

# 패킷 네트워크에서 지연과 버퍼 크기 한계를 추출할 수 있는 분석 도구

(An Analysis Tool for Deriving Bounds on Delay and  
Buffer Size in Packet Networks)

편 기 현 <sup>†</sup> 송 준 화 <sup>\*\*</sup> 이 흥 규 <sup>\*\*\*</sup>

(Kihyun Pyun) (Junehwa Song) (Heung-Kyu Lee)

**요약** 본 논문은 가변 패킷을 사용하는 일반적인 망에서 실시간 세션이 이종의 여러 스케줄러들을 통과할 때 네트워크가 보장하는 지연의 한계를 추출하는 범용 도구를 제안한다. 이 도구는 이질적인 각 스케줄러를 동질적인 서비스 곡선 서버로 특성화한다. 우리는 서비스 곡선 서버가 LR 서버보다 엄격히 큰 범주의 스케줄러들을 특성화할 수 있음을 보인다. 즉, 모든 LR 서버는 서비스 곡선 서버로 특성화할 수 있고, 또한 RC-EDF와 SC 알고리즘 조차도 서비스 곡선 서버로 정확히 특성화할 수 있음을 보인다. 이와 같이 서비스 곡선 서버는 LR 서버에 비해서 범용성 관점에서 큰 장점이 있으면서도 분석 결과가 정확하다. 우리는 LR 서버들로 분석할 수 있는 네트워크의 경우도 서비스 곡선 서버로 분석했을 때, 두 경우 모두 동일한 지연의 한계를 추출함을 수학적으로 증명한다.

**키워드** : 실시간 서비스, 보장 서비스, 패킷 스케줄링

**Abstract** In this paper, we propose a general analysis tool that derives guaranteed delay bounds for real-time sessions when those sessions pass through heterogeneous schedulers in general packet networks. This tool characterizes each different scheduler by a homogeneous service curve server. We show that service curve servers can characterize a strictly larger class of schedulers than LR servers. That is, we can characterize not only all LR servers but also RC-EDF and SC scheduler by service curve servers. Even with such advantage over LR servers, service curve servers provide accurate analysis results. We prove mathematically that if we analyze a network that can be analyzed by both LR servers and service curve servers, both cases yield the same delay bound.

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

## 1. 서론

통합 서비스 망에서 실시간 세션들에게 고품질의 서비스를 제공하기 위해서는 네트워크를 통과할 때 발생하는 종단간 지연의 한계를 보장해야 한다[1]. 이 보장 서비스는 각 라우터가 실시간 패킷 스케줄링 알고리즘을 채용할 때 제공될 수 있다. 많은 실시간 패킷 스케줄링 알고리즘들이 네트워크를 통과하는 각 세션에게 지

연의 한계를 보장할 수 있음을 보였지만, 대개의 알고리즘들은 모든 라우터에 자신의 알고리즘을 동일하게 사용하는 경우를 가정하고 있다[2,3]. 그러나, 통합 서비스 망에서는 지연의 한계를 보장할 수 있는 서로 다른 특성을 갖는 이종의 스케줄링 알고리즘들이 공존할 것이다. 이와 같은 환경에서는 실시간 세션이 이종의 여러 스케줄러들을 통과할 때 네트워크가 보장할 수 있는 지연의 한계를 추출하는 분석 도구가 필요하다. 특히 이런 도구가 유용하게 사용되기 위해서는, 가변 크기의 패킷을 사용하는 일반적인 망에서 적용할 수 있으면서 동시에 광범위한 범주의 스케줄러를 포함해서 분석할 수 있는 범용성이 중요하다.

현재까지 제안된 도구들은 범용성 결여의 문제점을 갖고 있다. [4]에서 제안된 LR 서버는 가변 크기를 갖는 일반적인 패킷망에서 각각의 라우터에 서로 다른 스케줄링 알고리즘을 사용되는 경우 실시간 세션에게 보장할

· 본 연구는 한국과학기술원 인공위성연구센터 기관고유 사업의 연구비 지원을 받았음

† 비 회 원 : 한국과학기술원 전기및전자공학 연구원  
khpyn@comis.kaist.ac.kr

\*\* 정 회 원 : 한국과학기술원 전자전산학과 교수  
junehwa@nclab.kaist.ac.kr

\*\*\* 종신회원 : 한국과학기술원 전자전산학과 교수  
hklee@rtlab.kaist.ac.kr

논문접수 : 2002년 12월 20일

심사완료 : 2003년 4월 21일

수 있는 지연의 한계를 추출할 수 있다. 그러나,  $LR$  서버가 특성화할 수 있는 스케줄러의 범주는 WFQ[5], VirtualClock[6,7], DRR[8] 등과 같이 하나의 서비스에 근거한 알고리즘에 국한된다. 따라서, RC-EDF[9], multirate[10], SCED[11], 그리고 SC[12]와 같이 시간 구간에 따라서 여러 서비스를 제공하는 일반적인 스케줄러들을 정확하게 특성화할 수 없다. 만일 이 스케줄러들을  $LR$  서버로 특성화시킨다면 분석된 지연의 한계는 실제 보장하는 것보다 훨씬 큰 값을 추출할 것이다. 또, [11,13]에서 제안된 방식은 큰 범주의 스케줄러들을 포함할 수 있지만, 특정 망에서만 적용할 수 있는 것이 문제다. 이 도구는 적용될 수 있는 망은 다음의 제한성을 갖는다. 모든 패킷이 동일한 크기를 갖고, 하나의 패킷을 전송하는데 걸리는 시간을 단위 시간(unit time)으로 설정한 이산 시간(discrete time)을 사용하며, 패킷 도착과 출발이 동기화(synchronization)되어 있다.

