

서버자원 이용도 측면에서 타임스탬프 기반 공평 패킷 스케줄러의 성능 비교 분석

김 태 준[†] · 안 효 범[†]

요 약

실시간 멀티미디어 응용의 서비스 품질을 보장하는 공평 패킷 스케줄링 알고리즘은 타임스탬프 계산시 사용되는 패킷의 기준시각 측면에서 종료시각 방식과 시작시각 방식으로 나뉜다. 전자는 트래픽 흐름의 예약속도 조정으로 원하는 레이턴시(latency) 값을 얻을 수 있어 IETF (Internet Engineering Task Force) 종합서비스 모델의 품질보장 라우터에 적용되었으나 인터넷 전화와 같이 흐름 속도가 낮지만 우수한 지연특성을 요구하는 트래픽의 경우 심각한 대역폭 손실이 발생하는 문제가 있다. 본 논문에서는 시작시각 방식을 품질보장 라우터에 적용하는 방안을 모색하기 위해 대역폭 및 페이로드(payload) 이용도 측면에서 두 가지 공평 패킷 스케줄링 방식의 성능특성을 비교 분석하였다. 분석결과 대역폭 이용도 측면에서 시작시각 방식이 우수하다는 것이 증명되었고, 모의실험결과 시작시각 방식이 최대 20%까지 높은 페이로드 이용도를 보였다.

키워드 : 패킷 스케줄러, 공평큐잉, 공평성, 레이턴시, 이용도

Performance Comparison of Timestamp based Fair Packet Schedulers in Server Resource Utilization

Tae-Joon Kim[†] · Hyo Beom Ahn[†]

ABSTRACT

Fair packet scheduling algorithms supporting quality-of-services of real-time multimedia applications can be classified into the following two design schemes in terms of the reference time used in calculating the timestamp of arriving packet: Finish-time Design (FD) and Start-time Design (SD) schemes. Since the former can adjust the latency of a flow with raising the flow's reserved rate, it has been applied to a router for the guaranteed service of the IETF (Internet Engineering Task Force) IntServ model. However, the FD scheme may incur severe bandwidth loss for traffic flows requiring low-rate but strong delay bound such as internet phone. In order to verify the usefulness of the SD scheme based router for the IETF guaranteed service, this paper analyzes and compares two design schemes in terms of bandwidth and payload utilizations. It is analytically proved that the SD scheme is better bandwidth utilization than the FD one, and the simulation result shows that the SD scheme gives better payload utilization by up to 20%.

Key Words : Packet Scheduler, Fair Queuing, Fairness, Latency, Utilization

1. 서 론

널리 보급 확산되고 있는 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)와 확장성이 우수한 차등서비스(DiffServ)의 두 가지 품질 지원 모델을 제시하였다[1]. IntServ 모델은 품질보장 라우터로 보장형 서비스의 요구 품질을 보장하는데, 품질보장 라우터는 트래픽 흐름의 요구 속도를 보장하고 요구 지연규격을

위반하지 않는 공평 패킷 스케줄러를 탑재한다. 여기서 트래픽 흐름 (이하 흐름)이란 실시간 멀티미디어 통신 서비스에 수반되는 각각의 미디어 스트림(stream), 예로 음성, 영상 스트림 등을 의미하고, 지연규격은 라우터, 즉 스케줄링 서버에서 그 흐름에게 허용되는 최대 통과(transit)지연시간을 의미한다.

이상적 패킷 스케줄링 알고리즘인 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 타임스탬프방식이고, 다른 하나는 WRR(Weighted Round Robin)[5]에 적용된 라운드로빈(round-robin)방식이다. 타임스탬프방식은 구현이 복잡한 반면 라운드로빈방식은 레이턴시(latency: 최대전달지연) 특성이 나쁜

[†] 정 회 원 : 공주대학교 정보통신공학부 부교수
논문접수 : 2005년 7월 8일, 심사완료 : 2006년 2월 13일

단점이 있다. 타임스탬프 방식은 또한 도착하는 패킷의 타임스탬프 계산시 사용되는 패킷의 기준시간 관점에서 예상되는 전송종료시각을 사용하는 종료시간 방식과 예상되는 전송시작시각을 사용하는 시작시간 방식으로 나뉜다. 본 논문에서는 편의상 전자를 FD(Finish-time based Design) 방식, 후자를 SD(Start-time based Design) 방식이라 부른다.

FD방식은 흐름의 속도에 반비례하는 레이턴시 특성[3]을 갖는 반면 SD 방식은 흐름의 속도에 무관하나 서버에 수용되는 흐름의 수에 비례하는 레이턴시 특성[4]을 갖는다. FD 방식은 흐름의 예약속도를 높여 원하는 레이턴시 값을 얻을 수 있어서 IETF IntServ 모델의 품질보장 라우터에 적용되었다[6]. 그러나 예약속도를 높일 경우 예약속도와 실제속도와의 차이, 즉 과잉예약분 만큼 속도자원이 손실된다. 특히 인터넷 전화와 같이 트래픽 흐름의 속도는 낮지만 우수한 종단간 지연특성을 요구하는 경우 과잉예약의 정도가 심해져 더 많은 자원이 손실되는 문제가 발생한다[4, 7, 8]. 이러한 자원손실 문제를 해결하기 위해 부족한 속도(대역폭)를 빌려오는 크레딧(credit) 방식[7]과 예약속도를 가변하는 최적 서비스 곡선(curve) 방식[8]이 연구되었는데, 이들은 모두 다른 흐름이 사용할 대역폭을 빌려서 사용하는 것으로 대역폭의 여유가 없는 고정 속도 트래픽의 경우 자원 손실의 개선 효과를 전혀 기대할 수 없다. 따라서 본 논문에서는 FD 방식 대신 속도-지연특성의 비결합(decoupling) 성질을 갖는 SD 방식을 IETF IntServ 품질보장 라우터에 적용하는 방안의 타당성을 검증하기 위해 네트워크 자원 이용도 측면에서 두 방식을 상세히 비교 평가하고자 한다.

