

공평 패킷 스케줄러의 대역폭 이용 효율 개선에 관한 연구

김 태 준[†] · 김 황 래^{††}

요 약

실시간 멀티미디어 응용의 서비스 품질을 지원하는 공평 패킷 스케줄러는 대부분 패킷의 예상되는 전송 종료시점을 그 패킷의 타임스탬프로 사용하는 종료시각 방식을 채택하고 있다. 이 방식은 예약속도로 레이턴시를 통제할 수 있는 장점을 가지나 속도-지연 결합 특성으로 인해 스케줄러 대역폭의 이용 효율이 저하되는 문제점이 있다. 본 연구에서는 지연 자원 개념을 도입하고, 속도-지연 결합 특성에 의해 손실되는 지연 자원을 속도 자원으로 변환하여 스케줄러 대역폭의 이용 효율을 높이는 스케줄링 방식을 제안한다. 성능 평가 결과 기존 방식보다 50% 이상 우수한 대역폭 이용도 특성을 보였다.

키워드 : 패킷 스케줄러, 공정 큐잉, 지연 자원, 레이턴시, 이용도

A study on improving the bandwidth utilization of fair packet schedulers

Tae-Joon Kim[†] · Hwang-Rae Kim^{††}

ABSTRACT

Most fair packet schedulers supporting quality-of-services of real-time multimedia applications are based on the finish time design scheme in which the expected transmission finish time of each packet is used as its timestamp. This scheme can adjust the latency of a flow with raising the flow's scheduling rate but it may suffer from severe bandwidth loss due to the coupled rate and delay allocation. This paper first introduces the concept of delay resource, and then proposes a scheduling method to improve the bandwidth utilization in which delay resource being lost due to the coupled allocation is transformed into bandwidth one. The performance evaluation shows that the proposed method gives higher bandwidth utilization by up to 50%.

Key Words : Packet Scheduler, Fair Queuing, Delay Resource, Latency, Utilization

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]를 구현하는 두 가지 접근방식이 있는데, 하나는 WFQ(Weighted Fair Queuing)[3]와 SFQ(Starting-time Fair Queuing)[4]에 적용된 타임스탬프 방식

이고, 다른 하나는 WRR(Weighted Round Robin)[5]에 적용된 라운드로빈(round-robin) 방식이다. 타임스탬프 방식은 구현이 복잡한 반면 라운드로빈 방식은 레이턴시 특성이 나쁜 단점이 있다. 타임스탬프 방식은 또한 도착하는 패킷의 타임스탬프 계산시 사용되는 패킷의 기준시각 관점에서 예상되는 전송 시작시각을 사용하는 시작시각 방식과 예상되는 전송 종료시각을 사용하는 종료시각 방식으로 나뉜다. 본 논문에서는 편의상 전자를 ST(Start-time Type), 후자를 FT(Finish-time Type) 방식이라 부른다.

수락 흐름의 수에 비례하는 레이턴시 특성[4]을 갖는 ST 방식의 경우 흐름 수의 제약에 따른 자원손실이 발생하고, 특히 개별 흐름의 지연특성을 분리할 수 없어 그 용도가 극히 제한된다. 반면 흐름의 속도에 반비례하는 레이턴시 특성[3]을 갖는 FT 방식은 흐름의 예약속도를 높여 원하는 레이턴시 값을 얻을 수 있어서 IETF IntServ 모델의 RSVP-라우터에 적용되었다[6]. 그러나 예약속도를 높일 경우 예약속도와 요구속도와의 차이, 즉 과잉 예약분 만큼 속도(대역폭)자원이 손실되는데, 특히 인터넷 전화와 같이 트래픽 흐

[†] 정 회 원 : 공주대학교 정보통신공학부 부교수

^{††} 종신회원 : 공주대학교 컴퓨터공학부 교수

논문접수: 2006년 2월 7일, 심사완료: 2006년 4월 5일

름의 속도는 낮지만 엄격한 종단간 지연특성을 요구하는 트래픽 흐름의 경우 과다한 과잉예약으로 인해 자원손실이 심각해진다[7, 8]. 이러한 문제점을 해결하기 위해 크레딧(credit) 방식[7]과 가변 예약속도 방식[8]이 연구되었다. 이들은 모두 어떤 흐름의 대역폭이 부족한 경우 다른 흐름이 사용할 대역폭을 잠시 빌리는 방식으로 대역폭의 여유가 없는 고정 속도 트래픽의 경우 효과를 기대할 수 없다. 본 연구에서는 지연자원 개념을 도입한 후 대역폭 자원과 지연 자원의 상호 자원변환을 통해 더 많은 흐름의 수락을 시도하는 새로운 접근 방식을 제안하고 성능특성을 분석 및 평가한다.

본 논문의 구성은 다음과 같다: 2장에서 관련연구의 소개와 문제점을 분석하고, 3장에서 제안된 방식을 기술하며, 4장에서 레이턴시 특성을 분석한다. 그리고 5장에서 성능을 평가하고 6장에서 결론을 맺는다.

2. 관련 연구 및 문제점 분석

FT 방식은 WFQ에 처음 적용되었다. WFQ는 FT 방식의 이상적인 레이턴시 특성을 제공하나 타임스탬프 계산이 너무 복잡한데, SCFQ(Self-Clocked Fair Queuing)[9]에서 자가클럭(self-clock)을 사용하여 이 문제를 해결하였다. 하지만 레이턴시 특성이 저하되는 또 다른 문제가 발생하였다. PRPS(Packet-by-packet Rate Proportional Servers)[10]에서 자가클럭 대신서버 가상시간을 도입하여 두 가지 문제점을 모두 해결할 수 있는 접근방식을 제안하였다. SPFQ(Starting Potential Fair Queuing)[11], MSPFQ(Medium Starting Potential Fair Queuing)[12] 및 NSPFQ(New Starting Potential Fair Queuing)[13] 등 최근까지 PRPS기반의 다양한 패킷 스케줄러가 개발되었는데, 이들은 모두 서버 가상시간의 운영과 유지방식을 개선하는데 주안점을 두었다. 한편 ST 방식은 SFQ(Starting-time Fair Queuing) 및 QLR(Queuing Latency Rate)[14]에 적용되었다. SFQ는 SCFQ와 마찬가지로 자가클럭을 사용하나 QLR은 자가클럭 대신 PRPS에서 제안된 서버가상시간을 사용한다.