우리는 이 논문에서 가변 패킷을 사용하는 일반적인 망에서 실시간 세션이 이종의 여러 스케줄러들을 통과할 때 네트워크가 보장하는 지연의 한계를 추출하는 범용 분석 도구를 제안한다. 이 도구는 [11,13]에서 제안된 방식을 가변 패킷 망에서 적용될 수 있도록 일반화한 것으로, 이질적인 각 스케줄러를 동질적인 서비스 곡선 서버로 특성화한다. 또한, 전송선 혹은 지연 요소와 같은 네트워크 요소들도 마찬가지로 서비스 곡선 서버로 특성화한다. 각 서비스 곡선 서버는  $LR$  서버에서 두 개의 인자로 특성화한 것과 달리, 보장 서비스 곡선(guaranteed service curve)이라 불리는 하나의 함수로 특성화한다. 보장 서비스 곡선은 네트워크 요소의 특성에 따라 지연과 여러 서비스를 모두 통합해서 표현할 수 있다. 우리는 서비스 곡선 서버가  $LR$  서버보다 엄격히 큰 범주의 스케줄러들을 특성화할 수 있음을 보인다. 즉, 모든  $LR$  서버는 서비스 곡선 서버로 특성화할 수 있고, 또한 RC-EDF와 SC 알고리즘조차도 서비스 곡선 서버로 정확히 특성화할 수 있음을 보인다. 이와 같이 서비스 곡선 서버는  $LR$  서버에 비해서 범용성 관점에서 큰 장점이 있으면서도 분석 결과가 정확하다. 우리는  $LR$  서버들로 분석할 수 있는 네트워크의 경우도 서비스 곡선 서버로 분석했을 때, 두 경우 모두 동일한 지연의 한계를 추출함을 수학적으로 증명한다.

이 논문의 구성은 다음과 같다. 2절에서는 우리가 고려하는 트래픽 특성화를 설명한다. 3절에서는 서비스 곡선 서버와 비교하기 위해서 [4]에서 제안된  $LR$  서버를 다시본다. 4절에서는 서비스 곡선 서버를 정의하고 분석할 때 사용되는 몇 가지 중요한 결과를 제시한다. 5절에서는 각 네트워크 요소들을 서비스 곡선 서버로 특성화하는 방법을 보인다. 6절에서는 서비스 곡선 서버를 사

용한 분석 결과와  $LR$  서버를 사용한 결과를 비교한다. 마지막으로, 7절은 이 논문의 결론을 맺는다.

## 2. 트래픽 특성화

우리는 임의의 시간 구간  $(s, t)$  동안 세션  $i$ 로부터 서버에 받은 패킷의 양과 전송한 패킷의 양을 각각  $A_i(s, t)$ 와  $W_i(s, t)$ 로 나타낸다.  $A_i(s, t)$ 는 패킷의 마지막 비트가 서버에 도착했을 때 증가한다. 비슷하게  $W_i(s, t)$ 는 패킷의 마지막 비트가 서버를 떠났을 때 증가한다. 또 표기의 편의성을 위해  $A_i(0, t)$ 와  $W_i(0, t)$ 를 각각  $A_i(t)$ 와  $W_i(t)$ 로 나타낸다.

각 실시간 세션  $i$ 는 함수  $b_i(\cdot)$ 로 자신의 트래픽을 특성화한다.  $b_i(t)$ 는 길이가  $t$ 인 임의의 시간 구간 동안 세션  $i$ 로부터 허용되는 최대 입력 트래픽 양을 의미한다. 따라서,  $b_i(\cdot)$ 는 감소하지 않는 함수이어야 하고, 우리는 모든  $u \leq 0$ 에 대해서  $b_i(u) = 0$ 으로 정의한다. 우리는 만일 임의의 시간 구간  $(s, t)$  동안 세션  $i$ 로부터 네트워크로 입력되는 데이터 양이  $b_i(t-s)$ 를 넘지 않을 때, 세션  $i$ 가  $b_i$ -smooth 하다고 말하거나 혹은 세션  $i$ 의 트래픽 한계 함수가  $b_i(\cdot)$ 이다라고 말한다.

함수  $b_i$ 의 한 특정한 경우로, 만일  $b_i(t) = \min_{k=1 \dots K} \{\sigma_i^k + \rho_i^k t\}$ 인 경우 우리는 세션  $i$ 가  $K(\sigma, \rho)$ -smooth 하다고 말하거나 혹은 세션  $i$ 의 한계 함수가  $K(\sigma, \rho)$ 이다라고 말한다.  $K(\sigma, \rho)$  함수는 VBR (Variable-Bit-Rate) 비디오의 트래픽 변화율을 잘 특성화할 수 있다[10,14]. 특히 IETF Int-Serv 워킹 그룹에 의해서는 현재  $K(\sigma, \rho)$  함수의 두 개의 특별한 함수를 받아들이고 있다[1]. 그 하나는 세션  $i$ 가  $(\sigma, \rho)$ -smooth한 경우이다. 여기서  $\sigma$ 는 세션  $i$ 에 허용된 균집성(burstiness)을 나타낸다.  $\rho$ 는 평균 데이터율이다. 나머지 하나는 세션  $i$ 가 트래픽 한계 함수  $\min\{\sigma_i + \rho_i t, P_i t\}$ 를 사용하는 경우이다. 여기서  $P_i$ 는 세션  $i$ 에게 순간적으로 허용된 최대 데이터율(peak rate)이다.

## 3. LR 서버 다시보기

이 장에서는  $LR$  서버의 정의를 다시보고, 나중에 서비스 곡선 서버의 분석 결과와 비교하기 위해서 [4]에서 제시된 몇 가지 분석 결과를 보인다.

$LR$  서버를 정의하기 위해서는 세션  $i$ 의 바쁜 기간(busy period)에 대한 설명이 먼저 필요하다. 세션  $i$ 에 할당된 서비스율이  $\rho_i$ 라고 가정하자. 세션  $i$ 의 버퍼가 비어있을 때 새로운 패킷이 도착하기 바로 이전 시간을  $\tau_i$ 이라고 표기하자. (즉,  $\tau_i$  시간에는 버퍼가 비어있고,

$\tau_1^+$  시간에는 그렇지 않다.) 또,  $\tau_1$  이후의 어떤 시간  $t$ 에 대해서, 시간 구간  $(\tau_1, t]$  동안에 도착한 세션  $i$ 의 축적된 패킷의 양, 다시 말해  $A_i(\tau_1, t)$ 가  $\rho_i(t - \tau_1)$  보다 크거나 같다고 가정하자. 그러면, 이 조건을 만족하는 가장 큰  $t$ 를  $\tau_2$ 라고 표기할 때  $(\tau_1, \tau_2]$ 를 세션  $i$ 의 바쁜 기간이라고 정의 한다.

