

클래스별 서비스 차등제공을 통한 패킷 지연 제어

정회원 정대인*

Packet delay control based on class-level prioritized service provision

Daein Jeong* *Regular Member*

요약

본 논문에서는, 트래픽으로부터 요구된 지연한계를 보장하는 스케줄링 알고리즘으로서, '클래스별 서비스 차등제공' 알고리즘을 제안하였다. 제안된 알고리즘은, 공평 큐잉과 우선순위 제어 개념의 절충형 패킷 스케줄링 알고리즘이다. 다수의 트래픽 클래스 환경에서, 각 클래스별 서비스 커브를 정의하며, 클래스별 지연 한계를 충족시키기 위한 각 서비스 커브의 형태, 그리고 서비스 커브간의 관계에 근거하여, 클래스별 차등 서비스를 제공한다. 또한, 동일한 클래스 내의 세션간에는 공평 큐잉을 제공함으로써, 세션별 최소한의 대역폭이 보장될 수 있도록 한다. 지연 제어를 위해 새로 정의된 변수는, 시스템 상태에 따라 그 값이 조정됨으로써, 자원의 효율적 활용을 가능하게 한다. 제안된 알고리즘은 가상적 fluid-flow 모델에서 디자인되었으며, 이를 실질적인 패킷 모드에서 구현하는 방안이 제시되었다. 아울러, 제안된 스케줄링 알고리즘에 수반되는, 결정적 형태의 지연 보장을 위한 호 수락 제어 조건을 유도함으로써, 수용된 세션들의 지연 성능이 확정적으로 보장될 수 있도록 하였다.

ABSTRACT

In this paper, we propose a scheduling algorithm that guarantees bounded delay of every connections. That algorithm combines the fair queueing framework and the prioritized delay control mechanism. With multiple classes of traffic, we define a set of service curves: one for each class. Constrained to the required profile of individual service curve for delay satisfaction and their relationships, the delay control is performed via an imposed class-wise service lagging, while the fair queueing discipline is preserved for sessions of the same class so that the protection from the misbehaving traffic is provided. It is noticeable that the value of the introduced control parameter is tunable rather than fixed in accordance with the varying system status. The proposed algorithm is derived based on the hypothetical fluid-flow traffic model. A method of implementing the proposed algorithm in the real packet mode is addressed. Moreover, the conditions for call admission that support the design objectives of the proposed algorithm are derived, so that the delay performance of the all admitted traffic is preserved.

I. 서론

인터넷을 중심으로 한 데이터 트래픽의 증가는, 망 용량의 제한성과 통신 프로토콜의 특성에 따라 서비스 품질에 많은 영향을 미치고 있다. 특히 통신 품질(QoS)의 다양성에 대한 요구는, 새로운 통신

서비스 모델의 창출이 활발해지면서 필연적으로 예견되는 특성이다. 다양한 통신 서비스 품질은, 우선 트래픽을 그 요구된 서비스 품질에 따라 분류(prioritize)하고, 이러한 분류에 따라 트래픽 관리를 차별적으로 수행(priority control) 함으로써 제공 가능하다. 적절한 트래픽 분류는 망 노드에서의 트래

* 한국외국어대학교(Hankuk University of Foreign Studies)

논문번호: K01113-0403, 접수일자: 2001년 4월 3일

※ 이 논문은 2001학년도 한국외국어대학교 학술연구비 지원에 의하여 이루어진 것임.

픽 관리 측면에서, 보다 역동적이고 효율적인 자원 관리를 가능케 하는 방안으로 활용되고 있다. 본 논문에서는, 망 노드에서의 패킷의 지연 한계 값을, 보장해 주어야 할 서비스 품질 요구사항으로 전제하고, 이러한 요구사항에 대하여 만족된 서비스 품질의 제공을 위한 새로운 패킷 스케줄링 알고리즘을 제안하고자 한다.

본 연구는, 일반적인 출력버퍼 구조 패킷 스위치에서, 하나의 출력포트에서 구현되는 패킷 스케줄링 알고리즘에 대한 연구이다. 각 출력포트는 여러 개의 입력과 버퍼를 갖춘 하나의 큐잉시스템으로서 모델링 될 수 있다. 망 노드에 인입되는 세션들은 각각 고유의 지연한계 요구 값을 갖는 것으로 가정하며, 그 요구 값에 따라 클래스로 구분하도록 한다. 즉, 동일한 클래스 내의 세션은 동일한 값의 지연 한계를 요구하는 것으로 분류한다. 이러한 환경에서, EDF(Earliest Deadline First)^[1] 는 지연 성능 관련 tail 확률분포를 최소화하는 특성으로 인하여 매우 유용한 스케줄링 방식으로 인식되고 있다. EDF는 각 패킷의 긴급성에 따라 패킷의 전송 순서를 정한다. 패킷의 긴급성은, 패킷의 도착시간과 허용 가능한 지연 한계 값을 더하여 구한 태그 값, 즉 서비스 완료 요구시간에 의해 판정된다. 태그 값이 작을수록 긴급성이 더해짐을 의미한다. 두 개의 트래픽 클래스, P_1, P_2 를 가정하자. 각각의 지연한계는 D_1 과 D_2 ($D_1 < D_2$) 로 한다. 그림 1에서는 이러한 가정에서의 EDF 방식에 따르는 패킷 스케줄링의 한 예를 보이고 있다. EDF 방식에 의하면, 두 클래스간의 지연 한계의 차이, 즉 $D_2 - D_1$ 만큼의 값이 상대적으로 클래스 2의 태그 값에 부가되고 있으며, 이는 결과적으로 서비스를 차등적으로 제공하는 효과를 낳는다. 이러한 부가적인 양, 즉 $D_2 - D_1$ 을 '부가지연' 이라고 하자. 부가지연은 클래스간의 긴급성의 차이로부터 정의되며, 단위는 시간으로 주어진다. 이러한 부가지연의 개념을, 클래스간의 차별화 된 서비스 제공의 수단으로서 적극적으로 활용하고자 하는 것이 본 제안의 특징이다. 제안된 알고리즘에서는, 클래스간의 상이한 지연한계 보장과 더불어, 동일한 클래스 내의 세션간에는 공평한 서비스 분배가 이루어지도록 한다. 공평성에 대한 필요는, 실질적 서비스 품질의 주요 요소로서 인식되고 있는 지연변이(delay variation) 특성을 고려하고자 하는데 있다. 서비스의 세션간 공평한 분배는 망 자원의 분배를 의미하며, 망 자원의