FT와 ST 방식의 레이턴시 특성을 상세히 살펴보자. 임의 흐름 i 의 예약속도를 R_i , 흐름 i 의 최대 패킷크기를 M_i , 모든 흐름의 최대 패킷크기를 M_{max} , 그리고 출력링크의 전송용량, 즉 스케줄러 대역폭을 G 로 표기한다. 참고로 흐름의 예약속도는 스케줄러가 흐름을 위해 투입하는 속도로 할당 속도 또는 스케줄링 속도라고도 하는데, 흐름 i 의 요구속도 r_i 를 보장해야 하므로 $R_i \geq r_i$ 이다. 참고로 본 논문에서 사용되는 두 가지 용어인 속도와 대역폭은 동일한 의미를 가지며 모두 bps(bit per second) 단위를 갖는다. PRPS에서 체계화된 FT 방식의 타임스탬프 계산방법을 살펴보면 다음과 같다: 시간 t 에 도착하는 흐름 i 의 k 번째 패킷 P_i^k 의 타임스탬프 T_i^k 는 P_i^k 의 가상 종료시각 F_i^k 와 같으며, 이는 $(\max(F_i^{k-1}, v(t)) + l_i^k / R_i)$ 로 계산된다. 여기서 F_i^{k-1} 는 흐름 i 의 $(k-1)$ 번째 패킷의 가상 종료시각이고, $v(t)$ 는 서버 가상시간이며, l_i^k 는 P_i^k 의 크기이다. 서버 가상시간이란 스케줄러의 활성구

간 동안, 즉 스케줄러가 지속적으로 패킷을 전송하는 구간의 시작시점부터 현재까지 전송된 전체 트래픽량을 나타내는 시간의 함수를 의미한다. FT 방식의 스케줄러에서 흐름 i 의 레이턴시 Q_i 는 다음과 같이 계산된다[10].

$$Q_i = \frac{M_i}{R_i} + \frac{M_{max}}{G} \quad (2.1)$$

참고로 임의 흐름 i 의 레이턴시는 스케줄러에서 그 흐름의 임의 패킷의 전송완료 시각을 T_s 라 하고, 레이턴시가 없는 대응 GPS 스케줄러에서 그 패킷의 전송완료 시각을 T_g 라 할 때 $(T_s - T_g)$ 의 최대값으로 정의되는 특수한 지연인데, 일반적으로 흐름내 패킷이 스케줄러를 통과할 때 발생할 수 있는 최대 지연의 의미로 해석된다.

ST 방식의 타임스탬프 계산방법[4]을 살펴보면 다음과 같다: P_i^k 의 타임스탬프 T_i^k 는 P_i^k 의 가상 시작시각 S_i^k 와 같으며, 이는 $\max(F_i^{k-1}, v(t))$ 로 계산된다. 여기서 $F_i^{k-1} = S_i^{k-1} + l_i^{k-1} / R_i$ 이다. 이러한 ST 방식의 스케줄러인 SPQ와 QLR에서 모든 흐름의 레이턴시 Q 는 다음과 같이 계산된다[4, 14].

$$Q = \frac{1}{G} \sum_{i=1}^N M_i \quad (2.2)$$

여기서 N 은 스케줄러에 수용되는 최대 흐름 수를 의미한다.

IETF IntServ 모델의 RSVP-라우터는 PATH 메시지로 통보되고 흐름의 요구속도를 의미하는 T_{spec} 내 토큰속도 r 을 보장하고, 라우터에서 발생하는 흐름의 레이턴시가 RESV 메시지로 통보된 R_{spec} 내 예약속도 R 과 탑재된 스케줄러의 C 와 D 값에 의해 결정되는 지연배분값(delay budget)(이하 지연규격 B)을 초과하지 않도록 패킷을 스케줄링을 해야 한다. 여기서 C 와 D 값은 대응 GPS 스케줄러에서 레이턴시와의 최대 오차를 결정하는 변수로 RFC 2212 문서[6]에 정의되어 있다. 흐름이 RSVP-라우터에 요구하는 서비스 요구조건, 즉 보장해야 할 속도 r 과 준수해야 할 지연규격 B 를 본 논문에서는 스케줄링 규격 $S(r, B)$ 로 표기한다.

먼저 ST 방식의 문제점을 분석해보자. $S(r, B)$ 의 스케줄링 규격을 갖는 임의 흐름의 수락 요청시 $B < Q$, 즉 결정된 레이턴시가 지연규격을 충족하지 못할 경우 (2.2)에 의해 흐름 수 N 을 줄여야 하는데, 이로 인해 대역폭 자원이 손실된다. 그리고 모든 흐름이 (2.2)로 결정되는 동일한 레이턴시 값을 가지므로 개별 흐름의 지연특성을 분리할 수 없는 문제가 있다. 다음 FT 방식의 경우를 살펴보자. (2.1)에 의해 결정된 레이턴시 Q 와 지연규격 B 와의 관계는 $Q=B$, $Q>B$ 및 $Q<B$ 의 세 가지 경우가 가능하다. $Q=B$ 의 경우 대역폭과 지연 자원의 손실 없이 흐름을 수용할 수 있다. $Q>B$ 의 경우 여유 대역폭 자원이 있다면 예약속도를 높여서 지연규격을 충족, 즉 $Q=B$ 가 되게 해야 한다. 이때 예약속도와 요구속도의 차이분, 즉 과잉예약분 만큼 대역폭 자원이 낭비되지만 지연 품질을 만족시키기 위해 피할 수 없다. $Q<B$ 의 경우 $Q=B$ 가 되게 하려면 예약속도를 요구속도 보다 낮게 해야

한다. 하지만 흐름의 요구속도를 보장해야 하기 때문에 예약속도를 낮출 수 없어 지연규격 보다 더 엄격한 레이턴시를 갖게 된다. 이 결과 (B-Q) 만큼의 지연 자원이 손실되어 스케줄러 자원 이용 효율이 저하된다.