LR 서버는 세션의 바쁜 기간에 보장 받는 서비스양에 기반해서 정의된다. 세션  $i$ 의  $j$ 번째 바쁜 기간이  $\tau$  시간에 시작한다고 가정하자. 시간 구간  $(\tau, t]$  동안에 서버  $S$ 에 의해서 세션  $i$ 의 버퍼로부터 전송된 데이터양을  $W_{i,j}^S(\tau, t)$ 로 표기하자. 그  $j$ 번째 바쁜 기간 동안에 세션  $i$ 의 버퍼로부터 마지막 비트가 떠난 시간을  $\tau^*$ 로 표기하자. 그러면, 서버  $S$ 가 LR 서버이기 위한 필요 충분 조건은 시간 구간  $(\tau, \tau^*]$  내에 속하는 모든 시간  $t$ 에 대해서 아래의 조건을 만족하는 어떤 음이 아닌 상수  $\theta_i^S$ 가 존재하는 것이다.

$$W_{i,j}^S(\tau, t) \geq \max(0, \rho_i(t - \tau - \theta_i^S)). \quad (1)$$

$\theta_i^S$ 는 세션  $i$ 의 각 바쁜 기간을 발생시키는 첫 번째 패킷에게 발생할 수 있는 최악의 지연을 의미하며, 이 LR 서버의 잠복 시간(latency)으로 정의된다. 또,  $\rho_i$ 는 LR 서버의 할당율(rate)로 정의된다.

다음의 정리는 LR 서버들로 구성된 네트워크에서 세션의 지연의 한계를 추출하는 방법을 보여준다[4].

**정리 1** 세션  $i$ 가  $M$ 개의 연속된 LR 서버들을 통과한다고 가정하자.  $j$ 번째 LR 서버의 세션  $i$ 에 대한 할당율과 잠복 시간이 각각  $g_j$ 와  $\theta_j^i$ 라고 하자. 만일 세션  $i$ 가  $(\sigma_i, \rho_i)$ -smooth하고  $g_j \geq \rho_i$ 인 경우 세션  $i$ 가 이  $M$ 개의 서버들을 통과할 때 걸릴 수 있는 최악의 지연  $D_i$ 는 다음과 같다.

$$D_i \leq \frac{\sigma_i}{g_i} + \sum_{j=1}^M \theta_j^i. \quad (2)$$

그렇지 않고 만일 세션  $i$ 의 트래픽 한계 함수가  $\min\{\sigma_i + \rho_i t, P_i t\}$ 인 경우, 세션  $i$ 가 이  $M$ 개의 서버들을 통과할 때 걸릴 수 있는 최악의 지연  $D_i$ 는 다음과 같다.

$$D_i \leq \left( \frac{P_i - g_i}{P_i - \rho_i} \right) \left( \frac{\sigma_i}{g_i} \right) + \sum_{j=1}^M \theta_j^i. \quad (3)$$

다음의 정리는 LR 서버들로 구성된 네트워크에서 세션 버퍼에 쌓일 수 있는 패킷양의 한계를 추출하는 방법을 보여준다[4].

**정리 2** 세션  $i$ 가  $(\sigma_i, \rho_i)$ -smooth하고,  $M$ 개의 연속된 LR 서버들을 통과한다고 가정하자.  $j$ 번째 LR 서버의 세션  $i$ 에 대한 할당율과 잠복 시간이 각각  $g_j$ ,

$g_j \geq \rho_i$ 와  $\theta_j^i$ 라고 할 때,  $M$ 번째 LR 서버에서 세션  $i$  버퍼에 쌓일 수 있는 최대 데이터양  $Q_i^m$ 는 다음과 같다.

$$Q_i^m \leq \sigma_i + \rho_i \sum_{j=1}^M \theta_j^i. \quad (4)$$

#### 4. 서비스 곡선 서버

서비스 곡선 서버는 각 세션  $i$ 에 대해서 비감소 함수  $S_i(\cdot)$ 로 특성화한다. 함수  $S_i(\cdot)$ 는 모든  $u \leq 0$ 에 대해서  $S_i(u) = 0$ 이다. 세션  $i$ 의 모든 패킷 출발 시간  $t$ 에 대해서 다음의 조건을 만족시키는 시간  $s$ 가 존재할 때 우리는 서버  $S$ 가 세션  $i$ 에게 서비스 곡선  $S_i(\cdot)$ 을 보장하는 서비스 곡선 서버라고 정의한다.

$$W_i(t) \geq A_i(s) + S_i(t - s). \quad (5)$$

식 (5)에서 한 가지 중요한 점은 시간  $s$ 가 시간  $t$  이전의 세션  $i$  저장 기간 (backlogged period)들의 시작 시점들 중 하나라는 것이다.<sup>1)</sup>  $S_i(t)$ 는 세션  $i$  저장 기간들 중 어떤 시작 시점으로부터 시간 구간  $t$  동안에, 세션  $i$ 를 위해 그 서버에 의해서 전송이 보장되는 최소한의 데이터양을 의미한다.

##### 4.1 서비스 곡선 서버 네트워크에서의 지연과 버퍼양 한계 분석

이 절에서는 서비스 곡선 서버들로 구성된 네트워크를 마치 하나의 서비스 곡선 서버로 특성화할 수 있음을 보이고, 주어진 서비스 곡선 서버에서 어떻게 지연과 버퍼 크기의 한계를 추출할 수 있는 지를 보인다. 이 결과들을 바탕으로 각 세션에게 보장되는 종단간 지연의 한계를 분석할 수 있다. 이 결과들은 [12]에서 특정 스케줄러에 대해서 도출된 것들을 이종의 스케줄러를 분석할 수 있는 일반적인 도구의 관점에서 재구성한 것이다.

세션  $i$ 가 통과하는 연속된 서비스 곡선 서버들을 고려하자. 이 서버들의 각각이 보장하는 서비스 곡선을 알고 있을 때 이 전체 서버들이 보장하는 서비스 곡선을 찾아낼 수 있다. 다시 말하면, 여러 개의 서버들은 그 찾아낸 서비스 곡선을 보장하는 하나의 서비스 곡선 서버로 간주될 수 있다. 이 방법은 정리 3에 제시되었다. 직관적으로, 이 정리는 세션  $i$ 에게 보장되는 네트워크

1) 우리의 정의는 [15]에 제시된 것과 동일하다. 그러나, [11]에서 제시된 것과는 다소 비슷한 형태이지만 다르다. [11]에서 제시된 것은 고정 크기의 패킷만을 사용하는 특정 네트워크에서만 적용할 수 있다. [11]에서 사용된 정의도 식 (5)를 동일하게 사용하지만, 시간  $t$ 가 패킷 출발 시간이 아니라 모든 시간을 의미하고 또한 시간  $s$ 가  $t$ 보다 작은 임의의 시간이다. 시간  $s$ 와  $t$ 의 조건이 두 정의 내에서 서로 다르기 때문에 어느 한 쪽의 정의가 다른 한쪽의 특별한 경우가 되지 않는다. 즉, [11]에서 사용된 정의에 의해서 서버  $S$ 가 서비스 곡선  $S_i(\cdot)$ 을 보장한다 할지라도, 서버  $S$ 가 우리의 정의에 의해서 서비스 곡선  $S_i(\cdot)$ 을 보장하는 것은 아니며, 그 반대의 경우도 성립하지 않는다.

