

# 인터넷에서 혼잡 제어를 위한 라우터 메커니즘

정회원 박재성\*, 신용철\*, 박희윤\*, 이재용\*

## A Router Mechanism for Congestion Control in Internet

Jaesung Park\*, Yong-chul Shin\*, Heui-youn Park\*, Jai-yong Lee\* *Regular Members*

### 요약

본 논문에서는 인터넷 트래픽의 자기 유사성 특성을 이용한 큐 관리 기법을 제안한다. 제안 기법은 LRD 프로세스의 특성을 이용하여 큰 타임 스케일에서 다음 혼잡 검출 주기 동안 평균 트래픽 입력율을 예측하고 이를 통해 혼잡 발생 여부를 미리 검출한다. 혼잡 검출시 입력 패킷을 확률적으로 폐기함으로써 망 내부에서 혼잡 발생을 사전에 방지한다. 제안 기법은 중단 시스템의 혼잡 제어 기법이나 각 플로우의 통신 환경에 무관하게 라우터의 평균 큐 길이와 큐 길이 변이를 작게 유지할 수 있으므로 지연과 지연 변이에 민감한 멀티미디어 서비스 제어에 적합하다. 또한 제안 기법은 입력율 예측을 통해 혼잡을 사전에 검출하기 때문에 중단간 혼잡 제어 기법을 사용하지 않는 플로우들에 의해 발생된 패킷들이 망 자원을 불평등하게 많이 사용하는 현상을 방지한다. 제안 기법은 각 플로우별 상태 정보를 관리하지 않으므로 확장성이 뛰어나며 이로 인해 소수의 PHB에 따라 각 홉별로 차등화된 서비스를 제공하는 Diffserv 구조에 적합하다. 또한 제안 기법은 기존 인터넷 구조에 변화를 주지 않고 라우터의 소프트웨어 업그레이드만으로 서비스가 가능하며, 필요한 버퍼의 양도 크지 않다는 장점을 가진다.(스타일-요약본문)

### ABSTRACT

In this paper, we propose an active queue management method which detect incipient congestion at a large time scale by using the predictability of an aggregate Internet traffic process. If an incipient congestion is detected, a router tries to avoid congestion by discarding incoming packets with packet discard probabilities which is set by the severity of the predicted incipient congestion. The proposed method can maintain an average queue length and its variation small regardless of the congestion control method in end systems and RTTs of flows. Moreover, the proposed method does not have bias against bursty flows and can protect well-behaving sources from non-adaptive sources by dropping packets from non-adaptive sources more aggressively. Therefore the proposed scheme can give incentives for the deployment of applications incorporating end-to-end congestion control mechanisms. This drives the network to desirable, congestion free operating regimes. The proposed scheme is scalable because it can provide QoS mechanism without maintaining per flow state. It is evolutionary because it can be easily deployed in current routers merely upgrading software and it is economic in that it does not require large buffer space.

### I. 서론

인터넷 서비스는 IP를 기반으로 하는 중단간 비연결형 패킷 서비스로 이루어져 왔다. 각 패킷은 서로 독립적으로 전송되고 최선형 서비스(Best Effort)

라 불리는 단일 서비스만을 제공받으며 패킷 전송 제어와 같은 복잡한 기능들은 중단 시스템의 수송 계층 프로토콜이 담당한다. 이와 같은 인터넷 구조는 간단하고, 유연성이 뛰어나지만 자원 관리 기능이 없기 때문에 사용자는 망으로부터 제공받는 서

\* 연세대학교 전기,전자공학과 네트워크 연구실(jspark@nasia.yonsei.ac.kr)

논문번호 : 00432-1103, 접수일자 : 2000년 11월

※ 본 연구는 한국과학재단 연구과제(1999-2-303-005-3) 지원으로 수행되었습니다.

비스의 품질을 예측할 수 없다. 특히 네트워크의 부하가 증가할수록 네트워크가 제공하는 서비스 품질은 급격히 하락하게 되고 최악의 경우 네트워크 전체가 마비되어 서비스 제공이 불가능해진다. 이와 같은 현상은 인터넷 초기에 발견되었으며<sup>[1]</sup> 네트워크 혼잡이라 불린다. 인터넷에서 혼잡 제어는 TCP 프로토콜이 담당해 왔다. TCP 혼잡 제어 방식은 패킷 손실을 검출할 때까지 점차적으로 전송율을 증가시키고 패킷 손실 검출시 전송율을 감소시키는 방법을 취한다<sup>[2]</sup>. TCP 혼잡 제어 방법은 네트워크 혼잡 해소에 중요한 역할을 수행해 왔으나 다음과 같은 문제점들을 가진다.