3. 제안방식

3.1 방식개요

본 논문에서 제안하는 공정패킷 스케줄링 알고리즘을 LOFQ (Latency-Optimized Fair Queuing)라 부르자. LOFQ는 임의 흐름 i 의 레이턴시를 결정하는 레이턴시 지수(Index) $\beta_i(\geq 0)$ 와 수락된 흐름의 수에 비례하고 흐름의 레이턴시 지수에 반비례하는 특성을 갖는 레이턴시 바이어스(Bias) $\Theta(\geq 0)$ 를 도입한다. 레이턴시 바이어스는 스케줄러가 지원하는 가장 엄격한 지연규격 B_{min} 을 초과할 수 없는, 즉 레이턴시 바이어스의 상한값(upper bound) Θ_{UB} 를 B_{min} 로 설정한다. 그러면 4장의 정리 2에 의해 흐름 i 의 레이턴시 Q_i 는 (3.1)과 같이 그 흐름의 레이턴시 지수의 함수로 표현되는데, 이를 그 흐름의 지연규격 B_i 와 일치하도록 한다.

$$Q_i = \Theta_{UB} + \beta_i \frac{M_i}{R_i} = B_i \quad (3.1)$$

그러면 LOFQ 스케줄러는 새로운 흐름 i 를 수락할 때 (3.1)에 의해 그 흐름의 β_i 값을 결정하고, 그 흐름 패킷의 타임스탬프 계산시 결정된 β_i 값을 적용하여 흐름의 지연규격과 동일한 레이턴시를 갖도록 하는, 즉 흐름의 레이턴시를 최적화 함으로서 지연 자원의 손실을 방지할 수 있다. 레이턴시 바이어스 Θ 는 4장의 정리 1에 의해 (3.2)와 같이 주어지며 M_{max}/G 의 최소값을 갖는다.

$$\Theta = \frac{\sum_{k \in B(t)} \max(0, 1 - \beta_k) M_k}{G} \quad (3.2)$$

참고로 모든 흐름의 레이턴시 지수가 1로 접근하면 (3.2)의 레이턴시 바이어스는 M_{max}/G 의 최소값으로 수렴하고, (3.1)의 레이턴시는 (2.1)에 수렴하므로 LOFQ는 바로 FT 방식이 되고. 반면에 모든 흐름의 레이턴시 지수가 0으로 접근하면 (3.2)와 (3.1)로부터 레이턴시는 (2.2)에 수렴하므로 LOFQ는 바로 ST 방식이 된다.

$S(r, B)$ 를 요구하는 임의 흐름을 수락하기 위해 필요한 자원 확보에 대해 살펴보자. 먼저 r 을 수용할 수 있는 가용 대역폭이 있어야 한다. 그리고 $Q=B$ 가 되게 하는 레이턴시 지수 β 를 확보할 수 있어야 한다. 1보다 작은 β 는 레이턴시 바이어스 Θ 를 증가시키는데(보조정리 2), 증가된 바이어스가 Θ_{UB} 를 초과할 경우 요구되는 β 를 확보할 수 없다. 따라서 레이턴시 바이어스를 지연규격 보장 측면에서 흐름의 수락에 필요한 자원으로 간주할 수 있는데, 본 논문에서는 이를 지연 자원이라 부른다. 현재의 레이턴시 바이어스는 바로 현

재 사용된 지연 자원 량을 의미하며, 지연 자원의 총 사용량은 Θ_{UB} 를 초과할 수 없다. 따라서 지연 자원은 총 사용량이 G 로 제한되는 대역폭 자원과 마찬가지로 방법으로 취급될 수 있다.

가용 대역폭 자원과 가용 지연 자원의 상호 관계를 살펴보자. 기 수락된 흐름의 예약속도를 높이는, 즉 가용 대역폭 자원을 희생할 경우 그 흐름의 레이턴시가 변경되지 않도록 하기 위해 (3.1)로부터 그 흐름의 레이턴시 지수를 증가시켜야 함을 알 수 있다. 레이턴시 지수가 증가하면 레이턴시 바이어스가 감소하므로 가용 지연 자원이 늘어나게 된다. 마찬가지로 가용 지연 자원을 희생할 경우 가용 대역폭 자원도 늘어날 수 있다. 이와 같이 대역폭과 지연 자원은 상호 변환이 가능한데, 이러한 성질을 이용하여 부족한 가용 자원을 보충할 수 있다.

3.2 LOFQ 스케줄러

LOFQ를 사용하는 스케줄러(이하 LOFQ 스케줄러)는 흐름의 수락 요청을 받아 수락여부를 판단하고 수락시 자원을 할당하는 흐름 수락제어와 수락된 흐름들의 트래픽을 그들의 스케줄링 규격에 맞게 서비스하는 패킷 스케줄링의 두 가지 기능을 갖는다. 수락된 임의 흐름 i 는 흐름 프로파일 $A_i(r, B, R, \beta, M)$ 를 갖는데, 여기서 r 과 B 는 스케줄링 규격이고, R 은 예약속도, β 는 레이턴시 지수, 그리고 M 은 최대 패킷 크기이다. 참고로 $A_i()$ 내 임의 요소 X 를 X_i 로 표기한다. 그리고 총 예약 대역폭 값을 갖는 R_{sum} 레지스터, 현재의 레이턴시 바이어스 값을 갖는 Θ 레지스터, 그리고 임시 흐름 프로파일 $A_{temp}()$ 버퍼가 사용된다.

3.2.1 흐름 수락 제어

$S(r, B)$ 를 요구하는 임의 흐름 i 에 대한 흐름 수락제어 알고리즘이 (그림 1)에 도시되어 있다. 먼저 흐름 i 의 레이턴시 지수를 구한 후 다음과 같이 흐름의 수용에 필요한 가용 자원이 있는지를 조사한다: 총 예약 대역폭에 흐름 i 의 요구속도를 더한 값 ($R_{sum} + r$) (이하 소요 대역폭) 와 G 를 비교하고, 레이턴시 바이어스에 흐름 i 에 의해 증가되는 바이어스를 더한 값 ($\Theta + \max(0, 1 - \beta_i) M_i / G$) (이하 소요 레이턴시 바이어스) 와 Θ_{UB} 를 비교하여 가용 자원이 충분한지를 조사한다. 가용 자원이 모두 부족하면 흐름의 수락을 거절하고, 모두 충분하면 흐름을 수락한다. 둘 중 어느 한쪽은 흐름을 수용하고도 남는 가용자원이 있지만 다른 쪽은 부족한 경우 남는 가용 자원을 부족한 자원으로 변환한 후 흐름의 수락을 재 시도한다.