서비스 곡선  $S_i^{net}(\cdot)$ 이 그 각각의 보장 서비스 곡선들을 구성하는 선 조각들을 기울기가 오름차순으로 붙임으로써 만들어짐을 말한다. 다음 정리들의 증명 과정은 부록에 제시되었다. (아래의 세 개의 정리와 비슷한 것들을 [11]에서 찾을 수 있지만, 그것들은 고정 크기의 패킷을 사용하는 특정 망에서만 적용될 수 있다.)

**정리 3** (네트워크 서비스 곡선 정리) 세션  $i$ 가  $M$ 개의 서비스 곡선 서버들을 통과한다고 가정하자.  $M$ 번째 서버가 세션  $i$ 에게 서비스 곡선  $S_i^m(\cdot)$ 을 보장한다고 하자. 그러면, 이 여러 개의 서버들은 세션  $i$ 에게 다음의 서비스 곡선  $S_i^{net}(\cdot)$ 을 보장한다.

$$S_i^{net}(t) = \min \left\{ \sum_{m=1}^M S_i^m(\Delta_m); \Delta_m > 0 \text{ and } \sum_{m=1}^M \Delta_m = t \right\}. \quad (6)$$

각 세션에 대한 서비스 곡선 서버가 주어지면 그 세션에 대한 지연과 버퍼양에 대한 한계를 추출할 수 있다.

**정리 4** (지연 한계 정리) 세션  $i$ 의 트래픽 한계 함수가  $b_i(\cdot)$ 이라고 하자. 만일 서비스 곡선 서버가 세션  $i$ 에게 서비스 곡선  $S_i(\cdot)$ 를 보장한다면, 그 서버에서의 패킷 지연은 결코 다음의 값보다 크지 않다.

$$\max_{k:k>0} \min \{ \Delta, \Delta > 0 \text{ and } b_i(k) \leq S_i(k + \Delta) \}. \quad (7)$$

**정리 5** (버퍼양 한계 정리) 세션  $i$ 의 트래픽 한계 함수가  $b_i(\cdot)$ 이라 하자. 만일 서비스 곡선 서버가 세션  $i$ 에게 서비스 곡선  $S_i(\cdot)$ 를 보장한다면, 어떤 순간에도 그 서버에서의 세션  $i$ 의 버퍼양은 결코 다음의 값보다 크지 않다.

$$\max_{k:k>0} \{ b_i(k) - S_i(k) \} + l^{\max}. \quad (8)$$

여기서  $l^{\max}$ 는 패킷들의 최대 크기이다.

### 5. 네트워크 요소들을 서비스 곡선 서버로 특 성화하기

우리는 네트워크를 세 종류의 네트워크 요소, 즉, 전송

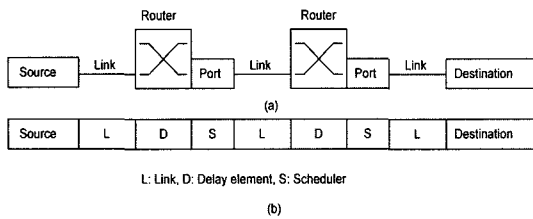


그림 1 (a) 진원지로부터 목적지까지의 네트워크 예  
(b) 그 네트워크를 전송선, 지연 요소, 그리고 스케줄러의 세 종류의 연속된 네트워크 요소로 모델링하기

선, 지연 요소, 그리고 스케줄러로 모델링한다. 그림 1은 진원지에서 목적지까지의 네트워크의 한 예와 이 네트워크를 어떻게 네트워크 요소로 모델링하는 지를 보여준다. 각 절에서는 각 종류의 네트워크 요소를 서비스 곡선 서버로 모델링하는 방법을 설명한다.

#### 5.1 전송선과 지연 요소

우리는  $W_i^{link}(s, t)$ 와  $W_i^{delay}(s, t)$ 을 각각 시간 구간  $(s, t)$  동안 전송선과 지연 요소로부터 출력된 세션  $i$ 의 데이터양으로 표기한다. 비슷하게,  $A_i^{link}(s, t)$ 와  $A_i^{delay}(s, t)$ 는 각각 시간 구간  $(s, t)$  동안 전송선과 지연 요소로부터 입력된 세션  $i$ 의 데이터양으로 표기한다.

전송선과 지연 요소는 대개 함께 붙어 있기 때문에 이들을 같이 묶어서 모델링하면 된다. 그러나, 그림 1의 목적지 바로 이전의 경우 처럼 전송선만 모델링 할 경우도 발생한다. 따라서, 전송선만 있는 경우와 전송선과 지연 요소가 함께 있는 경우 두 가지로 나누어 서비스 곡선 서버로 모델링 하는 방법을 설명한다.

첫째로, 우리는 전송선을 서비스 곡선 서버로 모델링 한다. 대역폭이  $C$  bps이고 최대 전파 지연이  $d^{pro}$ 인 선을 고려하자. 세션  $i$ 가 시간 구간  $(\tau, \tau^*]$  동안 내내 이 선을 점유해서 패킷을 전달한다고 가정하자. 이 경우 이 선은 시간 구간  $(\tau, \tau^*]$  동안 세션  $i$ 가 저장 기간 내에 있는 것으로 간주할 수 있다. 또, 시간 구간  $(\tau, \tau^*]$ 에 속하는 임의의 시간  $t$ 에 대해서 다음의 식이 성립한다.

$$W_i^{link}(\tau, t) \geq \max(0, C(t - d^{pro} - \tau)). \quad (9)$$

식 (9)는 다음과 같이 쓸 수 있다.

$$W_i^{link}(t) \geq W_i^{link}(\tau) + \max(0, C(t - d^{pro} - \tau)) \quad (10)$$

$$A_i^{link}(\tau) + \max(0, C(t - d^{pro} - \tau)). \quad (11)$$

식 (11)은  $\tau$ 가 세션  $i$  저장 기간의 한 시작 시점이기 때문에 성립한다. 식 (11)로부터 이 선은 식 (5)에 의해 세션  $i$ 에게 서비스 곡선  $S_i(t) = \max(0, C(t - d^{pro}))$ 을 보장하는 서비스 곡선 서버로 모델링 된다.