TCP 혼잡 제어 방법은 네트워크에서 혼잡이 발생한 후에 혼잡을 해소시키기 위한 방법이므로 혼잡이 해소되기까지 네트워크는 적절한 수준의 서비스 품질을 제공할 수 없다. 또한 이와 같은 방법은 모든 사용자들이 TCP 혼잡 제어 기법을 사용해야 한다는 것을 가정하고 있다. 최근에 등장하고 있는 인터넷 전화(VOIP)나 Realplayer와 같은 멀티미디어 서비스들은 수송 계층으로 TCP와 같은 종단간 혼잡 제어 기능을 사용하지 않는다. 이와 같은 플로우들과 TCP를 기반으로 한 플로우들이 동시에 사용되는 경우 혼잡이 발생하면 각 플로우들이 전송한 패킷들이 손실된다. 패킷이 손실 되는 경우 TCP 기반 플로우들은 전송율을 감소시키지만 종단간 혼잡 제어 기능을 사용하지 않는 플로우들은 전송율을 그대로 유지하므로 결과적으로 혼잡 제어 기능을 사용하지 않는 플로우들이 병목 지점의 자원을 독점하게 된다. 또한 일부 사용자들이 TCP 혼잡 제어 기능을 변경하여 네트워크 혼잡 신호를 무시하고 데이터 전송율을 계속 일정하게 유지하거나 증가시키게 되면 네트워크 혼잡 발생시 다른 사용자들이 전송율을 낮추어서 얻어지는 잉여 대역폭을 그렇지 않은 사용자들이 차지하게 되는 불공정한 현상이 발생한다.

따라서 플로우들이 종단간 혼잡 제어를 이용하도록 하기 위해서는 네트워크 내에서 이와 같은 불공정한 현상을 방지하기 위한 기법이 필요하다. 망 내에서 혼잡을 제어하기 위한 방법은 크게 각 플로우의 상태 관리를 통한 패킷 전송 순서를 제어하는 방법<sup>[14]</sup>과 다중화된 플로우들의 입력 여부를 결정하는 방법<sup>[6]</sup>으로 구분된다.

전자의 방법은 각 플로우별 개별 큐를 두고 이들의 전송 순서를 제어함으로써 각 플로우별로 대역폭을 보장함으로써 불공정한 대역폭 사용을 방지하는 방

법이다. 이와 같은 스케줄링 기법은 플로우들을 격리시켜서 처리하기 때문에 각 플로우별로 공정한 대역폭 할당이 가능하지만 평균 큐 길이에 대한 제어가 불가능하고 각 플로우별 개별 큐를 유지 관리해야 하기 때문에 확장성이 떨어진다는 단점을 가진다.

두 번째 방법은 같은 클래스에 속하는 모든 플로우들은 동일 큐를 공유하며, 예측되는 혼잡 정도에 따라 입력 패킷을 선택적으로 수용하는 방법으로서 큐 관리 기법이라 불린다. 큐 관리 기법은 혼잡을 예측하여 사전에 패킷을 폐기시키기 때문에 평균 큐 길이를 일정하게 유지할 수 있다. 따라서 큐 관리 기법은 각 플로우의 상태 관리 없이 각 패킷의 평균 큐잉 지연을 제어할 수 있으므로 확장성이 우수하다. 그러나 큐 관리 기법은 각 플로우별로 상태 정보를 관리하지 않으므로 공평한 대역폭 할당을 보장하지 못한다.

본 논문에서는 네트워크 혼잡 방지를 통해 평균 큐 길이와 평균 큐 길이의 변이를 제어하며, 플로우별 상태 관리 없이 공정한 대역폭 할당이 가능한 큐 관리 기법을 제안한다. 본 논문에서 제안하는 큐 관리 기법은 입력 트래픽의 자기 유사성 특성<sup>[10][11][13]</sup>을 이용한다. 자기 유사성 프로세스는 LRD 특성을 가지며 이와 같은 특성을 이용하면 큰 타임 스케일에서 향후 평균 입력 트래픽을 예측할 수 있다. 이처럼 예측된 평균 입력 트래픽을 이용하면 향후 혼잡 발생 가능성과 그 정도를 예측할 수 있으며 이를 이용하여 입력 패킷을 확률적으로 폐기시킴으로써 혼잡을 회피한다. 제안 기법은 종단간 혼잡 제어 기능을 사용하지 않는 플로우들의 패킷을 보다 많이 폐기시킴으로써 종단간 혼잡 제어 기능을 사용하는 플로우들과 그렇지 않은 플로우들이 네트워크를 공유하는 경우 각 플로우별 공평성을 제공한다. 제안 기법은 각 플로우별 상태 정보를 관리하지 않으므로 확장성이 뛰어나며 이로 인해 소수의 PHB(Per Hop Behavior)<sup>[17][18]</sup>에 따라 각 홉별로 차등화된 서비스를 제공하는 Diffserv 구조<sup>[3]</sup>에 적합하다. 또한 제안 기법은 기존 인터넷 구조에 변화를 주지 않고 라우터의 소프트웨어 업그레이드만으로 서비스가 가능하며, 필요한 버퍼의 양도 크지 않다는 장점을 가진다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존 큐 관리 기법들의 장단점을 정성적으로 분석한다. 3장에서는 본 논문에서 이용하고 있는 LRD 프로세스의 특성을 이용한 향후 평균 트래픽 입력을 예측

