상수의 데드라인 계산 비용으로 높은 네트웍 유용도를 얻는 서비스 곡선 할당 방식

(Service Curve Allocation Schemes for High Network Utilization with a Constant Deadline Computation Cost)

편 기 현 † 송 준 화 †† 이 흥 규 †††
(Kihyun Pyun) (Junehwa Song) (Heung-Kyu Lee)

요 약 통합 서비스망은 실시간 응용들에게 고품질의 서비스를 제공하기 위해서 종단간 지연의 한계를 보장해야 한다. 이러한 보장 서비스는 라우터의 출력 포트에 설치되는 실시간 스케줄러에 의해서 제공된다. 그러나 현재까지 연구된 스케줄링 알고리즘들은 네트워크 유용도 혹은 확장성(scalability)에 문제점을 갖고 있다. 여기서 네트워크 유용도는 얼마나 많은 실시간 세션들을 승인할 수 있는 지를 의미한다. 본 논문은 서비스 곡선 알고리즘에서 높은 네트워크 유용도와 확장성 양쪽을 모두 성취할 수 있는 서비스 곡선 할당 방식을 제안한다. 서비스 곡선 알고리즘의 가장 큰 특징은 서비스 곡선 할당 방식에 따라서 네트워크 유용도와 확장성 모두가 결정된다는 점이다. 일상적인 믿음과 반대로, 데드라인을 계산할 때 전체 서비스 곡선이 아닌 일부만이 사용됨을 증명한다. 이 사실로부터 우리는 데드라인을 계산하는 비용이 상수시간인 서비스 곡선 할당 방식을 제안한다. 또한, 수치결과를 통해서 제안하는 방식이 mutirate 알고리즘을 포함한 GPS 알고리즘들보다 더 높은 네트워크 유용도를 성취함을 보인다. 우리가 아는 한, 서비스 곡선 알고리즘이 제안하는 서비스 곡선 할당 방식을 채용하면 동일한 확장성을 갖는 스케줄링 알고리즘들 중에 가장 높은 네트워크 유용도를 성취한다.

키워드 : 실시간 서비스, 보장 서비스, 패킷 스케줄링

Abstract Integrated services networks should guarantee end-to-end delay bounds for real-time applications to provide high quality services. A real-time scheduler is installed on all the output ports to provide such guaranteed service. However, scheduling algorithms studied so far have problems with either network utilization or scalability. Here, network utilization indicates how many real-time sessions can be admitted. In this paper, we propose service curve allocation schemes that result in both high network utilization and scalability in a service curve algorithm.In service curve algorithms, an adopted service curve allocation scheme determines both network utilization and scalability. Contrary to the common belief, we have proved that only a part of a service curve is used to compute deadlines, not the entire curve. From this fact, we propose service curve allocation schemes that result in a constant time for computing deadlines. We show through a simulation study that our proposed schemes can achieve better network utilizations than Generalized Processor Sharing (GPS) algorithms including the multirate algorithm. To our knowledge, the service curve algorithm adopting our schemes can achieve the widest network utilization among existing scheduling algorithms that have the same scalability.

Key words : Real-time service, guaranteed service, packet scheduling

---

· 본 연구는 한국과학기술원 인공위성연구센터 기관고유 사업의 연구비 지원을 받았음

† 비 회 원 : 한국과학기술원 전기및전자공학 연구원
khpyun@comis.kaist.ac.kr
†† 정 회 원 : 한국과학기술원 전산학과 교수
junehwa@nclab.kaist.ac.kr
††† 종신회원 : 한국과학기술원 전산학과 교수
hklee@casaturn.kaist.ac.kr

논문접수 : 2003년 1월 16일
심사완료 : 2003년 3월 26일

1. 서 론

통합 서비스망은 실시간 응용들에게 고품질의 서비스를 제공하기 위해서 네트워크을 통과할 때 종단간 지연의 한계를 보장해야 한다[1]. 이러한 보장 서비스는 라우터의 출력 포트에 설치되는 실시간 스케쥴러에 의해서 제공된다. 실시간 스케쥴러를 선택할 때 중요한 선택 기준은 네트워크 유용도이다. 여기서 네트워크 유용도는 얼마나 많은 실시간 세션들을 승인할 수 있는 지를 의미한다. 또한, 동시에 많은 세션들을 지원할 수 있는 확장성(scalability)이 중요하다.

현재까지 연구된 스케쥴러들은 네트워크 유용도 혹은 확장성에 있어서 문제점을 안고 있다. 높은 네트워크 유용도를 성취하기 위한 한 중요한 조건은 주어진 트래픽 기술(traffic specification)에 대해서 서로 다른 지연의 한계를 보장할 수 있는 스케쥴러의 능력이다[2]. 또한, 확장성 있는 스케쥴러 구현을 위해서는 스케쥴링 복잡도가 세션 수에 의존하지 않아야 하고 세션 승인 절차가 너무 복잡하지 않아야 한다. 학계에서 연구된 지연의 한계를 보장할 수 있는 스케쥴링 알고리즘은 세 가지 경향, 즉, GPS(Generalized Processor Sharing) 알고리즘 [3,4,5,6,7], RC(Rate-Controlled) 알고리즘 [8,9,10], 그리고 서비스 곡선(Service Curve) 알고리즘 [11,12, 13,14]으로 나눌 수 있다. GPS 알고리즘에서 세션 승인 절차가 복잡하지 않게 하는 경우 지연에 한계는 기술된 트래픽의 데이타율에 의해서 결정된다[4]. 따라서, 성취할 수 있는 네트워크 유용도는 매우 낮다[10]. RC 알고리즘들 중에서 RC-EDF 알고리즘이 가장 높은 네트워크 유용도를 성취할 수 있다. 그러나, RC-EDF 알고리즘은 각 세션에게 레귤러이터를 할당해야 하므로 확장성에 문제점을 안고 있다[2,10]. 게다가 RC-EDF 알고리즘이 GPS 알고리즘보다 엄격히 더 높은 네트워크 유용도를 성취하기 위해서는 네트워크 전체를 고려해야 하므로 세션 승인 절차가 복잡하다[10]. 마지막으로 서비스 곡선 알고리즘의 경우 네트워크 유용도와 확장성 양쪽 모두 각 세션에 할당되는 서비스 곡선에 따라 결정된다. 만일 서비스 곡선 알고리즘이 [14]에서 제안된 서비스 곡선 할당 방식을 사용하면, 레귤레이터를 필요로 하지 않기 때문에 RC-EDF 알고리즘보다 더 좋은 확장성을 가지면서도 RC-EDF 알고리즘과 동일한 네트워크 유용도를 성취할 수 있다. 그러나, 그 할당 방식은 RC-EDF 알고리즘에 기반하고 있으므로[14], 높은 네트워크 유용도를 성취하기 위해서는 세션 승인 절차가 복잡한 문제점을 갖는다.

