논문 02-27-5C-4 한국통신학회논문지 '02-5 Vol.27 No.5C

지연의 상한 보장과 안정성을 고려한 패킷 스케쥴링 알고리즘

정회원 정 대 인

Packet scheduling algorithm for guaranteed bound and firewall property of delay performance

Daein Jeong *Regular Member*

요 약

본 논문에서는, 세션별 결정적 지연품질의 보장이 주어짐과 동시에, 세션간 서비스품질의 안정성이 지원되는 패킷 스케쥴링 알고리즘인 "클래스별 서비스 차등제공" (CSL: Class level Service Lagging) 알고리즘[6]에 대하여, 시뮬레이션을 통한 기능 검증과, 실질적 구현에 관련된 이슈를 분석하였다. 서비스품질의 안정성 면에서 CSL알고리즘과 EDD(Earliest Due Date) 방식과의 차별성을 확인하였다. 구현의 복잡성에 대한 검토에서, CSL알고리즘은 sorting을 제외하고는, $O(1)$의 복잡성을 갖도록 구현될 수 있음을 보였다. 공평큐잉의 기본 골격이 유지됨으로써, 다양한 형태의 공평큐잉 구현에 쉽게 적용될 수 있는 점은 CSL알고리즘이 갖는 주요 특징이다. TCP/IP 네트워크에서와 같이, 사용자의 의지에 의존하는 TCP 혼잡제어의 환경에서, 세션간 서비스품질의 안정성 보장은 필수적이며, 이러한 측면에서 CSL알고리즘이 갖는 적합성을 확인하였다.

ABSTRACT

In this paper, a novel packet scheduling algorithm, so-called the CSL algorithm is discussed, whereby the firewall property as well as the deterministic delay bound guarantee are supported in session level. Lots of simulation studies validate those properties of the CSL algorithm. The CSL algorithm is distingushable from the well- known EDD scheme in terms of the firewall property. Regarding the implementation complexity, the CSL algorithm turns out to be of $O(1)$ besides the sorting overhead. Owing to the maintained generic fair queueing structure in the CSL algorithm, a various fair queueing schemes can be applied with minor modification. For the TCP/IP network which is vulnerable to the misbehaving traffic sources, the firewall property of the CSL algorithm is quite useful for the advanced quality of services.

Ⅰ. 서 론

인터넷을 통한 통신서비스를 제공함에 있어, 시간이 흐를수록 증가하는 서비스 특성의 다양성에 대한 수요를 수용하기 위한 새로운 연구가 점증적으로 요구되고 있다. 특히, 음성 전화와 같은 실시간성의 서비스를 인터넷에서 수용하기 위해서는, 망 내부에서의 지능화된 트래픽 관리기능의 구현을 필요로 하고 있다. 서비스 특성의 다양화를 수용하기 위한 트래픽 관리기능의 다양화는, 한편으로는 망 구성 요소의 복잡성 증가 및 이에 따르는 비용의 문제가 있지만, 한정된 망 자원의 효율적 활용을 통해 서비스 수용의 다양성을 기함으로써, 향후 개발되어질 새로운 통신 서비스의 수용 가능성을 높이고, 이에 따른 다양한 수익 모델을 창출할 수 있다

* 한국외국어대학교 정보산업공과대학
논문번호 : 020020-0116, 접수일자 : 2002년 1월 16일
※본 연구는 한국과학재단 목적기초연구(R01-2000-00280) 지원으로 수행되었습니다.

는 점에서 매우 중요하게 다루어져야 할 연구이슈로 인식되고 있다.

인터넷에서의 서비스품질에 대한 고려는 지난 수년간 매우 활발하게 다루어지고 있으며, Diff/Serv[1] 개념은 실질적 구현의 용이성으로 인해, 인터넷 코어 망에서의 QoS 제공구조로 많은 주목을 받고 있다. 따라서, 인터넷 코어 망에서 운용될 초고속 라우터를 위한 새로운 트래픽 관리기능의 연구는, 이러한 Diff/Serv 구조에서 용이하게 구현 가능한 형태로 제시되어야 할 것이다. 즉, 트래픽을 생성하는 개별 세션단위에 대한 관리보다는, 특정한 기준에 따라 분류된 클래스 단위의 트래픽 관리가 요구되며, 더 나아가 미시적 QoS 지표에 대한 접근보다는, 트래픽 처리의 차별성 제공을 위한 트래픽 관리 구조의 정립을 필요로 하고 있다. 클래스 단위로 다양한 지연품질의 제공이 가능한 기존의 방식들로서, 본 논문과 연관성이 있는 스케쥴링 알고리즘들을 정리하면 다음과 같다.

Virtual Clock 알고리즘[2]은, TDM (Time Division Multiplexing)방식을 패킷 스위칭 환경에 유동적으로 적용하고자 하였으며, 이를 통해 세션별 다양한 대역폭의 보장, 세션간의 독립성(firewall) 등의 기능을 지원한다. 이러한 특성은, 유사한 성격을 갖는 공평 큐잉(fair queueing)의 개념과 비교되었고, 결과적으로 공평 큐잉의 구체적인 구현을 위한 방안으로 활용되었다[3]. 패킷은 시스템에 도착되면서, 서비스 순서가 결정 지워지는 가상시간(virtual clock) 값의 적절한 할당을 통해 고유의 대역폭 보장이 지원된다. 또한, 측정된 실시간(real time)은 트래픽 발생 패턴을 모니터링을 하기 위한 변수로 사용되며, 특정 세션이 갖는 지연의 우선권(priority)을 반영하는데에도 활용된다. 즉, 측정된 실시간 값에서 특정한 양의 우선권 반영변수 (T)를 감소시킨 값을, 해당 패킷의 서비스 순서를 결정짓는 가상시간의 산출에 적용하므로써, 스케쥴링이 우선되도록 지연 제어를 구현하였다. 그러나, 실제 각 세션이 요구하는 지연의 한계 조건 등에 연관되어, 위의 T 값을 유도하는 과정이 정의되지 않았으며, 또한 서로 다른 지연요구가 혼재하는 경우 그들간의 차별성을 구현하는 방법이 제시되어 있지 않다. 또한, 실시간을 변수로 사용하므로써, 각 세션별 트래픽 파라미터의 준수 여부는 간단히 모니터링 될 수 있는 반면, 상이한 세션간의 서비스 제공에 대한 공평성 지원은 매우 제한된 범위 내에서만 정의되는 단점이 있다. 세션간의 서비스 공평성에 대한 정의는, 동시에 서비스 요구에 참여하는 세션간에만 적용되어야 하는데 반하여, Virtual Clock에서는 각각의 세션이 제공받는 총 서비스 양의 공평성에 평가 기준을 두는 것이다. 정의에 부합되도록 세션간의 공평성을 보장하기 위해서는, 세션간의 서비스 제공 정도를 지속적으로 모니터링하는 새로운 변수가 필요하며, GPS (Generalized Processor Sharing)[3]에서는 이를 실시간 변수에 대체하므로써 명확한 공평 큐잉 알고리즘이 정의되었다.

