W_i^{R-CFS}(\tau, t_k) - W_i^{P-CFS}(\tau, t_k) \quad (14)$$

를 만족하므로 부등식(8)이 증명된다. □

세션 i가 시작 t일 때 R\_CFS와 P\_CFS로 받은 서비스 시 대기 패킷의 길이를 각각 양을 각각  $Q_i^{R-CFS}(t)$ 와

$Q_i^{P-CFS}(t)$ 라 하자.

**정리 4.** R\_CFS로 서비스 시 세션 i의 시작 t일 때 대기패킷의 길이와 P\_CFS로 서비스 시 세션 i의 시작 t일 때 대기패킷의 길이는 다음 부등식을 만족한다.

$$Q_i^{P-CFS}(t) - Q_i^{R-CFS}(t) \leq P_{max} \quad (15)$$

**증명.** 시간구간 $[0, t]$ 중 서버에 도착한 세션 i의 전체 트래픽을  $T_i(0, t)$ 라 놓으면

$$T_i(0, t) = W_i^{R-CFS}(0, t) + Q_i^{R-CFS}(t) \quad (16)$$

$$T_i(0, t) = W_i^{P-CFS}(0, t) + Q_i^{P-CFS}(t) \quad (17)$$

위 두식에서  $T_i(0, t)$ 를 소거한 후, 정리 3을 적용하면 부등식 (15)가 만족된다. □

### 5. 성능평가

제안된 모델과 알고리즘을 검증하기 위해 모의실험을 통해 성능평가를 수행하였다. 모의실험에서는 각 세션들이 자신의 가중치에 맞는 용량의 서비스를 받는 지 확인하는 실험, 서비스 개시시간이 다른 세션이 자신의 가중치에 비례하는 용량으로 서비스를 받는 지 확인하는 실험, 각 세션의 보장서비스 용량이 잘 지켜지는 지 확인하는 실험, 그리고 제시된 알고리즘과 PGPS(Pakcet GPS)간 누적 서비스 용량의 비교에 관한 실험을 수행하였다. 실험에는 세 종류의 트래픽, 멀티미디어, 웹, 파일전송이 사용되었다. 멀티미디어 트래픽은 매 33msec마다 8kbytes 패킷을 전송한다고 가정하였고, 웹 트래픽은 on/off 모델로 모델링 하였는데, on 주기와 off 주기는 파레토분포를 갖는다고 가정하였다. 그리고 파일 전송트래픽은 지속적으로 패킷을 생성한다고 가정하였다.

그림 3은 세 개의 세션 파일전송, 멀티미디어, 웹이 각각 가중치 0.1, 0.2, 0.3으로 서비스 될 때, 누적 서비스 용량을 보여준다. 서비스 개시 초반에는 약간의 변동성이 발생하지만, 시간이 지남에 따라 자신의 서비스 용량에 비례하는 누적 서비스 용량으로 서비스가 제공됨을 볼 수 있다.

그림 4는 세 개의 파일전송 세션이 각각 가중치 0.1, 0.2, 0.3으로 서비스 될 때, 각 세션의 서비스 개시 시간이 다른 경우에 대한 누적 서비스 용량을 보여준다. 파일 전송 세션 2와 3은 각각 시작 500과 1000에 서비스가 개시되었는데, 다른 세션이 장시간 서비스를 받지 못하는 기아현상(starvation)을 발생시키지 않고 점차 시간이 지남에 따라 자신의 가중치에 비례하는 서비스 용량으로 서비스됨을 볼 수 있다.

그림 5는 멀티미디어와 파일 전송 세션이 각각 가중치 0.1과 0.2로 서비스될 때, 각 세션의 패킷 각각이 받는 서비스 용량을 나타낸 것이다. 두 세션의 가중치가 각각 0.1, 0.2이므로 이 이상의 서비스 대역폭을 보장받