분배가 보장될 때 각 세션은 예측 가능한 지연변이의 특성을 확보할 수 있다. 특정 클래스에 할당된 망 자원에 대한 적절한 관리가 주어지지 않을 때, 요구된 지연한계 값 내에서의 각 세션의 패킷들이 겪는 지연 변이는 다른 세션들의 트래픽 발생 패턴에 전적으로 영향을 받는다. 이를 방지하기 위해서는 세션간에 공평한 자원의 분배가 이루어져야하며, 이러한 자원의 분배는 잘 알려진 바와 같이 공평 큐잉(fair queueing) 알고리즘이 갖는 중요한 특징이다. 자원을 공유하는 여러 세션간에 각각의 가중치에 비례하는 자원을 할당함으로써, 적절한 호 수락 제어(call admission control)를 거쳐 접속이 허용된 세션들에게, 요구되어진 서비스 품질의 지원을 가능케 하는 것이 공평 큐잉 개념이다. 공평 큐잉의 이해 및 해석은, fluid-flow 가상모델에서 정의되고 분석이 이루어진 GPS (Generalized Processor Sharing) 개념^[2,3]에 의해 체계적인 연구가 이루어졌다. 이러한 특성의 공평 큐잉은, 본 논문에서 제안하는 스케줄링 방안의 기초적 개념을 이룬다. 단, 가중치에 비례하는 대역폭의 할당에 근거하여 서비스를 제공하는 GPS 방식과는 달리, 본 논문에서는 기존의 EDF 알고리즘과 같이, 우선순위 제어형의 서비스 품질 제어 방식을 제안한다. 또한 그 접근방법에 있어, GPS 개념의 확장을 통해 정의되어진, 클래스 단위의 서비스 커브를 기반으로 스케줄링 알고리즘이 도출되었다. 세션 단위의 서비스 커브를 기반으로 하는 지연제어로는 SCED^[4] 방식이 있으며, 이에 비해 클래스 단위의 서비스 커브 활용은, 감소된 구현의 복잡성에 따른 확장성의 이득을 갖게 한다. EDF 방식과의 차이점은 적용된 공평 큐잉의 특성에서 찾을 수 있다. 두 개 클래스의 예에서 EDF 방식은, 시간이 아닌 서비스 양의 관점에서 다음과 같이 해석할 수 있다. 즉, 클래스 2에게 적용된 부가지연은 결과적으로 클래스 2로 하여금 특정한 만큼의 서비스 양을 클래스 1에게 양보하도록 하고 있다. 우리는, 부가지연을 이와 같이 시간개념이 아닌 서비스 양의 개념으로 인식하고, 부가지연, 즉 $D_2 - D_1$ 에 대응되는 서비스 양 단위의 변수를 정의하여, 클래스 1에게 클래스 2에 비해 우선적으로 제공하는 서비스 양으로서 적용하고자 한다. 이로써, 서비스 양의 영역에서 정의되는 공평 큐잉 개념과의 해석적 연동이 가능하게 된다. 이를 임의의 개수의 트래픽 클래스의 환경으로 확장시켜, 각 클래스별 지연 한계를 충족시키고자 하는 것이, 제안하는 스케줄링 방식의 개요이다. 여기서, 서비스의 양으로 정의된 새로운 변수의 값은 모든 클래스의 서비스 품질에 매우 밀접한 관계를 갖는다. 이 변수를 Δ 로 표시하기로 한다. Δ 의 정의는, 2장에서 서비스 커브의 개념을 기반으로 명확하게 설명되어진다.

본 논문의 구성은 다음과 같다. 2장에서는 입력 트래픽의 모델링을 제시하고, 또한 공평 큐잉 개념의 확장된 해석을 통해 클래스 레벨의 서비스 커브 개념을 설명한다. 이를 토대로 3장에서는 새로운 스케줄링 알고리즘을 제안하고, 이에 필요한 변수 λ 의 정의 및 관계식을 유도한다. 접속 허용된 모든 세션들의 안정적 서비스품질 보장을 위한 호 수락 제어조건도 아울러 유도한다. 실질적인 패킷 모드에서의 제안 알고리즘 구현을 위한 방안을 4장에서 논의하며, 5장에서는 결론을 정리한다.

II. 시스템 모델링

1. 입력 트래픽 모델

각 세션은 두 가지 파라미터, 즉 요구되는 지연 상한 값과 트래픽 특성 파라미터로 정의된다. 세션들은 요구된 지연 상한 값에 의해 분류되어 지는데, 동일한 지연 요구 값을 갖는 세션은 동일한 클래스로 분류된다. 이와 같은 분류의 결과, 지연 한계 값 D_i 를 요구하는 세션들은 클래스 i 에 속하게 되며, 클래스 i 의 j 번째 세션은 $s_{i,j}$ 로 표현하기로 한다. 한편, 본 연구에서와 같이 지연의 상한을 모든 트래픽에 대해 충족시키고자 하는 방식의 QoS 제어, 즉 결정적(deterministic) 특성의 QoS를 보장하기 위해서는 트래픽 특성 자체도 확률적 표현이 아닌 결정적 방식의 표현으로 정의되어야 한다^[1]. 따라서 본 연구에서는, "burstiness constraint" 개념^[6]을 적용하여 다음과 같은 트래픽 모델링을 가정한다. 임의의 $\tau(>0)$ 에 대해, $A_{i,j}(t, t+\tau)$ 를 $s_{i,j}$ 가 시간구간 $[t, t+\tau]$ 동안 생성한 데이터 양이라고 정의하였을 때, 임의의 $\rho_{i,j} \geq 0, \sigma_{i,j} \geq 0$ 값에 대하여

$$A_{i,j}(t, t+\tau) \leq \sigma_{i,j} + \rho_{i,j}\tau = A_{i,j}^*(0, \tau)$$

을 만족시키면, $s_{i,j}$ 는 "($\sigma_{i,j}, \rho_{i,j}$)-제약"을 만족시키는 것으로 정의한다. 여기서, $\sigma_{i,j}$ 는 $s_{i,j}$ 의 버스트 특성을 표현하고, $\rho_{i,j}$ 는 $s_{i,j}$ 의 평균 도착율의 상한 값을 나타내며, 편의상 '평균부하'로 부르기로 한다. $A_{i,j}^*(0, \tau)$ 는, 임의의 시간구간 $\tau(>0)$ 동안 $A_{i,j}(\cdot)$ 가 가질 수 있는 상한 함수를 나타내며, 이를 도착 커브라고 정의한다. 본 연구에서는 모든 $s_{i,j}$ 가 이러한 제약을 만족하는 각각의 ($\sigma_{i,j}, \rho_{i,j}$) 값을 갖는 것으로 가정하였다. 시스템에 $P(\geq 1)$ 개의 클래스가 존재하며, P 개의 지연 상한 값이 $D_1 < D_2 < \dots < D_P$

와 같은 관계를 갖도록 정의한다. 또한 N_i 는 클래스 i 에 속한 세션의 개수를, Π_i 는 클래스 i 에 속한 모든 세션의 집합을 나타낸다.

우리가 관심을 갖는 QoS 요소는 지연의 상한 값이며, 이는 시스템 내의 세션들로부터 유발될 수 있는, 가능한 모든 트래픽 발생 패턴에 대하여 항상 보장되는 서비스 품질이어야 한다. 따라서, 지연의 최대 값을 유발하는 트래픽의 발생 패턴을 먼저 정의하여야 한다. 이러한 트래픽 발생 패턴에서, 요구된 지연 한계 값이 보장된다면, 다른 어떠한 트래픽 발생 패턴에 대해서도 그 보장은 유지될 수 있기 때문이다. 참고문헌 [2]에서는, GPS 서버의 운용 시 서버 휴지상태에서, 모든 세션들이 허용된 최대한의 트래픽을 동시에 발생시킬 때, 최대의 지연 값을 모든 세션들이 겪게 됨을 증명하였으며, 이러한 트래픽 발생 패턴을 "all-greedy 가정"으로 정의하였다. 본 연구에서도 동일한 GPS 구조를 가정하고 있으므로, 별도의 증명 없이 all-greedy 가정을, 최악의 지연을 유발하는 트래픽 발생 패턴으로 적용하기로 한다. 즉, $A_{i,j} = A_{i,j}^*(\cdot)$ 로써, 휴지중인 서버에 접속된 모든 세션들이 동시($t=0$ 로 가정함)에, 허용된 최대의 데이터를 발생시키는 패턴을 본 연구에서의 기본적인 트래픽 발생 모델로 가정한다. 이를 도시하면, 모든 $A_{i,j}(\cdot)$ 커브는 $t=0$ 에서 시작하는 커브, 즉 $A_{i,j}^*(0, t) = \sigma_{i,j} + \rho_{i,j}t$ 와 같은 단순한 형태를 갖게 된다.