한편으로, 지연 성능에 대한 차등제어만을 효과적으로 수행하는 알고리즘인 EDD (Earliest Due Date)[4] 방안은, 각 세션별로 요구되는 지연성능을 자원 할당의 개념이 아닌, 우선권 제어에 의존하여 지원하는 형태를 갖는다. 즉, 각 패킷별로 긴급성을 표현하는 스탬프 값인 "도착시간+지연요구값"을 구하여 태깅한 후, 긴급성이 가장 급한 패킷을 먼저, 즉 태그 값의 오름차순으로 스케쥴링하는 방법이다. 이러한 EDD 방식은, 지연 성능을 보장할 수 있는 트래픽의 수용 능력 면에서 최적임이 이미 증명되었다[5]. 그러나, EDD방식에서는 동일한 클래스 내의 세션간에는 FIFO 방식의 서비스가 주어짐에 주목할 필요가 있다. 즉, 동일한 지연요구값을 갖는 트래픽 간에는 EDD방식에 의해 구해지는 태그 값이 도착하는 순서에 따라 증가하게 되어 있으므로, 도착한 순서와 서비스 받는 순서가 동일하게 되는 것이다. 앞에서 언급한 공평큐잉의 특성에 비하여 이러한 FIFO방식이 갖는 문제점은, 지연 성능에 대한 각 세션간의 독립성이 결여되어있다는 점이다. 공평큐잉에서는, 다른 트래픽의 발생 패턴에 독립적으로 최소한의 서비스 제공이 보장되는 반면, FIFO에서는 이러한 독립성이 보장되어있지 않다.

이러한 문제를 보완하기 위해, 참고문헌 [6]에서 CSL알고리즘이 제안된 바 있다. CSL알고리즘은, 위의 GPS에서 정의된 공평큐잉의 개념을 확장하여, 다양한 클래스의 트래픽으로 구성된 환경에 적용하는 것을 목적으로 한다. 즉, 동일한 클래스 내의 트래픽 간에는 공평큐잉을 적용하므로서 세션 단위의 독립적 지연품질을 지원하고, 상이한 클래스간에는 우선권에 따르는 차별화된 차등제어를 수행하는 것이다. 결과적으로, GPS에서 정의된 공평큐잉의 개념을 이용하여, EDD방식에서 관측되는 세션 단위의 지연품질에 대한 독립성 결여문제를 개선하고자 한 것이다. CSL알고리즘의 정의 및 해석적 분석은 참고문헌[6]에 상세히 소개되어있다. 본 논문은 참고문헌[6]의 후속 논문으로서, CSL알고리즘의 배경

에 대한 보완 설명과 그 실질적인 구현 방안 및 구현의 복잡성 검토, 그리고 시뮬레이션을 통한 특성 확인 및 다른 방식과의 비교 등, 참고문헌[6]에서 다루어지지 않은 주요 이슈들을 보완하므로써, CSL 알고리즘 연구의 완성도를 높이고자 하였다.

본 논문의 구성은 다음과 같다. 2장에서는, CSL 알고리즘의 정의 및 구현 방안에 대해 간략히 정리하였다. 3장에서는, CSL 알고리즘에 대한 시뮬레이션 결과를 정리하였으며, 여기에는 CSL알고리즘의 실질적 구현의 복잡성에 대한 검토와, 이의 개선을 위한 방안 검증이 포함되었다. 4장에서는, CSL알고리즘에 수반되는 호수락제어조건에 의해 규정되는, 서버의 가용 서비스영역에 대하여 논의하였고, 5장에서는 결론을 다루었다.

Ⅱ. CSL 알고리즘

2.1 공평큐잉 개념의 적용

CSL알고리즘의 기본개념은, 흐름모델에서 정의된 GPS알고리즘에 기반을 두고 있으며, 그 구현 방법은, GPS알고리즘을 패킷모드에서 구현하는 PGPS (Packet-by- packet GPS)[3] 알고리즘을 참조한다. PGPS알고리즘은 다음과 같이 정의된다. 우선, 세션 i 의 k 번째 패킷은 P_i^k , P_i^k 의 도착시간은 $t=t_i^k$, P_i^k 의 길이는 L_i^k , 세션 i 의 가중치는 ϕ_i 로 표시하기로 한다. 시간 $t=t_i^k$ 에 도착된 P_i^k 에게는, 가상종료시간 F_i^k 을 다음과 같이 산출하여 태그하며, 서버는 버퍼에 있는 패킷중 가장 작은 태그 값을 가진 패킷을 선택하여 서비스한다.

$$F_i^k = \frac{L_i^k}{\phi_i} + \max[v(t_i^k), F_i^{k-1}] \quad (1)$$

여기서, 함수 $v(t)$는, 버퍼에 대기중인 패킷을 갖고 있는 세션들에게 공통적으로 제공된 정규화된 서비스의 양을 나타내며, 흐름모델의 GPS동작을 추적하기 위한 함수로서 다음과 같은 특성을 갖는다.

$$v'(t) = \frac{r}{\sum_{i \in B(t)} \phi_i} \quad (2)$$

$B(t)$ 는 시간 t 에 버퍼에 저장된 패킷을 갖고 있는 세션들의 집합을 나타내며, r 은 서버의 용량이다. 수식 (1)을 분석하면, 우측의 첫 번째 항은 가상적인 GPS서버에 의해 P_i^k 가 서비스 받는 시간구간, max 항은 GPS서버에 의해 P_i^k 가 서비스를 제공받게 되는 시작시간을 나타낸다. 따라서, 수식 (1)의 F_i^k 값은, 가상적인 GPS 서버를 가정하였을 때, 패킷 P_i^k 가 서비스 종료되는 시간을 의미한다. 결국, F_i^k 값의 오름차순으로 패킷을 서비스한다는 것은, 가상적인 GPS서버의 동작을 모방하는 것이 되며, 수식 (1)은 이를 위한 기본 관계식이다. 수식 (2)와 같이 정의된 $v(t)$ 는, GPS서버가 버퍼링되어있는 세션들에게 제공하고 있는 서비스양의 정규화 값을 나타내는 함수로서, 세션간의 공평성이 유지되도록 수식 (1)에서와 같이 각 패킷의 서비스 시작시간을 결정하는 역할을 한다.