본 논문은 서비스 곡선 알고리즘에서 높은 네트워크 유용도와 확장성 양쪽을 모두 성취할 수 있는 서비스 곡선 할당 방식을 제안한다. 서비스 곡선 알고리즘에서 패킷의 데드라인은 각 세션에 할당된 서비스 곡선으로부터 계산되며 가장 이른 데드라인을 갖는 패킷 순으로 전송된다. 일상적인 믿음과 반대로, 우리는 데드라인을 계산할 때 전체 서비스 곡선이 아닌 일부만이 사용됨을 증명한다. 이 사실로부터 데드라인을 계산하는 비용이 상수 시간인 서비스 곡선 할당 방식을 제안한다. 또한, 수치결과를 통해서 제안하는 방식이 mutirate 알고리즘을 포함한 GPS 알고리즘들보다 더 높은 네트워크 유용도를 성취함을 보인다. 우리가 아는 한, 서비스 곡선 알고리즘이 제안하는 서비스 곡선 할당 방식을 채용하면 동일한 확장성을 갖는 스케쥴링 알고리즘들 중에 가장 높은 네트워크 유용도를 성취한다.

이 논문의 구성은 다음과 같다. 2 절은 서비스 곡선 알고리즘을 이해하는데 도움이 되는 배경을 설명한다. 3 절에서는 [14]에서 제안된 서비스 곡선 알고리즘의 수정판을 제시한다. 4 절은 서비스 곡선 알고리즘이 높은 네트워크 유용도를 성취할 수 있는 서비스 곡선 할당 방식을 제안한다. 5 절은 제안하는 서비스 곡선 할당 방식을 사용했을 때 데드라인 계산 복잡도가 상수 시간임을 보인다. 6 절은 수치 결과를 통해서 제안하는 방식을 채용하는 서비스 곡선 알고리즘이 다른 알고리즘들에 비해서 더 높은 네트워크 유용도를 가짐을 보인다. 마지막으로 7 절은 이 논문을 매듭짓는다.

2. 배 경

이 절은 [14]에서 제시한 서비스 곡선의 정의와 알려진 몇 가지 분석 결과를 다시 살펴본다. 또한 본 논문에서 고려하는 네트워크 모델링과 트래픽 특성화(traffic characterization) 방식도 설명한다.

2.1 네트워크 모델링과 트래픽 기술

네트워크는 연속된 라우터들(혹은 서버들)로 모델링된다. 논의의 편의를 위해서 전송선은 무시한다. 임의의 시간 구간 $(s, t]$ 동안 세션 i로부터 라우터에 도착한 패킷의 양과 출발한 패킷의 양을 각각 $R_i^{in}(s, t)$와 $R_i^{out}(s, t)$로 표기한다. 만일 $s \geq t$인 경우 $R_i^{in}(s, t) = R_i^{out}(s, t) = 0$으로 정의한다. 표기의 편의성을 위해 $R_i^{in}(0, t)$와 $R_i^{out}(0, t)$를 각각 $r_i^{in}(t)$와 $r_i^{out}(t)$로 나타낸다. 여러 개의 라우터가 있을 때 다른 라우터들을 구분하기 위해서 적절히 위첨자를 붙여서 표기를 달리한

다. 예를 들면, $R_i^{out,m}(s,t)$와 $r_i^{out,m}(t)$는 각각 세션 i가 통과하는 m번째 라우터로부터의 출력양을 의미한다. 만일 라우터에 전송할 세션 i의 패킷이 존재하는 경우 세션 i는 저장 기간(backlogged period) 내에 있다고 말한다.

각 실시간 세션 i는 함수 $b_i(\cdot)$로 자신의 트래픽을 특성화한다. $b_i(t)$는 길이가 t인 임의의 시간 구간 동안에 세션 i로부터 허용되는 최대 입력 트래픽 양을 의미한다. 따라서, $b_i(\cdot)$는 감소하지 않는 함수이어야 하고, 모든 $u\le 0$에 대해서 $b_i(u)=0$으로 정의한다. 만일 임의의 시간 구간 $(s,t]$ 동안 세션 i로부터 네트워크로 입력되는 데이타양이 $b_i(t-s)$를 넘지 않을 때, 세션 i가 b_i-*smooth* 하다고 말하거나 혹은 세션 i의 트래픽 한계 함수가 $b_i(\cdot)$이다라고 말한다.

함수 b_i의 한 특정한 경우로, 만일 $b_i(t)=\min_{k=1\cdots K}\{\sigma_i^k+\rho_i^k t\}$인 경우 우리는 세션 i가 $K(\sigma,\rho)$-*smooth*하다고 말하거나 혹은 세션 i의 트래픽 한계 함수가 $K(\sigma,\rho)$이다라고 말한다. $K(\sigma,\rho)$ 함수는 VBR (Variable-Bit-Rate) 비디오의 트래픽 변화율을 잘 특성화할 수 있다[15,7]. 특히 IETF Int-Serv 워킹 그룹에 의해서는 현재 $K(\sigma,\rho)$ 함수의 두 개의 특별한 경우를 받아들이고 있다[1,16]. 그 하나는 세션 i가 (σ,ρ)-*smooth*한 경우이다. 여기서 σ는 세션 i에 허용된 균집성(burstiness)을 나타낸다. ρ는 평균 트래픽율이다. 나머지 하나는 세션 i가 트래픽 한계 함수가 $\min\{\sigma_i+\rho_i t, P_i t\}$인 경우이다. 여기서 P_i는 세션 i에게 순간적으로 허용된 최대 데이타율(peak rate)이다.

2.2 서비스 곡선과 몇 가지 분석 결과들

모든 $u\le 0$에 대해서 $S_i(u)=0$인 비감소 함수 $S_i(\cdot)$를 고려하자. 세션 i의 임의의 패킷이 라우터를 떠나는 패킷 출발 시간 t에 대해서 다음의 조건을 만족시키는 시간 s가 존재할 때 우리는 이 라우터가 세션 i에게 서비스 곡선 $S_i(\cdot)$를 보장한다고 정의한다.

$$r_i^{out}(t)\ge r_i^{in}(s)+S_i(t-s) \tag{1}$$

식 (1)에서 한 가지 중요한 점은 시간 s가 시간 t 이전의 세션 i 저장 기간(backlogged period)들의 시작 시점들 중 하나라는 것이다. $S_i(t)$는 세션 i 저장 기간들 중 어떤 시작 시점으로부터 길이가 t인 시간 구간 동안에, 세션 i를 위해 그 라우터에 의해서 전송이 보장되는 최소한의 데이타양을 의미한다. 만일 임의의 시간 t보다 크지 않은 세션 i의 저장 기간들의 시작 시점들의 집합을 $B_i(t)$로 표기한다면, 식 (1)을 동격으로 다음과 같이 표현할 수 있다.

$$r_i^{out}(t)\ge \min_{s\in B_i(t)}\{r_i^{in}(s)+S_i(t-s)\} \tag{2}$$

다음의 두 정리는 보장 서비스 곡선에 적용할 수 있는 결과들이다. 이들의 증명은 [17]에서 찾을 수 있다.