둘째로, 그 전송선 뒤에 지연 요소가 따르는 경우 이 둘을 함께 하나의 서비스 곡선 서버로 모델링한다. 지연 요소에 의해서 발생할 수 있는 최대 지연을  $d^{\max}$ 라고 가정하자. 그러면, 이 지연 요소는 다음의 식을 만족시킨다.

$$W_i^{delay}(\tau, t) \geq A_i^{delay}(\tau, t - d^{\max}). \quad (12)$$

이 지연 요소는 그 전송선 뒤에 바로 붙어 있으므로  $A_i^{delay}(\cdot)$ 은  $W_i^{link}(\cdot)$ 와 동일하다. 따라서, 식 (12)를 다음과 같이 쓸 수 있다.

$$W_i^{delay}(\tau, t) \geq W_i^{link}(\tau, t - d_i) \geq \max(0, C(t - d^{\max} - d^{pro} - \tau)). \quad (13)$$

식 (13)으로부터 그 링크와 지연 요소는 세션  $i$ 에게 서비스 곡선  $S_i(t) = \max(0, C(t - d^{\max} - d^{\text{prom}}))$ 를 보장하는 하나의 서비스 곡선 서버로 간주할 수 있다.

5.2 스케줄러

이 절에서는 모든 LR 서버뿐만 아니라 LR 서버가 제대로 특성화할 수 없는 RC-EDF와 SC 스케줄러들도 서비스 곡선 서버로 특성화할 수 있음을 보인다. 이 사실은 서비스 곡선 서버가 LR 서버보다 엄격히 더 큰 범주의 스케줄러들을 포함해서 분석할 수 있음을 의미한다.

5.2.1 LR 서버

다음 정리는 LR 서버를 서비스 곡선 서버로 특성화하는 방법을 보인다.

정리 6 세션  $i$ 에 대한 할당율이  $\rho_i$ 이고 잠복 시간이  $\theta_i$ 인 LR 서버를 고려하자. 그러면, 이 LR 서버는 세션  $i$ 에게 다음의 서비스 곡선  $S_i^{\rho_i, \theta_i}(\cdot)$ 를 보장하는 서비스 곡선 서버로 모델링 된다.

$$S_i^{\rho_i, \theta_i}(t) = \max(0, \rho_i(t - \theta_i)). \tag{14}$$

정리 6의 증명

임의의 시간 구간  $(s, t]$  동안 이 LR 서버에 의해서 전송되는 세션  $i$ 의 데이터양을  $W_i(s, t)$ 로 표기하자.  $\tau_0$ 를 세션  $i$ 의 바쁜 기간들의 한 시작점이라고 하자.  $\tau_0$ 를 시작점으로 하는 그 바쁜 기간에 속하고 이 LR 서버를 떠나는 임의의 패킷의 출발 시간을  $t$ 라고 하자. 이 경우 LR 서버의 정의에 의해서 시간 구간  $(\tau_0, t]$  동안 세션  $i$ 는 적어도  $\max(0, \rho_i(t - \tau_0 - \theta_i))$  만큼의 서비스양을 보장 받는다. 다시 말해,

$$W_i(\tau_0, t) \geq \max(0, \rho_i(t - \tau_0 - \theta_i)) \tag{15}$$

$$= S_i^{\rho_i, \theta_i}(t - \tau_0). \tag{16}$$

LR 서버의 정의에 의해서 식 (16)을 만족시키는  $\tau_0$ 은 항상 존재한다.  $\tau_0$ 은 바쁜 기간들의 한 시작점이므로 동시에 세션  $i$ 의 저장 기간들 중 한 시작점이기도 하다. (바쁜 기간은 저장 기간들의 한 시작점에서 시작하고 여러 개의 저장 기간을 포함할 수 있다.) 그러므로, 서비스 곡선 서버의 정의에 의해 이 LR 서버는 세션  $i$ 에게 서비스 곡선  $S_i^{\rho_i, \theta_i}(\cdot)$ 을 보장하는 서비스 곡선 서버로 모델링 된다. □

5.2.2. RC-EDF

RC-EDF는 레귤레이터(regulator)들과 하나의 EDF 스케줄러로 구성된다. 각 세션  $i$ 는  $b_i$ -레귤레이터를 할당 받는다.  $b_i$ -레귤레이터는 세션  $i$ 로부터의 입력 트래픽에 관계 없이 출력 트래픽을 항상  $b_i$ -smooth하게 만든다[13]. 또, 각 레귤레이터는 입력되는 패킷에 데드라

인(deadline)을 붙인다. 데드라인은 레귤레이터를 떠나는 시간이나가 EDF 스케줄러를 통과할 때 발생할 수 있는 최악의 지연을 더한 값이 할당된다. EDF 스케줄러는 각 패킷을 데드라인 이내에 전송한다.

우리는 임의의 시간 구간  $(s, t]$  동안  $b_i$ -레귤레이터와 EDF 스케줄러로부터 출력되는 데이터양을 각각  $W_i^{\text{reg}}(s, t)$ 와  $W_i^{\text{EDF}}(s, t)$ 로 표기하고, 입력되는 데이터양은 각각  $A_i^{\text{reg}}(s, t)$ 와  $A_i^{\text{EDF}}(s, t)$ 로 표기한다.

세션  $i$ 에 대해서 RC-EDF를 서비스 곡선 서버로 모델링하자.  $b_i$ -레귤레이터의 정의에 의해 임의의 시간 구간  $(\tau, t]$ 에 대해서 다음의 식이 성립한다.

$$W_i^{\text{reg}}(\tau, t) = \min(A_i^{\text{reg}}(\tau, t), b_i(t - \tau)). \tag{17}$$

위의 식에서 만일  $\tau$ 를 세션  $i$  저장 기간의 한 시작점이고,  $t$ 가 그 저장 기간에 속한다고 가정하자. 그러면, 식 (17)은 다음과 같이 더욱 간소화된다.

$$W_i^{\text{reg}}(\tau, t) = b_i(t - \tau). \tag{18}$$

또, 세션  $i$ 의 패킷이 EDF 스케줄러를 통과할 때 발생할 수 있는 최악의 지연이  $d_i$  시간인 경우 그 시간 구간  $(\tau, t]$  동안 다음의 식이 성립한다.

$$\begin{aligned} W_i^{\text{EDF}}(\tau, t) &\geq A_i^{\text{EDF}}(\tau, t - d_i) \\ &= W_i^{\text{reg}}(\tau, t - d_i) \\ &= b_i(t - d_i - \tau). \end{aligned} \tag{19}$$

식 (19)에서  $\tau$ 는 세션  $i$ 의 저장 기간들 중 한 시작점이므로, RC-EDF는 세션  $i$ 에게 서비스 곡선  $S_i(t) = b_i(t - d_i)$ 를 보장하는 서비스 곡선 서버로 모델링된다.