가용 대역폭 자원에는 여유가 있지만 가용 지연 자원이 부족한 경우를 살펴보자. $0 < \beta < 1$ 의 레이턴시 지수를 갖는 기 수락 흐름에 대해 그의 예약속도를 ΔR 만큼 높이면 레이턴시 불변원칙에 의해 그의 레이턴시 지수를 $\Delta \beta$ 만큼 증가시켜야 하는데, $\Delta \beta = (\beta/R) \Delta R$ 가 되므로 이로 인해 감소하는 레이턴시 바이어스의 량 $\Delta \Theta$ 는 (3.2)에 의해 다음과 같이 계산된다.

$$\Delta \Theta = \frac{M}{G} \Delta \beta = \frac{1}{G} \frac{\beta M}{R} \Delta R \quad (3.3)$$

```

 $\beta_i = (B_i - \Theta_{UB})r_i / M_i$  // 레이턴시 지수 계산
if( $R_{sum} + r_i > G$  and  $\Theta + \max(0, 1 - \beta_i)M_i / G > \Theta_{UB}$ ) reject
else if( $\Theta + \max(0, 1 - \beta_i)M_i / G > \Theta_{UB}$ ) { // 가용 지연자원 확충
    while( $\Theta + \max(0, 1 - \beta_i)M_i / G > \Theta_{UB}$ ) {
         $j = \{k \mid \max_{k \in B(t) \text{ and } 0 < \beta_k < 1} (\beta_k M_k / R_k)\}$ 
         $\Delta R_j = \min(1 - \beta_j, (\beta_j / R_j)\Delta R_j)R_j / \beta_j$  //예약속도 증가분 산출
         $\Delta \beta_j = (\beta_j / R_j)\Delta R_j$  //레이턴시 지수 증가분 계산
        if( $R_{sum} + r_i + \Delta R_j > G$ ) reject
         $A_j(r, B, R, \beta, M) = A_j(r, B, R + \Delta R_j, \beta + \Delta \beta_j, M)$  //프로파일 갱신
         $R_{sum} += \Delta R_j, \Theta -= \Delta \beta_j M_j / G$ 
    }
} else if( $R_{sum} + r_i > G$ ) //가용 대역폭자원 확충
    while( $R_{sum} + r_i > G$ ) {
         $j = \{k \mid \min_{k \in B(t), r_k < R_k \text{ and } 0 < \beta_k \leq 1} (\beta_k M_k / R_k)\}$ 
         $\Delta \beta_j = \min(\beta_j, \min(R_j - r_j, (R_j / \beta_j)\Delta \beta_j)\beta_j / R_j)$  //레이턴시 지수 감소분 산출
         $\Delta R_j = (R_j / \beta_j)\Delta \beta_j$  //예약속도 감소분 계산
        if( $\Theta + \max(0, 1 - \beta_i)M_i / G + \Delta \beta_j M_j / G > \Theta_{UB}$ ) reject
         $A_j(r, B, R, \beta, M) = A_j(r, B, R - \Delta R, \beta - \Delta \beta, M)$  //프로파일 갱신
         $R_{sum} -= \Delta R_j, \Theta += \Delta \beta_j M_j / G$ 
    }
}
 $A_i(r, B, R, \beta, M) = A_{temp}(r_i, B_i, r_i, \beta_i, M_i)$  //흐름 프로파일 등록
 $R_{sum} += r_i$ 
 $\Theta += \max(0, 1 - \beta_i)M_i / G$ 

```

(그림 1) 흐름 수락제어 알고리즘

여기서 M과 R은 레이턴시 지수가 증가되는 기 수락 흐름의 최대 패킷크기와 스케줄링 속도이다.

가용 대역폭 자원의 희생 대비 확보되는 가용 지연 자원의 극대화, 즉 자원변환 효율의 극대화를 위해 $\beta M/R$ 값이 큰 흐름 순으로 자원을 변환하는데, 부족한 가용 지연 자원이 확보될 때까지 반복한다. 이때 증가시키는 ΔR 의 양은 $(\beta + \Delta \beta) \leq 1$ 의 조건을 위반하지 않는 그러한 양이다. 가용 지연 자원의 확보과정 중 총 예약 대역폭, 즉 R_{sum} 이 G를 초과하면 수락을 거절한다. 다음 가용 지연 자원에는 여유가 있지만 가용 대역폭 자원이 부족한 경우를 살펴보자. 위의 역과정을 통해 가용 대역폭 자원을 확보할 수 있는데, 구체적으로 살펴보면 다음과 같다: $R > r$ 이고 $0 < \beta \leq 1$ 의 레이턴시 지수를 갖는 기 수락 흐름을 대상으로 자원변환 효율을 극대화 하기 위해 $\beta M/R$ 값이 작은 흐름 순으로 가용 지연 자원을 대역폭 자원으로 변환하여 부족한 가용 자원을 보충한다. 이때 줄이는 $\Delta \beta$ 의 양은 $(\beta - \Delta \beta) \geq 0$ 과 $(R - \Delta R) \geq r$ 의 조건을 위반하지 않는 그러한 양이 된다. 가용 대역폭 자원의 보충 과정 중 레이턴시 바이어스의 값이 Θ_{UB} 를 초과하면 수락을 거절한다. 참고로 레이턴시 지수가 1보다 큰 흐름은 자원 변환의 대상이 될 수 없는데, 이는 요구속도와 동일한 예약속도를 갖기 때문이다.

여유 대역폭 자원을 가용 지연 자원으로 변환하기 위해 높이는 예약속도 ΔR 과 여유 지연 자원을 가용 대역폭 자원으로 변환하기 위해 줄이는 레이턴시 지수 $\Delta \beta$ 의 값에 대해

살펴보자. 값이 적을수록 변환효율은 좋아지지만 흐름 수락 제어 알고리즘의 수행 복잡도가 높아지고, 반대로 값이 클수록 변환효율은 나빠지지만 수행 복잡도가 낮아진다. 흐름 수락제어 알고리즘은 새로운 흐름이 수락을 요청할 때만 동작하므로 매 패킷당 일련의 연산동작에 대한 반복 횟수로 산정하는 스케줄러의 복잡성에 비해 스케줄러에 미치는 영향은 미미하다.

(그림 1)에 도시된 알고리즘은 부족한 가용 자원의 보충 시 기 수락된 흐름만 대상으로 하였는데, 수락요청 흐름도 그 대상에 포함시킬 수 있다. 그리고 B(t)는 현재 수락된 흐름의 집합을 의미한다.

3.2.2 패킷 스케줄링

수락된 흐름들이 발생시키는 패킷을 입력 받아 출력링크로 전송하는 패킷 스케줄링 기능은 패킷이 도착할 때 동작하는 패킷 큐잉(queuing) 기능과 흐름들의 큐에 대기중인 패킷을 출력링크를 통해 전송하는 패킷 큐 서비스 기능으로 구성한다.

임의 흐름 i의 k번째 패킷 P^k의 도착시 큐잉 기능의 동작 절차는 다음과 같다:

- 1) 적용된 서버 가상시간 운영방식에 따라 서버 가상시간 v(t)를 계산하고 필요시 갱신한다.

WFQ, SPFQ, MSPFQ 및 NSPFQ 등에서 다양한 서버 가상시간 운영 방식이 개발되었는데, 이들을 포함하여 기타

다른 서버 가상시간 운영방식도 본 논문에서 제안한 LOFQ 스케줄러에 적용될 수 있다.

