

# 최적 레이턴시 기반 공정 큐잉 알고리즘의 대역폭 이용도

김 태 준<sup>\*</sup>

요 약

널리 이용되고 있는 공정 큐잉 방식인 WFQ(Weighted Fair Queuing)는 특히 인터넷 전화와 같이 저속이지만 엄격한 지연한계를 요구하는 서비스에 대해 대역폭 이용도가 저하되는 고질적인 문제점을 갖고 있었다. 이러한 WFQ의 문제점은 흐름의 레이턴시가 최적화 되지 않았기 때문으로 최근에 밝혀졌고, 이어서 최적 레이턴시 기반 공정 큐잉 방식인 LOFQ(Latency-Optimized Fair Queuing)가 도입되었다. 그리고 LOFQ에 점유자원 최적화 기능을 추가하여 대역폭 이용도를 개선하고, 반복적으로 수행되던 자원변환 알고리즘을 개선하여 수행 복잡성을 줄이는 등 LOFQ의 성능특성이 향상되었다. 하지만 여전히 시뮬레이션에 의해 성능을 평가하고 있어 평가의 정확성과 소요시간 측면에서 어려움이 있다. 따라서 본 연구에서는 LOFQ의 대역폭 이용도를 수학적으로 계산하는 방법을 연구하고자 한다.

키워드 : WFQ, 공정 큐잉, 지연 자원, 최적 레이턴시, LOFQ

## Bandwidth Utilization in Latency-Optimized Fair Queuing Algorithm

Tae Joon Kim<sup>\*</sup>

ABSTRACT

WFQ (Weighted Fair Queuing) is the most popular fair queuing algorithm, but it had the inherent drawback of a poor bandwidth utilization, particularly under the traffic requiring a low rate but tight delay bound such as internet phone. It was recently identified that the poor utilization is mainly due to non-optimized latency of a flow and then LOFQ(Latency-Optimized Fair Queuing) to overcome the drawback was introduced. LOFQ was also improved through introducing an occupied resource optimization function and the implementation complexity of recursive resource transformation was reduced with revising the transformation scheme. However, the performance of LOFQ has been evaluated by means of simulation, so that there are some difficulties in evaluating the performance in the terms of the accuracy and evaluation time. In this paper, we develop how to analytically compute the bandwidth utilization in LOFQ.

Key Words : WFQ, Fair queuing, Delay resource, Optimal latency, LOFQ

### 1. 서 론

인터넷 전화, 인터넷 영상회의와 같은 고 수준의 서비스 품질을 요구하는 실시간 멀티미디어 통신 서비스를 수용하기 위해 IETF(Internet Engineer Task Force)에서 자원예약 기반의 종합서비스(IntServ) 모델을 제시하였다[1]. IntServ 모델에서 품질 보장형 서비스를 지원하기 위해 자원예약 기반의 RSVP(Reservation protocol) 라우터가 사용되는데, RSVP 라우터는 트래픽 흐름의 요구 속도를 보장하고 허용되는 레이턴시(latency), 즉 지연(delay) 규격을 위반하지 않기 위해 공평 패킷 스케줄러를 탑재한다.

유체모델 기반의 이상적 패킷 스케줄링 방식인 GPS(General Processor Sharing)[2]는 WFQ(Weighted Fair Queuing)

[3][4]에 의해 구현되었다. WFQ는 흐름간의 상호 간섭을 차단하여 각 흐름의 품질 특성이 다른 흐름으로부터 영향을 받지 않도록 하는 흐름 분리와 차별화된 품질 보장을 지원하는 등 공정 큐잉의 요구사항을 충실히 따르기 때문에 IETF IntServ 모델의 RSVP 라우터에 적용되었다[5]. 그러나 흐름의 스케줄링 속도를 높일 경우 스케줄링 속도와 요구속도와의 차이 만큼 속도(대역폭)자원이 손실되는데, 특히 인터넷 전화와 같이 트래픽 흐름의 속도는 낮지만 엄격한 종단간 지연특성을 요구하는 트래픽 흐름의 경우 대역폭 이용도의 저하가 심각해짐이 밝혀졌다[6].

WFQ의 이용도 저하 문제를 해결하기 위해 다양한 연구가 이루어졌다. 음성 트래픽의 지원을 위해 감시기를 갖는 WFQ가 제안되었는데[7], 이는 음성 흐름의 스케줄링 속도를 다소 과다하게 할당하는 대신 필요시 감시기가 패킷을 드롭(drop)하는 방식을 도입하였다. 한편 가변 스케줄링 속도를 사용하는 최적 네트워크 서비스 곡선 방식[8]과 크레디트

<sup>\*</sup> 정 회 원 : 공주대학교 정보통신공학부 부교수  
논문접수: 2006년 8월 14일, 심사완료: 2007년 1월 24일

(Credit) 기반 프로세서 공유방식[9]이 제안되었는데, 이는 일시적으로 예약된 속도보다 높은 스케줄링 속도를 요구하는 흐름들에 대해서 대역폭에 여유가 있는 다른 흐름의 대역폭을 빌려주는 방식이다. 참고로 본 논문에서 속도(rate)와 대역폭(bandwidth)는 같은 의미로 사용된다. 그러나 이러한 방식은 빌려줄 만큼 대역폭의 여유를 가진 흐름이 줄어들 경우 그 효과가 반감되는 문제가 있다.

그 동안 WFQ의 대역폭 저하 문제는 WFQ의 속도와 대역폭의 결합특성 때문이라고 이해되어 왔다[6-9]. 최근에 WFQ의 고질적인 문제의 근원은 레이턴시가 최적화 되지 않아서 발생하는 지연자원의 낭비 때문으로 밝혀졌고, 그러한 낭비를 방지하기 위해 레이턴시가 최적화되는 LOFQ (Latency-Optimized Fair Queuing) 방식이 연구되었다[10]. 그리고 점유자원 최적화 기능이 추가되어 대역폭 이용도가 더욱 개선되었고 반복적으로 수행되던 LOFQ의 자원변환 알고리즘이 개선되어 수행 복잡성도 대폭 줄어들었다[11]. 하지만 여전히 시뮬레이션에 의해 성능특성을 평가하고 있어 평가의 정확성과 소요시간 측면에서 어려움이 있다. 따라서 본 연구에서는 LOFQ의 대역폭 이용도를 수학적으로 계산하는 방법을 연구하고자 한다.

본 논문의 구성은 다음과 같다: 2장에서 관련 연구를 소개하고, 3장에서 LOFQ의 수락제어와 점유자원 최적화 알고리즘을 살펴보면, 4장에서 대역폭 이용도 측면에서 LOFQ의 특성을 분석하고 대역폭 이용도를 구하는 방법을 제시한다. 5장에서 수치해석과 시뮬레이션을 통해 대역폭 이용도를 확인하고, 6장에서 결론을 맺는다.