본 논문의 구성은 다음과 같다: 2장에서 관련연구를 소개하고, 3장에서 서버자원 이용도 측면에서 FD와 SD방식의 성능특성을 비교분석하며, 4장에서 모의실험을 통해 FD와 SD방식의 성능특성을 비교평가한다. 그리고 5장에서 결론을 맺는다.

2. 관련 연구

FD방식은 WFQ에 처음 적용되었다. WFQ는 우수한 레이턴시 특성을 제공하나 타임스탬프 계산이 너무 복잡하였다. SCFQ(Self-Clocked Fair Queuing)[9]에서 자가클럭(self-clock)을 도입하여 이 문제를 해결하였으나 레이턴시 특성이 저하되는 또 다른 문제점이 대두되었다. RPS(Rate Proportional Servers)[10]에서 자가클럭 대신 서버 가상시간을 도입하여 두 가지 문제점을 동시에 해결할 수 있는 방안을 제안하였다. RPS기반의 다양한 패킷 스케줄러가 개발되었는데[11-13], 이들은 모두 서버 가상시간의 운영과 유지방식을 개선하는데 주안점을 두었다. 한편 SD방식은 SFQ(Starting-time Fair Queueing) 및 QLR(Queueing latency Rate)[14]에 적용되었다. SFQ는 SCFQ와 마찬가지로 자가클럭을 사용하나 QLR은 자가클럭 대신 RPS에서 제안된 서버 가상시간을 사용한다.

[15]에서 공평 패킷 스케줄러가 갖추어야 할 특성을 다음

과 같이 제시하였다: 첫째 서비스 이용자가 요구하는 종단간 서비스 지연규격을 만족시킬 수 있도록 우수한 레이턴시 특성을 가져야 하고, 둘째 초고속 네트워크에 적용될 수 있도록 구현복잡성이 낮아야 하며, 셋째 스케줄링 서버에 여분의 대역폭이 있을 경우 이 대역폭을 각 흐름에 공평하게 분배해야 한다. 이러한 요구사항으로부터 레이턴시, 구현복잡성 및 공평성의 세가지 성능평가 지표가 도출되었고, 종래연구[6-8]에서 스케줄러의 성능평가에 사용되었다.

각 지표의 유용성을 생각해보자. 종단간 서비스 지연시간은 각 노드에 발생된 지연시간의 합으로 계산되는데, 각 노드의 지연시간을 평가하기 위해 레이턴시 지표가 필요하다. 스케줄링 서버의 전송용량은 출력링크의 대역폭뿐만 아니라 스케줄링 알고리즘을 동작시키는 컴퓨팅 능력에 의해서도 제한되므로 컴퓨팅 능력의 제약 정도를 평가하기 위해 구현복잡성 지표가 요구된다. 반면 공평 패킷 스케줄러는 기본적으로 각 흐름이 예약한 속도, 즉 각 흐름에 예약된 서버 대역폭을 완벽히 보장한다. 다만 서버에 여분의 가용 대역폭이 있어서 이를 각 흐름에 분배할 때 짧은 시간구간내에서 불공평성이 발생할 수 있는데, 이는 서비스 이용자에게는 아무런 영향을 주지 않고 서버 출력 트래픽의 간헐성(burstiness) 특성에 영향을 미치는 정도이다. 따라서 공평성 지표는 레이턴시와 구현복잡성 지표에 비해 그 유용성이 매우 낮다. 한편 구현복잡성은 타임스탬프계산 복잡성과 패킷 전송 복잡성에 의해 결정되는데[16], 타임스탬프 계산에 사용되는 패킷 기준시간의 차이가 타임스탬프계산 복잡성에 아무런 영향을 미치지 않기 때문에 FD방식과 SD방식은 동일한 구현 복잡성을 갖는다.

레이턴시 측면에서 FD와 SD방식 관련 기존연구를 살펴보자. 임의 흐름 i 의 요구속도를 r_i , 흐름 i 의 최대 패킷크기를 L_i , 모든 흐름의 최대 패킷크기를 L_{max} , 그리고 서버 대역폭을 C 라 하자. 참고로 서버 대역폭이란 서버의 출력링크의 전송용량을 의미하고, 본 논문에서 사용되는 속도와 대역폭은 같은 의미를 갖는다.

[10]에서 체계화된 FD방식의 타임스탬프 계산공식을 살펴보면 다음과 같다: 시간 t 에 도착하는 흐름 i 의 k 번째 패킷, 즉 p_i^k 의 타임스탬프 F_i^k 는 $\max(F_i^{k-1}, P(t)) + L_i^k / r_i$ 로 계산된다. 여기서 F_i^{k-1} 는 흐름 i 의 $(k-1)$ 번째 패킷의 가상 종료시간이고, $P(t)$ 는 시스템시간이며, L_i^k 는 p_i^k 의 크기이다. RPS는 서버 가상시간을 $P(t)$ 로 사용한다. 서버 가상시간이란 서버 활성구간 동안, 즉 서버가 지속적으로 패킷을 전송하는 구간의 시작시점부터 현재까지 전송된 전체 트래픽량을 나타내는 시간의 함수를 의미한다. FD방식을 적용하는 스케줄러에서 흐름 i 의 레이턴시는 다음과 같이 계산된다[10].

$$\frac{L_i}{r_i} + \frac{L_{max}}{C} \quad (2.1)$$

(2.1)에서 L_i / r_i 항은 대응 GPS 서버에서의 패킷 서비스 시간을 의미하고, L_{max} / C 항은 출력링크에서 최대 패킷전송

시간을 의미한다. 예외적으로 자가클럭 (전송중인 패킷의 타임스탬프)를 $P(t)$ 로 사용하는 SCFQ의 레이턴시 특성은 $L_i / r_i + VL_{\max} / C$ 로 계산되는데[9], (2.1)의 레이턴시에 비해 크게 저하됨을 알 수 있다.