(1)과 같이, Virtual Clock알고리즘에서의 실시간 변수가 $v(t)$ 로 대체되면서, 세션간의 서비스 공평성이 고려되는 스케줄링 성격을 갖게된 것이 PGPS 알고리즘이다. CSL알고리즘에서는, 위의 알고리즘을 확장하여, 여러 개의 클래스로 분류된 트래픽 환경을 가정하였다. 이를 위해 우선, 각 변수들의 표기를 다음과 같이 변형한다. 클래스 i 에 속한 세션 j 는 $s_{i,j}$ 로 표시하며, 이를 반영하여 위의 모든 변수들의 아래 첨자인 i 를 i,j 로 대체한다. 한편, 모든 세션은 (σ, ρ)-제한조건[7]을 만족하는 것으로 가정한다. 즉, 임의의 시간구간 $\tau(>0)$동안 임의의 세션 $s_{i,j}$로부터 발생되는 데이터의 총량을 나타내는 $A_{i,j}(t, t+\tau)$는 다음을 만족하도록 한다. 이는, 결정적 지연성능의 보장을 위해서는 결정적 형태의 트래픽 모델링이 필요하기 때문이다[8].

$$A_{i,j}(t, t+\tau) \le \sigma_{i,j} + \rho_{i,j}\tau \quad (3)$$

식 (3)에서, $\sigma_{i,j}$는 버스트 특성을, $\rho_{i,j}$는 평균 도착률 특성을 각각 정의하는 파라미터로 쓰인다. 각 세션의 평균 도착률인 $\rho_{i,j}$를, 각 세션의 가중치로 할당하는 방식은 RPPS(Rate Proportional Processor Sharing) 방식으로 불리며[3], 본 논문에서도 RPPS방식을 적용한다.

CSL알고리즘에서, 클래스간 지연 성능의 차별성은, Virtual Clock알고리즘에서와 같이 각 패킷의 태그 값을 계산하는 과정에 반영하는 방법을 취한다. 앞에서도 언급하였듯이 Virtual Clock알고리즘에서는, 서버가 태그 값의 오름차순으로 서비스한다는 점을 활용하여, 우선권이 있는 패킷의 태그 값을

인위적으로 특정한 양만큼 감소시키는 방식으로 우선권 제어를 구현하였다. 그러나, Virtual Clock알고리즘이 갖고 있는 공평성의 결여와, 우선권 제어를 위해 필요한 정량적 해석이 결여되어 있는 문제를 해결하기 위해, CSL알고리즘은 공평큐잉에 기반을 두고 있으며, 이로써 흐름모델에서의 해석적 분석이 가능하게 되었다[6].

2.2 흐름모델에서의 CSL알고리즘

CSL알고리즘의 기본 개념은 가상적인 흐름모델에서 정의되었다. 임의의 클래스 i 에 속한 세션과 클래스 $i+1$에 속한 세션이 지속적으로 버퍼링되어 있는 환경을 가정한다. 상위 클래스 i의 세션은 버퍼링 되는 시점에, 우선 Δ_{i+1}만큼의 서비스를 제공받은 후, 클래스 $i+1$세션과 서버를 공유하게 된다. 두 세션이 서버를 공유하게 된 시점 이후, 클래스 i 의 세션은 휴지기에 들기까지 클래스 $i+1$의 세션과 차별 없는 서비스를 받게 된다. 이와 같이, 고유의 클래스는 다르지만 서버를 차별 없이 공유하는 세션들의 집합을 '가상클래스'로 정의한다. 시스템에의 도착 후 각 세션은, 서비스를 제공받음에 따라 단계적으로 가상클래스 인덱스를 증가하게 된다. 예를 들어, 클래스 1의 세션은 도착 즉시 가상클래스 1로 전환되며, Δ_2만큼의 서비스를 받은 후 가상클래스 2로 전환되고, 기존의 가상클래스 2에 속한 세션들과 더불어 Δ_3만큼의 서비스를 받은 후 가상클래스 3으로 전환된다. 시스템 내에 $P(\geq 3)$개의 클래스를 가정할 때, 클래스 1의 세션이 버퍼링을 유지하는 한 가상클래스의 전환은 가상클래스 P에 이를 때까지 지속되며, 이러한 과정을 통해 클래스 단위의 차별적 서비스 제공이 이루어지는 것이다. 이를 정리하면, 흐름모델에서의 CSL 알고리즘은, $P(\geq 3)$개의 클래스가 서버를 공유하는 환경에서 다음과 같이 정의된다.

정의(CSL 알고리즘): 임의의 클래스 k 에 속한 세션으로부터의 데이터 도착시, CSL 서버는 다음과 같은 서비스루틴을 수행한다. i의 초기 값은 2이다.

1) 새로 도착된 세션은 가상클래스 k 로 전환된다.
2) 정규화 서비스 Δ_i를 가상클래스 $i-1$에게, 버퍼링 되어있을 경우 제공하며, Δ_i를 제공받은 세션들은 모두 가상클래스 i로 전환된다.
3) $i=i+1$을 수행하여, $i\leq P$ 인 경우 단계 2로 가며, $i=P+1$인 경우 모든 버퍼링 되어있는 세션들에게 RPPS로 공평한 서비스를 제공한다.

이러한 서비스루틴은, 모든 새로운 세션의 도착에 의해 즉시 1단계로부터 재 수행된다. 그림 1에서는, CSL알고리즘의 수행흐름도를 도시하였다. 가상클래스 k가 서비스를 받고 있는 상태에서 도착된 클래스 i 세션에 의해 수행되는 과정을 도시하였으며, 이후 도착된 세션이 없는 경우를 가정하였다. 임의의 다른 세션이 도착하게되면, 즉시 그림 1의 루틴은 다시 수행된다. 이와 같은 CSL알고리즘을 주의깊게 살펴보면, CSL서버는 항상 최상위의 가상클래스에 속한 세션만을 서비스하고 있음을 확인할 수 있다.