정리 1 (네트워크 서비스 곡선 정리) 세션 i가 M개의 라우터들을 통과한다고 가정하자. m번째 라우터가 세션 i에게 서비스 곡선 $S_i^m(\cdot)$을 보장한다고 가정하자. 그러면, 이 전체 라우터들은 세션 i에게 다음의 서비스 곡선 $S_i^{net}(\cdot)$을 보장한다.

$$S_i^{net}(t)=\min\{\sum_{m=1}^{M}S_i^m(\Delta_m):\Delta_m>0 \quad \text{and} \quad \sum_{m=1}^{M}\Delta_m=t\} \tag{3}$$

정리 2 (지연 및 버퍼량 한계 정리) 세션 i의 트래픽 한계 함수가 $b_i(\cdot)$라고 하자. 최대 패킷 크기를 $l^{\max}$로 표기하자. 어떤 라우터가 세션 i에게 서비스 곡선 $S_i(\cdot)$를 보장한다고 가정하자. 그러면, 세션 i의 패킷은 그 라우터에서 결코 다음의 값보다 크지 않은 지연을 겪는다.

$$\max_{k,\Delta>0}\ \min\{\Delta:\Delta>0 \quad \text{and} \quad b_i(k)\le S_i(k+\Delta)\} \tag{4}$$

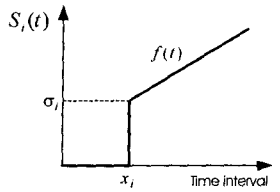
그리고, 어떤 순간에도 그 라우터에서의 세션 i의 버퍼량은 결코 다음의 값보다 크지 않다.

$$\max_{k,k>0}\{b_i(k)-S_i(k)\}+l^{\max} \tag{5}$$

3. 수정된 서비스 곡선 알고리즘

본 절에서 제안하는 알고리즘은 [14]에서 제시된 서비스 곡선 알고리즘을 패킷의 데드라인 계산이 분명하도록 수정한 것이다(수정된 부분은 나중에 자세히 설명한다). 2.2 절에서는 라우터가 세션에게 보장하는 서비스 곡선을 알고 있다면, 지연과 버퍼량 한계를 추출할 수 있다는 것을 보였다. 서비스 곡선 알고리즘은 이 사실을 역으로 이용한 것이다. 다시 말해, 세션에게 보장할 지연의 한계로부터 그 세션이 필요로 하는 보장 서비스 곡선을 찾을 수 있고, 서비스 곡선 알고리즘은 그 곡선을 보장한다.

서비스 곡선 알고리즘은 각 세션에게 서비스 곡선을 할당한다. 세션으로부터 입력되는 각 패킷은 그 할당된 서비스 곡선으로부터 계산된 데드라인이 붙여진다. 서비스 곡선 알고리즘은 가장 이른 데드라인을 갖는 패킷 순으로 비선점(non-preemptive) 방식으로 전송한다. 같은 데드라인을 갖는 패킷이 여러 개인 경우 그 순서는 임의로 결정한다.



$$S_i(t) = \begin{cases} 0, & 0 \le t < x_i \\ f(t), & otherwise \end{cases}$$

그림 1 예-계단 함수인 서비스 곡선 $S_i(\cdot)$. $f(t)$는 시간점 x_i^-에서 영이고 x_i에서 σ_i 값을 갖는 비감소 함수이다.

이 절 전체에 걸쳐서 다음의 표기법을 사용한다. 세션 I로부터 k번째 패킷을 p_i^k로 표기한다. 패킷 p_i^k가 라우터에 도착하는 시간, 출발하는 시간, 그리고 그 패킷 길이를 각각 a_i^k, d_i^k, 그리고 l_i^k로 표기한다. 또, 세션 i의 m번째 저장 기간의 시작점은 b_i^m으로 표기한다.

서비스 곡선 함수에서 데드라인 계산법을 알아 보자. 각 패킷의 데드라인은, 만일 모든 패킷이 그들의 데드라인까지 전송이 완료된다면 각 세션에게 할당된 서비스 곡선이 보장되도록 붙여진다. 패킷 p_i^k가 세션 i의 m번째 저장 기간 중에 도착한다고 가정하자. 서비스 곡선 $S_i(\cdot)$가 세션 i에게 할당되었다고 가정하고 그 패킷의 데드라인을 $D_i^{m,k}$로 표기하자. $D_i^{m,k}$는 다음과 같이 결정된다.

$$D_i^{m,k} = \min\{d \mid \min_{s \in B_i(b_i^m)} \{r_i^{in}(s) + S_i(d-s)\} \ge \sum_{j=1}^{k} l_i^j\} \quad (6)$$

([14]에서 패킷 p_i^k의 데드라인 $D_i^{m,k}$는 식 (6)과 달리 등호가 사용되지 않는다. 그러나, 수학적으로 그러한 데드라인이 존재한다 할지라도 우리는 그 정확한 값을 기술할 수 없다. 우리는 등호를 포함시키기 위해서 식 (8)에서 제시된 데드라인 곡선 유지 범위, 정리 4의 승인 제어, 정리 2의 지연의 한계, 그리고 식 (10)의 보장 서비스 곡선 추출을 수정하였다.)

식 (6)에서 부등식이 사용된 이유는 할당된 서비스 곡선이 계단 함수인 경우를 다루기 위해서이다. 그림 1은 계단 함수의 예를 보여 준다. 그림에서 서비스 곡선 $S_i(\cdot)$는 시간점 x_i 이전에는 영의 값을 가지고, 시점에 영보다 훨씬 큰 σ_i 값으로 뛰어 오른다. 만일 이 서비스 곡선 $S_i(\cdot)$가 세션 i에게 할당되고 패킷 p_i^k가 첫 번째 저장 기간 중에 도착하는 경우에 그 데드라인이 어떻게 할당되는 지를 살펴 보자. 만일 패킷 p_i^k를 포함한 입력된 축적량이 σ_i보다 작다면, (즉, $\sum_{j=1}^{k} l_i^j < \sigma_i$) 식 (6)으로부터 패킷 p_i^k의 데드라인은 x_i가 된다. 여기서 만일 식 (6)에서 등식만이 사용되었다면 이 경우 데드라인을 결정하는데 어려움을 갖는 다는 것을 주목해야 한다.

식 (6)의 데드라인을 계산하기 위해서 서비스 곡선 알고리즘은 각 세션 i에 대해서 다음과 같이 정의되는 데드라인 곡선 $D_i(\cdot)$를 유지한다.

$$D_i(t) = \min_{s \in B_i(t)} \{r_i^{in}(s) + S_i(t-s)\} \quad (7)$$

$D_i(\cdot)$는 세션 i가 처음으로 저장(backlogged)되었을 때 $S_i(\cdot)$로 초기화된다. 그 뒤를 이어 세션 i가 새롭게 저장될 때마다 아래와 같이 $D_i(\cdot)$가 갱신된다.

$$D_i(b_i^m; t) = \min\{D_i(b_i^{m-1}; t), r_i^{in}(b_i^m) + S_i(t - b_i^m)\}$$
$$\text{for} \quad t \ge b_i^m + x_i \quad (8)$$

식 (8)에서 x_i는 서비스 곡선 $S_i(\cdot)$가 처음으로 영이 아닌 값을 갖는 시점을 나타낸다. 그림 1은 할당된 서비스 곡선이 계단함수인 경우 x_i 값을 나타낸다. 여기서 $D_i(b_i^m; t)$가 오직 $t \ge b_i^m + x_i$에 대해서만 유지된다는 점을 주목해야 한다. 그 이유는 아래 정리가 말하듯이 데드라인 계산을 위해서는 이 영역에 대해서만 데드라인 곡선을 갱신하는 것이 충분하기 때문이다. 만일 $D_i(b_i^m; t)$를 $t \ge b_i^m$에 대해서 갱신하면, 이 영역에 대해서 데드라인 곡선 유지 비용이 불필요하게 높은 복잡도를 야기하는 문제점을 갖는다[14].