## 2. 관련 연구

먼저 스케줄러에 도착하는 흐름의 속성을 나타내는 흐름 규격  $F(r,b,M)$ 를 정의하는데,  $r$ 은 흐름의 요구속도,  $b$ 는 스케줄러에 할당되는 흐름의 지연예산(delay budget),  $M$ 은 흐름의 최대 패킷크기이다. 흐름의 지연예산이란 중단간 지연한계가 각 노드에 분배된 것으로 본 연구에서는 간단히 지연규격이라 칭한다. 참고로 공정 큐잉 알고리즘은 흐름의 요구속도  $r$ 과 지연규격  $b$ 를 만족시키기 위한 것이다. WFQ에서 흐름 규격  $F(r,b,M)$ 을 갖는 임의 흐름의 고유 레이턴시  $q^r$ 은 다음과 같이 주어진다[2].

$$q^r = \frac{M}{r} + \frac{M^{\max}}{C^B}, \quad (2.1)$$

여기서  $C^B$ 는 출력링크의 용량으로 스케줄러의 총 대역폭을 의미하고,  $M^{\max}$ 는 모든 흐름의 최대 패킷크기이다. WFQ는 요구속도와 지연규격을 동시에 만족시켜야 하므로 임의 흐름의 스케줄링 속도  $s$ 는 (2.1)로부터 다음과 같이 계산된다.

$$s = \max(r, r^C), \text{ 여기서 } r^C = \frac{MC^B}{bC^B - M^{\max}}. \quad (2.2)$$

(2.2)내  $r^C$ 를 임계속도라 한다. 그러면 WFQ에서 임의 흐름의 실제 레이턴시  $q$ 는 다음과 같이 계산된다.

$$q = \frac{M}{s} + \frac{M^{\max}}{C^B}. \quad (2.3)$$

참고로 WFQ의 경우 (2.2)와 (2.3)으로부터  $q \leq b$ 가 된다. (2.3)에서  $M^{\max}/C^B \ll M/s$  이므로 WFQ에서 각 흐름의 레이턴시와 스케줄링 속도는 서로 반비례하는 결합관계를 갖는다. 이러한 결합관계가 바로 WFQ의 성능저하의 주된 요인으로 이해되어 왔으나[6-9], 최근에 다음과 같이 지연자원의 낭비에 기인하는 것으로 밝혀졌다[10]. (2.1)로부터 고유 레이턴시  $q^r$ 와 지연규격  $b$ 는  $q^r=b$ ,  $q^r>b$  또는  $q^r<b$ 의 관계를 갖는다.  $q^r=b$ 의 경우 자원의 손실 없이 흐름을 수용할 수 있다.  $q^r>b$ 의 경우 여유 대역폭 자원이 있다면 스케줄링 속도를 높여서 지연규격을 충족, 즉  $q=b$ 가 되게 해야 한다. 이때 스케줄링 속도와 요구속도의 차이분 만큼 대역폭 자원이 손실되지만 지연 품질을 만족시키기 위해 피할 수 없다.  $q^r<b$ 의 경우  $q=b$ 가 되게 하려면 스케줄링 속도를 요구속도보다 낮게 해야 한다. 하지만 흐름의 요구속도를 보장해야 하기 때문에 스케줄링 속도를 낮출 수 없어 지연규격보다 더 엄격한 레이턴시를 갖게 된다. 이 결과  $(b-q)$ 만큼의 지연 자원이 손실되어 스케줄러 자원 이용 효율이 저하된다는 것이다. 그리고 이러한 지연자원의 낭비를 근본적으로 차단하는 LOFQ가 연구되었다[10].

LOFQ는 레이턴시 지수  $\beta \geq 0$ 와  $[0, D^B]$ 내, 여기서  $D^B$ 는 흐름 중 가장 엄격한 지연규격 값으로 설정됨, 어떤 값을 갖는 레이턴시 바이어스(bias)  $D$ 를 도입한 후 (2.4)와 같이 각 흐름의 레이턴시  $q$ 를  $\beta$ 와 스케줄링 속도  $h$ 의 함수가 되게 하였다. 그리고  $\beta$ 와  $h$ 를 조정하여  $q$ 를  $b$ 와 동일하게, 즉 항상 지연규격과 동일한 최적의 레이턴시를 갖도록 하는데, 대역폭 점유를 최소화 하기 위해  $h$ 를 우선적으로 줄인다. 참고로  $h$ 란 바로 그 흐름이 점유하는 대역폭이 된다.

$$q = D^B + \beta \frac{M}{h} = b. \quad (2.4)$$

레이턴시 바이어스  $D$ 는 흐름을 많이 수용할수록 증가하지만 그 값이  $D^B$ 를 초과할 수 없다. 따라서 대역폭(BW: BandWidth)과 마찬가지로 레이턴시 바이어스는 일종의 자원으로 취급될 수 있는데, 이를 지연자원(DW: Delay Width)이라고 한다. 그리고  $D^B$ 와  $D$ 를 각각 스케줄러의 DW 용량과 스케줄러의 점유 DW라 한다. 그리고  $\beta$ 의 레이턴시 지수를 갖는 흐름에 의해 증가되는 레이턴시 바이어스 값, 즉 그 흐름의 점유 DW가 되는  $d$ 는 다음과 같이 주어졌다.

$$d = \max(0, 1-\beta)M/C^B. \quad (2.5)$$

그리고 스케줄러의 점유 DW, 즉 레이턴시 바이어스  $D$ 는 다음과 같이 주어졌다.

$$D = \max\left(\frac{M^{\max}}{C^B}, \frac{\sum_{B(t)} \max(0, 1-\beta)M}{C^B}\right), \quad (2.6)$$

여기서 B(t)는 수용된 흐름 집합이다.

(2.4)와 (2.5)로부터 흐름의 점유 BW h와 점유 DW d는 서로 반 비례 하므로, 점유 BW를 늘리는 대신 점유 DW를 줄일 수 있고, 반대로 점유 DW를 늘리는 대신 점유 BW를 줄일 수 있는, 즉 DW와 BW가 상호 변환될 수 있음을 알 수 있다. 여기서 전자를 B2D 변환, 후자를 D2B 변환이라 한다. 그리고 늘어나는 BW 대비 줄어드는 DW의 비율을 B2D 변환 효율  $E^{B2D}$ , 늘어나는 DW 대비 줄어드는 BW의 비율을 D2B 변환 효율  $E^{D2B}$ 라 한다. WFQ의 경우 스케줄러의 점유 BW가  $C^B$ 에 도달하면 더 이상 흐름을 수용할 수 없다. 하지만 LOFQ의 경우 다음과 같이 자원변환을 통해 대역폭 이용도를 높인다: 새로운 흐름 수용에 필요한 DW와 BW가 모두 부족하면 흐름 수락을 중단한다. 하지만 DW는 부족하나 BW는 충분할 경우 B2D 변환을 통해 부족한 DW를 확보하고, 반대로 BW는 부족하나 DW는 충분할 경우 D2B 변환을 통해 부족한 BW를 확보한 후 그 흐름을 수락한다.