5.2.3 SC 스케줄러

SC 스케줄러는 데드라인에 기반한 알고리즘이다[12]. 이 스케줄러는 각 세션  $i$ 에게 서비스 곡선  $S_i(\cdot)$ 을 할당하고, 패킷들은 할당된 서비스 곡선에 기반해서 데드라인이 계산된다. SC 스케줄러는 데드라인이 가장 이른 순으로 패킷을 전송한다. 만일 대역폭이  $r$  bps이고 최대 패킷 크기가  $l^{\max}$ 인 경우, SC 스케줄러는 세션  $i$ 에게 서비스 곡선  $S_i(\cdot)$ 을 오른쪽으로  $l^{\max}/r$  만큼 이동시킨 곡선을 보장하는 서비스 곡선 서버로 모델링된다[12].

6. 분석 결과 비교

서비스 곡선 서버를 사용한 기법은 LR 서버를 사용한 것보다 훨씬 큰 범주의 스케줄러를 분석할 수 있다. 또한, 우리의 기법은 LR 서버 방식보다 분석할 때 더 편리하다. 즉, 서비스 곡선 서버 방식에서는 트래픽 한계 함수와 보장 서비스 곡선만 주어지면, 지연과 버퍼 양의 한계를 정리 4와 5를 사용함으로써 바로 추출할

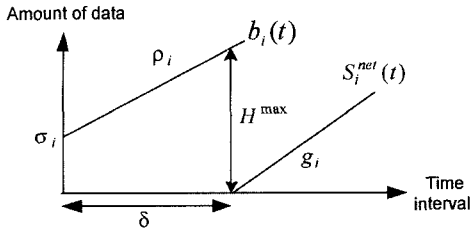


그림 2 세션  $i$ 에 대한 종단간 지연  $D_i^{SC}$

수 있다. 대조적으로 LR 서버 기법은 트래픽 한계 함수마다 별도의 지연과 버퍼양의 한계를 추출해야 하는 불편이 따른다. 예를 들어, 세션  $i$ 가  $(\sigma, \rho)$ -smooth한 경우와  $2(\sigma, \rho)$ -smooth한 경우를 고려한다면, 그 각각의 경우에 별도의 지연의 한계를 추출하는 수고가 필요하다[4]. 이러한 장점뿐만 아니라 서비스 곡선에 기반한 분석 결과는 매우 정확하다. 우리는 LR 서버들을 서비스 곡선 서버로 모델링 했을 때 두 분석 기법에서 추출된 지연의 한계가 동일함을 보인다.

세션  $i$ 가 통과하는  $M$  개의 LR 서버들로 구성된 네트워크를 고려하자.  $M$ 번째 LR 서버의 세션  $i$ 에 대한 할당율이  $g_i$ 이고 잠복 시간이  $d^m$ 이라 하자. 세션  $i$ 가 이 LR 서버들을 지날 때 종단간 지연의 한계를  $D_i^{LR}$ 로 표기하자. 만일 세션  $i$ 가  $(\sigma_i, \rho_i)$ -smooth 하고  $g_i \geq \rho_i$ 인 경우 정리 1로부터 다음의 식이 성립한다.

$$D_i^{LR} \leq \frac{\sigma_i}{g_i} + \sum_{j=1}^M d^j. \quad (20)$$

그렇지 않고 만일 세션  $i$ 의 트래픽 한계 함수가  $\min\{\sigma_i + \rho_i t, P_i t\}$ 인 경우는 다음의 식이 성립한다.

$$D_i^{LR} \leq \left( \frac{P_i - g_i}{P_i - \rho_i} \right) \left( \frac{\sigma_i}{g_i} \right) + \sum_{j=1}^M d^j. \quad (21)$$

이제 이 LR 서버들을 서비스 곡선 서버들로 모델링 하자. 세션  $i$ 가 이 서비스 곡선 서버들을 통과할 때 발생하는 종단간 지연의 한계를  $D_i^{SC}$ 라고 표기하자. 정리 6으로부터  $M$ 번째 LR 서버는 세션  $i$ 에게 다음의 서비스 곡선  $S_i^m(\cdot)$ 을 보장하는 서비스 곡선 서버로 모델링 된다.

$$S_i^m(t) = \begin{cases} 0, & t \leq d^m \\ it, & t > d^m. \end{cases} \quad (22)$$

또한, 정리 3으로부터 그 모델링된 서비스 곡선 서버들은 다음의 네트워크 서비스 곡선  $S_i^{net}(\cdot)$ 을 보장하는 하나의 서비스 곡선 서버로 간주할 수 있다.

$$S_i^{net}(t) = \begin{cases} 0, & t \leq \sum_{m=1}^M d^m \\ it, & t > \sum_{m=1}^M d^m. \end{cases} \quad (23)$$

만일 세션  $i$ 가  $(\sigma_i, \rho_i)$ -smooth한 경우 정리 4에 의해 그 종단간 지연  $D_i^{SC}$ 는  $D((\sigma_i, \rho_i) \parallel S_i^{net})$ 이다. 즉,

$$D_i^{SC} \leq \frac{\sigma_i}{g_i} + \sum_{j=1}^M d^j. \quad (24)$$

마찬가지로 만일 세션  $i$ 의 트래픽 한계 함수  $b_i(t)$ 가  $\min\{\sigma_i + \rho_i t, P_i t\}$ 인 경우도 정리 4에 의해 그 종단간 지연  $D_i^{SC}$ 는 그림 2에 제시되었듯이  $D(b_i \parallel S_i^{net})$ 이다. 그림에서  $\sum_{m=1}^M d^m$ 는 표기상 편의를 위해  $\delta$ 로 나타내었다. 또한,  $t$ 은 그 한계 함수를 구성하는 그 두 함수가 서로 교차하는 시간이다. 즉,

$$\sigma_i + \rho_i t = P_i t. \quad (25)$$

식 (25)로부터  $t$ 은 다음과 같이 쓸 수 있다.

$$t = \frac{\sigma_i}{P_i - \rho_i}. \quad (26)$$

또한,  $x_i$ 는 서비스 곡선  $S_i^{net}(\cdot)$ 이  $\sum_{m=1}^M d^m$  시간 이후에  $\rho_i t$ 와 같아지는 지점이다. 즉,

$$g_i x_i = P_i t. \quad (27)$$

식 (27)로부터  $x_i$ 는 다음과 같이 쓸 수 있다.

$$x_i = \frac{P_i}{g_i} \left( \frac{\sigma_i}{P_i - \rho_i} \right). \quad (28)$$

그러므로,

$$D_i^{SC} \leq \sum_{j=1}^M d^j + x_i - t \quad (29)$$

$$= \sum_{m=1}^M d^m + \frac{P_i}{g_i} \left( \frac{\sigma_i}{P_i - \rho_i} \right) - \left( \frac{\sigma_i}{P_i - \rho_i} \right) \quad (30)$$

$$= \sum_{m=1}^M d^m + \left( \frac{P_i - g_i}{P_i - \rho_i} \right) \left( \frac{\sigma_i}{g_i} \right). \quad (31)$$

우리는 식 (20), (21), (24), 그리고 (31)로부터 두 분석 기법에서 추출된 지연의 한계가 동일함을 결론짓는다.