정리 3 세션 i에게 할당된 서비스 곡선이 $S_i(\cdot)$라고 가정하자. 서비스 곡선 $S_i(\cdot)$가 처음으로 영이 아닌 값을 갖는 첫 번째 시점을 x_i로 표기하자. 세션 i의 마지막 저장 기간의 시작점 b_i^m이후에 도착하는 세션 i의 패킷들을 고려하자. 그러면, 이 패킷들의 데드라인은 항상 $b_i^m + x_i$ 보다 크다.

증명. 그 시점 b_i^m 이후에 도착하는 임의의 패킷 p_i^k를 고려하자. 그러면,

$$\begin{aligned}
D_i^{m,k} &= \min\{t \mid D_i(b_i^m; t) \ge \sum_{j=1}^{k} l_i^j\} \\
&= \min\{t \mid \min\{D_i(b_i^{m-1}; t), r_i^{in}(b_i^m) + S_i(t - b_i^m)\} \ge \sum_{j=1}^{k} l_i^j\} \\
&= \max\{\min\{t \mid D_i(b_i^{m-1}; t) \ge \sum_{j=1}^{k} l_i^j\}, \\
&\qquad \min\{t \mid r_i^{in}(b_i^m) + S_i(t - b_i^m) \ge \sum_{j=1}^{k} l_i^j\}\} \\
&\ge \min\{t \mid r_i^{in}(b_i^m) + S_i(t - b_i^m) \ge \sum_{j=1}^{k} l_i^j\} \\
&\ge \min\{t \mid S_i(t - b_i^m) > 0\} \quad (\sum_{j=1}^{k} l_i^j > r_i^{in}(b_i^m) \text{이므로}) \\
&\ge b_i^m + x_i
\end{aligned}$$

□

다음 정리는, 만일 세션들에게 할당된 서비스 곡선의

합이 대역폭을 초과하지 않는다면, 각 패킷의 지연은 한계 값을 가짐을 말한다. 또한, 정리 4는 승인 제어(admission control)에 사용될 수 있다. 이 정리의 증명은 [17]에서 제시된 것을 바탕으로 쉽게 유추할 수 있으므로 생략한다. ([17]에서 제시된 증명은 모든 패킷은 자신의 데드라인에다가 l^{max}/r을 더한 시간이전(등호는 포함안함)까지 전송완료됨을 증명하였다. 그러나, 우리의 정리는 등호를 포함시켜도 됨을 나타낸다.)

정리 4 대역폭이 r bps인 서버에서 N개의 세션을 처리하는 서비스 곡선 알고리즘을 고려하자. 각 세션 i에게 할당된 서비스 곡선을 $S_i(\cdot)$라고 가정하자. 최대 패킷 크기를 l^{max}로 표기하자. 만일 다음의 조건을 만족하면 모든 패킷들은 그것들의 데드라인에다가 l^{max}/r를 더한 시간까지는 모두 전송이 완료된다.

$$\sum_{i=1}^{N} S_i(t) \le rt \tag{9}$$

서비스 곡선 알고리즘이 세션 i에게 서비스 곡선 $S_i(\cdot)$를 할당할 때 서비스 곡선 알고리즘이 다음의 $\hat{S}_i(\cdot)$를 보장한다.

$$\hat{S}_i(t) = \begin{cases} 0, & t < l^{max}/r \\ S_i(t - l^{max}/r), & otherwise \end{cases} \tag{10}$$

식 (10)은 만일 세션 i가 보장 서비스 곡선으로 $S_i(\cdot)$를 필요로 한다면 서비스 곡선 알고리즘이 $S_i(\cdot)$를 왼쪽으로 l^{max}/r 만큼 이동시킨 곡선을 할당하면 됨을 의미한다. 이 사실의 증명은 [17]에서 제시된 것을 바탕으로 쉽게 유추됨으로 생략한다.

4. 서비스 곡선 할당 방식

이 절에서는 높은 네트워크 유용도를 성취할 수 있도록 세션에 서비스 곡선을 할당하는 법을 제안한다. 세션이 통과하는 각 라우터에게 동일한 종단간 지연의 한계를 보장하는 여러 가지 서비스 곡선을 할당 방식들이 존재한다. 그러나, 그 방식들은 서로 다른 네트워크 유용도와 스케쥴링 복잡도를 갖는 다는 점에 주목해야 한다.

4.1 지연 분배 방식

지연 분배 방식은 세션의 종단간 지연 요구를 각 라우터에게 균등하게 분배하고, 각 라우터에 존재하는 서비스 곡선 알고리즘은 세션의 트래픽 기술 함수와 그 지역 지연 요구로부터 서비스 곡선을 구성한다. 고려하는 라우터의 대역폭을 r bps라고 가정하자. 최대 패킷 크기를 l^{max}로 표기하자. 이 라우터에서 세션 i의 트래픽 한계 함수와 지역 지연 요구가 각각 $b_i(\cdot)$와 $(d_i + l^{max}/r)$라고 가정하자. 이 경우 세션 i는 다음과 같은 서비스 곡선 $S_i(\cdot)$가 할당된다.

$$S_i(t) = \begin{cases} 0, & 0 \le t < d_i \\ b_i(t - d_i), & t \ge d_i \end{cases} \tag{11}$$

2.1 절에서 말했듯이 $b_i(\cdot)$는 $K(\sigma,\rho)$ 함수, 즉, $b_i(t) = \min_{k=1\cdots K} \{\sigma_i^k + \rho_i^k t\}$이다.