[4]에서 체계화된 SD방식의 타임스탬프 계산공식을 살펴보면 다음과 같다: p_i^k 의 타임스탬프 S_i^k 는 $\max(F_i^{k-1}, P(t))$ 로 계산된다. 여기서 S_i^k 는 p_i^k 의 가상시작시각이고, $F_i^{k-1} = S_i^{k-1} + l_i^{k-1} / r_i$ 이다. QLR에서는 서버 가상시간을 $P(t)$ 로 사용하나 SFQ에서는 자가클럭을 $P(t)$ 로 사용한다. 이러한 SD 방식을 적용하는 SPQ와 QLR에서 흐름의 레이턴시는 다음과 같이 계산된다[4, 14].

$$\frac{1}{C} \sum_{i=1}^{\nu^{SD}} L_i, \tag{2.2}$$

여기서 ν^{SD} 는 SD방식을 사용하는 스케줄링 서버에 수용되는 최대 흐름 수를 의미한다.

3. 성능비교 분석

서버자원 이용도 측면에서 FD방식과 SD방식의 성능특성을 비교분석한다. 다수의 지연규격을 요구하는 트래픽도 각 지연규격을 요구하는 트래픽으로 분리하여 생각할 수 있으므로 본 논문에서는 모든 트래픽 흐름은 하나의 지연규격 Q 를 요구하는 것으로 가정한다.

3.1 대역폭이용도

본 논문에서 스케줄링 서버의 대역폭 이용도(utilization)를 다음과 같이 정의한다: 트래픽 처리에 할당된 서버 대역폭 중 트래픽이 요구하는 속도뿐만 아니라 요구 지연규격을 모두 만족시키면서 실제 트래픽 전송에 이용된 대역폭의 비율. FD방식과 SD방식을 사용하는 서버들, 즉 FD서버와 SD서버의 대역폭 이용도를 구해보자. 흐름의 요구속도를 $[a, b]$ 내 분포하는 랜덤변수 R 로 가정한다.

FD 서버에서 요구속도가 r_i 인 임의의 흐름 i 에 대해 그의 요구지연규격을 만족시키기 위해 필요한 최저속도를 흐름 i 의 임계속도라 부르고, 흐름 i 의 임계속도와 예약속도를 각각 r_i^{crit} 와 r_i^{res} 로 표기한다. 흐름의 요구속도가 임계속도보다 빠르면 요구속도로 예약하고, 느리면 임계속도로 예약해야 하므로 $r_i^{res} = \max(r_i, r_i^{crit})$ 가 된다. 따라서 요구속도가 임계속도보다 작을, 즉 $r_i < r_i^{crit}$ 인 경우 $(r_i^{crit} - r_i)$ 만큼 더 많은 서버대역폭을 예약해야 하는 과잉예약 현상이 일어나서 대역폭이 손실되는데, 이를 예약손실이라 부른다. $(r_i^{crit} - r_i)$ 를 흐름 i 의 미달속도라 부르고 r_i^{def} 로 표기한다.

총 예약속도가 서버대역폭 C 를 초과할 수 없으므로 FD서버가 수용할 수 있는 최대 흐름 수 ν^{FD} 는 다음과 같이 계산된다.

$$\nu^{FD} = \{k \mid \sum_{i=1}^k r_i^{res} \leq C \text{ and } \sum_{i=1}^{k+1} r_i^{res} > C\}. \tag{3.1.1}$$

복잡성을 피하기 위하여 모든 흐름의 최대 패킷크기는 L_{\max} 로 동일하다고 가정하자. 그러면 ν^{FD} 는 $\lfloor C / r^{res} \rfloor$ 로 계산되고, (2.1)로부터 모든 흐름은 모두 똑 같은 임계속도를 갖게 되는데, 이를 r^{crit} 로 표기하면 다음과 같이 계산된다.

$$r^{crit} = CL_{\max} / (CQ - L_{\max}). \tag{3.1.2}$$

도착하는 흐름의 예약속도는 $[r^{crit}, b]$ 내 분포하는 새로운 랜덤변수 $\max(R, r^{crit})$ 로 표현되며, 이의 평균값 r^{res} 는 다음과 같이 계산된다.

$$r^{res} = E[\max(R, r^{crit})] = r^{crit} F_R(r^{crit}) + \int_{r^{crit}}^b k f_R(k) dk, \tag{3.1.3}$$

여기서 f_R 과 F_R 은 랜덤변수 R 의 pdf(probability density function)와 cdf(cumulative distribution function)이다. 참고로 도착하는 흐름의 미달속도는 새로운 랜덤변수 $\max(0, (r^{crit} - R))$ 가 되고, 이의 평균값 r^{def} 는 $\int_0^{r^{crit}} (r^{crit} - k) f_R(k) dk$ 로 계산되는데, 이는 r^{res} 에서 r^{req} 를 뺀 결과와 일치한다. (3.1.3)으로부터 FD서버의 대역폭이용도 ρ_{BW_FD} 는 다음과 같이 계산된다.

$$\rho_{BW_FD} := \frac{1}{C} \sum_{i=1}^{\nu^{FD}} r_i = \frac{r^{req}}{C} \left\lfloor \frac{C}{r^{res}} \right\rfloor. \tag{3.1.4}$$

여기서 $r^{req} = E[R]$.

SD서버에서 최대 흐름 수 ν^{SD} 를 구해보자. 예약속도 수용조건으로부터 구해지는 ν^{FD} 와 달리 ν^{SD} 는 예약속도 수용과 지연규격 수용의 두 가지 조건으로부터 구해진다. 복잡성을 피하기 위하여 모든 흐름의 최대 패킷크기는 L_{\max} 로 동일하다고 가정하자. 예약속도 수용측면에서 흐름수의 상한치 ν_R^{SD} 는 다음과 같이 계산된다.

$$\nu_R^{SD} = \{k \mid \sum_{i=1}^k r_i \leq C \text{ and } \sum_{i=1}^{k+1} r_i > C\} = \left\lfloor \frac{C}{r^{req}} \right\rfloor. \tag{3.1.5}$$

한편 지연규격 수용측면에서 흐름 수의 상한치 ν_Q^{SD} 는 (2.2)로부터 다음과 같이 계산된다.

$$\nu_Q^{SD} = \{k \mid \sum_{i=1}^k L_i \leq CQ \text{ and } \sum_{i=1}^{k+1} L_i > CQ\} = \left\lfloor \frac{CQ}{L_{\max}} \right\rfloor. \tag{3.1.6}$$

따라서 V^{SD} 는 다음과 같이 결정된다.