2.3 패킷 모드에서의 CSL알고리즘

흐름모델에서 정의된 CSL알고리즘의 고유 특성을 유지하면서, 실질적인 패킷 단위의 데이터 처리가 이루어지는 환경에서 구현하는 방안은 다음과 같다.

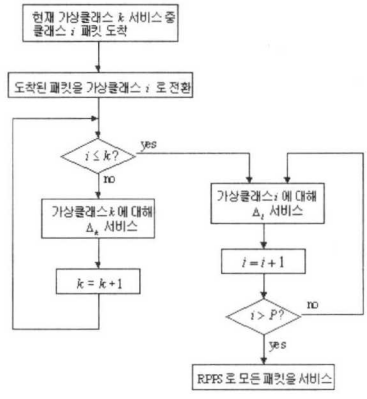

그림 1. CSL 알고리즘의 동작흐름도

1) 시간 $t=t_{i,j}^{k}$에 도착한 패킷 $P_{i,j}^{k}$ 에 대해, 버퍼에 저장되기 이전에 다음과 같은방식으로 태그값 $F_{i,j}^{k}$을 산출하여 태그한다.

$$F_{i,j}^{k}=\frac{L_{i,j}^{k}}{\rho_{i,j}}+\max[v(t_{i,j}^{k})+\sum_{m=1}^{i}\Delta_m, F_{i,j}^{k-1}],$$

$$F_{i,j}^{0}=0. \quad (4)$$

2) 서버는 버퍼에 저장된 패킷이 없을 때를 제외하고는 계속적으로 서비스를 제공하며(work-conserving 방식), 버퍼에 있는 패킷 중 태그 값의 오름차순으로 패킷을 선택하여 서비스한다.

3) 도착된 패킷을 고려하여 서비스 목적함수인 $v(t)$를 추적한다.
4) 버퍼에 패킷이 남아있지 않아 서버가 휴지상태에 들어서면, 함수 $v(t)$는 0로 초기화되며, 모든 태그값 F들과 패킷 순서 인덱스인 k도 0으로 초기화한다.

위의 식 (4)는, 식 (1)에 비하여 새로운 변수 Δ_m으로 구별되며, 이는 앞 절에서도 설명되었듯이 클래스간의 차별화된 서비스의 제공을 위한 변수로서, 클래스 $m-1$이 클래스 m에 비해 차등적으로 제공받는 서비스의 양을 나타낸다. 클래스 1이 가장 우선권이 높은 클래스로서, $\Delta_1=0$ 로 정의되고, 우선권이 낮을수록(i가 증가) $\sum_{m=1}^{i}\Delta_m$값은 누적되어, 결과적으로 우선권이 낮은 패킷은 상대적으로 큰 태그값을 할당받음으로서, 서버의 서비스를 상대적으로 지연되어 받는 것이다. 식 (4)에서 max항은, Virtual Clock알고리즘에서의 max항과 동일한 의미를 갖는다. Virtual Clock알고리즘에서는 *real time* $-T$의 형태로 우선권제어를 수용한 반면, CSL에서는 $v(t)+\sum\Delta$의 방식으로 수용하고 있음을 볼 수 있다. 주목할 점은, *real time* $-T$가 갖는 단위가 실시간(예, *ms*)임에 반하여, $v(t)$는 가상시간으로 불리지만 그 단위는 서비스의 양과 같다. 즉, 상이한 세션들에게 제공되어진 서비스의 양을 각 세션의 가중치인 ϕ로 정규화시킨 양이다. 여기에, 클래스 단위로 주어진 Δ 값들을 더함으로써, 클래스간에 인위적인 서비스의 차별성을 유도하는 것이다. 이로부터, Virtual Clock알고리즘은 TDM방식에 근접한 스케줄링 방식임에 반해, CSL 알고리즘은 서비스 제공 양의 차별성에 근거하고 있음을 확인할 수 있다.

변수 Δ_m값들은, 각 클래스별로 요구되어지는 지연성능의 상한값을 차별적으로 충족시킬 수 있도록 산출되며, 시스템에 새롭게 참여하고자 하는 세션의 연결 요청이 있을 경우, 또는 기존 세션의 종료가 발생되는 경우에 새로운 값으로 재 산출된다. Δ_m 값을 구하는 관계식은 참고문헌 [6]에 상세히 기술되어 있으며, 다음과 같이 주어진다.

$$\Delta_{k+1}=\max\left\{0,\ \frac{1}{\sum_{j\in\Pi_{k+1}}\rho_{k+1,j}}\left(\sum_{i=1}^{k}W_i^k+\left(\frac{\sigma}{\rho}\right)_{\max}^{k}\sum_{j\in\Pi_{k+1}}\rho_{k+1,j}-CD_k\right)\right\} \quad (5)$$

여기서, D_i는 클래스 i의 지연 상한 요구값이며, Π_i는 클래스 i에 속한 모든 세션들의 집합을 말한다. 그리고, $\left(\frac{\sigma}{\rho}\right)_{\max}^{i}$ 는 클래스 i내의 세션들이 갖는 $\frac{\sigma_{i,j}}{\rho_{i,j}}$ 중 가장 큰 값을 나타내며, W_i^k는 클래스 1부터 클래스 k까지의 지연 상한값을 만족시키기 위해 시간구간 $[0,\ D_k]$동안 클래스 $i(\le k)$에게 제공되어야 하는 최소한의 서비스 양을 나타낸다.

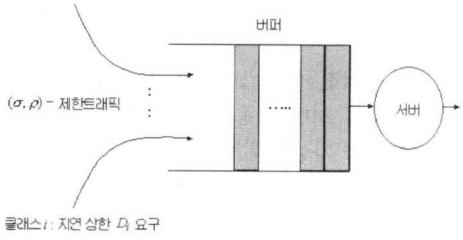

그림 2 출력버퍼 스위치의 출력포트 큐잉모델

Ⅲ. 시뮬레이션

3.1 성능 및 안정성 분석

본 연구에서는, 위에 소개된 CSL알고리즘의 특성을 확인하기 위하여 여러 가지 시뮬레이션을 수행하였다. 하나의 노드에서의 지연성능 측정에 주안점을 두었으며, 노드는 출력 버퍼 구조를 갖는 라우터, 또는 스위치 등 망 노드를 가정하였다. 하나의 출력포트는, 여러 개의 입력 소스로부터 인입되는 패킷을 버퍼에 저장하고, 서버가 가용해짐과 동시에 버퍼내의 특정 패킷을 선택하여 전송하는, 그림 2와 같은 큐잉모델로 도시될 수 있다. 각각의 입력 세션은, 요구하는 지연 상한값에 따라 4개의 클래스(동일 클래스 내의 세션들은 동일한 지연 상한값을 요구)로 분류되는 것을 가정하였으며, 각 클래스 별로 10개의 세션을 설정하여, 총 40개의 입력 세션이 서버를 공유하도록 설정하였다. 각 세션 $s_{i,j}$ 의 트래픽 특성은, 식 (3)과 같은 (σ,ρ)-제한조건을 갖는 것으로 가정하였다. 시뮬레이션에 적용된 스케줄링 알고리즘으로는 CSL 뿐만 아니라, EDD와 FIFO 방식 모두를 수행하였으며, 특히 FIFO는 트래픽에 대한 제어가 전혀 없는 환경에서의 지연성능 참조를 위해 수행하였다. 시간축은 하나의 패킷