[10]에서 연구된 LOFQ는 초기버전으로서 점유 자원이 최적화 되지 못하고, D2B 변환과 B2D 변환시 변환될 자원량을 알 수 없어서 미소량 단위로 변환동작을 반복 수행함으로써 인해 자원변환의 수행 복잡도가 너무 높다는 문제가 제기되었다[11]. 이어서 스케줄러의 점유자원 최적화를 통해 이용도 특성을 대폭 개선하였고, D2B 변환과 B2D 변환시 변환될 수 있는 최대 자원량을 산정하여 흐름 별로 한번의 자원변환만 수행 함으로써 자원변환의 횟수를 최소화 하였다[11].

### 3. LOFQ 알고리즘

LOFQ의 대역폭 이용도를 분석하기에 앞서 LOFQ 알고리즘을 통해 LOFQ의 기능을 살펴본다. LOFQ 스케줄러는 자신의 점유 BW와 DW를 나타내는 두 레지스터 C와 D를 운영하고, 수락된 임의 흐름 i에 대해 흐름 프로파일  $P_i(r_i, b_i, \beta_i, M_i)$ 를 할당한다. 참고로 P()내 변수의 아래첨자는 생략될 수 있다. 관련되는 LOFQ의 기능은 흐름 수락제어와 점유자원 최적화이다.

#### 3.1 흐름 수락제어 기능

흐름 수락제어 기능에 대한 c언어 형태의 알고리즘이 그림 1에 정리되어 있다.  $F(r_i, b_i, M_i)$ 를 갖는 임의 흐름 i에 대한 수락요청이 입력되면 수락제어 기능이 동작한다. 먼저 스케줄링 속도  $h_i$ 를 WFQ에서의 스케줄링 속도인  $s_i$ 로 설정하고 레이턴시 지수  $\beta_i$ 를 계산한 후  $\beta_i \in (0, 1]$  이면  $h_i$ 가 최소가 되도록 조정된 후  $\beta_i$ 를 갱신한다. 이는 스케줄링 속도를 우선적으로 줄이기 때문이다. 그리고 가용 자원을 조사

하여 BW와 DW 모두 충분하면 흐름을 수락하고, 모두 부족하면 수락을 거절하며, DW만 부족하면 B2D변환, BW만 부족하면 D2B변환 후 흐름을 수락한다. 그림 1에서  $P_i()$ 는 임시 프로파일이다. 참고로 LOFQ 스케줄러는 항상 새로운 흐름을 수락하기 전에 점유자원 최적화를 완료한다.

그림 1과 같이 수락제어 기능은 각각 B2D 변환과 D2B 변환을 위한 두 서브알고리즘 B2DT()와 D2BT()을 포함한다. B2D 변환에서 투입되는 BW 량  $\Delta h_j^{B1}$  과  $\Delta h_j^{B2}$  는 다음과 같이 주어진다[11].

$$\Delta h_j^{B1} = \min\left(\frac{h_i}{\beta_i} - h_i, \frac{\Delta D}{E_i^{B2D}}, C^B - C\right) \quad \text{및} \quad \Delta h_j^{B2} = \min\left(\frac{h_i}{\beta_i} - h_i\right). \quad (3.1)$$

그리고 D2B 변환에서 감소되는 레이턴시 지수  $\Delta \beta_j^B$  량은 다음과 같다[11]

$$\Delta \beta_j^B = \min\left(\beta_i, \frac{h_i - r_i}{h_i} \beta_i, \frac{\Delta B}{E_i^{D2B}} \frac{C^B}{M_i}, \frac{(D^B - D)C^B}{M_i}\right). \quad (3.2)$$

수락된 임의 흐름 i에 대한 해제 요청 시 다음과 같이 흐름의 점유 BW와 DW를 반환한다:  $C = -h_i$  과  $D = -d_i$ . 그리고 흐름의 프로파일을 제거한다.

#### 3.2 점유자원 최적화 기능

점유자원 최적화란 스케줄러에 아무런 영향을 주지 않고 스케줄러의 점유 BW 또는 DW를 최소화 하는 것으로서 예를 통해서 그 의미를 살펴본다.  $P_i(10Kbps, 61ms, 10Kbps, 0.6, 1000bit)$ 인 흐름 i와  $P_j(10Kbps, 21ms, 30Kbps, 0.6, 1000bit)$ 인 흐름 j를 생각해보자. 흐름 i의 경우 B2D 변환으로  $\beta_i$ 를 최대 0.4만큼 늘릴 수 있는데 이로 인해 줄어드는 흐름 i의 점유 DW는  $400/C^B$  이고 늘어나는 점유 BW는  $6/(100C^B) * 400/C^B = 24Hz$ 가 된다.  $0.4M_i/C^B$ 의 DW를 흐름 j에 투입하여 D2B 변환을 하면 흐름 j의 줄어드는 BW는  $(400/C^B) * 50C^B = 2000Hz$ 가 된다. 이 결과 스케줄러의 점유 DW 변화 없이 점유 BW를 1976Hz 만큼 줄이게 된다.

최적화 기능에 대한 c언어 형태의 알고리즘이 그림 2에 정리되어 있다. 가장 효율이 좋은 B2D 변환가능 흐름 I에 대한 B2D 변환에서  $\Delta d_i^B$ 의 DW를 확보한 후 이를 가장 효율이 좋은 D2B 변환가능 흐름 j에 투입하여  $\Delta d_i^B E_j^{D2B}$ 의 BW를 확보하는 것으로서  $E_i^{D2B} * E_j^{B2D}$ 으로 계산되는 이득이 1 이하일 경우 더 이상 점유자원을 줄일 여지가 없으므로 중단한다. 그리고 B2D 변환에서 확보하는 DW 량  $\Delta d_i^B$ 은 다음과 같이 주어진다[11].