만일 종단간 지연을 각 라우터에게 분배하는 경우로 국한시키면, 서비스 곡선 알고리즘과 다른 알고리즘과의 네트워크 유용도 비교는 전체 네트워크 대신 단일 서버에서의 비교로 지역화할 수 있다. 이제 단일 서버로 한정시킨 경우 서비스 곡선 알고리즘이 지연 분배 방식을 사용할 때 가장 높은 네트워크 유용도를 가짐을 보이자. [18]에서 RC-EDF 알고리즘이 단일 서버의 경우에 있어서 최고의 네트워크 유용도를 가짐을 증명하였다. 따라서, RC-EDF에 의해서 지연 보장이 승인된 세션들의 집합은 지연 분배 방식을 사용하는 서비스 곡선 알고리즘도 역시 승인됨을 보인다. 만일 RC-EDF에 의해서 N개의 세션들이 승인되었다면 다음 식이 만족되어야 한다[19].

$$\sum_{i=1}^{N} b_i(t - d_i) + l^{max} \le rt \text{ for } t \ge \min_{i=1\cdots N} \{d_i\} \tag{12}$$

위의 식에서 $b_i(\cdot)$는 세션 i의 레귤레이터의 출력 함수를 의미한다. 단일 서버로 한정하는 경우에 각 레귤레이터 함수는 트래픽 한계 함수와 동일하다[10]. 그러나, 서버들의 네트워크인 경우에 있어서는 레귤레이터 함수가 트래픽 한계 함수와 같지 않을 수 있음을 주목해야 한다[10]. 식 (12)는 다음과 같이 줄여진다.

$$\sum_{i=1}^{N} b_i(t - d_i) \le rt \text{ for } t \ge \min_{i=1\cdots N} \{d_i\} \tag{13}$$

이제 지연 분배 방식에 의해서 구성된 서비스 곡선 $S_i(\cdot)$가 세션 i에게 할당되었다고 가정하자. 식 (11)로부터 $S_i(\cdot) = b_i(t - d_i)$ for $t \ge d_i$이기 때문에, 식 (13)으로부터 다음 식이 성립한다.

$$\sum_{i=1}^{N} S_i(t) \le rt \text{ for } t \ge \min_{i=1\cdots N} \{d_i\} \tag{14}$$

또한 $S_i(t) = 0$ for $t < d_i$ 이므로 다음 식이 성립한다.

$$\sum_{i=1}^{N} S_i(t) \le rt \quad \text{for } t < \min_{i=1\cdots N} \{d_i\} \tag{15}$$

식 (14)와 (15)로부터 길이가 t인 임의의 시간 구간에 대해서 $\sum_{i=1}^{N} S_i(t) \le rt$를 만족한다. 그러므로, 정리 4에 의해서 그 N개의 세션들은 지연 분배 방식을 사용하는 서비스 곡선 알고리즘에 의해서 승인된다.

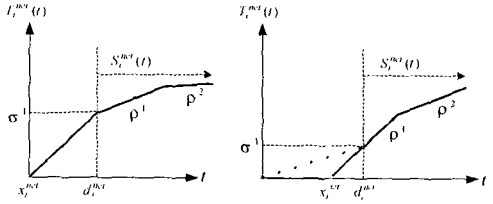
지연 분배 방식을 채용하는 서비스 곡선 알고리즘은 [7]에서 제안된 multirate 알고리즘보다 엄격히 더 높은 네트워크 유용도를 성취할 수 있다. Multirate 알고리즘은 시간구간에 대해서 여러 개의 서비스율을 제공하는 알고리즘으로 [20,3]에서 제안된 WFQ 알고리즘이 확장된 것이다. [7]의 저자들은 multirate 알고리즘이 WFQ보다 더 높은 네트워크 유용도를 성취할 수 있음을 보였다. multirate 알고리즘은 세션의 종단간 지연을 그 세션이 통과하는 각 라우터의 지역 지연 한계로 분배한다. 서비스 곡선 알고리즘이 지연 분배 방식을 사용할 때 단일 서버에서 최고이므로 multirate 알고리즘보다 더 높은 네트워크 유용도를 갖는다. 즉, multirate 알고리즘에 의해서 승인된 세션들의 집합은 모두 그 서비스 곡선 알고리즘에 의해서 승인되지만, 그 역은 성립하지 않는다. 그 주된 이유는 multirate 알고리즘은 동일한 트래픽 한계 함수를 가지는 세션들에게 동일한 지연의 한계만을 제공하기 때문이다.

4.2 네트워크 서비스 곡선 분배 방식

서비스 곡선 알고리즘이 지연 분배 방식을 사용할 때 단일 서버에서는 최고의 네트워크 유용도를 성취할 수 있지만, 서버들의 네트워크에서 항상 높은 네트워크 유용도를 성취할 수 있는 것은 아니다. 서버들의 네트워크인 경우에 높은 네트워크 유용도를 성취하기 위해서는 네트워크 대역폭, 즉, 경로상이 있는 라우터들의 전체 대역폭을 모두 고려해서 최적화해야 한다. 지연 분배 방식은 오직 하나의 라우터의 대역폭만을 최적화한다는 점에 주목해야 한다.

우리는 서버들의 네트워크인 경우에 대해서 높은 네트워크 유용도를 성취하기 위한 네트워크 서비스 곡선 분배 방식, 줄여서 ND(Network service curve Distribution) 방식을 제안한다. 우리는 세션 I가 m개의 서버들을 통과할 때 ND 방식에 의해 어떻게 각 라우터가 그 세션 i에 대한 서비스 곡선을 할당하는 지를 보이는 데 초점을 맞춘다. 우리는 m번째 서버에서의 대역폭을 r^m으로 표기한다. 세션 i의 트래픽 한계 함수가 $b_i(\cdot) = \min_{k=1\cdots K}\{\sigma_i^k+\rho_i^k\}$이고 필요로 하는 종단간 지연의 한계가 $\hat{d}_i^{net}$ 이라고 하자. $l^{\max}$가 최대 패킷 크기라고 할 때 ($\hat{d}_i^{net} - \sum_{m=1}^{M} l^{\max}/r^m$)을 d_i^{net}으로 표기하자. d_i^{net}은 각 서버에서 비선점 효과를 무시했을 때 종단간 지연의 한계를 의미한다. 제안하는 서비스 곡선 할당 절차는 다음과 같이 세 단계로 설명될 수 있다.

첫 번째 단계는 네트워크가 세션 i에게 할당해야 할



(a) $\sigma_i^1/d_i^{net} > \rho_i^1$ 인 경우 (b) 그렇지 않은 경우

그림 2 목적 서비스 곡선 $T_i^{net}(\cdot)$ 구성

최소 네트워크 서비스 곡선을 추출한다. 우선 정리 2로부터 네트워크가 보장해야 할 최소 네트워크 서비스 곡선 $\hat{S}_i^{net}(\cdot)$ 은 다음과 같다.

$$\hat{S}_i^{net}(t) = \begin{cases} 0, & 0 \le t < \hat{d}_i^{net} \\ b_i(t-\hat{d}_i^{net}), & otherwise \end{cases} \quad (16)$$

따라서, 네트워크가 세션 i에게 $\hat{S}_i^{net}(\cdot)$을 보장하기 위해서는 다음의 서비스 곡선 $S_i^{net}(\cdot)$을 할당해야 한다.

$$S_i^{net}(t) = \begin{cases} 0, & 0 \le t < d_i^{net} \\ b_i(t-d_i^{net}), & otherwise \end{cases} \quad (17)$$

두번째 단계는 네트워크가 할당해야 할 최소 서비스 곡선 $S_i^{net}(\cdot)$에 기반해서 목적 네트워크 서비스 곡선 $T_i^{net}(\cdot)$을 추출한다. 그림 2는 $S_i^{net}(\cdot)$로부터 $T_i^{net}(\cdot)$를 구성하는 법을 보여준다.