한편, 지연 성능에 대한 분석을 위해서는, 시스템에 유입되는 데이터 양을 나타내는 커브 $A_{i,j}(\cdot)$ 와 더불어, 서버로부터 제공되는 서비스의 양을 나타내는 별도의 커브가 정의되어야 하며, 이를 서비스 커브라고 한다. 본 연구에서와 같이 서버가 공평 큐잉 서버인 경우, 공평 큐잉의 해석적 분석을 가지적으로 나타내어, 그 해석을 용이하게 해주는 서비스 커브가 참고문헌 [7]에서 처음 제시되었고, 참고문헌 [2]에서는 보다 체계화되어 universal-서비스 커브라는 이름으로 정의되었다. 본 연구에서는 이를 간단하게 '서비스 커브'로 정의하고자 하며, 본 연구에서 적극적으로 활용하고 있으므로, 다음절에서 서비스 커브에 대하여 살펴보고자 한다.

2. 서비스 커브

클래스의 구별이 없는 환경에서 유도된 서비스

커브에 대해서는, 참고문헌 [2]에 자세히 논의되어 있다. 이 절에서는, 그러한 서비스 커브에 대한 면밀한 이해를 바탕으로, 클래스 별로 서비스 커브가 추어지는 환경에서, 즉 다수의 서비스 커브가 정의된 환경에서, 특히 지연 성능과 관련된 점을 논의하고자 한다. 공평 큐잉 기반의 서비스 제공구조에서, 클래스 단위로 서비스 커브를 정의할 경우, 각 클래스별 지연 요구사항을 만족하기 위한 조건은, 해당 클래스의 서비스 커브에 대한 다음과 같은 필요조건으로 표현되어질 수 있다. 여기서, 각 세션의 가중치로는 자신의 평균부하 $\rho_{i,j}$ 를 사용하는 RPPS (Rate Proportional Processor Sharing) 방식을 적용하도록 한다. C 는 서버의 성능을, $W_{i,j}(t)$ 는 세션 $s_{i,j}$ 가 시간 t 까지 제공받은 서비스의 양을 나타낸다. 또한, 표현의 편의를 위하여, 임의의 단조증가함수 $f(t)$ 에 대하여, $t_1 < t_2$ 일 때, $f(t_1, t_2) \equiv f(t_2) - f(t_1)$ 과 같은 관계를 정의한다. $v_i(t)$ 는 클래스 i 에게 정의된 서비스 커브이며, 이는 각 세션의 가중치가 반영되기 이전의, 클래스 단위의 공통적 서비스 제공 추이를 도시한다. 즉, $v_i(t_1, t_2)$ 는 시간구간 $[t_1, t_2]$ 동안 계속적으로 버퍼링을 유지한, 임의의 $s_{i,j}$ 가 제공받은 정규화(normalized)된 서비스의 양, $\frac{W_{i,j}(t_1, t_2)}{\rho_{i,j}}$ 을 나타내며, 이 커브가 정의되는 평면을 $\frac{1}{\rho}$ -scaled 평면이라고 정의한다. 이러한 평면에는, 각 세션의 도착 커브 또한 $\frac{A_{i,j}(t_1, t_2)}{\rho_{i,j}}$ 의 형태로 동시에 도시할 수 있다. 그림 2에서는, all-greedy 도착 패턴을 가정하여, 특정 클래스 i 에 속한 임의의 세션들의 도착 커브와, $v_i(t)$ 를 $\frac{1}{\rho}$ -scaled 평면에 같이 도시하였다. 서비스 커브는, 도착 커브와의 교차점에서만 기울기의 변화를 갖는 선형 커브이다. 이는, all-greedy 가정에서, 임의의 세션이 더 이상 버퍼에 데이터를 갖지 않게 되면(본 논문에서는 이를 '버퍼링으로부터의 해소'라고 한다.), 그 세션이 소모하는 대역폭이 평균부하 값으로 감소하여, 서버에게는 여분의 대역폭이 더해지기 때문이다. 이 평면에서, 시간 τ 에 도착한 $s_{i,j}$ 의 데이터가 겪는 지연 값, $D_{i,j}(\tau)$ 는 다음과 같이 표현된다.

$$D_{i,j}(\tau) = \min \left\{ t \geq \tau : v_i(0, t) = \frac{A_{i,j}(0, \tau)}{\rho_{i,j}} \right\} - \tau$$

즉, 각 세션별 도착 커브와 $v_i(t)$ 와의 수평적 거리가 해당 세션의 각 도착시간에 따른 지연을 나타내는 것이다. 이를, 모든 세션의 도착 커브와 모든 클래스의 서비스 커브가 도시된 $\frac{1}{\rho}$ -scaled 평면을 가정하여 적용시키면, 다음과 같은 결과를 얻을 수 있다. 시간 $t=0$ 에서 시작하는 all-greedy 도착 패턴에서, 다음과 같은 조건이 클래스 i , $i=1, \dots, P$ 에 대해 모두 만족할 때, 클래스 i 의 모든 트래픽이 지연 상한 값 D_i 을 항상 만족할 수 있다.

$$v_i(D_i) \geq \max_{i,j \in \Pi_i} \left\{ \frac{\sigma_{i,j}}{\rho_{i,j}} \right\} \equiv \left(\frac{\sigma}{\rho} \right)_{\max}$$

여기서 $\left(\frac{\sigma}{\rho} \right)_{\max}$ 는, 클래스 i 에 속한 세션 중에서 가질 수 있는 가장 큰 $\left(\frac{\sigma}{\rho} \right)$ 값을 나타낸다. 그림2에서 보듯이, $\frac{1}{\rho}$ -scaled 평면에서는, 이러한 세션이 가장 큰 도착 커브를 갖는다. 위의 조건이 갖는 의미는, 임의의 클래스 i 에 대해, 도착 커브와 서비스 커브 간의 수평적 거리가 D_i 를 초과하지 않아야 함을 의미하며, 이는 $v_i(D_i) \geq A_{i,j}(t - D_i) / \rho_{i,j}$ 의 조건과 동일하게 해석될 수 있다. 단, 유의할 점은, 위의 조건과 더불어 $v_i(D_i) \geq 1$, $i=1, \dots, P$ 가 필요하다. 이는 $t = D_i$ 이후에도, 각 세션의 도착 커브와 서비스 커브와의 수평적 간격이 D_i 를 초과하지 않도록 하기 위한 조건으로서, $\frac{1}{\rho}$ -scaled 평면에서 모든 도착 커브의 기울기는 all-greedy 가정 아래 1 이고 서비스 커브의 기울기는 단조 증가하는 특성에 근거를 둔다. 본 논문에서 제안하는 스케줄링 알고리즘은, 결국 이러한 조건을 모든 클래스 별로 만족시키도록, 해당 $v_i(t)$ 를 제공하는 역할을 하게 된다. $v_i(t)$ 제공을 위한 여러 가지 방법 중에서, 우리는 소위 '클래스별 서비스 차등제공' 방식 알고리즘을 제안하며, 이를 편의상 CSL(Class-level Service Lagging) 알고리즘으로 부르기로 한다. 그 자세한 내용은 3장에서 보인다.