처리시간에 해당하는 슬롯 단위로 나누었으며, 패킷의 길이를 1,000bit, 링크 용량을 1Mbps로하므로써, 하나의 슬롯은 1ms로 하였다.

표 1. 시뮬레이션을 위한 트래픽 파라미터

	σ(패킷)	ρ(패킷/slot)	지연 상한 요구값 (D_i : ms)
클래스1	2~4	0.02	50
클래스2	6~12	0.02	280
클래스3	10~19	0.02	800
클래스4	14~27	0.02	1100

표 1에 주어진 클래스별 지연요구사항과 트래픽 파라미터, 출력 서버의 용량 등은, EDD 및 CSL이 갖는 스케쥴링 가능영역[5][6] 내에 포함됨을 확인하였다. 이러한 시험환경이 적용된 시뮬레이션을 통해 측정된, 클래스1 내의 임의의 세션의 지연분포를 그림 3에 도시하였다. EDD와 CSL은 클래스1의 지연 상한 값을 만족시킨 반면, FIFO는 사실상 클래스의 구분 없이 50 슬롯을 상회하는 지연 값을 나타냄으로서, 표 1에 주어진 트래픽 부하에 대해 특별한 제어가 수반되지 않으면 클래스1의 지연 요구가 만족될 수 없음을 보여주고 있다.

스케쥴링 알고리즘이 제공하는, 트래픽 간 지연성능의 안정성을 살펴보기 위하여 다음과 같은 두 번째 시뮬레이션을 수행하였다. 즉, 40개의 세션 중, 주어진 (σ, ρ)-제한조건을 준수하지 않는 하나의 비정상적인 세션을 설정하고, 이러한 세션이 타 세션의 지연성능에 미치는 영향을 관측하였다.

주목할 점은, 인터넷의 환경에서 트래픽의 흐름을 제어하는 TCP 혼잡제어 메커니즘을 준수하지 않고

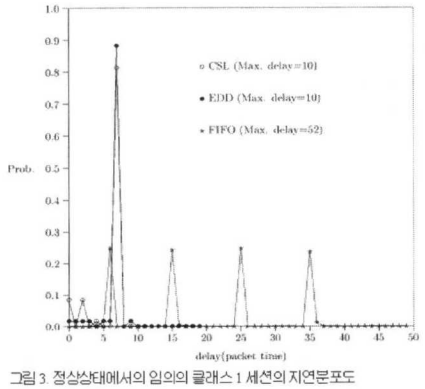

그림 3. 정상상태에서의 임의의 클래스 1 세션의 지연분포도

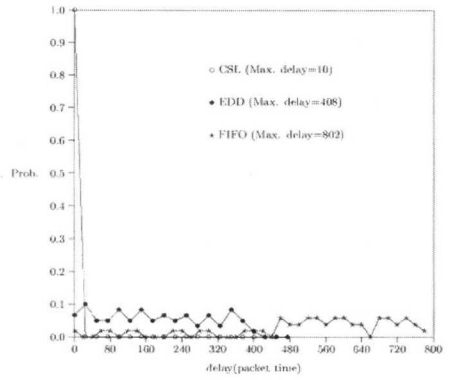

그림 4 비정상 버스트 트래픽에 대한 임의의 클래스 1 세션의 지연분포도

과다하게 데이터를 생성, 인입하는 세션이 얼마든지 있을 수 있다는 점에서, 이와 같은 상황에서의 지연성능 분석은 매우 실질적인 이슈로 인식된다. 본 시뮬레이션에서는, 특정의 비정상적인 세션이 1,000 슬롯 동안 지속적으로 패킷을 생성하도록 하였으며, 이 세션이 속한 클래스는 클래스 1부터 클래스 4까지 번갈아서 시뮬레이션을 수행하였다. 비정상적인 세션이 클래스 1에 속한 경우, 그림 3의 지연분포를 보였던 세션이 갖게 되는, 변화된 지연 분포를 그림 4에 도시하였다. 비정상적인 세션이 다른 클래스에 속한 경우에도 마찬가지의 결과를 확인할 수 있었으므로 생략한다. 비정상적인 세션은, 시스템이 안정화된 시점 이후, 2,000번째 슬롯부터 3,000번째 슬롯까지 매 슬롯마다 패킷을 생성하도록 하였다. 한편, 각 세션별 지연값의 취득은 안정화 시점부터 5,000 슬롯까지의 시간동안 측정된 것이며, 따라서 이 기간 중간에 발생한 비정상적인 세션에 의한 영향을 충분히 반영한 결과를 얻을 수 있었다. 그림 4에 따르면, EDD와 FIFO의 경우 클래스1에 요구된 지연 상한 값을 과도하게 초과하고 있음을 볼 수 있는 반면, CSL은 비정상적인 세션으로부터의 영향을 전혀 받지 않고 있음을 확인할 수 있다. 이는 곧, 특정 세션의 비정상적인 동작에 대한 독립성이 CSL알고리즘에서는 명확히 지원되고 있음에 반하여, EDD와 FIFO방식에서는 전혀 제공되고있지 않음을 보여주는 것이다.