이제 LR 서버의 네트워크와 그에 대응하는 서비스 곡선 서버에서의 버퍼양 분석 결과를 비교하자.  $M$ 번째 LR 서버에서 세션  $i$ 의 버퍼양을  $Q_i^{m,LR}$ 로 표기하자. 만일 세션  $i$ 가  $(\sigma_i, \rho_i)$ -smooth한 경우  $Q_i^{m,LR}$ 은 정리 2로부터 다음의 식을 만족한다.

$$Q_i^{m,LR} \leq \sigma_i + \rho_i \sum_{j=1}^m d^j. \quad (32)$$

$m$ 번째 LR 서버를 서비스 곡선 서버로 모델링 했을 때 세션  $i$ 의 최대 버퍼양을  $Q_i^{m,SC}$ 로 표기하자. 이 경우 정리 5로부터 최대 버퍼양  $Q_i^{m,SC}$ 는 첫 번째 서버부터  $m$ 번째 서버들로 구성된 네트워크가 보장하는 네트워크 서비스 곡선과 세션  $i$ 의 트래픽 한계 함수 사이의 최대 수직 거리이다가 최대 패킷 크기  $l^{max}$ 를 더한 값을 넘

지 않는다. 다시 말해,

$$Q_i^{m,sc} \leq \sigma_i + \rho_i \sum_{j=1}^m d^j + l^{\max}. \quad (33)$$

식 (32)와 (33)으로부터, 우리의 버퍼양 분석 결과가 LR 서버에 의한 것보다 다소 부정확하다. 그러나, 그 오차는 단지 작은 상수인 최대 패킷 크기  $l^{\max}$ 를 넘지 않는다.

## 7. 결론

본 논문에서는 가변 패킷 망에서 이종의 스케줄러뿐만 아니라 전송선, 지연 요소와 같은 네트워크 요소들을 서비스 곡선 서버로 모델링 하여 네트워크를 통과할 때 발생하는 최악의 지연과 버퍼양의 한계를 추출하는 분석 도구를 제안하였다. 제안하는 도구의 가장 큰 특징은 기존의 것보다 큰 범주의 스케줄러를 포함하여 분석할 수 있으면서도 분석결과가 정확함을 보였다. 향후 과제로는 좀 더 많은 스케줄러, 예를 들면 multirate 스케줄러, 를 서비스 곡선 서버 방식으로 특성화하는 것이다.

## 참고 문헌

- [1] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service, September 1997. RFC 2212.
- [2] 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.
- [3] Pawan Goyal, Harrick M. Vin, and Haichen Cheng. Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks. *IEEE/ACM Tran. Networking*, 5(5):690-704, October 1997.
- [4] D. Stiliadis and A. Varma. Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms. *IEEE/ACM Tran. Networking*, 6(5):611-624, October 1998.
- [5] 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.
- [6] L. Zhang. VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks. *ACM Transactions on Computer Systems*, 9(2):101-124, May 1991.
- [7] N. Figueira and J. Pasquale. An Upper Bound on Delay for the VirtualClock Service Discipline. *IEEE/ACM Tran. Networking*, 3(4):399-408, August 1995.
- [8] M. Shreedhar and George Varghese. Efficient Fair

Queueing Using Deficit Round-Robin. *IEEE/ACM Tran. Networking*, 4(3):375-385, June 1996.

- [9] 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.
- [10] 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.
- [11] 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.
- [12] Kihyun Pyun, Juneha 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.
- [13] R. L. Cruz. Quality of Service Guarantees in Virtual Circuit Switched Networks. *IEEE JSAC*, 13(6):1048-1056, 1995.
- [14] 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 Tradeoffs. *IEEE/ACM Tran. Networking*, 4(3):352-362, June 1996.
- [15] 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.

## 부 록

### 정리 3의 증명

우리는  $m$ 번째 서버에서  $t$ 시간까지 세션  $i$ 로 부터 받은 데이터양과 출력한 데이터양을 각각  $A_i^m(t)$ 와  $W_i^m(t)$ 로 표기한다. 표기상 편의를 위해 우리는 세션  $i$ 의 진원지를 0번째 서버로 표기하면  $W_i^0(\cdot)$ 는  $A_i^1(\cdot)$ 와 같다.

마지막  $M$ 번째 서버로부터 세션  $i$ 의 한 패킷이 떠나는 시간  $t$ 를 고려하자. 이 시간  $t$ 를 기준으로  $m$ 번째 서버에 사용될 시간점  $t_m$ 을 아래와 같이 정의할 수 있다. 첫째로  $M$ 번째 서버에서  $t_M = t$ 로 정의하자. 이 고정된  $t_M$ 으로부터  $(M-1)$ 번째 서버에 사용될  $t_{M-1}$ 은  $t_M$ 보다 작으면서 다음을 만족하는 시간으로 정의할 수 있다.

$$W_i^M(t_M) \geq W_i^M(t_{M-1}) + S_i^M(t_M - t_{M-1}). \quad (34)$$

$M$ 번째 서버는 세션  $i$ 에게  $S_i^M(\cdot)$ 을 보장하기 때문에 위의  $t_{M-1}$ 은 반드시 존재하고,  $t_{M-1}$ 은 세션  $i$  저장

시간의 한 시작 시점이다. 식 (34)로부터 세션  $i$  버퍼는 시간  $t_{M-1}$ 에는 비어있으므로  $W_i^M(t_{M-1})$ 는  $A_i^M(t_{M-1})$ 과 동일하다. 따라서,

$$W_i^M(t_M) \geq A_i^M(t_{M-1}) + S_i^M(t_M - t_{M-1}) \quad (35)$$

$$= W_i^{M-1}(t_{M-1}) + S_i^M(t_M - t_{M-1}). \quad (36)$$

위의 단계를 재귀적으로 따르면, 우리는 식 (36)과 유사한 식을 만족하는  $t_{M-2}, t_{M-3}, \dots, t_0$  시간점들을 정의할 수 있다. 그러므로, 시간  $t$ 에 대해서 그  $(M+1)$ 개의 시간점들이 주어지면, 다음의 식이 성립한다.