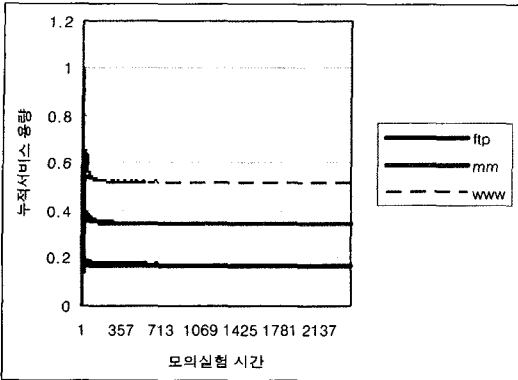


그림 3 파일전송, 멀티미디어, 웹 세션의 누적 서비스 용량의 비교

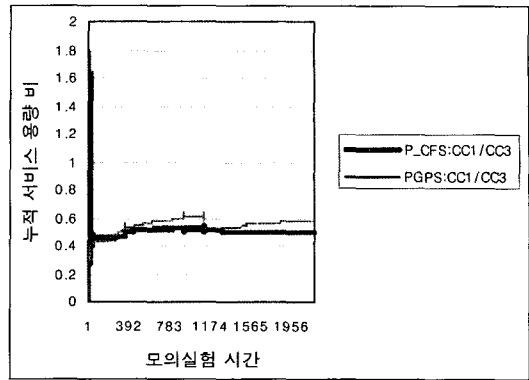


그림 6 P\_CFS와 PGPS의 누적 서비스 용량 비 비교

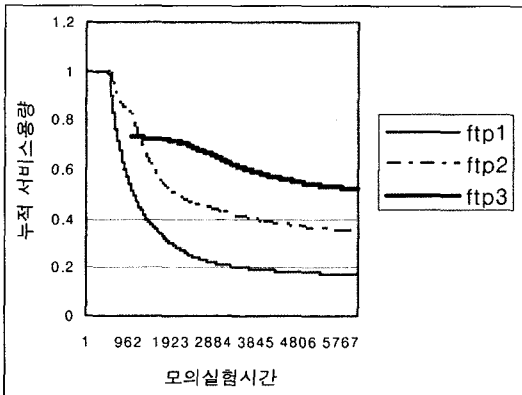


그림 4 서비스 개시시점이 다른 파일전송 세션의 누적 서비스 용량 비교

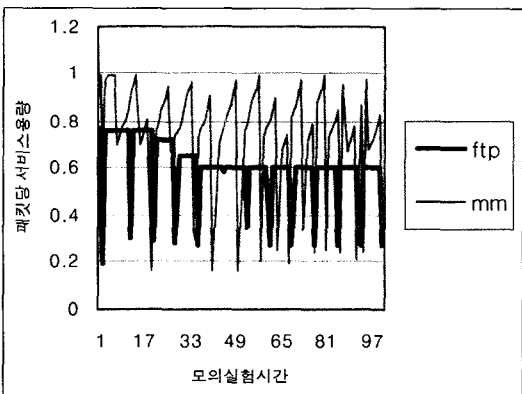


그림 5 파일 전송과 멀티미디어 세션의 패킷 당 제공되는 서비스 용량

아야 되는데, 그림에서 보듯이 멀티미디어와 파일전송세션의 패킷들이 각각 자신의 보장 용량 0.1과 0.2 이상의 서비스 용량으로 서비스됨을 볼 수 있다.

그림 6은 하나의 파일전송과 두 개의 웹 세션이 각각 가중치 0.1, 0.1, 0.2로 서비스될 때, P\_CFS와 PGPS 알고리즘에서의 가중치 0.2 인 웹 세션과 가중치 0.1인 파일전송 세션의 상대적인 누적 서비스 용량 비를 보여준다. 이론적으로 가중치 0.2인 세션을 기준으로 다른 세션을 비교하였으므로 누적 서비스 용량은 0.5가 될 때 이상적인 서비스가 이루어졌다고 볼 수 있다. P\_CFS를 수행 시, 가중치 0.2인 세션의 누적용량(CC3)과 파일전송 세션의 누적 용량 비(CC1)는 초반 약간의 변동성이 발생하였지만, 시간이 지남에 따라 이상적인 용량 비인 0.1/0.2=0.5에 수렴함을 볼 수 있다. 그러나 PGPS의 경우 CC3와 CC1의 누적 서비스 용량 비는 전반적으로 0.5보다 큰 값을 보여주고 있다.