$$\Delta d_i^B = \min\left(\frac{(1-\beta_i)M_i}{C^B}, \frac{\beta_j M_j}{C^B}, \frac{(h_j - r_j)\beta_j M_j}{h_j C^B}\right). \quad (3.2)$$

```

Admit() { /*** 흐름 수락제어 기능 ***
     $h_i = s_i$  and  $\beta_i = (b_i - D^B)h_i/M_i$  //  $\beta$  계산
    if(  $\beta_i = 0$  )  $h_i = r_i$  //  $h$  조정 및  $\beta$ 갱신
    if(  $\beta_i > 1$  )  $h_i = \max(r_i, M_i/(b_i - D^B))$  and  $\beta_i = (b_i - D^B)h_i/M_i$ 
    if(  $C+h_i > C^B$  and  $(D+d_i) > D^B$  ) return fail //모든 자원 부족이면 거절
    else if(  $(D+d_i) > D^B$  ) if (B2DT( -1, D+d_i - D^B, 1) return fail //DW만 부족이면 B2D변환
    else if(  $C+h_i > C^B$  ) if(D2BT(-1, C+h_i - C^B) return fail //BW만 부족이면 D2B변환
     $P_i(r, b, h, \beta, M) = P_i(r_i, b_i, r_i, \beta_i, M_i)$  //수락되면 프로파일 할당
     $C+ = h_i$  and  $D+ = d_i$  //점유 BW 및 DW 갱신
}

D2BT(j,  $\Delta B$ ) { // D2B 변환기능: DW 늘리는 대신  $\Delta B$ 의 BW 줄임
    While( $\Delta B > 0$ ) {
        if( $j \geq 0$ )  $j = -1$  and  $j = \{k \mid \max_{k \in B(t), r_k < R_k \text{ and } 0 < \beta_k \leq 1} E_k^{D2B}\}$  //흐름 지정 안되면 효율 순서
        if( $j < 0$  or  $\Delta \beta_j^B \leq 0$ ) return fail //실패시 에러 리턴
         $\Delta h_j = h_j \Delta \beta_j^B / \beta_j$  and  $\Delta B- = \Delta h_j$  //BW 감소분 계산,  $\Delta B$ 갱신
         $C- = \Delta h_j$  and  $D+ = M_j \Delta \beta_j^B / C^B$  //점유 BW와 DW 갱신
         $P_j(r, b, h, \beta, M) = P_j(r, b, h - \Delta h_j, \beta - \Delta \beta_j^B, M)$  //흐름 j 프로파일 갱신
    }
}

B2D(j,  $\Delta D$ , flag) { // B2D 변환기능: BW 늘리는 대신  $\Delta D$ 의 DW 줄임, flag  $\geq 0$ 이면  $C > C^B$  허용
    While( $\Delta D > 0$ ) {
        if( $j \geq 0$ )  $j = -1$  and  $j = \{k \mid \max_{k \in B(t) \text{ and } 0 < \beta_k < 1} E_k^{B2D}\}$  //흐름 지정 안되면 효율 순서
        if( $j < 0$  or  $\Delta h_j^B \leq 0$ ) return fail //실패시 에러 리턴
         $\Delta h_j^B = \Delta h_j^{B1}$  and if(flag  $>= 0$ )  $\Delta h_j^B = \Delta h_j^{B2}$ 
         $\Delta \beta_j = \beta_j \Delta h_j^B / h_j$  and  $\Delta D- = M_j \Delta \beta_j / C^B$  //  $\beta$  증가분 계산,  $\Delta D$ 갱신
         $C+ = \Delta h_j^B$  and  $D- = M_j \Delta \beta_j / C^B$  //점유 BW와 DW 갱신
         $P_j(r, b, h, \beta, M) = P_j(r, b, h + \Delta h_j^B, \beta + \Delta \beta_j, M)$  //흐름 j 프로파일 갱신
    }
}

```

그림 1. 흐름 수락제어 알고리즘

```

While() {
    i = {k | max_{k ∈ E(t) and 0 < β_k < 1} E_k^{B2D}} //최고 B2D변환효율의 흐름선택
    j = {k | max_{k ∈ E(t), r_k < R_k and 0 < β_k ≤ 1} E_k^{D2B}} //최고 D2B변환효율의 흐름선택
    if(E_i^{D2B} * E_j^{B2D} ≤ 1) stop; //이득이 없으면 중단
    B2DT(i, Δd_i^B, 1) //점유 BW 제약 없이 B2D 변환
    D2BT(j, Δd_j^B E_j^{D2B}) //확보된 Δd_i^B 투입하여 D2B 변환
}
    
```

그림 2. 점유자원 최적화 알고리즘

참고로  $C=C^B$ 인 상태에서도 점유자원 최적화가 가능하도록 B2D 변환시 스케줄러의 점유 BW의 제약, 즉  $C \leq C^B$ 의 조건을 무시하도록 한다.

#### 4. 대역폭 이용도

##### 4.1 기본 특성 분석

흐름  $i$ 의 점유 DW 변화량을  $\Delta d_i$ , 점유 BW 변화량을  $\Delta h_i$ 로 표기한다. 먼저 스케줄러에 수용된 각 흐름 내부에서 BW와 DW간의 상호변환, 즉 흐름 내 (intra-flow) 자원변환과 그의 효율을 살펴보자.

**보조정리 1:** 임의 D2B 변환가능 흐름  $i$ 가 D2B 변환을 통해 자신의 점유 DW를  $\Delta d_i$  만큼 증가시키는 대신 자신의 점유 BW를  $\Delta h_i$ 만큼 줄일 수 있다. 여기서  $\Delta h_i = E_i^{D2B} \Delta d_i$  및  $E_i^{D2B} = C^B / (b_i - D^B) = h_i C^B / (\beta_i M_i)$ . 그리고 임의 B2D 변환가능 흐름  $i$ 도 B2D 변환을 통해서 자신의 점유 BW를  $\Delta h_i$  만큼 늘리는 대신 자신의 점유 DW를  $\Delta d_i$  만큼 줄일 수 있다. 여기서  $\Delta d_i = E_i^{B2D} \Delta h_i$  및  $E_i^{B2D} = 1 / E_i^{D2B}$ .

**증명:** (2.5)로부터 흐름  $i$ 의 점유 DW를  $\Delta d_i$ 만큼 증가시키기 위해 변화되어야 하는 레이턴시 지수 값  $\Delta \beta_i$  는

$$\Delta \beta_i = -(C^B / M_i) d_i \tag{4.1}$$

$q_i$ 가  $b_i$ 와 동일한 값을 갖도록 해야 하므로 (2.4)로부터  $\Delta h_i$  는 다음과 같이 전개된다.

$$\Delta h_i = -\frac{M_i}{b_i - C^B} \Delta \beta_i = -\frac{C^B}{(b_i - C^B)} \Delta d_i = -\frac{h_i C^B}{\beta_i M_i} \Delta d_i \tag{4.2}$$

(4.2)로부터 보조정리가 증명된다.

모든 흐름이 모두 D2B 변환과 B2D 변환의 대상이 되지 않는다. 임의 흐름이 이들 변환의 대상이 되는, 즉 D2B 변환가능과 B2D 변환가능이 되기 위한 조건을 살펴보자.