$$W_i^M(t) = W_i^M(t_M)$$

$$\geq W_i^{M-1}(t_{M-1}) + S_i^M(t_M - t_{M-1})$$

$$\geq W_i^{M-2}(t_{M-2}) + \sum_{m=M-1}^M S_i^m(t_m - t_{m-1})$$

$$\dots$$

$$\geq W_i^0(t_0) + \sum_{m=1}^M S_i^m(t_m - t_{m-1})$$

$$= A_i^0(t_0) + \sum_{m=1}^M S_i^m(t_m - t_{m-1})$$

$$\geq A_i^0(t_0) + S_i^{net}(t_M - t_0). \quad \square$$

**정리 4의 증명**

세션  $i$ 로부터 임의의 패킷  $p$ 가 그 서버를 떠나는 시간을  $t$ 라고 하자. 그 패킷  $p$ 의 도착 시간과 지연을 각각  $a$ 와  $d$ 로 표기하자( $t = a + d$ ). 그 패킷  $p$ 가 겪을 수 있는 최악의 지연을 추출해 보자. 그 서버는 세션  $i$ 에게 서비스 곡선  $S_i(\cdot)$ 를 보장하기 때문에 다음의 조건을 만족하는 시간  $s$ 가 존재한다.

$$W_i(t) \geq A_i(s) + S_i(t-s). \quad (37)$$

여기서 시간  $s$ 는 세션  $i$  저장 기간들의 시작 시점 중 하나이다. 패킷  $p$  이전에 도착한 패킷들과 패킷  $p$ 는 패킷  $p$ 의 출발 시간인  $t$  이전에 모두 전송된다. 그러나, 만일 패킷  $p$ 의 도착 시간  $a$ 에 여러 개의 패킷들이 동시에 도착한다면, 그 패킷들은 시간  $t$ 까지 전송이 완료되지 않는다. 따라서,

$$A_i(a) \geq W_i(t). \quad (38)$$

식 (38)과 (37)로부터

$$A_i(a) - A_i(s) \geq S_i(t-s). \quad (39)$$

시간  $s$ 는 저장 기간들의 마지막 시작 시점보다 결코 크지 않기 때문에 패킷  $p$ 의 도착시간  $a$ 는 시간  $t$ 보다 크다. 즉,  $a > s$ . 또한, 세션  $i$ 의 트래픽 한계 함수는  $b_i(\cdot)$ 이기 때문에 다음의 식이 성립한다.

$$b_i(a-s) \geq A_i(a) - A_i(s). \quad (40)$$

식 (39)와 (40)부터

$$b_i(a-s) \geq S_i(t-s)$$

$$= S_i(a-s+d). \quad (41)$$

식 (41)을 만족시키기 위해서는  $S_i(\cdot)$ 가 감소하지 않는 감소 함수이므로 다음의 식이 성립한다.

$$d \leq \min \{ \Delta, \Delta > 0 \text{ and } b_i(a-s) \leq S_i(a-s+\Delta) \}$$

$$\leq \max_{k:k>0} \min \{ \Delta, \Delta > 0 \text{ and } b_i(k) \leq S_i(k+\Delta) \}. \quad (42)$$

식 (42)로부터, 패킷  $p$ 에 대한 최악의 지연은  $\max_{k:k>0} \min \{ \Delta, \Delta > 0 \text{ and } b_i(k) \leq S_i(k+\Delta) \}$ 이다. 이 최악의 지연은 패킷  $p$ 의 출발 시간  $t$ 와 독립적이므로 패킷  $p$ 를 포함한 모든 패킷이 이 지연의 한계를 넘지 않음을 알 수 있다.  $\square$

**정리 5의 증명**

임의의 시간  $t$ 에 발생하는 세션  $i$ 의 버퍼양을  $B_i(t)$ 로 표기하자. 세션  $i$ 의 임의의 패킷  $p$ 가 그 서버를 떠나는 시간을  $t$ 라고 할 때 이 시간  $t$ 에 발생할 수 있는 최악의 버퍼양을 추출하자. 우리는 나중에 모든 (패킷 출발 시간과 또한 그렇지 않은) 시간들을 고려한다. 그 서버는 세션  $i$ 에게 서비스 곡선  $S_i(\cdot)$ 를 보장하기 때문에 다음의 조건을 만족하는 시간  $s$ 가 존재한다.

$$W_i(t) \geq A_i(s) + S_i(t-s). \quad (43)$$

여기서 시간  $s$ 는 세션  $i$  저장 기간들의 시작 시점 중 하나이다. 또, 시간  $t$ 에 발생한 버퍼양은  $t$  시간까지 도착한 양에서  $t$ 시간까지 출력된 양을 뺀 것이다. 따라서,

$$B_i(t) = A_i(t) - W_i(t)$$

$$\leq A_i(t) - A_i(s) - S_i(t-s) \quad (\text{식 (43)으로부터})$$

$$= A_i(s, t) - S_i(t-s)$$

$$\leq b_i(t-s) - S_i(t-s)$$

$$\leq \max_{k:k>0} (b_i(k) - S_i(k)). \quad (44)$$

식 (44)로부터 패킷 출발 시간  $t$ 에 발생할 수 있는 최악의 버퍼양은  $\max_{k:k>0} (b_i(k) - S_i(k))$ 이다. 이 최악의 버퍼양은 패킷  $p$ 의 출발 시간  $t$ 와 독립적이므로 이 버퍼양의 한계는 모든 패킷 출발시간에 대해서도 성립한다.

이제, 패킷 출발 시간뿐만이 아닌 모든 시간을 고려하자. 패킷 출발 시간들 사이에는 오직 하나의 패킷만이 떠난다. 따라서, 모든 패킷 출발 시간들에 대해서 식 (44)를 만족하기 위해서는, 모든 시간을 고려한 최악의 버퍼양은  $\max_{k:k>0} (b_i(k) - S_i(k)) + l^{\max}$ 를 초과할 수 없다.  $\square$

편 기 현

정보과학회논문지 : 정보통신  
제 30 권 제 4 호 참조

송 준 화

정보과학회논문지 : 정보통신  
제 30 권 제 4 호 참조

이 흥 규

정보과학회논문지 : 정보통신  
제 30 권 제 4 호 참조