방법을 수학적으로 정의한다. 4장에서는 제안하는 큐 관리 기법의 알고리즘과 구현 가능성을 설명하고 5장에서는 모의 실험을 통해 제안 기법의 성능을 평가한 후 6장에서 결론을 맺는다.

## II. 관련 연구

큐 관리 기법은 큐가 포화된 이후 패킷을 폐기하는 drop-on-full 기법과 큐가 포화되기 이전에 패킷을 폐기하는 random early drop 기법으로 구분할 수 있다. drop-on-full 큐는 큐가 포화될때까지 패킷을 수용하고 이후의 패킷을 폐기시키는 기법으로서 패킷을 폐기하는 위치에 따라 drop-tail, drop-front, random drop 기법으로 나누어진다. FIFO 큐는 drop-tail 큐의 대표적인 예로서 큐가 포화되면 여유 공간이 생길 때까지 이후 입력 패킷들을 폐기시킨다. FIFO 큐는 구현이 간단하여 지난 수십년간 인터넷 라우터에 사용되어 오고 있으나 소수의 플로우가 큐를 독점하게 되고 큐가 항상 포화 상태 근처에서 유지되는 등의 많은 단점들을 가지고 있다<sup>[4][5]</sup>. drop-front 큐와 random-drop 큐는 FIFO 큐의 변형이다. 큐 포화시 패킷이 입력되면 drop-front 큐는 큐 앞단의 패킷을 폐기하고 random-drop 큐는 큐 내의 임의의 패킷을 폐기하여 입력 패킷을 수용한다. 이와 같은 큐들은 패킷을 보다 빨리 폐기시킴으로써 TCP의 FRTX(fast Retransmit) 시간을 단축시키는 것을 목적으로 하지만, 실제의 경우 큐 관리를 위한 부하에 비해 성능 향상이 크지 않으며 큐 길이가 항상 최대 큐 길이 근처에서 유지된다는 단점을 가진다. 버퍼가 항상 포화 상태 근처로 유지되는 경우 패킷이 버스터하게 입력되면 다수의 패킷들이 폐기되며 이로 인해 global synchronization<sup>[4]</sup> 문제가 발생되어 네트워크 이용률이 심하게 진동하게 된다.

Random Early drop 기법<sup>[6][7]</sup>은 큐가 포화되기 전에 패킷을 폐기시킴으로써 평균 큐 길이를 일정하게 유지시키며 버스티한 패킷 입력을 수용하여 전체적인 패킷 손실율을 감소시키는 것을 목적으로 한다. ERD 기법<sup>[15]</sup>은 큐 길이가 임의의 임계값을 초과하면 고정 확률로 입력 패킷을 폐기시킨다. 그러나 ERD 기법은 혼잡 신호에 반응하는 플로우와 그렇지 않는 플로우를 구분하지 못하며 이로 인해 혼잡 신호에 따라 전송율을 조절하지 않는 플로우들이 전송율을 조절하는 플로우들보다 많은 양의 큐를 차지하게 된다. RED<sup>[6]</sup>는 drop-tail 큐의 단점을 극복

하고 혼잡 회피 기능을 제공하기 위해 제안되었다. RED는 평균 큐 길이를 계산하여 이를 최대 및 최소 임계값과 비교한다. 만일 평균 큐 길이가 최소 임계값 이하이면 입력 패킷은 폐기되지 않으며 평균 큐 길이가 최대 임계값 이상이면 모든 입력 패킷은 폐기된다. 반면 평균 큐 길이가 최대, 최소 임계값 사이이면 평균 큐 길이에 따라 패킷 폐기 확률이 결정되며 이에 따라 입력 패킷의 폐기여부가 확률적으로 결정된다. 이상적인 환경에서 RED는 평균 큐 길이를 작게 유지하고 global synchronization 문제를 야기 시키지 않는다. 그러나 RED는 자기 유사성 프로세스와 같이 입력 트래픽 레벨이 급격하게 변하는 경우 적응성이 떨어지며, 이상적인 동작을 위해서는 큰 큐가 필요하고, 여러 파라미터들이 혼잡 상황에 최적화 되어야 한다<sup>[7]</sup>. 또한 RED는 자원을 공유하는 플로우들별로 공정한 대역폭 할당을 할 수 없다<sup>[8][16]</sup>.