$$V^{SD} = \min(V_R^{SD}, V_Q^{SD}) = \min\left(\left\lfloor \frac{CQ}{L_{\max}} \right\rfloor, \left\lfloor \frac{C}{r^{req}} \right\rfloor\right). \quad (3.1.7)$$

$V_Q^{SD} = V_R^{SD}$ 가 되는 최대 패킷크기를 임계패킷크기라 부르고 L_{crit} 로 표기하자. (3.1.7)로부터 SD서버의 대역폭이용도 ρ_{BW_SD} 는 다음과 같이 계산된다.

$$\rho_{BW_SD} := \frac{1}{C} \sum_{i=1}^{V^{SD}} r_i = \frac{r^{req}}{C} \min\left(\left\lfloor \frac{CQ}{L_{\max}} \right\rfloor, \left\lfloor \frac{C}{r^{req}} \right\rfloor\right). \quad (3.1.8)$$

대역폭 손실량을 분석해보자. FD와 SD서버에서 대역폭 손실량을 각각 L^{FD} 와 L^{SD} 라 하자. 서버의 가용 대역폭이 부족하여 새로운 트래픽 흐름을 수용할 수 없을 경우 그 대역폭은 손실되는데, 이를 잔존손실이라 부르고, 그 량을 L^{FD_rsd} 라 하자. 과잉예약, 즉 미달속도에 의해 발생하는 예약손실량을 L^{res} 라 하자. 그러면 $L^{FD_rsd} < r^{res}$ 이고 $L^{res} \leq V^{FD} r^{def}$ 이므로 $L^{FD} < V^{FD} r^{def} + r^{res}$ 가 된다. 한편 SD서버에서 $V_R^{SD} < V_Q^{SD}$, 즉 $L_{\max} < L_{crit}$ 이면 잔존손실만 발생하는데, 그 량을 L^{SD_rsd} 라 하자. 이는 r^{req} 보다 작다. 반면에 $V_R^{SD} \geq V_Q^{SD}$, 즉 $L_{\max} \geq L_{crit}$ 이면 흐름 수의 제약으로 인해 V_Q^{SD} 개의 흐름을 수용하고 남은 대역폭은 모두 손실된다. 이를 흐름손실이라 부르고 그 량을 L^{flow} 라 하자. $L^{SD_rsd} < r^{req}$ 이고 $L^{flow} \leq (C - V_Q^{SD} r^{req})$ 이므로 SD서버에서 대역폭 손실량은 다음과 같이 표시된다.

$$L^{SD} = \begin{cases} L^{SD_rsd} < r^{req}, & \text{for } L_{\max} < L_{crit} \\ L^{flow} \leq C - V_Q^{SD} r^{req}, & \text{for } L_{\max} \geq L_{crit}. \end{cases} \quad (3.1.9)$$

3.2 이용도 비교

[정리 3.1] 동일한 트래픽 부하와 지연규격하에서 SD방식의 대역폭이용도 ρ_{BW_SD} 는 FD방식의 대역폭 이용도 ρ_{BW_FD} 와 같거나 더 높다. 즉,

$$\rho_{BW_FD} \leq \rho_{BW_SD}. \quad (3.2.1)$$

[증명] FD방식과 SD방식에서 최대 수용 흐름의 수, 즉 V^{FD} 와 V^{SD} 를 비교해보자. V^{SD} 가 예약속도 수용조건에 의해 결정되는 경우 정의에 의해 $r_i^{res} > r_i$ 가 되고 동일한 트래픽 부하가 가해지므로 (3.1.1)과 (3.1.5)로부터

$$\begin{aligned} V^{FD} &= \{k \mid \sum_{i=1}^k r_i^{res} \leq C \text{ and } \sum_{i=1}^{k+1} r_i^{res} > C\} \\ &\leq \{k \mid \sum_{i=1}^k r_i \leq C \text{ and } \sum_{i=1}^{k+1} r_i > C\} = V^{SD}. \end{aligned} \quad (3.2.2)$$

다음은 V^{SD} 가 지연규격 수용조건에 의해 결정되는 경우에 대해 살펴보자. FD방식에서 (2.1)로부터 $r_i^{crit} = CL_i / (CQ - L_{\max})$ 로 계산되고, 정의에 의해 $r_i^{res} \geq r_i^{crit}$ 가 되며, 또한 예약속도 수용조건에 의해 $\sum_{i=1}^{V^{FD}} r_i^{res} \leq C$ 가 된다. 따라서 $\sum_{i=1}^{V^{FD}} r_i^{crit} \leq \sum_{i=1}^{V^{FD}} r_i^{res} \leq C$ 가 성립하므로 다음 부등식이 유도된다.

$$\sum_{i=1}^{V^{FD}} L_i \leq (CQ - L_{\max}). \quad (3.2.3)$$

한편 SD방식의 경우 (2.2)로부터 다음 부등식이 유도된다.

$$\sum_{i=1}^{V^{SD}} L_i \leq CQ. \quad (3.2.4)$$

동일한 트래픽 부하가 가해지므로 (3.2.3)과 (3.2.4)로부터

$$\begin{aligned} V^{FD} &= \{k \mid \sum_{i=1}^k L_i \leq (CQ - L_{\max}) \text{ and } \sum_{i=1}^{k+1} L_i > (CQ - L_{\max})\} \\ &\leq \{k \mid \sum_{i=1}^k L_i \leq C \text{ and } \sum_{i=1}^{k+1} L_i > C\} = V^{SD}. \end{aligned} \quad (3.2.5)$$

(3.2.2)와 (3.2.5)로부터 항상 $V^{FD} \leq V^{SD}$ 관계가 성립한다. 따라서 (3.1.4)와 (3.1.8)의 대역폭이용도 정의에 의해 (3.2.1)이 증명된다. □

한편 패킷은 사용자 데이터인 페이로드(payload)가 담기는 바디부분과 오버헤드(overhead)가 되는 헤더부분으로 구성되므로 패킷의 오버헤드와 페이로드 부분이 모두 포함된 대역폭 이용도 보다 사용자 입장에서 볼 때 의미가 없는 오버헤드 부분을 제외한 대역폭 이용도, 즉 페이로드 부분만 고려한 실제적 대역폭 이용도가 더 의미가 있다. 본 논문에서는 이러한 이용도를 페이로드 이용도라 부른다. 패킷의 오버헤드를 L_o 로 표기하자. 예로 RTP(Real Time Protocol) 경우 L_o 가 54바이트(432비트)인데, 이는 14바이트의 이더넷 헤더, 20바이트의 IP 헤더, 8바이트의 UDP 헤더 및 12바이트의 RTP 헤더로 구성되기 때문이다. 트래픽 전송에 사용된 대역폭 중 페이로드의 전송에 사용된 대역폭의 비율을 페이로드 비율이라 하자. 그러면 페이로드 이용도는 대역폭 이용도와 페이로드 비율의 곱으로 정의된다. 페이로드 이용도는 대역폭 이용도에 비례하므로 다음의 추론이 도출된다.