$T_i^{net}(\cdot)$은 영이 아닌 값을 갖는 영역, 즉, 그 그림의 경우 $t \ge x_i^{net}$에 대해서 오목한 선형 곡선(concave piecewise linear curve) 임에 주목해야 한다. 서비스 곡선 $S_i(t)$는 만일 임의의 두 시간 t_1과 t_2에 대해서 그리고 0과 1 사이의 값을 갖는 임의의 α에 대해서 $S_i(\alpha t_1+(1-\alpha)t_2) \ge \alpha S_i(t_1)+(1-\alpha)S_i(t_2)$인 경우 오목하다고 정의한다. 그렇지 않으면 $S_i(t)$는 볼록(convex)하다고 정의한다. 네트워크 서비스 곡선 할당 방식에 의해서 구성된 목적 네트워크 서비스 곡선은 오직 영보다 큰 영역에 대해서만 오목하며 원점으로부터는 그렇지 않다는 점을 주목해야 한다. 우리는 $T_i^{net}(\cdot)$이 영이 아닌 값을 갖는 첫번째 시간점을 x_i^{net}으로 표기한다. 좀 더 구체적으로 나타내면 $T_i^{net}(t)$은 $t \ge d_i^{net}$인 경우에 $S_i^{net}(t)$과 동일하다. 그러나, $t < d_i^{net}$인 경우 $T_i^{net}(\cdot)$은 d_i^{net}과 σ_i^1의 값에 따라서 두 가지 서로 다른 방식으로 구성된다. 그 하나는 $\sigma_i^1/d_i^{net} > \rho_i^1$인 경우이

다. 이 경우 $T_i^{net}(\cdot)$은 d_i^{net} 시점까지 σ_i^1/d_i^{net} 기울기를 가지고, x_i^{net}의 값은 영이다. 그렇지 않은 경우를 고려해 보자. 이 경우 $T_i^{net}(\cdot)$은 d_i^{net} 시점까지 가장 급한 기울기인 ρ_i^1이 사용되고, x_i^{net}의 값은 $(d_i^{net}-\sigma_i^1/d_i^{net})$이다. 다시 말해, 만일 $t<x_i^{net}$이라면 $T_i^{net}(t)=0$이고 $x_i^{net}\le t<d_i^{net}$이라면 $T_i^{net}(t) = \sigma_i^1+\rho_i^1(t-d_i^{net})$이다.

마지막 단계는 $T_i^{net}(\cdot)$를 사용해서 각 라우터들에게 서비스 곡선을 할당한다. $1\le m\le M$일 때 x_i^{net}을 $\sum_{m=1}^{M} x_i^m = x_i^{net}$을 만족하는 어떤 값이라고 가정하자. 가장 간단한 방법은 x_i^{net} 값을 골고루 분배하도록 x_i^{net}을 모두 x_i^{net}/M으로 설정하는 것이다. 그러면, m번째 라우터는 $T_i^{net}(\cdot)$을 왼쪽으로 $(x_i^{net}-x_i^m)$만큼 이동시킨 곡선을 세션 i에게 할당한다. 다시 말해, m번째 라우터는 세션 i에게 다음의 서비스 곡선 $S_i^m(\cdot)$을 할당한다.

$$S_i^m(t)=\begin{cases}0, & 0\le t<x_i^m\\ T_i^{net}(t+x_i^{net}-x_i^m), & otherwise\end{cases} \qquad (18)$$

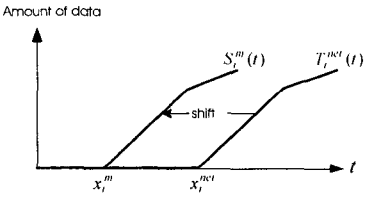


그림 3 m번째 서버에서 x_i^m과 $T_i^{net}(\cdot)$이 주어졌을 때 세션 i에게 할당된 서비스 곡선 $S_i^m(\cdot)$

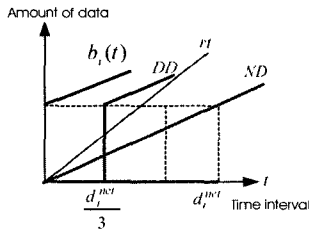


그림 4 예-지연 분배 방식과 네트워크 서비스 곡선 할당 방식에 의해서 세션 i에게 할당된 서비스 곡선. 각각 DD와 ND로 표기. $b_i(\cdot)$와 d_i^{net}은 각각 세션 i의 트래픽 한계 함수와 비선점 효과를 무시한 종단간 지연의 한계를 나타냄. 대역폭은 r bps

그림 3은 주어진 x_i^m과 $T_i^{net}(\cdot)$으로부터 $S_i^m(\cdot)$를 구성하는 법을 보여준다. 우리는 서비스 곡선 알고리즘이 ND 방식을 따랐을 때 그 할당된 서비스 곡선들을 사용해서 전체 네트워크가 보장하는 서비스 곡선을 추출함으로써 각 세션이 필요로 하는 종단간 지연의 한계가 보장됨을 쉽게 검증할 수 있다.

우리는 간단한 예제를 통해서 ND 방식이 네트워크인 경우에 지연 분배 방식보다 더 낫게 되는 직감을 주고자 한다. 그림 4는 세션 i가 세 개의 라우터들을 통과할 때 각 서버에서 지연 분배 방식과 ND 방식에 의해서 할당되는 서비스 곡선을 보여준다. 정리 4로부터 ND 방식에 의해서 할당된 서비스 곡선은 승인되는 반면, 지연 분배 방식에 의해서 할당된 서비스 곡선은 거절됨을 주목해야 한다.

5. 구현 복잡도

만일 서비스 곡선이 아무런 제약없이 할당된다면 데드라인 계산이 높은 복잡도를 필요로 한다. 그러나, 만일 서비스 곡선이 제안한 방식대로 할당된다면 데드라인 계산시 단지 상수 시간만을 필요로 함을 보인다. 그러면, 정렬된 우선 순위 큐를 사용하는 다른 알고리즘들, 예들 들면 WFQ 알고리즘과 비교하면, 제안한 서비스 곡선 할당 방식을 채용하는 서비스 곡선 알고리즘은 동일한 스케쥴링 복잡도를 가지면서 더 나은 네트워크 유용도를 가진다.

서비스 곡선 알고리즘의 스케쥴링 복잡도는 두 개의 부분, 즉, 입력되는 패킷의 데드라인을 계산하는 부분과 그 패킷들을 전송하는 부분으로 구성된다. 패킷 전송을 위해서는 정렬된 우선 순위 큐가 유지된다. 이 유지 비용은 N이 세션 수를 나타낼 때 $O(\log N)$ 복잡도이다. 이 비용은 시퀀서(sequencer)[21] 혹은 시스톨릭 어레이(systolic array)[22]와 같은 특별 하드웨어를 사용하는 경우 $O(1)$으로 줄어든다. 여기서 주목할 점은 어떤 서비스 곡선이 사용된다 할지라도 이 패킷 전송 비용은 변화하지 않는다는 것이다.