2) 다음 식을 사용하여 P_i^k 의 타임스탬프 T_i^k 를 계산한 후 P_i^k 에 부착하고, T_i^k 가 부착된 P_i^k 를 해당 흐름의 큐에 수록한다.

$$T_i^k = F_i^k = \max(v_i(t), v(t)) + \beta_i l_i^k / R_i \quad (3.4)$$

여기서 흐름 i 의 가상시간을 의미하는 $v_i(t)$ 는 F_i^{k-1} 의 값을 가지며 흐름 i 의 백로그시 0의 값으로 초기화된다.

패킷 큐 서비스 기능의 동작절차는 다음과 같다:

1) 수락된 흐름 큐의 선두 패킷 중 가장 앞선 타임스탬프를 갖는 패킷을 찾은 후 출력링크로 전송되도록 한다.

2) 적용된 서버 가상시간 운영방식에 따라 $v(t)$ 를 갱신한다.

패킷 스케줄링 기능의 구현 복잡도를 살펴보자. 서버 가상시간 운영 부담에 기인하는 패킷 큐잉기능의 복잡도와 수락된 흐름 큐의 선두 패킷 중 가장 앞선 타임스탬프를 갖는 패킷을 찾는 분류작업에 기인하는 패킷 큐 서비스 기능의 복잡도에 의해 결정된다. 전자는 가장 낮은 복잡도를 갖는 NSPFQ의 서버 가상시간 운영 방식을 사용하면 $O(1)$ 로 줄일 수 있다. 하지만 후자의 복잡도가 $O(\log N)$ 이기 때문에 패킷 스케줄링 기능의 복잡도는 $O(\log N)$ 으로 주어진다.

4. 레이턴시 분석

3장에서 살펴본 바와 같이 흐름의 레이턴시 지수가 모두 1이면 LOFQ는 바로 FT 방식이 되므로 FT 방식을 체계화한 PRPS의 레이턴시 특성[10]을 이용하여 LOFQ의 레이턴시를 분석한다. 먼저 역순서 바이어스와 간섭 바이어스라 불리는 두 가지 레이턴시 바이어스 개념을 도입한다: PRPS에서 어떤 패킷의 도착에 의해 백로그 되는 흐름의 경우 패킷 도착 시점부터 도착한 패킷의 타임스탬프를 계산하고 타임스탬프가 부착된 패킷을 그 흐름 큐의 선두에 수록할 때까지 아주 짧은 시간 구간 동안 도착한 패킷보다 늦은 타임스탬프 값을 갖는 다른 패킷이 먼저 전송되는 역순서 전송이 일어날 수 있고, 이로 인해 (2.1)의 두 번째 항, 즉 M_{\max}/G 만큼 레이턴시가 증가함이 증명되었다[10]. 이를 역순서 바이어스라 부른다. LOFQ에서 1보다 작은 레이턴시 지수를 갖는 흐름이 수락되면 그 흐름의 패킷은 (3.4)에 의해 RPRS의 경우 보다 앞선 타임스탬프 값을 가져 더 빨리 전송되므로 타 흐름의 패킷의 전송 시작 시점을 늦추는, 즉 타 흐름의 레이턴시에 어떤 바이어스를 가하는 결과를 가져온다. 이를 간섭 바이어스라 부른다.

[보조정리 1] LOFQ에서 도착 패킷에 의해 흐름이 백로그되고 그 흐름에 가해지는 간섭 바이어스가 M_{\max}/G 보다 작을 경우 역순서 전송이 일어날 수 있으며, 이때 발생하는 역순서 바이어스 Θ^{DB} 는 (4.1)과 같이 주어진다.

$$\Theta^{DB} = \frac{M_{\max}}{G} \quad (4.1)$$

[증명] 시간 t 시점에 도착한 임의 흐름 i 의 k 번째 패킷 P_i^k 에 의해 백로그되는 경우를 생각해보자. 도착한 P_i^k 가 흐름 i 의 큐의 선두에 위치하는 시각을 $(t+\Delta t)$ 라 하자. 시간구간 $(t, t+\Delta t)$ 동안 P_i^k 를 전송할 수 없기 때문에 역순서 전송이 일어날 수 있다. 스케줄러가 지원하는 최소 패킷크기 $M_{\min}(>G\Delta t)$ 에 대해 $\Delta t < M_{\min}/G$ 이므로 역순서로 전송될 수 있는 패킷은 하나만 존재한다. 따라서 역순서 바이어스 값은 최대크기의 패킷 하나를 전송하는 시간, 즉 M_{\max}/G 가 된다. P_i^k 도착 이전에 이미 흐름 i 가 백로그된 경우를 생각해보자. $\Delta t < M_{\min}/G \leq l_i^{k-1}/G$ 이므로 역순서 전송이 일어날 수 없다. P_i^k 의 도착에 의해 흐름 i 가 백로그 되지만 $\Theta_i^{DB} \geq M_{\max}/G$ 인 간섭 바이어스 Θ_i^{RB} 가 가해질 때 $\Delta t < M_{\min}/G \leq M_{\max}/G$ 이므로 역순서 전송이 일어날 수 없다. 따라서 보조정리 1이 증명된다.

[보조정리 2] LOFQ에서 임의 흐름에 가해지는 간섭 바이어스 Θ^{RB} 는 다음과 같이 주어진다.

$$\Theta^{RB} = \frac{\sum_{k \in B(t)} \max(0, 1 - \beta_k) M_k}{G} \quad (4.2)$$

[증명] 임의의 두 흐름 i 와 j 를 생각하고, 흐름 j 의 레이턴시 지수를 β_j , 흐름 j 의 m 번째 도착 패킷을 P_j^m 이라 하자. 그때 P_j^m 의 타임스탬프 T_j^m 는 (3.4)에 의해 다음과 같이 표현된다.

$$T_j^m = F_j^{m-1} + \beta_j (l_j^m / R_j) = F_j^{m-1} + l_j^m / R_j - (1 - \beta_j) l_j^m / R_j \quad (4.3)$$

(4.3)에 의해 $\beta_j < 1$ 의 경우 P_j^m 는 $\beta_j = 1$ 일 때 보다 $(1 - \beta_j) l_j^m / R_j$ 만큼 앞선 타임스탬프를 가지므로 그만큼 빨리 전송되는데, 이는 흐름 j 가 최대 $(1 - \beta_j) M_j$ 만큼의 트래픽을 먼저 서비스 받을 수 있음을 의미한다. LOFQ 스케줄러는 작업보존 시스템이므로 트래픽 $(1 - \beta_j) M_j$ 의 전송시간, 즉 $(1 - \beta_j) M_j / G$ 만큼 흐름 i 의 패킷 전송 시작 시점이 늦어질 수 있다. 따라서 흐름 j 가 수락되기 전과 수락된 후의 간섭 바이어스를 각각 Θ_j^{RB} 와 Θ_j^{-RB} 라 할 때 다음의 관계가 성립한다.