[추론 3.1] 동일한 동일한 트래픽 환경과 지연규격하에서 SD방식의 페이로드 이용도 ρ_{PL_SD} 는 FD 방식의 페이로드 이용도 ρ_{PL_FD} 와 같거나 더 높다. 즉,

$$\rho_{PL_FD} \leq \rho_{PL_SD}. \quad (3.2.7)$$

3.3 최적 패킷크기

페이로드 이용도를 최대화하는 최적 패킷크기와 최대 페이로드 이용도를 구해보자. 복잡성을 피하기 위해 모든 흐름

은 모두 동일한 크기 l 의 패킷을 사용하는 것으로 가정한다.

3.3.1 FD방식

모든 흐름의 패킷크기가 l 이므로 페이로드 이용도 ρ_{PL_FD} 는 다음과 같이 전개된다.

$$\rho_{PL_FD} := \frac{(l-L_o)}{l} \rho_{BW_SD} = \frac{(l-L_o)}{l} \frac{V^{FD} r^{req}}{C}. \quad (3.3.1)$$

$l \leq L_o$ 이면 페이로드를 담을 공간이 없으므로 $\rho_{PL_FD} = 0$ 가 되고, $l \geq CQ/2$ 이면 $r^{crit} > C$, 즉 임계속도가 서버대역폭을 초과하여 V^{FD} 가 0이 되므로 역시 $\rho_{PL_FD} = 0$ 이 된다. 따라서 l 의 범위는 $(L_o, CQ/2)$ 이 된다. V^{FD} 는 l 에 대한 계단형함수가 되어 미분이 불가능하므로 연속적인 함수인 C/r^{res} 로 근사화 하자. 그러면 근사 페이로드 이용도 ρ_{PLA_FD} 는 다음과 같이 전개될 수 있다.

$$\rho_{PLA_FD} := g(l)h(l),$$

여기서 $g(l) = \frac{l-L_o}{l}$ 이고 $h(l) = \frac{r^{req}}{r^{res}}$ 인. (3.3.2)

참고로 $\rho_{PLA_FD} > 0$ 이 되는 l 의 범위는 (L_o, CQ) 이 된다. 패킷크기 변화에 따른 페이로드 이용도의 특성을 살펴보기 위해 l 에 대한 ρ_{PLA_FD} 의 1차 도함수를 전개하면 다음과 같이 표시된다.

$$\frac{d\rho_{PLA_FD}}{dl} = h(l) \frac{dg(l)}{dl} + g(l) \frac{dh(l)}{dl}. \quad (3.3.3)$$

(3.3.2)에서 r^{req} 는 l 에 대해 상수이고, r^{res} 는 (3.1.3)과 같이 l 의 함수인 r^{crit} 의 함수가 된다. $h(l)$ 의 도함수를 직접 구하기가 어려우므로 r^{res} 의 도함수로부터 구해보자. r^{crit} 에 대한 r^{res} 의 도함수를 구하기에 앞서 $r^{res} := f(r^{crit})$ 로 표기하고 $f(r^{crit} + \Delta r^{crit})$ 를 전개하면

$$\begin{aligned} f(r^{crit} + \Delta r^{crit}) &= (r^{crit} + \Delta r^{crit}) F_R(r^{crit} + \Delta r^{crit}) + \int_{(r^{crit} + \Delta r^{crit})}^b k f_R(k) dk \\ &= f(r^{crit}) + \Delta r^{crit} F_R(r^{crit} + \Delta r^{crit}) \\ &\quad + r^{crit} (F_R(r^{crit} + \Delta r^{crit}) - F_R(r^{crit})) - \int_{r^{crit}}^{(r^{crit} + \Delta r^{crit})} k f_R(k) dk. \end{aligned} \quad (3.3.4)$$

(3.3.4)의 결과를 이용하여 r^{crit} 에 대한 r^{res} 의 도함수를 구하면

$$\frac{dr^{res}}{dr^{crit}} = \lim_{\Delta r^{crit} \rightarrow 0} \frac{f(r^{crit} + \Delta r^{crit}) - f(r^{crit})}{\Delta r^{crit}} = F_R(r^{crit}). \quad (3.3.5)$$

(3.1.2)의 r^{crit} 정의로부터 $dr^{crit}/dl = C^2Q/(CQ-l)^2$ 이므로 l 에 대한 r^{res} 의 도함수를 구하면 $dr^{res}/dl = F_R(r^{crit})C^2Q/(CQ-l)^2$ 가 된다. $h(l)$ 의 정의로부터 $r^{res}dh(l)/dl + h(l)dr^{res}/dl = 0$ 가 되므로 여기에 $h(l)$ 과 dr^{res}/dl 를 대입하고 전개하면 다음과 같은 $h(l)$ 의 도함수가 구해진다.

$$\frac{dh(l)}{dl} = -\frac{r^{req}}{(r^{res})^2} \frac{F_R(r^{crit})C^2Q}{(CQ-l)^2}. \quad (3.3.6)$$

$g(l)$ 의 정의로부터 이의 도함수는 $dg(l)/dl = L_o/l^2$ 가 된다. $h(l)$ 와 $g(l)$ 의 도함수를 (3.3.3)에 대입하면

$$\frac{d\rho_{PLA_FD}}{dl} = \frac{r^{req}}{r^{res}l} \left(\frac{L_o}{l} - \frac{F_R(r^{crit})(l-L_o)C^2Q}{r^{res}(CQ-l)^2} \right). \quad (3.3.7)$$