그림 5부터 그림 7까지는, 위와 동일한 환경의 시뮬레이션 결과로서, 비정상적인 세션으로부터 발생되는 데이터의 양이 누적되는 시간 흐름에 맞추어, 각 클래스별로 겪게되는 최대 지연값이 변화하는 추이를, 스케쥴링 방식별로 도시하였다. 시스템 안정화 시점 이후 2,000번째 슬롯으로부터 3,000번

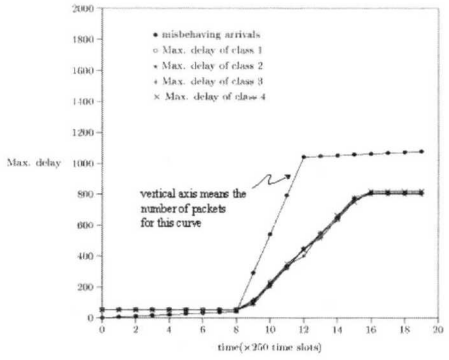

그림 5. 비정상 버스트 트래픽에 대한 클래스별 최대지연 값 변화 FIFO의 경우

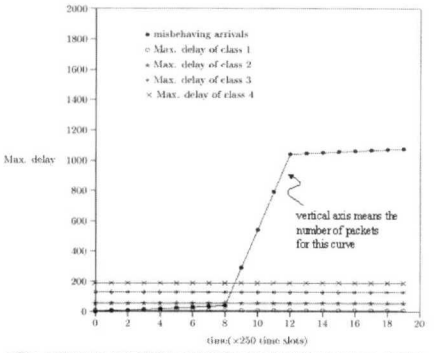

그림 7. 비정상 버스트 트래픽에 대한 클래스별 최대지연 값 변화 CSL의 경우

째 슬롯까지 비정상적으로 증가하는 특정세션의 트래픽 커브와 비교되도록, FIFO, EDD, 그리고 CSL 방식이 주는 결과를, 비정상적인 기간을 포함하는 5,000 슬롯 동안 취합하여 각각 그림 5, 그림 6, 그리고 그림 7에 도시하였다.

그림 5와 6으로부터, FIFO와 EDD방식에서는 비정상적인 세션의 영향이 모든 클래스에게 매우 민감하게 작용됨을 볼 수 있으며, 결과적으로 지연 요구값이 보장되지 못함을 확인할 수 있다. FIFO방식은 클래스에 무관하게, 비정상적 데이터가 모든 세션에 고르게 영향을 미치고 있음을 보여주고 있으며, EDD방식은 클래스간의 차별화는 유지하지만 비정상적인 트래픽의 영향으로부터 독립성 기능을 지원하지 못함을 보여주고 있다. 이에 반하여, 그림 7에서 보듯이, CSL알고리즘은 클래스간의 차별화 지원은 물론, 비정상적인 세션에 대한 독립성이 명확하게 지원되고 있음을 보인다. 이는 곧, Virtual Clock알고리즘으로부터 출발한, 제반 공평큐잉 방식이 갖고 있는 세션간의 독립성이 CSL알고리즘에서도 그대로 유지되고 있음을 의미한다.

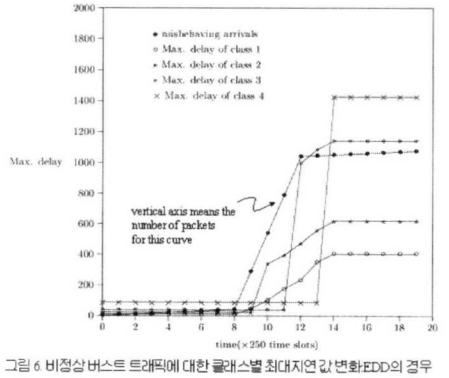

그림 6. 비정상 버스트 트래픽에 대한 클래스별 최대지연 값 변화 EDD의 경우

이와 같은 독립성의 또 다른 측면으로서, 위의 비정상적인 세션은 어떠한 지연 성능을 갖는지를 살펴보았다. 시뮬레이션 결과, FIFO에서는 802 슬롯, EDD에서는 394 슬롯, 그리고 CSL방식에서는 3,525 슬롯이 됨을 확인하였다. 이 세션이 클래스1에 속한다는 점에 비추어, 그림 5, 6, 7에 비교해보면 다음과 같은 정리를 할 수 있다. FIFO방식에서는, 특정 세션의 비정상성에 대한 대가가, 비정상적인 세션을 포함한 모든 세션에게 균등하게 작용하며, 따라서 결과적으로 비정상적인 세션이 이득을 보게된다는 기존의 FIFO방식에 대한 평가가 확인되었다. EDD방식의 경우에도, 동일한 클래스 내에서는 FIFO와 다를 것이 없다는 2장에서의 언급과 같이, 비정상적인 세션을 포함한 클래스1의 모든 세션에게 비정상성의 대가가 고르게 적용되고 있음을 확인할 수 있다. 반면에, CSL알고리즘에서는, 비정상성의 대가가 오직 비정상적인 세션에게만 적용되고 있다는 점에서, 독립성의 적절한 지원이 다시 한 번 확인된다.

3.2 구현의 단순화에 따르는 성능 분석

CSL알고리즘은, 기존의 PGPS알고리즘이 갖고 있는 구현의 복잡도(complexity)를 그대로 갖고 있다. 즉, 패킷의 태그 값을 계산하는 식(4)에서, 공평한 서비스의 제공을 위해 지속적으로 추적해야 하는 함수 $v(t)$를 구하기 위해 $O(N)$의 복잡성을 갖고 있는 것이다[9]. 이는, 시스템에 존재하는 세션의 총 개수 N에 비례하는 복잡도로서, 흐름 모델에서 정의된 GPS방식을 가상적으로 에뮬레이션 할 때, 임의의 기간동안에 식(2)의 세션집합 B 가 최대 N번의 변화를 가질 수 있음에 근거하여 주어진 복잡도이다.

그러나, 매우 엄격한 계산 과정을 통해 함수 $v(t)$에 대한 정확한 추적이 가능하더라도, 흐름 모델의 GPS에 비해 패킷 모드의 시스템 운용에서 갖는 공평성의 오차는 피할 수 없다[3]. 따라서, 이러한 오차가 피할 수 없는 것이라면, 공평성 오차를 줄이려는 노력보다는 GPS에뮬레이션이 갖는 복잡성을 줄임으로써, 어느 정도의 허용된 오차 내에서 구현의 가능성을 높이려는 노력이 GPS이후 매우 다양하게 이루어져 왔다[9,10,11]. 특히, 세션간의 공평성 자체 보다는 세션간의 지연성능 안정성 보장에 주안점을 두고 있는 본 논문에서는, 미세적 공평성 보장보다는, 거시적 척도에서 여전히 공평을 유지하면서 그 구현을 가장 간단($O(1)$)한 방식으로 구현한 SCFQ (Self-Clocked Fair Queueing) 방식[9]을 적용하여, $v(t)$추적이 불필요한 형태의 CSL 알고리즘 구현방안을 검토하였다. 즉, 패킷별 태그값 계산 과정에 SCFQ방식을 활용, 관계식 (4)를 다음과 같은 간단한 형태로 변경하였다.

$$F_{i,j}^{k} = \frac{L_{i,j}^{k}}{\rho_{i,j}} + \max[b(a_{i,j}^{k}) + \sum_{m=1}^{i} \Delta_m, F_{i,j}^{k-1}] \quad (6)$$

여기서 $b(a_{i,j}^{k})$ 는, 시간 $t = a_{i,j}^{k}$ 에서 서비스를 받고 있는 패킷의 태그 값을 나타낸다.