III. 클래스별 차등서비스 제공방식

상이한 트래픽 클래스간에, 서비스 제공의 차등을 줌으로써 지연제어가 이루어지는 과정은, 클래스별

로 정의된 서비스 커브를 활용하여 쉽게 이해될 수 있다. 먼저, 시간 $t=0$ 로 부터 시작하는 all-greedy 입력 패턴을 가정한다. 2장에서, all-greedy 가정에 모든 클래스의 지연 한계가 보장되기 위해서는 다음과 같은 두 조건이 만족되어야 함을 보였다. 즉, $i=1, \dots, P$ 에 대해서,

$$C_1 \quad v_i(D_i) \geq \max_{i,j \in \Pi_i} \left\{ \frac{\sigma_{i,j}}{\rho_{i,j}} \right\} \equiv \left(\frac{\sigma}{\rho} \right)_{\max},$$

$$C_2 \quad v_i'(D_i) \geq 1.$$

그림 3은, 본 논문에서 제안하는 CSL 알고리즘의 개념을 잘 보여주는 그림이다. 설명의 편의를 위해, 두 개의 트래픽 클래스만 주어진 환경에서 CSL 알고리즘이 적용된 일례를 보였으며, 이를 통해 CSL 알고리즘의 내용을 설명하고자 한다. 그림 3에서, 실선으로 표현된 커브는 클래스 1의 서비스 커브를, 점선은 클래스 2의 서비스 커브를 나타낸다. 그림에서 보듯이, 클래스 1은 자신의 서비스 커브 값이 Δ 에 이르기까지 서버의 서비스를 독점하며, 이 후에는 클래스 2와 서버의 서비스를 공유함을 보이고 있다. 서비스의 공유는, 각 세션별 기중치 값의 산출이 필요 없는 단순한 RPPS 방식에 의해 이루어지는 것으로 한다. CSL에서 지연제어는 우선 순위 제어방식에 의해 이루어지므로, 대역폭 제어는 불필요하기 때문이다. 그림 3을 참조하여, 두 클래스 경우의 CSL 알고리즘을 정의하면 다음과 같다. 임의의 데이터가 시스템에 도착하면, 서버는 Δ 만큼의 서비스를 클래스 1에게 우선 제공하고, 그 이후 클래스 1과 클래스 2를 동일하게 RPPS 방식으로 서비스한다. 클래스 2와 서비스를 공유하기 시작하는 시점인 $t=r$ 에서 클래스 2의 서비스 커브는 증가하기 시작하며, 반면에 클래스 1의 서비스 커브는 갑작스런 증가율 감소를 보이게된다. 클래스 2와 의 서버 공유로 인해 클래스 1에게 제공되는 서버의 용량이 감소하기 때문이다. 그림 3으로부터, Δ 값에 따라 각 클래스의 지연 성능이 크게 영향 받게 됨을 알 수 있다. 즉, 큰 값의 Δ 는 클래스 2의 서비스 제공을 크게 늦추게 되어 클래스 2에 대한 지연 한계 값을 증가시키며, 반면에 작은 Δ 값은 클래스 1의 서비스 커브의 증가율을 너무 일찍 감소시켜 클래스 1의 지연 값을 증가시키는 것이다. 이러한 특성은 다음의 그림 4에서 명확히 보여진다. 그림 4에서는, 7개의 클래스 1 세션과 임의의 개수

의 클래스 2 세션을 가정하여, 클래스 1 서비스 커브의 변화 추이를 두 가지 경우에 대하여 도시하였다. 클래스 2 세션은 그림의 복잡성을 피하기 위하여 도시되어 있지 않지만, 클래스 1이 클래스 2와 서비스를 공유하는 시점에서 클래스 1의 서비스 커브의 변화율이 크게 감소하는 형태로 도시함으로써 그 영향을 반영하였다. 실선(또는 점선)으로 표현된 서비스 커브의 경우, 서버는 클래스 1에게 Δ (또는 Δ') 만큼의 서비스 제공 후 클래스 2에게도 서비스를 제공하기 시작한 경우이다. 그림 4에서 보듯이 실선의 경우, 클래스 1의 지연 한계 값을 만족시키기 위한 조건 C_1 을 충족하고 있음을 확인할 수 있는 반면, 점선의 경우에는 만족시키지 못함을 볼 수 있다. 다음 절에서는, 이와 같이 지연 성능 충족 여부에 큰 영향을 미치는 Δ 값을 구하는 방안을 제시하였다. 이에 앞서, 지연 한계를 만족시키기 위한 필요조건 중, C_2 즉 $v_i'(D_i) \geq 1, i=1, \dots, P$ 에 대한 논의를 우선 하기로 한다. 이 조건은, 시간 $t=D_i$ 이후의 시간 영역에서 클래스 i 의 서비스 커브의 증가율이, 인입되는 각 세션별 데이터양의 증가율에 비해 작지 않게 유지함으로써, 두 커브간의 수평적 간격, 즉 지연 값이 D_i 를 넘지 않도록 하기 위한 조건이다. 이 조건은, 전체 시스템 부하와 서버 성능간의 관계에 대한 다음과 같은 기본적인 조건에 의해서 항상 만족됨을 보이고자 한다.

$$\sum_{i=1}^P \sum_{j=1}^M \rho_{i,j} < C \tag{1}$$

즉, 시스템의 평균 부하는 서버의 성능을 넘지 않도록 하는, 시스템이 안정되기 위한 기본 조건을 가정하기로 한다. 이러한 조건아래, 임의의 시간 t 에서 클래스 i 에게 할당된 서비스 커브의 기울기 r_i 는, 그 값이 0 이 아닐 때, 다음과 같이 표현된다.

$$r_i > \frac{C - \sum_{i,j \in B(t)} \rho_{i,j}}{\sum_{i,j \in B(t)} \rho_{i,j}}$$

여기서, $B(t)$ 는 시간 t 의 순간에 버퍼링 되어 있는 세션의 집합이다. 여기에 (1)을 적용하면, $C > \sum_{i=1}^P \sum_{j=1}^M \rho_{i,j} = \sum_{i,j \in B(t)} \rho_{i,j} + \sum_{i,j \notin B(t)} \rho_{i,j}$ 의 관계식에 의해 항상 $r_i > 1$ 임을 알 수 있다. 즉, 서비스가 제공되고 있는 클래스의 서비스 커브는 항상 1 보다 큰 증가율을 가지고 있으며, 반면에 동일한 평면

에서 정의되는 모든 도착 커브는 그 증가율이 1 로서 고정되어 있으므로, 두 커브간의 수평적 간격, 곧 지연 값은 증가하지 않게 된다. 따라서 조건 C_2 는, 서버용량에 대한 조건 (1)을 항상 가정함으로써, 고려할 필요가 없게 된다. (1)은 참고문헌 [2]에서 소위 feasible ordering의 존재조건으로서 요구되고 있다. Feasible ordering의 존재는 큐잉 지연이 유한함을 말하며, 이는 곧 시스템의 안정성을 나타낸다고 볼 수 있다. (1)을 호 수락 제어조건의 하나로서 고려함으로써, 제안된 스케줄링 방식의 안정성을 확보할 수 있다.