(3.3.7)에서 $d\rho_{PLA_FD}/dl|_{l=L_o^+} > 0$ 이고 $d\rho_{PLA_FD}/dl|_{(CQ)^-} < 0$ 이므로 ρ_{PLA_FD} 는 극대값을 가지며, ρ_{PLA_FD} 가 극대값을 갖도록 해주는 l 의 값은 다음의 방정식에서 구할 수 있다.

$$L_o r^{res}(CQ-l)^2 - l F_R(r^{crit})(l-L_o)C^2Q = 0. \quad (3.3.8)$$

(3.3.8)과 (3.3.2)로부터 근사 최적패킷크기 L_{opt}^{app} 와 근사 최대 페이로드 이용도 $\rho_{PLA_FD}^{max}$ 를 각각 구할 수 있다.

참고로 흐름의 요구속도가 균등분포를 가질 경우 근사 최적패킷크기와 최대 페이로드 이용도를 구해보자. 정의에 의해 $F_R(r^{crit})$ 는 다음과 같이 계산된다.

$$F_R(r^{crit}) = \begin{cases} \frac{(r^{crit} - a)}{(b - a)}, & \text{for } r^{crit} < b \\ 1, & \text{otherwise.} \end{cases} \quad (3.3.9)$$

(3.1.3)과 (3.3.9)로부터 예약속도 평균값 r^{res} 는 다음과 같이 구해진다.

$$\begin{aligned} r^{res} &= r^{crit} F_R(r^{crit}) + \frac{b + \min(r^{crit}, b) - \min(r^{crit}, b)}{2} \\ &= \begin{cases} \frac{(r^{crit})^2 - 2r^{crit}a + b^2}{2(b-a)}, & \text{for } r^{crit} < b \\ r^{crit}, & \text{otherwise.} \end{cases} \end{aligned} \quad (3.3.10)$$

$r^{crit} \geq b$ 경우 (3.3.8)은 다음과 같이 표시된다.

$$(L_o + CQ)l - 2L_o CQ = 0. \quad (3.3.11)$$

따라서 근사 최적패킷크기 L_{opt}^{app} 는 $2L_o CQ/(L_o + CQ)$ 가 되고 근사 최대 페이로드 이용도는 다음과 같이 계산된다.

$$\rho_{PLA_SD}^{max} = \frac{l-L_o}{l} \frac{b+a}{2} \frac{CQ-L_o}{2CL_o} \quad (3.3.12)$$

$r^{crit} < r_b$ 경우 (3.3.8)을 전개하면 다음과 같은 l 의 3차방정식이 구해진다.

$$\begin{aligned} &-(2aC + 2aC^2Q + L_oC^2 + 2C^3Q + b^2)l^3 \\ &+ (4aC^2Q + 2aL_oC^2Q + 2aC^3Q^2 + 3L_oC^3Q + 3b^2CQ)l^2 \\ &- (2aC^3Q^2 + 2aL_oC^3Q^2 + 3b^2C^2Q^2)l + \\ &+ b^2C^3Q^3 = 0. \end{aligned} \quad (3.3.13)$$

(3.3.13)으로부터 근사 최적패킷크기를 구할 수 있고, 이를 (3.3.2)에 대입하여 근사 최대 페이로드 이용도를 구할 수 있다.

3.3.2 SD방식

모든 흐름의 패킷크기가 l 이므로 페이로드 이용도 ρ_{PL_SD} 는 다음과 같이 표시된다.

$$\rho_{PL_SD} := \frac{(l-L_o)}{l} \rho_{BW_SD} = \frac{(l-L_o)}{l} \frac{V^{SD} r^{req}}{C} \quad (3.3.14)$$

$l \leq L_o$ 이면 페이로드를 담을 공간이 없으므로 $\rho_{PL_SD} = 0$ 가 되고, $l \geq CQ$ 이면 (3.1.5)와 (3.1.7)로부터 V^{SD} 가 0이 되므로 역시 $\rho_{PL_SD} = 0$ 가 된다. 따라서 l 의 범위는 (L_o, CQ) 이 된다. (3.1.7)의 V^{SD} 는 패킷크기와 r^{req} 에 대한 계단형함수가 되어 미분이 불가능하므로 V_R^{SD} 를 C/r^{req} 로, V_Q^{SD} 를 CQ/l 로 각각 근사화 하자. 그러면 근사 페이로드 이용도 ρ_{PLA_SD} 는 다음과 같이 표시된다.

$$\rho_{PLA_SD} := \frac{l-L_o}{l} \frac{r^{req}}{C} \min\left(\frac{CQ}{l}, \frac{C}{r^{req}}\right) \quad (3.3.15)$$

근사화된 V_R^{SD} 과 V_Q^{SD} 를 함께 하는 패킷크기를 근사 임계 패킷크기라 부르고 L_{crit}^{app} 로 표기하자. $C/r^{req} = CQ/L_{crit}^{app}$ 로부터 L_{crit}^{app} 는 $r^{req}Q$ 로 계산된다. (3.3.15)의 ρ_{PLA_SD} 는 다음과 같이 전개된다.

$$\rho_{PLA_SD} = \begin{cases} \frac{l-L_o}{l} \frac{r^{req}Q}{l}, & \text{for } l \geq L_{crit}^{app} \\ \frac{l-L_o}{l}, & \text{for } l < L_{crit}^{app}. \end{cases} \quad (3.3.16)$$

패킷크기 변화에 따른 페이로드 이용도의 특성을 살펴보기 위해 l 에 대한 ρ_{PLA_SD} 의 1차 도함수를 구하면 다음과 같다.