이 값의 산출은 $O(1)$의 복잡성을 가지며, 이러한 단순성이 SCFQ의 핵심이다. 이와 같이, 함수 $v(t)$를 변수 $b(a_{i,j}^{k})$ 로 대체하여 구현한 CSL알고리즘에서, 세션간의 독립성이 여전히 보장되고 있는지에 대한 검토는, 그 구현의 단순성에 비추어 매우 흥미로운 이슈이다. 따라서, 위의 식 (6)이 적용된, $O(1)$의 복잡성을 갖는 CSL알고리즘에 대하여 4장에서와 같은 시뮬레이션을 수행하였으며, 다음과 같은 결과를 얻었다.

그림 8에서는, 비정상적인 동작을 하는 세션의 영향에 따른 임의의 클래스 1세션의 지연분포를 보여주고 있다. 또한 그림 9에서는, 비정상적인 세션의 도착 패턴에 동기되어 각각의 클래스가 갖는 최대 지연값의 변화 추이를 도시하였다. 두 그림에서 볼 수 있듯이, 흐름모델에 기초한 $v(t)$ 대신에 단순한 구현 방안인 SCFQ의 기본적 개념을 적용하여도, CSL알고리즘은 세션간의 독립성을 명확하게 지원하고 있음을 확인하였다. 이로부터, 결정적 지연성능의 보장이 주요 QoS 요구사항인 상황에서, 세션간의 공평성 자체보다는 공평 큐잉이 갖는 기본 특성을 유지하여, 비정상적인 세션의 영향으로부터 차단된 결정적 지연 성능의 보장을 $O(1)$의 복잡성으로 구현할 수 있다는 점에서, CSL알고리즘의 우수성을 볼 수 있다.

이에 기초하여, CSL알고리즘의 전체적인 구현 복잡성을 다음과 같이 명시할 수 있다. 각각의 도착되는 패킷 단위의 복잡성은, 위에서 살펴본 바와 같이 SCFQ의 개념을 적용하는 방법으로 $O(1)$에 그친다. 반면에, 새로운 세션의 시스템 참여에 따르는 호수락제어 기능의 구현에는 Δ값의 변경에 소요되는 복잡성이 요구되어진다. Δ값은 클래스별 서비스 제공의 차별성을 규정하는 파라미터로서, 서버를 공유하는 세션들의 집합에 변화가 있을 때에만 새롭게 계산되어지는 변수이다. 이는 [6]에서 보인 바와 같이 $O(N)$에 이르며, 이와 같은 호 수락제어 기능은 스케줄링 알고리즘의 구현과는 별개의 기능으로서, line-speed로 이루어져야 하는 제약이 적용되지 않는다.

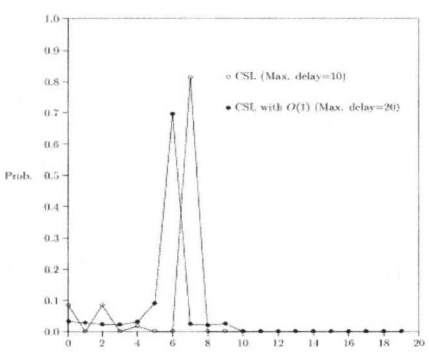

그림 8 CSL 구현방안별 비교 비정상 버스트 트래픽 발생시 임의의 클래스 1 세션의 지연분포도 비교

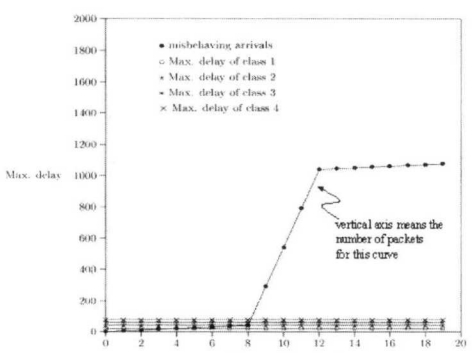

그림 9 비정상 버스트 트래픽에 대한 클래스별 최대지연값 변화 $O(1)$의 복잡도로 구현된 CSL의 경우

Ⅳ. 서비스 가용 영역 검토

본 장에서는, EDD방식과 CSL알고리즘에 수반되는 호수락제어 기능에 의해 수용되어질 수 있는 서비스영역, 즉 서비스 제공범위를 비교하였다. 단순히 결정적 지연성능의 보장이라는 목적아래에서는 EDD가 최적의 서비스 영역을 확보하고 있음이 증명되어 있다[5]. CSL알고리즘이 EDD방식과 다른 점 중의 하나는, 각각의 세션별 서비스커브를 서비스 목표함수로 정의하지 않고, 각 클래스내의 세션별 서비스커브 중 최대의 서비스 커브를 해당 클래스에 대한 서비스 목표함수로 정의하여, 클래스 레벨의 서비스를 제공하는 것이다. 따라서, 동일 클래스내의 세션별 서비스 커브의 편차가 클수록 비최적화 요인이 커지게 되는 것이다. 본 논문에서 확인한, 세션간의 지연성능에 대한 독립성은 이러한 비최적화된 자원활용의 대가로 지원되는 것으로 볼 수 있다.