1. Δ 값의 결정

CSL 알고리즘에 의하면, 모든 클래스의 지연성능은 Δ 값에 의해 제어됨을 앞에서 확인하였다. Δ 는, P 개의 트래픽 클래스가 있는 환경에서 P 개, 즉 $\Delta_i, i=1, \dots, P$, 가 필요하며, 이러한 환경에서 Δ 에 대한 확장된 정의는 다음과 같다.

$\Delta_i \equiv$ 클래스 $i-1$ 이 클래스 i 에 비해, 임의의 트래픽 도착 시 차등적으로 제공받는 서비스 양. 위의 정의에 따르면, $\Delta_1=0$ 임을 알 수 있다. 이러한 Δ_i 가 활용되어, CSL 알고리즘은 다음과 같이 정의된다.

정의(CSL 알고리즘) : 임의의 데이터 도착 시, CSL 서버는 다음과 같은 서비스 과정을 반복한다.

- 1) $i=2$ 로 둔다.
- 2) CSL 서버는 Δ_i 만큼의 서비스를 클래스 1 부터 $i-1$ 까지 RPPS 방식으로 제공한다.
- 3) $i=i+1$ 로 한다. $i \leq P$ 이면 2)의 과정으로 돌아가고, $i > P$ 이면 모든 세션들에게 RPPS 방식의 서비스 제공을 유지한다.

이를, all-greedy 가정아래 적용하면, $t=0$ 로 부터 클래스 1이 Δ_2 만큼의 서비스를 독점적으로 받은 후, 클래스 1과 클래스 2가 동시에 서비스를 공유한다. 이 후, 클래스 2가 Δ_3 만큼의 서비스를 받으면, 클래스 3에게도 서비스가 허용된다. 이러한 방식으로 클래스간의 차등적 서비스가 이루어지는 것이다.

이 절에서는, 두 개의 트래픽 클래스가 정의된 시스템에서, Δ 의 결정과정과 이에 수반되는 호 수락 제어조건의 유도 과정을 설명한다. 임의의 P 개의 트래픽 클래스가 있는 경우에 대한 분석은, 두 클레

스 경우에 대한 분석의 반복되는 적용이므로, 이 장의 끝에 그 결과만을 정리하였다.

클래스 1 세션의 서브셋인 M_1^t 을 다음과 같이 정의한다.

$$M_1^t = \left\{ s_{1,j} : \frac{\sigma_{1,j}}{\rho_{1,j}} + D_1 < \left(\frac{\sigma}{\rho}\right)_{\max}^1, s_{1,j} \in \Pi_1 \right\}$$

조건 $v_1(D_1) \geq \left(\frac{\sigma}{\rho}\right)_{\max}^1$ 을 만족시키기 위해서는, 위의 M_1^t 에 속한 모든 세션들은 $t = D_1$ 이전에, 각각의 도착 커브가 서비스 커브와 교차함으로써, 해당 데이터가 버퍼에 더 이상 존재하지 않은 상태가 되어야 한다. All-greedy 가정에서, 모든 세션들은 자신의 평균부하율로 데이터를 생성하는데, 각각의 도착 커브와 서비스 커브가 교차된 이후에는 최소한 평균부하율 만큼의 서비스가 보장되므로, 더 이상 버퍼링 되지 않는다. 이는 fluid-flow 모델에서 가능한 개념이다. 한편으로, 동일한 시간구간, 즉 $[0, D_1]$ 동안 M_1^t 에 속하지 않은 세션들은, 서비스 커브의 정의에 의하여 $v_1(D_1)$ 만큼의 서비스를 받으며, 이는 클래스 1의 지연성능 보장을 위한 조건 $v_1(D_1) \geq \left(\frac{\sigma}{\rho}\right)_{\max}^1$ 에 의해, 최소한 $\left(\frac{\sigma}{\rho}\right)_{\max}^1$ 만큼씩의 서비스를 받는다. 이로부터, all-greedy 가정에서, 시간구간 $[0, D_1]$ 동안에 클래스 1의 각 세션들에게 제공되어야 하는 최소한의 서비스의 총 양을 W_1^t 이라고 하면, 다음과 같이 표현할 수 있다. 여기서, $\frac{1}{\rho}$ -scaled 평면에서 정의된 값은 정규화(normalize) 된 값이며, 따라서 ρ 를 곱해야 실질적인 서비스의 양을 나타냄을 반영하였다.

$$W_1^t = \sum_{j \in M_1^t} (\sigma_{1,j} + D_1 \rho_{1,j}) + \left(\frac{\sigma}{\rho}\right)_{\max}^1 \sum_{j \notin M_1^t} \rho_{1,j}$$

시간구간 $[0, D_1]$ 동안 서버가 제공할 수 있는 총 데이터의 양은 CD_1 이므로, 클래스 1만을 고려할 때 다음과 같은 관계식이 클래스 1에 대한 호 수락 제어조건이 된다.

$$CD_1 \geq W_1^t \tag{2}$$

주어진 클래스 1의 트래픽 부하에서 (2)가 만족된다면, 이는 서버의 용량이 클래스 1의 지연한계를 만족시키기에 충분함을 의미하며, 남은 유휴 용량을 활용하여 구간 $[0, D_1]$ 내에서도 클래스 2에게 서비스를 제공 가능하게 한다. 이 때, 얼마만큼의 서

비스를 클래스 1에게 제공한 후 클래스 2에게도 서비스를 허용할 것인가를 결정하는 것이 중요하며, 이 값은 앞에서 Δ_2 로 정의하였다. 그림 4에서 확인하였듯이, Δ_2 의 결정 과정에서는 우선 클래스 1의 지연 성능 충족이 고려되어야 할 것이다. 자원 활용을 최대화하기 위해서, 클래스 1의 지연 한계를 충족시켜주는 최소한의 서비스, 즉 $v_1(D_1) = \left(\frac{\sigma}{\rho}\right)^1_{\max}$ 을 제공하면, 클래스 1의 지연 성능을 유지하면서 클래스 2에의 서비스를 최대화할 수 있다. 따라서, 클래스 1에 비해 Δ_2 만큼의 서비스를 적게 받는 클래스 2의 서비스 커브는 다음과 같이 표현될 수 있다.

$$v_2(D_1) = \left(\frac{\sigma}{\rho}\right)^1_{\max} - \Delta_2.$$

시간구간 $[0, D_1]$ 동안 제공된 서비스의 양은, 동일기간동안 서버의 휴식이 없었다면 CD_1 이며, 이는 클래스 1에게 주어진 서비스와 클래스 2에게 주어진 서비스로 나누어 다음과 같이 구성된다.

$$CD_1 = W_1^1 + \left(\left(\frac{\sigma}{\rho}\right)^1_{\max} - \Delta_2\right) \sum_{j \in \mathcal{H}_2} \rho_{2,j}.$$

이 관계식에서 적절히 이항을 하면, Δ_2 에 대한 다음과 같은 관계식을 얻는다.

$$\Delta_2 = \frac{1}{\sum_{j \in \mathcal{H}_2} \rho_{2,j}} \left\{ W_1^1 + \left(\frac{\sigma}{\rho}\right)^1_{\max} \sum_{j \in \mathcal{H}_2} \rho_{2,j} - CD_1 \right\}$$

C 가 충분히 커서 위의 결과가 $\Delta_2 < 0$ 와 같다면, $\Delta_2 = 0$ 로 둔다. 이 경우, 클래스 1은 별도의 차동적 서비스를 받지 않아도, 지연 요구 값이 만족될 수 있을 만큼 서버의 용량이 충분함을 의미한다. 따라서, 이러한 경우에는 $t=0$ 부터, 클래스 1과 클래스 2 모두 동시에 공평 큐잉 기반의 서비스를 제공받는다.