**보조정리 2:**  $P_i(r, b, h, \beta, M)$ 를 갖는 임의 흐름  $i$ 에 대해  $\beta_i \in (0, 1]$ 이고  $h_i > r_i$ 이면 D2B 변환가능이 되고,  $\beta_i \in (0, 1)$ 이면 B2D 변환가능이 된다.

**증명:** 흐름  $i$ 에 대해 (2.5)로부터  $\beta_i \in (0, 1]$ 이면  $\beta_i$ 를 줄여서  $d_i$ 를 늘릴 수 있다. 하지만  $h_i = r_i$ 이면 요구속도 보장 때문에  $h_i$ 를 감소시킬 수 없다. 따라서 유효한  $E_i^{D2B}$  값을 가지려면  $\beta_i \in (0, 1]$ 인 레이턴시 지수와  $h_i > r_i$ 인 스케줄링 속도를 가져야 하고, 그러면 보조정리 1로부터 D2B 변환의 대상이 될 수 있다. 한편 (2.4)로부터  $\beta_i > 0$ 이면  $\beta_i$ 를 증가시켜  $h_i$ 를 늘릴 수 있고, (2.5)로부터  $\beta_i \in (0, 1)$ 이면  $\beta_i$ 를 증가시켜  $d_i$ 를 줄일 수 있다. 따라서  $\beta_i \in (0, 1)$ 이면 유효한  $E_i^{B2D}$  값을 갖게 되므로 보조정리 1로부터 B2D 변환의 대상이 될 수 있다.

보조정리 2로부터 다음의 추론이 도출된다.

**추론 1:**  $P_i(r, b, h, \beta, M)$ 를 갖는 임의 흐름  $i$ 에 대해  $\beta_i \notin (0, 1]$ 이거나  $h_i = r_i$ 이면 D2B 변환 불가능이 되고,  $\beta_i \notin (0, 1)$ 이면 B2D 변환 불가능 된다.

**추론 2:**  $P_i(r, b, h, \beta, M)$ 를 갖는 임의 흐름  $i$ 의  $\beta_i = 1$ 이고  $h_i > r_i$ 이면 흐름  $i$ 는 D2B 변환 가능이고 B2D 변환 불가능이다.

다음은 서로 다른 흐름 사이에 BW와 DW의 상호 변환, 즉 흐름 간 (inter-flow) 자원변환에 대해 살펴보자. 흐름  $i$ 에 대한 B2D 변환을 통해  $\Delta d_i$ 의 DW를 먼저 확보한 다음 이를 흐름  $j$ 에 대한 D2B 변환에 투입하는 것이다. 그러면 흐름  $i$ 의 점유 BW가 늘어나는 반면 흐름  $j$ 의 점유 BW가 줄어든다. 물론 스케줄러의 점유 DW는 변화가 없다. 이러한 연속변환을 B2B 변환이라 하고, 그 효율을  $E_{ij}^{B2B}$ 로 표기한다. 그러면  $E_{ij}^{B2B} = E_i^{B2D} E_j^{D2B}$ .

**보조정리 3:**  $E_{ij}^{B2B} > 1$ 인 임의 B2D 변환가능 흐름  $i$ 와 D2B 변환가능 흐름  $j$ 가 존재하는 한 LOFQ 스케줄러는 B2B 변환을 통해 자신의 점유 BW를 줄일 수 있다.

**증명:** 보조정리 1로부터 흐름 i에 대한 B2D 변환은  $\Delta d_i = -E_i^{B2D} \Delta h_i$ 로 계산되는  $\Delta d_i$ 의 DW를 산출한다. 이 DW를 흐름 j에 대한 D2B 변환에 투입하면 흐름 j의 점유 BW는  $\Delta h_j = -E_j^{D2B}(-\Delta d_i) = E_j^{B2B} \Delta h_i$ 로 계산되는  $\Delta h_i$ 만큼 줄어든다.  $E_j^{B2B} > 1$ 이면 이러한 B2B변환은 스케줄러의 점유 BW를  $(|\Delta h_j| - |\Delta h_i|) > 0$  만큼 줄어든게 하므로 보조정리가 증명된다.

보조정리 1로부터 임의의 D2B 변환 가능 흐름에 대한 D2B 변환은 그 흐름의 스케줄링 속도가 그의 요구속도와 일치할 때까지 계속될 수 있고, 임의의 B2D 변환 가능 흐름에 대한 B2D 변환은 그 흐름의 레이턴시 지수가 1이 될 때까지 계속될 수 있다. 이로부터 스케줄링 속도와 요구속도가 서로 동일하여 더 이상 D2B 변환을 할 수 없을 때까지 수행한 D2B 변환을 D2B 완전변환이라 하고, 레이턴시 지수가 1이 되어 더 이상 B2D 변환을 할 수 없을 때까지 수행한 B2D 변환을 B2D 완전변환이라 하자. 그러면 다음과 같은 보조정리가 도출된다.

**보조정리 4:**  $F(r_i, b_i, M_i)$  를 갖는 임의의 D2B 가능 흐름 i가 D2B 완전변환 되면 그 흐름의 점유 BW는  $r_i$ 가 되고, 레이턴시 지수는  $(b_i - D^B)r_i/M_i \leq 1$ 가 되며, 점유 DW는  $(M - (b_i - D^B)r_i)/C^B$ 가 된다.

**증명:** 흐름 i가 D2B 완전변환 되면 정의에 의해  $h_i = r_i$ 이므로 점유 DW는  $r_i$ 가 된다. (2.4)로부터  $\beta_i = (b_i - D^B)r_i/M_i$ 가 되며, 보조정리 2로부터 D2B 변환이 종료될 때까지 D2B 변환가능해야 하므로  $\beta_i \in (0, 1]$ 이어야 한다. 따라서  $\beta_i = (b_i - D^B)r_i/M_i \leq 1$ . 그리고 (2.5)로부터 점유 DW  $d_i = (M - (b_i - D^B)r_i)/C^B$ .

**보조정리 5:**  $F(r_i, b_i, M_i)$ 를 갖는 임의의 B2D 가능 흐름 i가 B2D 완전변환 되면 그 흐름의 점유 BW는  $h_i^1 \equiv M_i/(b_i - D^B)$ 인  $h_i^1$ 가 되고, 점유 DW는 0이 된다.

**증명:** 흐름 i가 B2D 완전변환 되면 정의에 의해  $\beta_i = 1$ 이므로 (2.4)로부터  $h_i = M_i/(b_i - D^B)$ .  $\beta_i = 1$ 이므로 (2.5)로부터 점유 DW  $d_i = 0$ .

한편 D2B 변환 불가능 흐름과 B2D 변환 불가능 흐름의 점유자원을 살펴보면 다음 보조정리와 같다.