그러나, 데드라인 계산 복잡도는 할당된 서비스 곡선에 의존한다. 서비스 곡선 알고리즘이 데드라인을 계산하기 위해서는 각 세션에 데드라인 곡선을 유지하고 패킷의 데드라인은 그 데드라인 곡선의 역함수로부터 구한다.

다음 정리는 세션 i의 데드라인 곡선 $D_i(\cdot)$를 갱신하는 법을 보인다. 이 정리에서 한가지 주목할 점은 $S_i(\cdot)$는 지연 분배 방식 혹은 ND 방식으로 세션 i에

게 할당된 서비스 곡선이라는 것이다.

정리 5 세션 i에게 할당된 서비스 곡선 $S_i(\cdot)$가 다음과 같다고 하자.

$$S_i(t)=\begin{cases}0, & 0\le t< d_i\\ \min\limits_{k=1\cdots K}\{\sigma_i^k+\rho_i^k(t-d_i)\}, & otherwise\end{cases} \quad (19)$$

그러면, 세션 i의 데드라인 곡선은 새롭게 저장될 때마다 다음과 같이 K 조각의 선형 곡선으로 갱신된다.

$$D_i(b_i^m;t)= \min_{k=1\cdots K}\{C_i^{m,k}+\rho_i^k t\} \text{ for } t\ge b_i^m+d_i$$

이고

$$C_i^{m,k}=\begin{cases}\sigma_i^k-\rho_i^k(b_i^1+d_i), & m=1\\ \min\{C_i^{m-1,k},\ \sigma_i^k-\rho_i^k(b_i^m+d_i)+r_i^{in}(b_i^m)\}, & m\ge 2\end{cases} \quad (20)$$

증명. 이 정리를 세션 i의 저장된 횟수에 대한 구조적 귀납법을 통해서 증명한다. 기본 단계로, $t\ge b_i^1+d_i$인 경우,

$$\begin{aligned}D_i(b_i^1;t)&= S_i(t-b_i^1)\\ &= \min_{k=1\cdots K}\{\sigma_i^k+\rho_i^k(t-b_i^1-d_i)\}\\ &= \min_{k=1\cdots K}\{C_i^{1,k}+\rho_i^k t\}\end{aligned}$$

귀납 가정에 의해서, $t\ge b_i^m+d_i$에 대해서 다음이 성립한다고 가정하자.

$$D_i(b_i^m;t)= \min_{k=1\cdots K}\{C_i^{m,k}+\rho_i^k t\} \quad (21)$$

그러면, 유추 단계에서, $t\ge b_i^{m+1}+d_i$에 대해서,

$$\begin{aligned}D_i(b_i^{m+1};t)&= \min\{D_i(b_i^m;t), S_i(t-b_i^{m+1})+r_i^{in}(b_i^{m+1})\}\\ &= \min\{\min_{k=1\cdots K}\{C_i^{m,k}+\rho_i^k t\},\\ &\quad \min_{k=1\cdots K}\{\sigma_i^k+\rho_i^k(t-b_i^{m+1}-d_i)\}+r_i^{in}(b_i^{m+1})\}\\ &= \min_{k=1\cdots K}\{\min\{C_i^{m,k},\sigma_i^k-\rho_i^k(b_i^{m+1}+d_i)\}\\ &\quad +r_i^{in}(b_i^{m+1})\}+\rho_i^k t\}\\ &= \min_{k=1\cdots K}\{C_i^{m,k}+\rho_i^k t\}\end{aligned}$$

□

정리 5로부터 각 $C_i^{m,k}$는 O(1) 복잡도로 계산되므로 $D_i(b_i^m;\cdot)$은 O(K) 복잡도로 계산된다. 대부분 K는 작은 상수로 제한되므로 데드라인 갱신 복잡도를 상수시간으로 간주할 수 있다. 또, 갱신된 데드라인이 오목한 선형 함수이므로 각 패킷의 데드라인도 상수 시간 이내에 계산될 수 있다.

6. 수치 결과

이 절은 ND 방식을 채용하는 서비스 곡선 알고리즘이 높은 네트워크 용도를 성취할 수 있음을 보인다. 우리는 네 개의 스케쥴링 알고리즘, 즉, 지연 분배 방식을 사용하는 서비스 곡선 알고리즘, ND 방식을 사용하는 서비스 곡선 알고리즘, WFQ (혹은 PGPS라 불리는) 알고리즘, 그리고 multirate 알고리즘을 고려한다. 네트워크 유용도를 비교하기 위해서 다른 수의 라우터와 다양한 종단간 지연 요구에 대해서 각 알고리즘이 승인할 수 있는 쥬라기 공원 비디오 수를 측정하였다. (다른 비디오에 대한 실험 결과는 이와 비슷하기 때문에 생략하였다.) 이 실험에 사용된 쥬라기 공원 비디오 트레이스(trace)는 ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG에서 찾을 수 있으며 min {5924 + 220 Kt, 9461 + 211 Kt} 함수로 특성화될 수 있다. 모든 서버들은 100 *Mbps*의 대역폭을 갖고 최대 패킷 크기가 1500 바이트라고 가정하였다. 단, multirate 알고리즘은 53 바이트의 고정된 크기만을 사용하는 ATM 망에서 연구되었으므로, 이 경우 모든 패킷의 크기는 53 바이트라고 가정하였다.

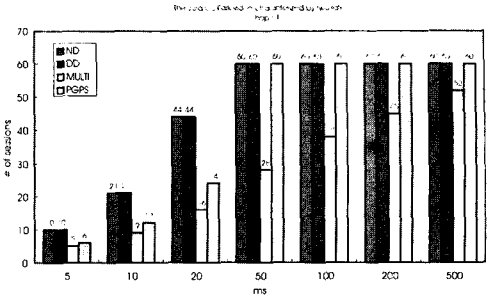


그림 5 라우터 수가 한 개일 때

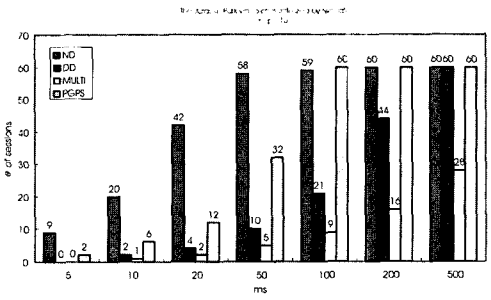


그림 6 라우터 수가 열 개일 때

그림 5와 그림 6은 각각 라우터 수가 한개 일 때와 열 개일 때의 실험 결과를 나타낸다. 그림에서, 라우터 수가 하나 일 때 지연 분배 방식은 항상 다른 것들보다 더 나은 네트워크 유용도를 갖는다. 그러나, 라우터 수가 증가하고 지연 요구가 짧아질수록 ND 방식이 다른 알고리즘들보다 훨씬 높은 네트워크 유용도를 갖는다. 따라서, ND 방식의 장점은 온라인 실시간 응용, 예를

들면, 화상 전화기와 화상 회의의 경우에 명확해 진다. 이러한 응용들은 대개 매우 짧은 종단간 지연의 한계를 요구한다. 주문형 비디오(video-on-demand)와 같이 저장 응용들의 경우는 큰 종단간 지연을 인내 할 수도 있다. 그러나, ND 방식은 이러한 저장 응용들의 경우에 있어서도 이득이 있다. 이 응용들이 요구하는 종단간 지연은 크다고 할지라도 WAN(wide-area-network)에서는 여러 개의 라우터들을 거치므로 각 라우터에 요구되는 지연의 한계는 짧기 때문이다.