2. 호 수락 제어 조건

지금까지는 클래스 1의 지연성능만을 고려한 시스템 분석이다. 시간 구간 $[0, D_1]$ 에서 클래스 1의 지연 성능이 고려되었듯이 시간구간 $[0, D_2]$ 에서는 클래스 2의 지연성능이 고려되어야 한다. 클래스 2에게 요구되는 조건도 역시 $v_2(D_2) \geq$

$\left(\frac{\sigma}{\rho}\right)^2_{\max}$ 이다. 이 조건은 우선 클래스 1 내에서 다음을 만족하는 세션들, 즉

$$M_1^2 = \left\{ s_{1,j} : \frac{\sigma_{1,j}}{\rho_{1,j}} + D_2 < v_1(D_2) \right. \\ \left. = \Delta_2 + \left(\frac{\sigma}{\rho}\right)^2_{\max}, j=1, \dots, N_1 \right\}$$

에 속한 클래스 1의 세션들의 버퍼링이 $t = D_2$ 이전에 해소되는 것을 요구한다. 그 이유는, $v_1(t)$ 는 Δ_2 만큼의 독립적 서비스 제공 이후 클래스 2와 동시에 서비스를 받게 되어, $v_2(t)$ 와 동일한 기울기로 증가하게 되므로, $v_2(D_2) = \left(\frac{\sigma}{\rho}\right)^2_{\max}$ 이라면 $v_1(D_2) = \Delta_2 + \left(\frac{\sigma}{\rho}\right)^2_{\max}$ 가 되기 때문이다. 아울러 위의 조건은, 클래스 2 세션 중 다음을 만족하는 세션들의 버퍼링이 $t = D_2$ 이전에 해소되는 것을 요구한다.

$$M_2^2 = \left\{ s_{2,j} : \frac{\sigma_{2,j}}{\rho_{2,j}} + D_2 < v_2(D_2) \right. \\ \left. = \left(\frac{\sigma}{\rho}\right)^2_{\max}, j=1, \dots, N_2 \right\}$$

따라서, 시간구간 $[0, D_2]$ 동안에는, M_1^2 와 M_2^2 에 속한 세션들에 대한 서비스 제공이 충족되어야 클래스 2의 지연 한계가 보장될 수 있다. 이들 세션들에 대하여 요구되는 서비스의 양을 각각 W_1^2 , W_2^2 라고 하면 다음과 같다.

$$W_1^2 = \sum_{j \in M_1^2} (\sigma_{1,j} + D_2 \rho_{1,j}) \\ + \left\{ \Delta_2 + \left(\frac{\sigma}{\rho}\right)^2_{\max} \right\} \sum_{j \in M_1^2} \rho_{1,j}$$

$$W_2^2 = \sum_{j \in M_2^2} (\sigma_{2,j} + D_2 \rho_{2,j}) + \left(\frac{\sigma}{\rho}\right)^2_{\max} \sum_{j \in M_2^2} \rho_{2,j}$$

이로부터, 다음과 같은 서버에 대한 요구조건이 얻어진다.

$$C \geq \frac{W_1^2 + W_2^2}{D_2}.$$

이는, 시간구간 $[0, D_2]$ 동안에 요구되는, 클래스 2의 지연요구를 충족시키기 위한 서버 성능의 조건이다. 한편, 이와는 별도로 시간구간 $[0, D_1]$ 동안에 클래스 1의 지연 요구 충족을 위한 조건 (2)를 유도한 바 있다. 이 두 조건을 모두 만족시키기 위한 조건은 다음과 같다.

$$C \geq \max \left\{ \frac{W_1^1}{D_1}, \frac{W_1^2 + W_2^2}{D_2} \right\}. \quad (3)$$

(3)은, CSL 알고리즘에서 클래스 1과 클래스 2가 동시에 지연요구사항을 보장받기 위한 호 수락 제어 조건이다. 새로운 세션으로부터 호 접속 요구가 주어질 때마다, 호 수락 제어조건 (3)의 충족 여부가 검증되어야 하며, 접속 허용이 될 경우 새로운 Δ_2 값을 계산하여 스케줄링 알고리즘에 적용하여야 한다.

위의, 두 클래스 경우의 과정을 임의의 P 개의 트래픽 클래스 환경에 적용하기 위해서는, 다음과 같은 정의가 우선 필요하다. $1 \leq i \leq k \leq P$ 에 대해, all-greedy 가정 아래,

M_i^k : 조건 C_1 에 의하여, $t = D_k$ 이전에 버퍼링이 해소되는, 클래스 i 의 세션들.

$$M_i^k = \left\{ s_{i,j} : \frac{\sigma_{i,j}}{\rho_{i,j}} + D_k < \sum_{m=i+1}^k \Delta_m + \left(\frac{\sigma}{\rho} \right)^k_{\max}, j=1, \dots, N_i \right\}$$

W_i^k : 조건 C_1 을 만족시키기 위하여, 시간구간 $[0, D_k]$ 동안 클래스 i 에게 제공되어야 하는 최소한의 서비스 양.

$$W_i^k = \sum_{j \in M_i^k} (\sigma_{i,j} + D_k \rho_{i,j}) + \left(\sum_{m=i+1}^k \Delta_m + \left(\frac{\sigma}{\rho} \right)^k_{\max} \right) \sum_{j \in M_i^k} \rho_{i,j}$$

두 클래스의 경우에 정의한 M_i^2 , W_i^2 등은 모두 위의 정의에 부합됨을 확인할 수 있다. 이러한 정의를 사용하여, Δ_i 값의 결정과 호 수락 제어 조건은 다음과 같은 반복적 과정을 통해 주어진다. 이는, 새로운 호 접속 요구가 있을 때마다 수행된다.

- 1) $k=1$, $\Delta_1=0$ 로 둔다.
- 2) 다음과 같은 관계식이 만족되면, 다음의 3단계로 간다.

$$C \geq \max \left\{ \frac{W_1^1}{D_1}, \frac{W_1^2 + W_2^2}{D_2}, \dots, \frac{\sum_{j=1}^k W_j^k}{D_k} \right\}$$

이 조건을 만족 못하면, 호 접속 요구는 거절되며, 이번 과정을 통해 계산되어진 Δ 값들은 폐기되고 과정을 종료한다.

- 3) 다음과 같이 Δ_{k+1} 을 계산한다.

$$\Delta_{k+1} = \max \left\{ 0, \frac{1}{\sum_{j \in \mathcal{H}_{k+1}} \rho_{k+1,j}} \left(\sum_{i=1}^k W_i^k + \left(\frac{\sigma}{\rho} \right)^k_{\max} \sum_{j \in \mathcal{H}_{k+1}} \rho_{k+1,j} - C D_k \right) \right\}$$

$k=k+1$ 로 두어, $k \leq P$ 이면 2단계로 간다. $k > P$ 이면 과정을 끝내고, 호 접속 요구를 수락하며, Δ 값들은 이 과정에서 구한 값들로 대체한다.