**보조정리 6:**  $F(r_i, b_i, M_i)$ 를 갖는 임의의 흐름 i가 D2B 변환 불가능이거나 B2D 변환 불가능일 경우 그 흐름의 점유 BW는  $r_i$ 가 되고, 점유 DW는  $\beta_i = 0$ 일 경우  $M_i/C^B$ 이고  $\beta_i > 1$ 일 경우 0이 된다.

**증명:**  $\beta_i = 0$ 이면 (2.4)로부터  $h_i$ 는 임의의 값을 가질 수 있는데, 흐름의 요구속도를 보장하면서 최소의 값을 가져야 한다. 따라서  $h_i = r_i$ . 그리고  $\beta_i > 1$ 이면 그림 1의 수락제어 알고리즘에 의해  $h_i = r_i$ 가 된다. 한편 (2.5)로부터  $\beta_i = 0$ 이면  $d_i = M_i/C^B$ 이

고  $\beta_i > 1$ 이면 0이 된다. 따라서 보조정리가 증명된다.

#### 4.2 이용도 계산

임의 흐름의 규격  $F(r, b, M)$ 에 대해 살펴보자. 흐름의 최대 패킷 크기인 M의 경우 대표적인 QoS 트래픽인 H.323에서 음성, 오디오 및 비디오의 패킷크기는 대부분 2Kbit 전후로 알려져 있다[12]. 따라서 본 연구에서는 복잡성을 피하기 위해 M은 상수로 가정하고, 요구속도를  $[r^{max}, r^{min}]$  사이에 분포하는 랜덤변수 R, 지연규격을  $[b^{min}, b^{max}]$  사이에 분포하는 랜덤변수 B로 가정한다. 그러면 흐름의 규격은 R과 B의 랜덤벡터로 표현된다.

먼저 WFQ의 대역폭 이용도를 구해보자. 랜덤 흐름의 요구속도의 기대값  $E[R]$ 은 다음과 같이 표현된다.

$$E[R] = \int_{r^{min}}^{r^{max}} \int_{b^{min}}^{b^{max}} r f_{R,B}(r, b) db dr \quad (4.3)$$

(2.1)로부터 지연규격 B를 만족시키기 위해 요구되는 임계속도는 다음과 같이 랜덤변수  $R^C$ 가 된다.

$$R^C = M / (B - \frac{M^{max}}{C^B}) \quad (4.4)$$

지연규격 B와 요구속도 R을 동시에 만족시켜야 하므로 WFQ하에서 스케줄링 속도는 다음과 같이 랜덤변수 S가 된다.

$$S = \max(R, R^C) = \max(R, M / (B - \frac{M^{max}}{C^B})) \quad (4.5)$$

따라서 WFQ하에서 랜덤 흐름의 스케줄링 속도의 기대값  $E[S]$ 은

$$E[S] = \int_{r^{min}}^{r^{max}} \int_{b^{min}}^{b^{max}} r f_{R,B}(r, b) db dr = \int_{r^{min}}^{r^{max}} \int_{b^{min}}^{b^{max}} \max(r, \frac{M}{b - \frac{M^{max}}{C^B}}) f_{R,B}(r, b) db dr \quad (4.6)$$

[13]로부터 WFQ의 대역폭 이용도는  $E[R]/E[S]$ 가 된다.

이제 LOFQ 스케줄러의 대역폭 이용도를 구해보자. 3장에서 살펴본 바와 같이 LOFQ 스케줄러는 각각 B2D 및 D2B 변환효율이 가장 큰 흐름인 흐름 i와 j에 대해  $E_i^{B2B} = E_i^{B2D} * E_j^{D2B} > 1$  인 한 자원 최적화 기능, 즉 흐름 i에 대한 B2D 변환과 흐름 j에 대한 D2B 변환을 수행하여 가용자원을 늘리고, 가용자원이 있는 한 흐름 수락제어 기능에 의해 새로운 흐름을 수락한다. 보조정리 1로부터 임의의 흐름의  $E^{B2D}$  와  $E^{D2B}$ 는 일정한 값을 갖고  $E^{B2D} = 1/E^{D2B}$ 의 관계를 가지므로 복잡한 LOFQ 스케줄러의 동작은 다음과 같이 간단히 모형화 될 수 있다: 첫째 자원 최적화 기능에 의해  $E^{D2B}$ 가  $E^{D2B,C}$ 로 표기되는 임계 D2B 변환 효율보다 큰 흐름은 모두 D2B 완전변환 되고, 그 외 B2D 변환 가능 흐름은 모두 B2D 완전변환 된다. 둘째 이러한 한 D2B 및 B2D 완전변환은 스케줄러의 가용 자원이 모두 고갈될 때까지 계속된다.

편의상 D2B 완전변환 되는 흐름을 그룹 A, B2D 완전변환 되는 흐름을 그룹 B, 기타 흐름을 그룹 C로 분류한다. 그러면 그룹 C는 D2B 변환 불가능이면서 B2D 변환 불가능인 흐름, 그리고  $E^{D2B} \leq E^{D2B,C}$ 인  $E^{D2B}$ 를 갖는 D2B 변환 가능이면서 B2D 변환 불가능인 흐름의 두 가지 유형으로 구성되는데, 전자는 추론 1로부터 모두 0 또는 1보다 큰 레이턴시 지수를 갖고, 후자는 추론 2로부터 모두 1의 레이턴시 지수를 갖는다.

스케줄러에 도착하는 흐름은 i.i.d(independent and identically distributed) 랜덤변수가 되고, 최대의 흐름이 수용될 경우 각 흐름의 평균 점유 DW는  $D^B/N^{LOFQ}$ , 여기서  $N^{LOFQ}$ 는 최대 수용 흐름 수, 가 된다. 따라서 스케줄러의 BW 용량인  $C^B$  값에 제약을 가하지 않을 경우 스케줄러의 점유 DW가  $D^B$ 의 값이 될 때까지 흐름이 수락되므로 다음과 같은 절차에 의해 LOFQ 스케줄러의 대역폭 이용도를 구할 수 있다.

- (단계 1) 랜덤 흐름의 점유 DW의 기대값,  $\bar{d}$ 로 표기함, 이  $D^B/N^{LOFQ}$ 가 되도록 하는  $E^{D2B,C}$  값을 구한다. 여기서  $N^{LOFQ}=C^B/E[H]$ 인데,  $E[H]$ 는 랜덤 흐름의 점유 BW의 기대값임,  $E[H]$ 를 알 수 없으므로  $N^{LOFQ}$ 의 초기값을  $C^B/E[R]$ 로 설정한다.
- (단계 2) 구해진  $E^{D2B,C}$  값을 이용하여  $E[H]$ 를 계산한다.
- (단계 3)  $N^{LOFQ}=C^B/E[H]$ 의 수식으로  $N^{LOFQ}$ 를 갱신한다.
- (단계 4)  $N^{LOFQ}$ 가 수렴할 때까지 위 3단계를 반복하고, 수렴하면  $E[R]/E[H]$ 로 대역폭 이용도를 구한다.