7. 맺음말

본 논문은 동일한 확장성을 갖는 알고리즘들 중에서 우리가 아는 한 가장 높은 네트워크 유용도를 성취할 수 있는 서비스 곡선 알고리즘과 두 개의 서비스 곡선 할당 방식인 지연 분배 방식과 네트워크 서비스 곡선 할당 방식을 제안하였다. 지연 분배 방식은 단일 서버의 경우에 있어서 최고인 반면 네트워크 서비스 곡선 방식은 네트워크인 경우에 있어서 더 나은 네트워크 유용도를 성취할 수 있다. 두 방식은 모두 데드라인을 계산할 때 상수시간만을 필요로 한다. 제안하는 서비스 곡선 알고리즘이 제안한 서비스 곡선 할당 방식을 사용하면, WFQ이나 multirate 알고리즘에 비해서 동일한 구현 복잡도를 가지면서 더 높은 네트워크 유용도를 성취하는 장점을 가진다.

참 고 문 헌

[1] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service, September 1997. RFC 2212.

[2] H. Zhang. Service Disciplines for Guaranteed Performance Service in Packet Switching Networks. *Proc. IEEE*, 3(4):391 430, 1995.

[3] A. K. J. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Service Networks: The Single Node Case. *IEEE/ACM Tran. Networking*, 1(3):344 357, June 1993.

[4] A. K. J. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Service Networks: The Multiple Node Case. *IEEE ACM Tran. Networking*, 2(2):137 150, April 1994.

[5] J.C.R. Bennett and H. Zhang. Hierarchical Packet Fair Queueing Algorithms. *IEEE ACM Tran. Networking*, 5(5):675 689, October 1997.

[6] D. Stiliadis and A. Varma. Rate Proportional Servers: A Design Methodology for Fair Queueing Algorithms. *IEEE/ACM Tran. Networking*, 6(2): 164 174, April 1998.

[7] Debanjan Saha, Sarit Mukherjee, and Satish K. Tripathi. Multirate Scheduling of VBR Video Traffic in ATM Networks. *IEEE JSAC*, 15(6):1132 1147, August 1997.

[8] H. Zhang and D. Ferrari. Rate Controlled Service Disciplines. *Journal of High Speed Networks*, 3(4):389 412, 1994.

[9] D. Ferrari and D. Verma. A Scheme for Real Time Channel Establishment in Wide-Area Networks. *IEEE JSAC*, 8(4):368 379, April 1990.

[10] Leonidas Georgiadis, Roch Guerin, Vinod Peris, and Kumar N. Sivarajan. Efficient Network QoS Provisioning Based on per Node Traffic Shaping. *IEEE/ACM Tran. Networking*, 4(4):482-501, August 1996.

[11] R. L. Cruz. Quality of Service Guarantees in Virtual Circuit Switched Networks. *IEEE JSAC*, 13(6):1048-1056, 1995.

[12] Ion Stoica, Hui Zhang, and T. S. Eugene Ng. A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services. *IEEE/ACM Tran. Networking*, 8(2):185 199, 2000.

[13] Hanrijanto Sariowan, Rene L. Cruz, and George C. Polyzos. SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service. *IEEE/ACM Tran. Networking*, 7(5):669-684, October 1999.

[14] Kihyun Pyun, Junehwa Song, and Heung Kyu Lee. The Service Curve Service Discipline with the Service Curve Discipline for the Rate-Controlled EDF service discipline in Variable-Sized Packet Networks. In *IEEE International Conference on Communications*, pages 1135 1141, 2002.

[15] Dallas E. Wrege, Edward W. Knightly, Hui Zhang, and Jorg Liebeherr. Deterministic Delay Bounds for VBR Video in Packet Switching Networks: Fundamental Limits and Practical Trade Off. *IEEE/ACM Tran. Networking*, 4(3): 352 362, June 1996.

[16] J. Wroclawski. Specification of the Controlled Load Network Element Service, September 1997. RFC 2211.

[17] Kihyun Pyun, Junehwa Song, and Heung Kyu Lee. The Service Curve Service Discipline with the Service Curve for the Rate Controlled EDF service discipline in Variable Sized Packet Networks. Technical Report 01 11 002, Advanced Information Technology Reserch Center (AITrc), 2001.

[18] Leonidas Georgiadis, Roch Guerin, and Abhay Parekh. Optimal Multiplexing on a Single Link: Delay and Buffer Requirements. *IEEE Trans. Information Theory*, 43(5):1518-1535, September 1997.

[19] Victor Firoiu, Jim Kurose, and Don Towsley. Efficient Admission Control of Piecewise Linear Traffic Envelopes at EDF Schedulers. *IEEE/ACM Tran. Networking*, 6(5):558-570, October 1998.

[20] A. Demers, S. Kesha, and S. Shenkeer. Analysis and Simulation of a Fair Queueing Algorithm. In *ACM SIGCOMM*, pages 3:12, 1989.

[21] Massoud R. Hashemi and Alberto Leon-Garcia. The Single-Queue Switch: A Building Block for Switches with Programmable Scheduling. *IEEE JSAC*, 15(5):785-794, June 1997.

[22] P. Lavoie and Y. Savaria. A systolic architecture for fast stack sequential decoders. *IEEE Tran. Communications*, 42(2/3/4):324-335, Feb./Mar./Apr. 1994.

편 기 현

1995년 인하대학교 전자계산공학과(학사) 1997년 KAIST 전산학과(석사). 2002년 KAIST 전산학과(박사). 2002년~현재 KAIST 전기및전자공학, 박사후 연구원(Post Doctor). 관심분야는 유무선 고품질 서비스, 패킷 스케쥴링, 자원 관리

송 준 화

1988년 서울대학교 계산통계학과 학사 1990년 State University of NY, Computer Science 석사. 1997년 University of Maryland, Computer Science 박사. 1994년~1996년 IBM T.J. Watson Research Center, S/W Engineer, Digital Library Group. 1996년~1997년 IBM T.J. Watson Research Center, S/W Engineer, Digital Media Solutions Dept. 1997년~2000년 IBM T.J. Watson Research Center, Research Staff Member, Parallel Commercial Systems Dept. 2000년~현재 한국과학기술원 전자전산학과 조교수. 관심분야는 전자상거래, 웹서비스, 그리드 컴퓨팅

이 홍 규

1978년 서울대학교 전자공학과(학사) 1981년 KAIST 전산학과(석사). 1984년 KAIST 전산학과(박사). 1984년~1986년 U. of Michigan, research scientist 1986년~현재까지 KAIST 전산학과 교수. 1989년~1997년 인공위성연구센터 연구기획실장. 관심분야는 디지털워터마킹, 영상처리, 실시간컴퓨팅