이러한 P 클래스 환경으로의 확장은, 앞에서 자세히 설명된 두 클래스 경우의 과정을 반복적으로 수행하여 구한 것이다. 그 과정에서, CSL의 기본 개념을 반영하는 다음과 같은 관계, 즉 $1 \leq i \leq k \leq P$ 에 대해 $v_i(D_k) = \Delta_{i+1} + v_{i+1}(D_k)$ 가 중요하게 활용된다.

IV. CSL 알고리즘의 구현

앞 절에서 정의된 CSL 알고리즘은, 가상적 fluid-flow 모델에서 정의되었으며, 따라서 패킷 단위로 서비스되는 실질적 시스템에서의 구현을 위해서는 적합한 접근방법이 강구되어야 한다. 참고문헌 [2]에서는, 가상적 fluid-flow 모델에서 정의된 GPS 개념을 실질적 패킷 모드의 시스템에서 구현하기 위해 PGPS(Packet-by-packet GPS) 방식을 제시하였다. 이후 많은 논문에서 논의되었던, 패킷 모드에서의 각종 변형된 공평 큐잉 알고리즘의 구현은 PGPS 방식을 기본으로 하여 제시되고 있다. 본 논문에서도 PGPS 알고리즘에 기반을 두고자 한다. PGPS 알고리즘은, 가상종료시간(virtual finishing time)의 개념을 활용하여 구현되었다. 즉, 도착된 패킷에 대해 가상종료시간을 구하여 태그한 후, 서버는 태그 값의 작은 순서대로 서비스하는 것이다. $P_{i,j}^k$ 를 세션 $s_{i,j}$ 의 k 번째 패킷, $F_{i,j}^k$ 는 $P_{i,j}^k$ 의 가상종료시간, $L_{i,j}^k$ 는 $P_{i,j}^k$ 의 길이, $a_{i,j}^k$ 는 $P_{i,j}^k$ 의 도착시간으로 가정한다. PGPS에서는 $F_{i,j}^k$ 를 다음과 같은 방식으로 구한다.

$$F_{i,j}^k = \frac{L_{i,j}^k}{\rho_{i,j}} + \max \{ F_{i,j}^{k-1}, v(a_{i,j}^k) \}$$

여기서 $v(t) = \frac{C}{\sum_{j \in \mathcal{B}_i(t)} \rho_{i,j}}$ 이며, $F_{i,j}^0 = 0$

이다. 이와 같이 주어진 PGPS에 기반을 두고, 본 논문에서 정의한 변수 Δ 를 고려하여 새로운 형태의 공평 큐잉 알고리즘의 구현방안을 논의한다. 설명의 편의를 위하여, 두 개의 클래스만이 있는 경우를 예로 든다. EDF 방식과의 비교를 위하여, 그림 1에서 살펴보았던 두 클래스 경우의 도착 패턴을 그림 5에 다시 도시하였다. 변수 Δ_2 의 도입에 따

른 영향은 그림 5의 세 번째와 다섯 번째 시간 축 그림의 비교에서 명확히 드러난다. 세 번째 시간 축은 클래스의 구별이 없는 일반적 공평 큐잉 알고리즘에 의해 재 정렬된 서비스 순서이고, 다섯 번째 시간 축은 Δ_2 를 활용한 클래스별 차동적 서비스 제공에 따라 재 정렬된 서비스 순서를 나타낸다. 패킷 $P_{i,j}^k$ 가 시간 $a_{i,j}^k$ 에 도착했을 때 계산되어지는 태그 값은, 도착된 패킷이 클래스 2에 속할 경우 $\Delta_2 + v(a_{i,j}^k)$ 보다 작지 않은 값을 갖도록 지정되는 반면, 클래스 1에 속한 패킷은 $v(a_{i,j}^k)$ 보다 작지 않은 값을 갖게 된다. 결과적으로 클래스 2는 클래스 1에 비하여 Δ_2 만큼의 가상종료시간이 인위적으로 지연된 결과를 갖는다. 이를 그림 1의 EDF의 경우와 비교하면, 클래스 2에 대한 인위적 지연이라는 기본구조는 동일하나, EDF에서는 시간 단위의 인위적 지연이 주어진 반면, CSL에서는 가상종료시간, 즉 서비스 양의 단위로 인위적 지연이 주어지고 있음을 확인할 수 있다. 이러한 특성을 고려하면서, 다음과 같은 태그 값 산출방식을 살펴보자.

$$F_{i,j}^k = \frac{L_{i,j}^k}{\rho_{i,j}} + \max\{F_{i,j}^{k-1}, v(a_{i,j}^k)\} + \sum_{l=1}^i \Delta_l$$

이 관계식을, 편의상 클래스 1과 2에 대해서만 살펴보면, 클래스 1의 경우 $\Delta_1=0$ 이므로 PGPS 방식에서의 산출 식과 같은 반면, 클래스 2의 경우, 이웃된 패킷간에 취할 수 있는 최소한의 가상종료시간의 간격은 $\frac{L_{i,j}^k}{\rho_{2,i}} + \Delta_2$ 가 된다. 여기서, 이웃 패킷간의 가상종료시간 간격의 의미를 살펴보기 위해 PGPS 방식에서의 해당되는 값 $\frac{L_{i,j}^k}{\rho_{i,j}}$ 을 참고하자. 설명의 편의를 위하여 모든 패킷의 길이를 일정하다고 가정하면, 이 값은 각 세션의 평균부하 $\rho_{i,j}$ 에 관여되는 값이다. 상대적으로 큰 $\rho_{i,j}$ 에 대해서는 그 값이 작아지는데, 이는 곧 가상종료시간이 상대적으로 조밀하게 주어짐을 의미하며, 따라서 서비스 받을 기회가 많이 주어지게 된다. 즉, 이웃 패킷간의 가상종료시간 간격은, 제공하는 대역폭과 직접적인 관련을 갖고 있음을 알 수 있다. 따라서 일정한 양의 Δ_2 값이 이 간격에 더해진다는 것은 결과적으로 클래스 2에게 할당되는 대역폭의 감소를 의미하는 것이며, 결국 대역폭 제어에 의한 지연제어 방식의 형태를 띄게 된다. 이는 CSL 알고리즘의 기본 개념을 올바르게 나타내지 못한 구현 방

식이다. 이러한 이해를 바탕으로, 다음과 같은 태그 산출방식을 생각해보자.

$$F_{i,j}^k = \frac{L_{i,j}^k}{\rho_{i,j}} + \max\{F_{i,j}^{k-1}, v(a_{i,j}^k) + \sum_{l=1}^i \Delta_l\} \quad (4)$$