이제  $\bar{d}$ 와  $E[H]$ 만 구할 수 있으면 위의 절차에 의해 LOFQ의 대역폭 이용도를 계산할 수 있다. 먼저  $\bar{d}$ 를 구하는 방법을 살펴보자. 임의 그룹 B 흐름은 B2D 완전변환 되므로 보조정리 5로부터 점유 DW가 0이 되고,  $\beta \geq 1$ 인 레이턴시 지수를 갖는 임의 그룹 C 흐름도 보조정리 6으로부터 점유 DW가 0이 된다. 반면 임의 그룹 A 흐름과 0의 레이턴시 지수를 갖는 임의 그룹 C 흐름은 DW를 점유한다. 그런데 그룹 A 흐름은 D2B 완전변환 되므로 보조정리 4로부터 다음과 같은 레이턴시 지수  $\beta$ 를 갖게 된다:  $\beta=(b-D^B)r/M \leq 1$ . 따라서 그룹 A 흐름은 모두  $r \leq r^A \equiv M/(b-D^B)$ 인 요구속도  $r$ 을 가져야 한다. 참고로  $r^A$ 는 보조정리 5에서 정의된  $h^1$ 과 같은 값이 된다. 또한 그룹 A 흐름은 모두  $E^{D2B,C}$ 보다 큰  $E^{D2B}$ 를 가지므로  $E^{D2B,C}=C^B/(b^C-D^B)$ 에 의해 결정되는  $b^C$  값보다 작은, 즉  $b < b^C=(D^B+C^B/E^{D2B,C})$ 인  $b$ 를 가져야 한다. 결과적으로 그룹 A 흐름은 모두  $b \in (b^{\min}, b^C)$ 인 지연규격  $b$ 와  $r \in [r^{\min}, \min(r^A, r^{\max})]$ 인 요구속도  $r$ 을 갖게 된다. 참고로  $r^A=r^{\max}$  일때  $b$ 를  $b^T$ , 즉  $b^T \equiv M/r^{\max}+D^B$ 라 하면  $b < b^T$ 인 지연규격  $b$ 를 갖는 그룹 A 흐름의 요구속도는  $[r^{\min}, r^{\max}]$ 내 임의의 값을 가질 수 있다.

임의 그룹 A 흐름이 점유하게 되는 DW는 보조정리 4로부터  $(M-(b-D^B)R)/C^B$ 로 표현되는 랜덤변수가 되는 반면  $\beta=0$ 인 임의 그룹 C 흐름의 점유 DW는 보조정리 6으로부터  $M/C^B$ 가 된다. 따라서  $\bar{d}$ 는 다음과 같이 전개된다.

$$\begin{aligned} \bar{d} &= \int_{r^{\min}}^{\min(r^{\max}, r^A)} \int_{b^{\min}}^{b^C} (M-(b-D^B)r)f_{R,B}(r,b)dbdr + O(b^{\min}) \\ &= \begin{cases} K + \int_{r^{\min}}^{r^A} \int_{b^T}^{b^C} (M-(b-D^B)r)f_{R,B}(r,b)dbdr + O(b^{\min}), & \text{for } K > D^B/N^{LOFQ} \\ \int_{r^{\min}}^{r^{\max}} \int_{b^{\min}}^{b^T} (M-(b-D^B)r)f_{R,B}(r,b)dbdr + O(b^{\min}), & \text{for } K \leq D^B/N^{LOFQ} \end{cases} \\ &, \text{ where } K \equiv \int_{r^{\min}}^{r^{\max}} \int_{b^{\min}}^{b^T} (M-(b-D^B)r)f_{R,B}(r,b)dbdr \text{ 및 } O(b^{\min}) \equiv \int_{r^{\min}}^{r^{\max}} \frac{M}{C^B} f_{R,B}(r, b^{\min})dr. \end{aligned} \tag{4.7}$$

$E[H]$ 를 구하기에 앞서 임의 흐름의 지연규격  $b$ 와 스케줄링 속도  $h$ 를 분석해보자. 그룹 A 흐름은 위에서 살펴본 바와 같이  $b \in (b^{\min}, b^C)$ 인 지연규격  $b$ 를 갖고 D2B 완전변환 되므로 보조정리 4로부터  $h$ 의 BW를 점유하게 된다. 반면 그룹 B 흐름은  $E^{D2B,C}$ 보다 작거나 같은  $E^{D2B}$ 를 가지므로  $b \in [b^C, b^{\max}]$ 인 지연규격  $b$ 를 가지며, B2D 완전변환 되므로 보조정리 5로부터  $h^1=M/(b-D^B)$ 의 BW를 점유하게 된다. 한편 그룹 C 흐름은 보조정리 6으로부터 모두  $h$ 의 BW를 점유한다. 따라서  $E[H]$ 는 다음과 같이 전개된다.

$$\begin{aligned} E[H] &= \int_{r^{\min}}^{r^{\max}} \int_{b^{\min}}^{b^{\max}} h f_{R,B}(r,b)dbdr \\ &= \int_{b^{\min}}^{b^C} \int_{r^{\min}}^{r^{\max}} r f_{R,B}(r,b)drdb + \int_{b^C}^{b^{\max}} \int_{r^{\min}}^{r^{\max}} \max(r, \frac{M}{b-D^B}) f_{R,B}(r,b)drdb. \end{aligned} \tag{4.8}$$

한편  $D^B$ 가 줄어들어  $M^{\max}/C^B$ 로 수렴하면 (4.8)의  $E[H]$  역시 줄어들어 WFQ에서의 랜덤 흐름의 스케줄링 속도의 기대값  $E[S]$ 에 수렴하게 되는데, 이는 (4.7)에서  $b^C$ 가  $b^{\min}$ 로 수렴하기 때문이다. 이는 WFQ가 바로  $M^{\max}/C^B$ 의 지연자원 용량을 갖는 LOFQ로 간주될 수 있음을 의미한다.