$$\Theta_j^{RB} = \Theta_j^{-RB} + \max(0, 1 - \beta_j) M_j / G \quad (4.4)$$

흐름 i 를 제외한 수락흐름 집합 $B(t)$ 내 흐름은 모두 흐름 i 에 가해지는 간섭 바이어스를 증가시키므로 흐름 i 에 가해지는 간섭 바이어스 Θ_i^{RB} 는 다음과 같이 표현된다.

$$\Theta_i^{RB} = \frac{\sum_{k \in B(t), k \neq i} \max(0, 1 - \beta_k) M_k}{G} \leq \frac{\sum_{k \in B(t)} \max(0, 1 - \beta_k) M_k}{G} \quad (4.5)$$

따라서 임의 흐름에 가해지는 간섭 바이어스 Θ^{RB} 는 (4.2)와 같이 주어진다.

[정리 1] LOFQ에서 임의 흐름에 가해지는 레이턴시 바이어스 Θ 는 다음과 같이 표현된다.

$$\Theta = \max\left(\frac{\sum_{k \in B(i)} \max(0, 1 - \beta_k) M_k}{G}, \frac{M_{\max}}{G}\right) \quad (4.6)$$

[증명] 흐름 i 의 k 번째 패킷 P_i^k 를 생각해보자. P_i^k 의 도착에 의해 흐름 i 가 백로그 되고 $\Theta_i^{RB} \leq M_{\max}/G$ 인 간섭바이어스 Θ_i^{RB} 가 흐름 i 에 가해지면 보조정리 1에 의해 흐름 i 에 가해지는 레이턴시 바이어스 Θ_i 는 다음과 같다: $\Theta_i = \max(\Theta_i^{DB}, \Theta_i^{RB}) = M_{\max}/G$. 한편 $\Theta_i^{RB} > M_{\max}/G$ 인 경우 Θ_i 는 다음과 같다: $\Theta_i = \max(\Theta_i^{DB}, \Theta_i^{RB}) = \Theta_i^{RB}$. 따라서 보조정리 2에 의해 임의 흐름에 가해지는 레이턴시 바이어스는 (4.6)과 같다.

이제 (3.1)로 표현되는 임의 흐름 i 의 레이턴시를 증명해보자.

[보조정리 3] LOFQ에서 β_i 의 레이턴시 지수를 갖는 흐름 i 를 제외한 모든 흐름의 레이턴시 지수가 1이고, 흐름 i 에 가해지는 레이턴시 바이어스가 없을 경우 흐름 i 의 레이턴시는 다음과 같다.

$$Q_i = \beta_i \frac{M_i}{R_i} \quad (4.7)$$

[증명] LOFQ에서 임의 흐름 i 의 k 번째 패킷 P_i^k 의 타임스탬프 T_i^k 는 (3.4)에 의해 $T_i^k = S_i^k + \beta_i (k^k/R_i)$ 가 된다. 임의 흐름 i 를 제외한 모든 흐름의 레이턴시 지수가 1이므로 P_i^k 는 PRPS에서 패킷 크기가 $\beta_i k^k$ 이고 예약속도가 R_i 인 흐름의 패킷으로 간주할 수 있다. 따라서 LOFQ에서 가해지는 레이턴시 바이어스가 없는 흐름 i 의 레이턴시는 PRPS에서 가해지는 역순서 바이어스가 없고 최대 패킷크기 $\beta_i M_i$ 인 흐름의 레이턴시와 동일하므로 (4.7)이 증명된다.

[정리 2] LOFQ에서 임의 흐름 i 의 레이턴시 Q_i 는 (3.1)과 같이 주어진다.

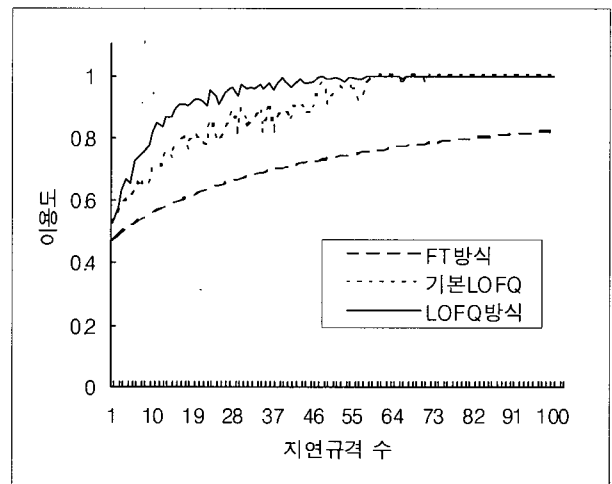
[증명] 레이턴시 바이어스가 없는 경우 임의 흐름 i 의 레이턴시는 보조정리 3으로부터 $\beta_i M_i/R_i$ 로 주어진다. 흐름 i 에 가해지는 레이턴시 바이어스는 정리 1과 같고, 그의상한값 Θ_{UB} 를 초과하지 않는다. 따라서 임의 흐름 i 에 가해지는 레이턴시는 이들의 합을 초과하지 않으므로 (3.1)이 증명된다.

5. 성능평가

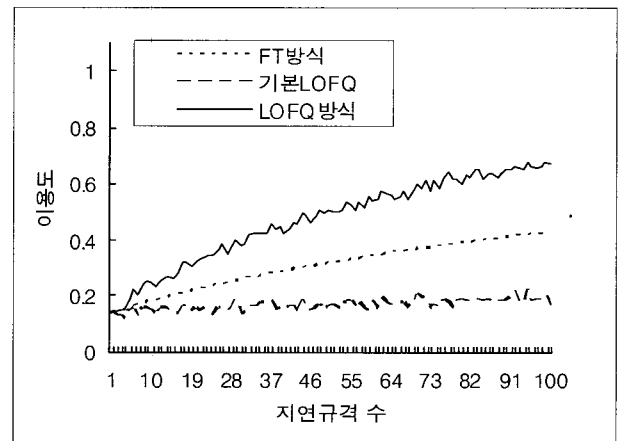
성능평가에 사용할 흐름의 요구속도와 지연규격 분포, 즉 임의 흐름의 스케줄링 규격 $S(r, B)$ 에서 r 과 B 의 분포를 고찰해보자. 인터넷에서 유통되는 미디어는 8Kbps급의 저속 음성부터 1.5Mbps급의 고속 영상까지 다양하므로 다음과 같은 두 가지 서비스 모형을 고려한다: 8Kbps의 음성, 128Kbps의 오디오 및 1.5Mbps의 영상의 3가지 흐름이 동일한 확률