$$\rho_{PLA_SD}' = \begin{cases} \frac{2L_o-l}{l^3} r^{req}Q, & \text{for } l \geq L_{crit}^{app} \\ \frac{L_o}{l^2}, & \text{for } l < L_{crit}^{app}. \end{cases} \quad (3.3.17)$$

l 의 증가시 ρ_{PLA_SD} 의 특성을 살펴보자. $l < L_{crit}^{app}$ 이면 $\rho_{PLA_SD} > 0$ 이 되나 $l \geq L_{crit}^{app}$ 이면 다음과 같이 L_o 값에 따라 달라진다: $L_o \geq L_{crit}^{app}/2$ 이면 $L_{crit}^{app} \leq l < 2L_o$ 에서 $\rho_{PLA_SD} > 0$ 이 되고 $l = 2L_o$ 에서 $\rho_{PLA_SD} = 0$ 이 된다. 반면 $L_o < L_{crit}^{app}/2$ 이면 $L_{crit}^{app} \leq l < CQ$ 구간에서 $\rho_{PLA_SD} < 0$ 이 된다. 따라서 ρ_{PLA_SD} 를 최대화하는 근사 최적패킷크기 L_{opt}^{app} 는 다음과 같이 된다.

$$L_{opt}^{app} := \max(L_{crit}^{app}, 2L_o) \quad (3.3.18)$$

(3.3.18)를 (3.3.16)에 대입하면 다음과 같이 근사 최대 페이로드 이용도 $\rho_{PLA_SD}^{max}$ 가 계산된다.

$$\rho_{PLA_SD}^{max} := \begin{cases} \frac{r^{req}Q}{4L_o}, & \text{for } L_o \geq \frac{L_{crit}^{app}}{2} \\ \frac{L_{crit}^{app}-L_o}{L_{crit}^{app}}, & \text{for } L_o < \frac{L_{crit}^{app}}{2}. \end{cases} \quad (3.3.19)$$

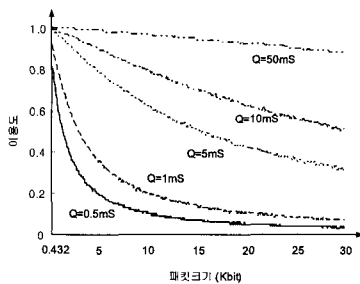
4. 성능 비교평가

성능 비교평가에 사용할 흐름의 요구속도 분포를 살펴보면 다음과 같다: IETF IntServ 모델에서 품질 보장형 서비스는 Tspec으로 자원예약을 요청하는데, Tspec은 토큰버킷(token bucket) (버킷크기와 버킷속도에 의해 정의됨), 피크속도(peak rate) 등으로 명시되며, 피크속도는 버킷속도와 같거나 더 빠른 값을 갖는다[6]. 복잡성을 피하기 위해 본 논문에서는 피크속도를 버킷속도와 같은 값을 갖도록 한다. 그러면 스케줄러에 도착하는 흐름의 요구속도는 바로 버킷속도가 된다. [6]에서 버킷속도를 [1bps, 40Tbps]내 임의속도로 정의하였지만, 본 논문에서는 Kbps급 음성부터 Mbps급 비디오의 보편적인 멀티미디어 흐름을 수용할 수 있는 [5Kbps, 14482Kbps]내에 균등분포하는 것으로 한다. 그리고 서버는 10Gbps의 고속 출력링크를 가지며 L_{OH} 는 RTP 오버헤드인 432비트(54바이트)로 한다. 모든 흐름의 패킷의 크기는 동일한 것으로 가정한다.

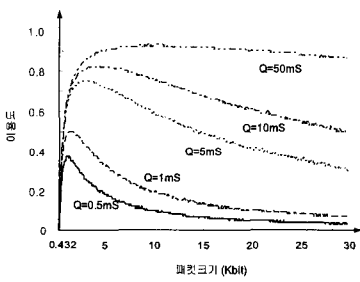
먼저 FD 방식과 SD 방식의 대역폭 및 페이로드 이용도 특성을 수치해석(numerical analysis)으로 살펴보자. (그림 1)과 (그림 2)는 5가지 지연규격에 대해 패킷크기에 따른 FD 방식의 대역폭 이용도와 페이로드 이용도 변화를 보여준다. (그림 1)에서 패킷크기가 클수록, 지연규격이 엄격할수록 대역폭 이용도가 나빠지는데, 이는 임계속도의 증가에 의해 예약손실량이 증가하기 때문이다. (그림 2)에서 페이로드 이용도는 모두 위로 볼록한 커브형태를 갖는데, 이는 패킷크기가 작아지면 패킷의 오버헤드가 상대적으로 증가하여 페이로드 이용도가 저하되고, 반대로 패킷크기가 커지면 (그림 1)과 같이 대역폭 이용도가 낮아져서 역시 페이로드 이용도가 저하되기 때문이다.

SD방식의 대역폭 이용도와 페이로드 이용도는 각각 (그

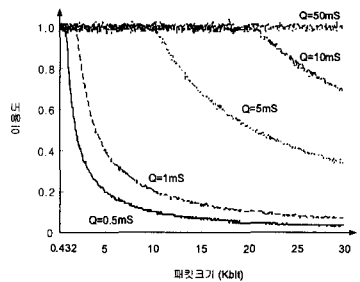
림 3)과 (그림 4)에 도시되어 있다. (그림 3)에서 대역폭 이용도가 급격히 저하되는 현상을 보이는데, (3.1.9)에 의해 다음과 같이 설명된다: 패킷크기가 임계패킷크기보다 작으면 무시할 수 있을 정도로 그 양이 적은 잔존손실이 발생하나 임계패킷크기보다 커지게 되면 흐름손실이 증가하므로 대역폭 이용도가 급격히 저하된다. (그림 4)에서 SD방식의 페이로드 이용도 역시 (그림 2)의 FD방식과 마찬가지로 위로 볼록한 커브형태가 됨을 보여준다. (그림 5)는 패킷크기의 증가에 따른 FD방식과 SD방식의 페이로드 이용도 변화를 직접 비교한 결과를 보여준다. [추론 3.1]에서 살펴본바와 같이 SD 방식이 FD방식보다 항상 우수함을 확인할 수 있다.