### 5. 대역폭 이용도 확인

수치해석과 시뮬레이션을 통해 4장에서 제시한 대역폭 이용도 분석 방법을 검증한다. 먼저 흐름의 규격  $F(r,b,M)$ 에서 요구속도  $r$ 과 지연규격  $b$ 의 분포를 고찰해보자. 인터넷에서 유통되는 미디어는 8Kbps급의 저속 음성부터 1Mbps급의 고속 영상까지 다양하므로 [8,1000]Kbps내 균등하게 분포하는  $r$ 을 고려한다. 널리 이용되는 VoIP의 경우 이용자의 만족을 위해서 단 방향 종단간 지연시간이 300ms 이내가 되어야 하고[14], 대부분의 경우 종단간 라우터의 수가 20개를 넘지 않을 것이므로 각 라우터에 할당되는 지연규격  $b$ 는 직관적으로 [1,11]ms내 균등 분포하는 것으로 가정한다. 그러면 종단간 총 큐잉 지연은 약 [20,220]ms내에 분포하므로 종단간 전송 선로상의 총 지연은 약 [80, 280]ms내에 분포하게 된다. 그리고 모든 흐름의 패킷 크기는 모두  $M$  bit,  $G=10Gbps$ ,  $D^B=1ms$ 로 가정한다. 시뮬레이션은 SMPL(Simulation Model Programming Language)[15]에 공평 패킷 스케줄러 루틴을 추가하여 수행한다.

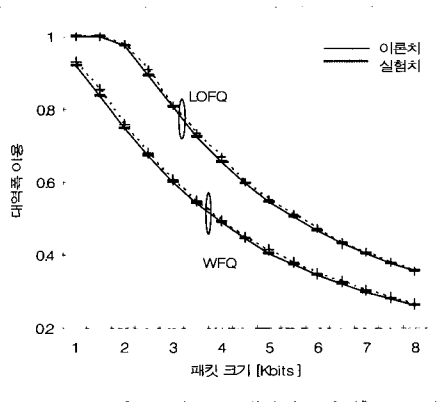


그림 3. 대역폭 이용도

패킷 크기 변화에 따른 LOFQ와 WFQ의 대역폭 이용도가 그림 3에 도시되어 있다. 이 그림으로부터 이론치와 시뮬레이션(실험치) 결과가 2% 오차범위 내에서 일치함을 확인할 수 있다.

### 6. 결 론

널리 이용되고 있는 공정 큐잉 방식인 WFQ(Weighted Fair Queuing)의 대역폭 저하 문제를 해결한 LOFQ 방식이 출현하였고, 일부 기능이 개선되었다. 하지만 여전히 시뮬레이션이 의해 성능특성을 평가하고 있어 평가결과의 정확성과 소요시간 측면에서 어려움이 있어왔다. 본 연구에서는 이러한 문제를 해결하기 위해 LOFQ의 대역폭 이용도를 수학적 방법으로 구하는 방법을 제시하였다.

먼저 LOFQ의 수락제어와 점유자원 최적화 알고리즘을 다음과 같이 모델링 하였다: 각 흐름의 프로파일이 어떻게 변화하던 상관없이 D2B 변환 효율이 어떤 임계값 보다 큰 흐름은 모두 D2B 완전변환 되고, 그 외 B2D 변환 가능 흐름은 모두 B2D 완전변환 되는데, 스케줄러의 가용 BW와 DW가 고갈될 때까지 계속된다. 그리고 랜덤 흐름의 점유 지연 자원의 기대값이 총 지연자원을 최대 흐름 수로 나눈 값과 일치하는 그러한 임계값을 구한 후 랜덤 흐름의 스케줄링 속도의 기대값을 계산하는데, 계산된 기대값으로 최대 흐름 수를 갱신하는 과정을 반복하여 정확한 스케줄링 속도의 기대값을 구할 수 있고 이로부터 대역폭 이용도를 구할 수 있었다.

균등분포의 요구속도와 균등분포의 지연규격 트래픽 환경 하에서 시뮬레이션 결과와 수학적 계산결과를 비교한 결과 2% 이내에서 서로 일치하였는데, 이로부터 시뮬레이션과 수학적 계산과정이 모두 정확함을 검증할 수 있었다.

### 참 고 문 헌

[1] X. Xiao and L. M. Ni, "Internet QoS: A Big Picture," IEEE Network, Vol.13, No.2, pp.8~18, 1999.  
 [2] A.K. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks," PhD

dissertation, Massachusetts Institute of Technology, Feb., 1992.  
 [3] A. Demers, S. Keshav, and S. Shenker, "Design and analysis of a fair queuing algorithm," Proc. ACM SIGCOMM'89, pp.1~12, 1989.  
 [4] S. Keshav, "An Engineering Approach to Computer Networking," Addison Wesley professional computing series, 1997.  
 [5] S. Shenker, C. Partridge, and R. Guerin, Specification of Guaranteed Quality of Service, RFC 2212, IETF, 1997.  
 [6] M. Baldi and F. Risso, "Efficiency of Packet Voice with Deterministic Delay," IEEE Comm. Mag, 2000. pp.170~177  
 [7] A. Francini and F.M.Chiussi, "A Weighted Fair Queuing Scheduler with Decoupled Bandwidth and Delay Guarantees for the Support of Voice Traffic," Proc. GLOBECOM, Vol. 3, 2001.  
 [8] Do Sung Jun, Jinwoo Choe and Alberto Leon Garcia, "Credit based Processor Sharing for Decoupled Delay and Bandwidth Allocation", IEEE Comm. Letters, Vol.5, No. 4, pp.178~180, 2001.  
 [9] Jens Schmitt. "Optimal Network Service Curves under Bandwidth Delay Decoupling," IEE Electronics Letters, Vol.38, No.6, pp.297~299, 2002.  
 [10] 김태준, 김황래, "공평 패킷 스케줄러의 대역폭 이용 효율 개선에 관한 연구", 정보처리학회논문지 제 13권 C 제3호, pp.331~338, 2006.  
 [11] 김태준, "최적 레이턴시 기반 공정 큐잉 알고리즘 개선에 관한 연구", 멀티미디어학회 논문지, 제 10권 1호. pp.83~93, 2007  
 [12] Hani ElGebaly, "Characterization of Multimedia Streams of an H.323 Terminal," Intel Technology Journal, Vol. 2 Issue 2, 1998  
 [13] T. J. Kim, "Bandwidth Utilization in Sorted Priority Schedulers," LNCS, Vol.3936, 2006, pp.1080~1085  
 [14] ITU T Recommendation G.114: One way transmission time, 2003.  
 [15] M. H. MacDougall, Simulating Computer Systems, Techniques and Tools, MIT Press, 1987.



김 태 준

e-mail : tjkim@kongju.ac.kr

1980년 2월 경북대학교 전자공학과 졸업  
 1982년 2월 한국과학기술원 전자공학 석사  
 1999년 8월 한국과학기술원 전자공학 박사  
 1982년 1996년 한국전자통신연구원  
 1996년~2005년 천안공업대학 교수

2005년 3월~현재 공주대학교 정보통신공학부 부교수  
 관심분야: 고속통신망, VoIP, 트래픽 품질 제어