로 존재하는 멀티미디어 응용과 이들 음성, 오디오 및 영상 흐름이 1:0.5:0.1의 비율로 존재하는 인터넷 전화. 대표적인 품질보장 서비스인 인터넷 전화의 종단간 지연이 국내의 경우 150ms이하의 단일 품질등급만 정의되었으나 일본의 경우 A등급(100ms이하), B등급(150ms이하), 및 C등급(400ms이하)의 3 등급으로 정의되어 있다[15]. 앞으로 많은 사업자가 출현하여 인터넷 전화 서비스의 다양화, 고급화 등을 추구할 경우 종단간 지연은 한층 더 차별화 될 것으로 예상되므로 본 논문에서는 지연규격의 수 N_B 를 최대 100개, n 번째 지연규격의 값을 $(1+(n-1)*0.1)$ ms로 하고, 임의 흐름의 지연규격 B 는 N_B 개의 지연규격에 균등하게 분포하는 것으로 가정한다. 그리고 라우터는 1Gbps의 출력링크를 가지며 모든 흐름의 패킷 크기는 모두 M 으로 가정한다. 수치해석과 시뮬레이션 방법을 이용하여 기존 FT 방식과의 비교 측면에서 LOFQ 방식의 성능을 평가한다. 시뮬레이션은 SMPL(Simulation Model Programming Language)[16]에 공평 패킷 스케줄러 루틴을 추가하여 수행한다.

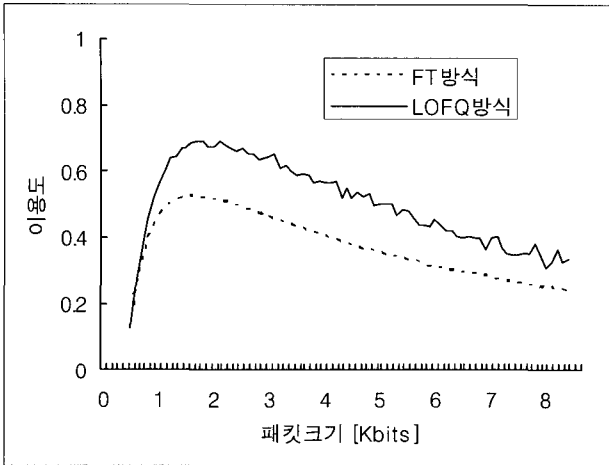
$M=1000$ 비트 및 $N_B=1 \sim 100$ 하에서 각 서비스 모형에 대한 대역폭 이용도를 비교한 결과가 (그림 2)와 (그림 3)에 도시되어



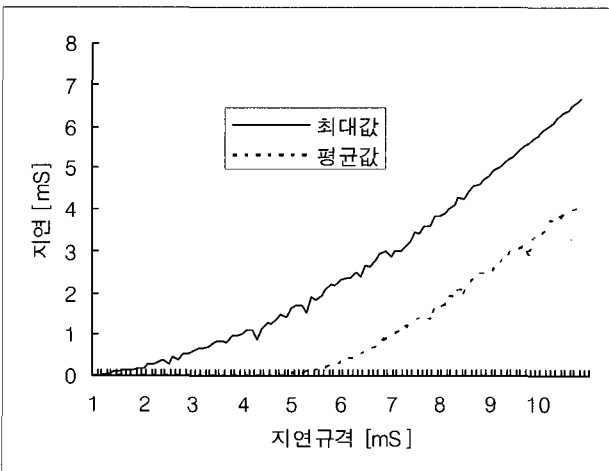
(그림 2) 대역폭 이용도(멀티미디어 응용)



(그림 3) 대역폭 이용도(인터넷 전화)



(그림 4) 유효 대역폭 이용도



(그림 5) 레이턴시 측정 결과

있다. 두 가지 서비스 모형 모두 LOFQ 방식이 FT 방식보다 최대 50% 이상 우수한 대역폭 이용도를 보여준다. (그림 3) 이 그림 2 보다 낮은 이용도를 보이는데, 이는 인터넷 전화 모형의 평균 요구속도가 멀티미디어 응용 모형의 그것 보다 낮기 때문이다. $N_B=1$ 인 경우 모든 흐름의 레이턴시 지수가 0의 값을 가지므로 이 경우 LOFQ 방식의 이용도는 바로 ST 방식의 이용도가 된다. 따라서 그림 2와 3으로부터 $N_B>1$ 인 실제적인 환경하에서 ST 방식의 스케줄러 대역폭 이용 효율은 FT 방식과 LOFQ 방식의 그것에 비해 매우 열악함을 확인할 수 있다. 이들 그림에서 기본 LOFQ 커브는 자원변환 기능이 없는 기본 LOFQ 방식에 대한 것으로 LOFQ의 자원 변환 효과를 살펴보기 위해 도시한 것이다. 평균 요구속도의 값이 낮은 인터넷 전화 모형의 경우 기본 LOFQ 방식은 매우 낮은 이용도 특성을 보이는데, 이는 대역폭 자원에 비해 지연 자원의 고갈 속도가 더 빠르기 때문이다. 하지만 여유 대역폭 자원을 고갈된 지연 자원의 확충에 투입 함으로서 대역폭 이용도를 높일 수 있는데, 바로 LOFQ 방식 커브가 그것을 입증하고 있다.

패킷은 헤드부분과 페이로드(payload)가 담기는 바디부분

으로 구성된다. 따라서 사용자 입장에서 볼 때 의미가 없는 헤드부분을 제외한 대역폭 이용도, 즉 페이로드 부분만 고려한 실제적 대역폭 이용도가 더 의미가 있다. 본 논문에서는 이러한 이용도를 유효 대역폭 이용도라 부른다. 패킷의 헤드부분의 크기를 M_0 로 표기한다. 그러면 유효 대역폭 이용도는 대역폭 이용도에 $(M-M_0)/M$ 을 곱한 값으로 계산된다. 예로 RTP(Real Time Protocol) 경우 M_0 가 54바이트(432비트)인데, 이는 14바이트의 인터넷 헤더, 20바이트의 IP(Internet Protocol) 헤더, 8바이트의 UDP(User Datagram Protocol) 헤더 및 12바이트의 RTP 헤더로 구성되기 때문이다. 멀티미디어 응용 모형에서 패킷 크기 변화에 따른 유효 대역폭 이용도가 (그림 4)에 도시되어 있다. 모두 볼록한 커브형태를 갖는데, 이는 패킷크기가 작아지면 패킷의 헤드부분의 크기가 상대적으로 증가하여 이용도가 저하되고, 반대로 패킷크기가 커지면 대역폭 이용도가 낮아져서 역시 이용도가 저하되기 때문이다. 역시 LOFQ 방식이 우수한 특성을 보이는데, 이는 유효 대역폭 이용도가 대역폭 이용도에 비례하기 때문이다. 다음은 시뮬레이션을 통해서 LOFQ 스케줄러의 레이턴시를 실제 관찰해 본다. $M=1000$ 비트, $N_B=100$ 및 멀티미디어 응용 모형 하에서 과부하 트래픽을 가하면서 1.5초 동안 측정된 최대지연과 평균지연을 (그림 5)에 도시하였다. 이 그림으로부터 LOFQ 방식이 지연규격을 보장함을 확인할 수 있다.