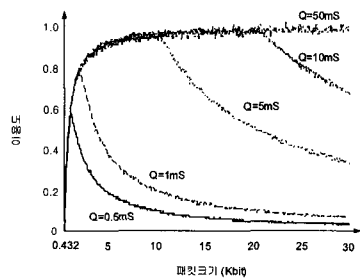
(그림 1) FD 방식의 대역폭 이용도



(그림 2) FD 방식의 페이로드 이용도

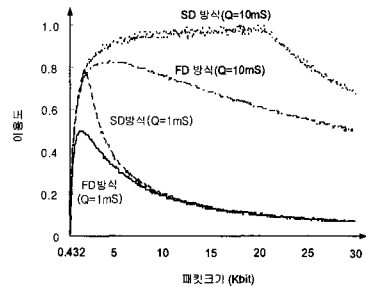


(그림 3) SD 방식의 대역폭 이용도

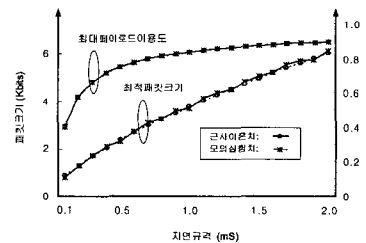


(그림 4) SD 방식의 페이로드 이용도

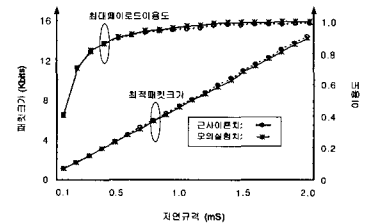
다음은 페이로드 이용도를 극대화 하는 최적 패킷 크기와 그때의 페이로드 이용도, 즉 최대 페이로드 이용도 특성을 비교 평가 해보자. 3절에서 구한 근사이론치와 모의실험 (simulation) 결과를 이용하여 평가한다. 본 논문에 사용된 모의실험은 SMPL(Simulation Model Programming Language) [17]에 공평 패킷 스케줄러 루틴을 추가하여 수행하였다. FD방식과 SD방식에서 지연규격의 변화에 따른 최적 패킷 크기와 최대 페이로드 이용도 특성이 (그림 6)과 (그림 7)에 각각 도시되어 있다. 최대 페이로드 이용도의 경우 근사이론치가 모의실험 결과에 2% 오차 이내로 근접함을 보인다. (그림 8)은 지연규격에 따른 FD방식과 SD방식의 최대 페이로드 이용도를 직접 비교한 결과를 보여준다. (그림 8)로부터 SD방식의 최대 페이로드 이용도가 FD방식의 그것에 비해 10~20% 정도 우수함을 관찰할 수 있다.



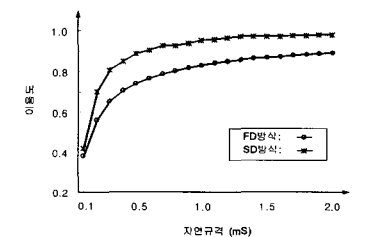
(그림 5) 페이로드 이용도 비교



(그림 6) FD방식에서 최적 패킷크기와 최대 페이로드 이용도



(그림 7) SD방식에서 최적 패킷크기와 최대 페이로드 이용도



(그림 8) 최대 페이로드 이용도 비교

5. 결 론

타임스탬프기반 공평 패킷 스케줄러의 두 가지 설계 방식, 즉 패킷의 가상 종료시각을 사용하는 FD방식과 가상 시작시각을 사용하는 SD방식에 대해 서버자원 이용도 관점에서 성능특성을 비교하였다. 먼저 각 방식에 대해 대역폭 손실 원인을 분석하였는데, FD방식은 예약속도에 의해 결정되는 레이턴시가 요구지연을 만족시키지 못할 경우 발생하는 과잉예약으로 인해 대역폭이 손실되는 반면 SD방식은 흐름수의 제약에 의해 가용 대역폭을 이용할 수 없어서 대역폭이 손실됨을 알았다. 다음은 각 방식에 대해 대역폭 이용도를 분석하였고, SD방식의 대역폭 이용도가 FD방식의 그것에 비해 우수함을 증명하였다. 또한 각 방식에 대해 근사 최적 패킷크기와 근사 최대 페이로드 이용도를 구하였다. 그리고 모의실험으로 두 방식의 성능특성을 비교 평가하였는데, SD방식의 최대 페이로드 이용도가 F D방식의 그것에 비해 10~20% 정도 우수함을 확인하였고, 근사 최대 페이로드 이용도가 2% 오차이내로 실험치에 근접함을 확인하였다.

이러한 성능 비교 결과로부터 IETF IntServ 품질보장 라우터의 설계에 있어 기존의 FD 방식 대신 SD 방식기반 스케줄러를 사용하여 네트워크 자원의 이용도를 높일 필요가 있음을 유추할 수 있다. 그리고 본 논문에서는 스케줄링 서버가 하나의 지연규격을 지원하는 경우를 대상으로 연구하였는데, 실제환경에서는 다수의 지연규격이 요구될 것이므로 향후 다수의 지연규격을 동시에 지원하는 서버에 대한 성능특성 분석이 필요하다.

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, 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.D Dissertation, Curtin University of Technology, Australia, 2002.
 [15] R. Geurin, and V. Peris, "Quality-of-Service in packet networks: Basic mechanisms and directions," Computer Networks, No.31, pp.169~189, 1999.
 [16] J. Xu and R. J. Lipton, "On Fundamental Tradeoffs between Delay Bounds and Computational Complexity in Packet Scheduling Algorithms," Proc. ACM SIGCOMM'2002, Pittsburgh, 2002.
 [17] M. H. MacDougall, "Simulating Computer Systems, Techniques and Tools," MIT Press, 1987.



김 태 준

e-mail : tjkim@kongju.ac.kr
 1980년 경북대학교 전자공학과
 1982년 한국과학기술원 전자공학 석사
 1999년 한국과학기술원 전자공학 박사
 1982년 한국전자통신연구원
 1996년 천안공업대학 교수

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



안 호 범

e-mail : hbahn@kongju.ac.kr
 1992년 단국대학교 전자계산학과(이학사)
 1994년 단국대학교 전산통계 대학원 석사
 (이학석사)
 2002년 단국대학교 전산통계학과 대학원
 박사(이학박사)

1997년~2005년 천안공업대학 정보통신학과 재직
 2005년~현재 공주대학교 정보통신학부 부교수
 관심분야 : Ad-hoc Network, 네트워크 보안, 네트워크 성능평가