Δ 는 max항 내에 놓여 있다. 실제 PGPS 방식에서 나타나는 max 항은 가상적 서비스 시작시간을 의미한다. Max 항 내의 $F_{i,j}^{k-1}$ 은, $P_{i,j}^k$ 바로 전에 입력된 패킷, 즉 $P_{i,j}^{k-1}$ 이 서비스를 종료하는 시점이 되며, $v(a_{i,j}^k)$ 는 $P_{i,j}^k$ 가 서버에 도착하였을 때의 시스템의 서비스 커브 값을 가상적 시간 개념으로 표현한 값이다. $P_{i,j}^k$ 는 이 두 값 중 어느 값보다도 앞서서 서비스를 시작할 수 없으므로, $P_{i,j}^k$ 의 서비스 시작시간은 위 두 값 중 나중 값에 의해 결정되는 것이다. 이와 같이, max 항이 $P_{i,j}^k$ 의 가상적 서비스 시작시간으로 이해될 때, Δ 는 그 정의에 비추어, max 항, 즉 서비스 시작시간에 영향을 미쳐야 한다는 점이 명확해진다. 서비스 시작시간에 대한 재조정은 우선순위 제어방식의 지연 제어를 의미하며, 이는 기본적으로 EDF와 동일한 제어구조를 보여준다. 이렇게 함으로써, 클래스 i 가 갖는, 이웃 패킷간의 가상종료시간 간격은, 자신의 평균부하에 대응되는 값($\frac{L_{i,j}^k}{\rho_{i,j}}$)을 그대로 유지하면서, 클래스 1에 비해 서비스 시작시간을 가상서비스시간 단위로 $\sum_{l=1}^i \Delta_l$ 만큼 지연 받게 됨을 볼 수 있다. 즉, 대역폭의 제어와는 직접적인 관련 없이 지연제어가 이루어진다. 이러한 태그 값 산출방식을 그림 5에서 가정된 도착 패킷들에게 적용하면, 다섯 번째와 여섯 번째의 시간 축 상에 도시되어 있는 형태로 서비스 순서가 재조정되는 것을 확인할 수 있으며, 이는 곧 본 논문에서 제안한 CSL 알고리즘의 기본 개념과 일치되는 것이다. 따라서, CSL 알고리즘의 구현방안으로서, (4)와 같은 태그 산출방식이 적용되어야 한다.

V. 토의 및 결론

본 논문에서는, 클래스 단위의 지연 제어를 위한 우선순위 제어방식의 스케줄링 알고리즘을 제안하였다. 각각의 클래스에 제공되어지는 서비스 커브의 이해 및 응용에 기반을 두었으며, 이러한 커브가 정

의되는 영역, 즉 정규화 서비스 개념의 영역 분석을 통해 CSL 알고리즘이 제시되었다. 시간의 영역에서 정의된 EDF 방식에 비추어, 제안된 알고리즘은 정규화 서비스 영역에서 정의됨으로써, 동일 클래스 내의 세션간에 공평 큐잉 기반 서비스 제공이 이루어진다는 점이 그 특성 중의 하나이다. 이는 곧, 각 세션별로 대역폭의 제공이 보장되는 구조로서, 비정상적인 특성의 트래픽으로부터 받을 수 있는 지연 성능의 침해를 차단할 수 있음을 의미한다.

새로이 정의된 변수 Δ 는 그 결정과정에서, 활성화된 세션간에 망 자원이 최대한 활용될 수 있도록 고려하였다. 더불어 제시한 호 수락 제어 조건은, 새로운 호 접속 요구 시 기존의 시스템 내 세션들의 지연 성능을 유지하면서, 새로운 세션의 결정적 지연요구에 대한 충족가능여부를 확인하기 위한 조건으로서, 주어진 입력 특성을 전제로 할 때 그 효율성이 최대가 되도록 정의하였다. 이러한 연구는, 클래스 레벨의 망 자원 제어 방식을 통해 수행되었으며, 그 기본적인 개념은 클래스 단위의 서비스 커브를 정의하는데 두고 있다. 각 클래스 단위의 서비스 커브를 정의하고, 그들 간의 관계를 정립함으로써, 제안된 알고리즘의 변수 Δ 의 결정과 호 수락 제어조건의 유도가 가능하였다. 새로이 정의된 변수 Δ 는 시스템 상황에 연계되어 동적으로 튜닝되어지며, 이는 클래스간의 효율적인 망 자원 공유 매커니즘을 나타낸다.

Fluid-flow 모델에 기초함으로써 초래하는 구현의 복잡성, 즉 시스템의 서비스 커브를 추적하기 위한 복잡성은 극복이 되어야 하는 문제이다. 이의 해결을 위해, 제안되어 있는 여러 가지 변형된 공평 큐잉 알고리즘과의 연계성을 찾기 위한 연구는, 향후

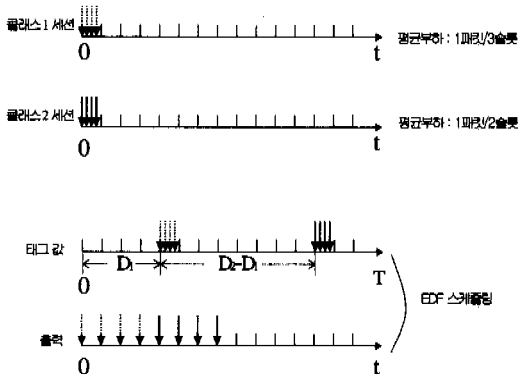


그림 1. EDF 스케줄링 방식 서비스 예(버스트 입력)

지속적으로 수행할 연구 중 하나이다. 또한, 결정적 지연요구를 충족하기 위해서는, 입력특성 자체가 결정적 형태로 표현되어야 하며, 이러한 결정적 형태의 입력특성과 실질적 입력특성과의 차이로 인하여, 주어진 호 수락 제어조건에 효율성이 제한되어진다. 따라서, 실질적 트래픽 발생 특성에 보다 접근된 트래픽 모델링 방안이 지속적으로 모색되어야 한다.

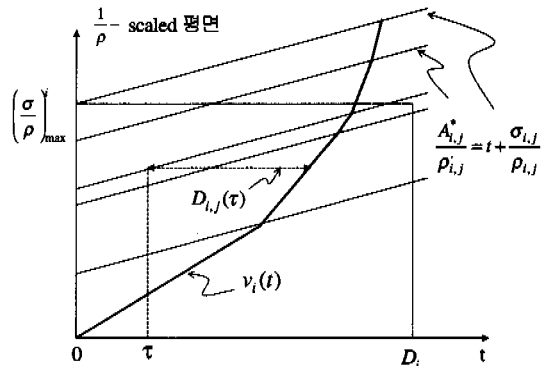


그림 2. $1/\rho$ -scaled평면에서의 도착커브와 서비스 커브

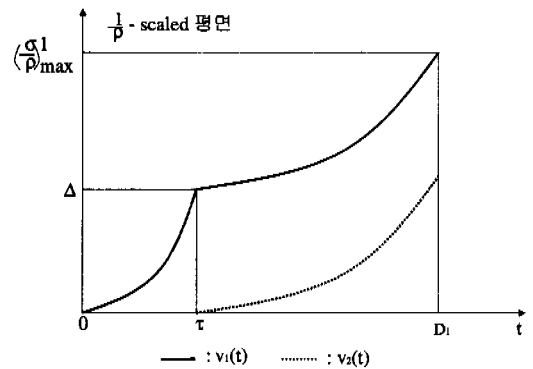


그림 3. CSL알고리즘 방식에 의해 제공되는 서비스 커브 형태(두 클래스의 경우)

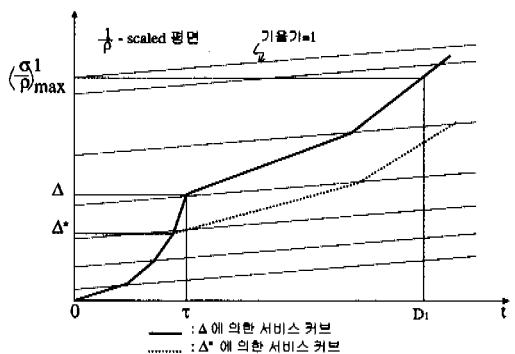


그림 4. Δ 값에 따른 지연성능 변화 예시