주목할 점은, EDD방식에서와 같이 세션 레벨의 서비스커브를 목표함수로 갖고, 세션간의 독립성까지도 지원하는 알고리즘으로서, 참고문헌[12]에서 제시된 SCED(Service Curve based Earliest Deadline first) 알고리즘이 있다는 점이다. SCED에서는, 각 세션별로 요구되는 고유의 서비스 커브를 세션 단위로 각각 보장하므로써, 위의 두 문제, 즉 결정적 지연 성능의 보장 및 세션간 독립성의 제공을 동시에 해결하고 있다. 즉, CSL알고리즘과 동일한 기능을, 최적의 수용 가능 서비스영역을 갖고 지원하는 것이다. 그러나, 이러한 방법은 세션 단위의 서비스 커브의 추적을 요구하고 있으며, 이를 위해 sorting 기능 이외에도 패킷단위로 처리되어야 하는 스케줄러 자체의 복잡성이 증가하는 문제점을 갖는다. 특히, 인터넷에서의 QoS 제공 구조로서 적용될 Diff/Serv 방식이, 개별 세션이 아닌 트래픽 클래스에 기반한 차별적 서비스 제공의 특성을 갖는다는 점에서, SCED방식은 그 적용의 한계를 갖는 알고리즘으로 평가된다.

이에 반하여, CSL알고리즘에서는 SCFQ 방식의 적용시 $O(1)$으로 스케줄링 기능이 구현될 수 있으며, 이는 세션 단위의 서비스 커브가 아닌 클래스 단위의 서비스 커브를 목적함수로 다루는 특성에 기인하고 있다. 또한, 최선의 환경으로서, 각 클래스내의 세션들이 동일한 파라미터로 (σ, ρ)-제한조건을 만족시킬 때, 클래스 단위의 서비스커브와 그 클래스에 속한 각 세션의 서비스 커브가 동일하게 되므로, CSL은 EDD 또는 SCED와 동일한 서비스 영역을 갖게 된다. 즉, 클래스 단위로 동일한 특성의 트래픽 환경에서는 CSL도 최적의 서비스 영역을 확보하게 된다.

Ⅴ. 결 론

본 논문에서는, 결정적 지연성능의 보장 및 세션간 성능의 안정성이 보장되는 CSL 알고리즘의 기능 검증과, 그 구현에 관련된 이슈를 분석하였다. 시뮬레이션을 통하여 CSL알고리즘의 기능 및 성능의 적합성을 확인하였다. 클래스 단위의 서비스 커브를, 차별화된 패킷 스케줄링에 대한 목적 함수로 정의하였으며, 이러한 측면에서 EDD, SCED와 접근방법을 달리하고 있다. 특히, 이들과는 달리 공평큐잉의 기본 골격을 유지하므로써, EDD에서 결여된 세션간 성능의 안정성이 보장될 수 있도록 하였다. 또한, CSL알고리즘 구현의 복잡성에 대한 검토에서, SCFQ가 갖는 공평성 척도 내에서의 세션간 공평성을 유지하면서, 단지 $O(1)$의 복잡성으로, 위의 모든 디자인 요구사항이 만족되도록 구현될 수 있음이 확인되었다. 다만, 세션간의 지연 성능의 안정성 보장은 대가 없이 얻을 수 있는 기능은 아니며, 서비스 영역의 손실에 그러한 대가가 반영되고 있음을 논하였다.

TCP/IP환경에서, 각 사용자의 자발적 준수 여부에 의존할 수밖에 없는 트래픽 생성 구간에서의 제어의 한계로 인하여, TCP의 혼잡제어 메커니즘에 따르지 않는 비정상적인 트래픽 소스는 충분히 예견될 수 있음을 감안할 때, CSL과 같이 세션간 서비스 성능의 안정성이 간단히 구축될 수 있는 스케줄링 알고리즘의 활용성은 매우 높을 것으로 예상할 수 있다. 미시적 측면에서의 공평성의 손상은, Diff/Serv와 같이 상대적 방식으로 QoS가 지원되는 인터넷 환경에서는 주요 이슈가 되지 않으며, 클래스 레벨의 제어 구조를 갖는 CSL방식이 실제 네트워크에 적용되기에 알맞은 특성으로 평가될 수 있다.

참 고 문 헌

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "Architecture for Differentiated Services", RFC 2475, IETF,

Dec. 1998.

[2] L. Zhang, "Virtual Clock: A new traffic control algorithm for packet switching", ACM Transactions on Computer Systems, vol. 9, no. 2, pp. 101-124, May 1991.

[3] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case," ACM/IEEE Trans. Networking, vol. 1, no. 3, pp. 344-357, June 1993.

[4] H. M. Goldberg, "Analysis of the earliest due date scheduling rule in queueing systems", Math. Oper. Res., vol. 2, no. 2, 1977.

[5] Jorg Liebeherr, D. W. Wrege and D. Ferrari, "Exact admission control for networks with a bounded delay service", IEEE Trans. Networking, vol. 4, no. 6, pp. 885-901, Dec. 1996.

[6] 정대인, "클래스별 서비스 차등제공을 통한 패킷 지연 제어", 한국통신학회논문지, 제26권, 제6A호, pp. 1008-1018, June, 2001.

[7] R. L. Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation," IEEE Trans. Information Theory, vol. 37, no. 1, pp. 114-131, Jan. 1991.

[8] N. Figueira and J. Pasquale, "An Upper Bound on Delay for the Virtual Clock Service Discipline," ACM/IEEE Trans. Networking, vol. 3, no. 4, pp. 399-408, Aug. 1995

[9] S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications," in Proc. IEEE INFOCOM'94, Toronto, Canada, June 1994.

[10] J. Bennet and H. Zhang, "WF^2Q : Worst-case fair weighted fair queueing," in Proc. IEEE INFOCOM'96, pp. 120-128, April 1996.

[11] D. Stiliadis and A. Varma, "Latency-rate servers:A general model for analysis of traffic scheduling algorithms," in Proc. IEEE INFOCOM '96, pp. 111-119, April 1996.

[12] H. Sariowan, R. L. Cruz, and G. C. Polyzos, "SCED:A generalized scheduling policy for guaranteeing quality-of-service", ACM/IEEE Trans. Networking, vol. 7, no. 5, pp. 669-684, Oct. 1999.

정 대 인(Daein Jeong) 정회원

1984년2월: 서울대학교 제어계측공학과 졸업
1986년2월: 서울대학교 제어계측공학과 석사
1987년5월~1999년8월: 한국통신 선임연구원
1998년1월: Polytechnic Univ. 박사(E.E)
1999년9월~2000년8월: 홍익대학교 과학기술대학 조교수
2000년9월~현재: 한국외국어대학교 정보산업공과대학 조교수

<주관심 분야> IP와 ATM에서의 QoS, 차세대인터넷, 초고속 라우터, 멀티미디어통신