### 6. 결론

본 논문에서는 긴 시간의 관점에서 세션에게 공평한 용량으로 서비스할 수 있는 모델과 이 모델을 참조하여 수행하는 알고리즘을 제안하였다. 이 모델과 알고리즘을 통해 세 종류의 서비스 즉, 세션에게 자신의 가중치에 비례하는 용량의 서비스 제공, 일정 대역폭 보장을 원하는 세션에게 보장서비스 제공, 그리고 지연 민감 세션과 처리 지향 세션 간 경쟁 시, 지연 민감 세션에게 우선 서비스 제공을 실현하였다. 그리고 모델과 알고리즘의 서비스 지연, 진행되는 서비스 양의 차이, 그리고 대기 중인 패킷 양의 차이 분석을 통해 알고리즘이 모델의 동작과 최대패킷크기 정도의 작은 차이를 가지고 수행함을 검증하였다. 마지막으로, 성능평가를 통해 알고리즘이 세션에게 긴 시간 관점에서 가중치에게 비례하는 용량을 제공하며, 항상 보장 대역폭 이상의 서버용량이 제공됨을 보였다.

앞으로의 연구 계획은 무선, 이동환경에서의 세션 간 공정서비스 제공에 관한 것이다. 무선 및 이동환경에서는 잡음, 신호의 간섭이 많이 발생하기 때문에, 각 세션

에게 순시적인 공정 서비스를 제공하기가 쉽지 않다. 이런 문제를 본 논문의 제안방법인 긴 시간의 관점에서 공정 서비스 제공으로 접근해 보고자 한다.

**참고 문헌**

- [1] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp.344-357, 1993.
- [2] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm," in *Proc. of ACM SIGCOMM*, pp. 1-12, 1989.
- [3] P. Goyal, H. Vin, and H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks," *IEEE/ACM Trans. on Networking*, vol. 5, no.5, pp. 690-704, 1997.
- [4] J. Bennett and H. Zhang, "WF2Q: Worst-case Fair Weighted Fair Queueing," in *Proc. of IEEE INFOCOM*, pp. 120-128, 1996.
- [5] S. Golestani, "A Self-clocked Fair Queueing Scheme for Broadband Applications," in *Proc. of IEEE INFOCOM*, pp. 636-646, 1994.
- [6] S. Lu, V. Bharghavan, and R. Srikant, "Fair Scheduling in Wireless Packet Networks," *IEEE/ACM Trans. on Networking*, vol. 7, no. 4, pp. 473-489, 1999.
- [7] P. Ramanathan and P. Agrawal, "Adapting Packet Fair Queueing Algorithms to Wireless Networks," in *Proc. of MOBICOM '98*.
- [8] R. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Network," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1048-1056, 1995.
- [9] H. Sariowan, "SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service," *IEEE/ACM Trans. on Networking*, vol. 7, no. 5, pp. 669-684, 1999.
- [10] J. Bennett and H. Zhang, "Hierarchical Packet Fair Queueing Algorithm," *IEEE/ACM Trans. on Networking*, vol. 5, no. 5, pp.676-689, Oct. 1997.



**이 주 현**  
 1992년 서울대학교 공과대학 컴퓨터공학과 학사. 1994년 서울대학교 공과대학 컴퓨터공학과 석사. 1996년~ 서울대학교 공과대학 전기컴퓨터 공학부 박사과정. 관심분야는 스케줄링, 무선랜, 이동통신



**박 경 호**  
 1987년~1991년 서울대학교 컴퓨터공학과 학사. 1991년~1993년 서울대학교 컴퓨터공학과 석사. 1993년~현재 서울대학교 컴퓨터공학과 박사과정. 관심분야는 스케줄링



**황 호 영**  
 1989년~1993년 서울대학교 컴퓨터공학 공학사. 1993년~1995년 서울대학교 컴퓨터공학 공학석사. 1995년~2003년 서울대학교 전기컴퓨터공학 공학박사. 2003년~현재 안양대학교 디지털미디어학부 조교수. 관심분야는 정보통신, 광통신 및 이동통신망, 멀티미디어시스템 등



**민 상 렬**  
 1983년 서울대학교 전자계산기공학과, 공학사. 1985년 서울대학교 전자계산기공학과 공학석사. 1989년 워싱턴대학 전산학과 박사. 1989년~1990년 IBM Watson 연구소 객원 연구원. 1990년~1992년 부산대학교 컴퓨터공학과 조교수. 1992년~현재 서울대학교 컴퓨터공학과 조교수. 관심분야는 컴퓨터 구조, 병렬처리 시스템, 캐쉬 메모리 시스템 등