6. 결 론

본 논문에서는 스케줄러에서 발생하는 흐름의 레이턴시를 허용되는 최대 레이턴시, 즉 지연규격과 동일하게 하여 스케줄러 대역폭의 이용 효율을 개선할 수 있는 레이턴시가 최적화되는 공평 패킷 스케줄러를 제안하고 성능특성을 분석하였다.

레이턴시가 지연규격을 초과할 경우 과잉예약에 의해 흐름의 레이턴시를 줄일 수만 있는, 즉 단 방향 레이턴시 통제 수단만 갖는 기존 방식에 비해 제안된 방식은 레이턴시 지수와 예약속도의 조정으로 흐름의 레이턴시를 줄이는 것은 물론 늘릴 수도 있다. 이러한 양 방향 레이턴시 통제수단으로 스케줄러에서 발생하는 흐름의 레이턴시를 그 흐름의 지연규격에 일치시키는, 즉 흐름의 레이턴시를 최적화 할 수 있으므로 지연 자원의 손실을 방지할 수 있다.

레이턴시 바이어스로 사용된 지연 자원을 정량화하고 가용 대역폭 자원과 가용 지연 자원을 상호 변환할 수 있게 하였으며, 자원 변환 효율을 극대화 하는 방법도 제시하였다. 기존 방식의 경우 가용 대역폭 자원이 부족하면 더 이상 새로운 흐름을 수락할 수 없는데 반해 제안된 방식은 가용 자원 변환 기능으로 부족한 가용 자원을 보충할 수 있으므로 가용 대역폭 자원과 가용 지연 자원이 모두 부족한 경우가 아니면 흐름을 수락할 수 있기 때문에 보다 많은 흐름을 수락하여 스케줄러 대역폭의 이용 효율을 높일 수 있다.

수치해석과 시뮬레이션 방법으로 제안된 방식의 성능 특

성을 기존 FT 방식과의 비교를 통해 평가하였다. 자원변환 기능을 가동하지 않을 경우 가용 대역폭 자원은 여유가 많은데도 불구하고 가용 지연 자원의 부족으로 인해 FT 방식보다 낮은 이용도를 보였다. 하지만 자원변환 기능을 가동하면 여유 대역폭 자원으로 부족한 가용 지연 자원을 보충하여 더 많은 흐름을 수락하기 때문에 FT 방식보다 높은 대역폭 이용도 특성을 보인다는 것을 확인하였다. 그리고 시뮬레이션을 통해 실제 레이턴시를 관찰해본 결과 지연규격을 충족시켰다.

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] P. Goyal, H. M. Vin, and H. Cheng, "Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks," *IEEE/ACM Trans. Networking*, Vol.5, No.5, pp.690~704, 1997.

[5] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," *IEEE JSAC*, Vol.9, pp.1265~1279, 1991.

[6] S. Shenker, C. Partridge, and R. Guerin, *Specification of Guaranteed Quality of Service*, RFC 2212, IETF, 1997.

[7] 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.

[8] Jens Schmitt. "Optimal Network Service Curves under Bandwidth-Delay Decoupling," *IEE Electronics Letters*, Vol.38, No.6, pp.297~299, 2002.

[9] S.J. Golestani, "A Self-Clocked Fair Queuing Scheme for Broadband Applications," *Proc. IEEE INFOCOM '94*, pp.636~646, 1994.

[10] D. Stiliadis and A. Varma, "Rate Proportional Servers: A Design Methodology for Fair Queueing Algorithms," *IEEE/ACM Trans. Networking*, Vol.6, No.2, pp.164~174, 1998.

[11] D. Stiliadis and A. Varma, "Efficient Fair Queueing Algorithms for Packet-Switched Networks," *IEEE/ACM Trans. Networking*, Vol.6, No.2, pp.175~185, 1998.

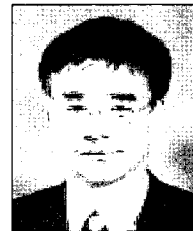
[12] Dong-Yong KWAK, Nam-Seok KO, Hong-Shik PARK, "Medium Starting Potential Fair Queueing for High-Speed Networks," *IEICE Trans. Communications*, Vol.E87-B, No.1, pp.188~19, 2004.

[13] Dong-Yong Kwak, Nam-Seok Ko, Bongtae Kim, and Hong-Shik Park, "A New Starting Potential Fair Queueing Algorithm with O(1) Virtual Time Computation Complexity," *ETRI Journal*, Vol.25, No.6, pp.475~488, 2003.

[14] H. Zhou, "Real-Time Services over High Speed Network," Ph.DCurtin University of Technology, Australia, 2002.

[15] 이은근, "인터넷 전화 국내의 제도화 현황 및 정책적 시사점", 정보통신정책, 제16권, 제17호, pp.1~17, 2004.

[16] M. H. MacDougall, *Simulating Computer Systems, Techniques and Tools*, MIT Press, 1987.



김 태 준

e-mail : tjkim@kongju.ac.kr
 1980년 경북대학교 전자공학과(학사)
 1982년 한국과학기술원 전자공학(석사)
 1999년 한국과학기술원 전자공학(박사)
 1982년 한국전자통신연구원
 1996년 천안공업대학 교수
 2005년~현재 공주대학교 정보통신공학부 부교수
 관심분야 : 고속통신망, VoIP, 트래픽 제어



김 황 래

e-mail : plusone@kongju.ac.kr
 1982년 중앙대학교 컴퓨터공학과(공학사)
 1991년 중앙대학교 컴퓨터공학과
 (공학석사)
 1994년 중앙대학교 컴퓨터공학과
 공학박사 수료
 1983년~1994년 한국전자통신연구원 선임연구원
 1994년~현재 공주대학교 컴퓨터공학과 교수
 관심분야 : 컴퓨터네트워크, 네트워크 보안, 컴퓨터 시뮬레이션