본 논문에서 제안하는 큐 관리 기법은 입력 트래픽의 자기 유사성 특성을 이용한다. 혼잡 검출은 일정 주기마다 큰 타임 스케일에서 이루어진다. 각 주기의 끝에서 자기 유사성 트래픽 특성에 따라 다음 주기 동안의 평균 입력율을 예측하고 이에 따라 혼잡 여부를 결정하게 된다. 제안 기법은 큰 타임 스케일에서 혼잡 발생 이전에 패킷을 폐기시키므로 네트워크 혼잡 상황에 들어가는 것을 방지 할 수 있으며, 이로 인해 필요한 큐 길이를 줄일 수 있고 평균 큐 길이와 큐 길이 변화폭을 감소시킬 수 있다. 또한 혼잡 검출은 입력율을 기반으로 이루어지므로 혼잡 신호에 따라 전송율을 조절하지 않는 플로우의 패킷들이 폐기될 확률이 커지며 이로 인해 각 플로우들별로 공정한 대역폭 할당을 할 수 있다.

## III. 평균 입력율 예측기

제안 기법은 입력 트래픽의 자기 유사성 특성을 이용하여 향후 평균 트래픽 입력율을 예측한다. 다양한 환경에서 트래픽 측정 및 분석 결과 인터넷 트래픽은 네트워크 상태, 시간, 트래픽 제어 기법 등에 무관하게 자기 유사성 특성을 가진다는 것이 증명되어 왔다<sup>[10][11][12][13]</sup>. 자기 유사성 프로세스는 LRD 특성을 가지며 LRD 특성을 이용하면 향후 트래픽의 평균 입력율을 예측할 수 있다. 이전 관측값에 기반하여 향후  $m$ 개의 관측값의 평균값을 예측하는 것은 매우 어려운 문제이며 최적의 예측기는 이전 관측값과 향후 관측할 값들의 연관 관계를 알고 있

어이만 하므로 현실적으로 구현 불가능하다<sup>9)</sup>. 따라서 본 논문에서는 최적의 예측기가 아닌 구현이 간단한 다음과 같은 예측기를 이용한다.

$\{X_i; i \in Z_+\}$ 를 측정 데이터 트래픽을 나타내는 시간 열이라고 하고 aggregated series  $X_i^{(m)}$ 을 다음과 같이 정의한다. 여기서  $m$ 은 블록의 수를 나타내며  $i$ 는 몇번째 블록인가를 나타낸다.

$$X_i^{(m)} = \frac{1}{m}(X_{i-m+1} + \dots + X_{im}) \quad (1)$$

$X_i$ 에 대해 이전  $n$ 개의 관측값의 평균을  $\bar{X}_n$ 이라고 하고 향후  $m$ 개의 실제 평균 값을  $\bar{X}_m$  예측 평균을  $\hat{X}_m$ 이라고 하면, 예측 평균은 다음과 같이 각각의 과거 값에 대한 조건부 평균을 구하는 대신에 평균값에 대한 조건부 평균으로 주어진다.

$$\hat{X}_m = E[\bar{X}_m | \bar{X}_n] \quad (2)$$

이 기법은 시간 영역에서 이루어지며 Bayesian 예측법의 하나로 생각할 수 있다<sup>9)</sup>. 위의 조건부 평균을 계산하기 위해서는 조건부 확률  $\Pr\{X_i^{(m)} | X_i^{(n)}\}$ 을 구하여야 한다. 이를 구하기 위해 다음과 같은 변수를 정의한다.

$$V_1 = \sum_{i=1}^{T_1} X_i \quad (3-1)$$

$$V_2 = \sum_{i=1}^{T_2} X_i \quad (3-2)$$

이제  $\Pr\{X_i^{(m)} | X_i^{(n)}\}$ 을 구하는 문제는 다음과 같은 조건부 확률,  $\Pr\{V_2 = b | V_1 = a\}$ 를 구하는 문제로 다시 정리할 수 있다. 랜덤 변수  $V_1, V_2$ 는 각각 시 구간  $[t - T_1, t), [t, t + T_2)$ 동안의 트래픽 입력율이므로 문제를 보다 수월하게 하기 위해 이들의 평균과 분산을 이용하여 단조 증가 함수  $L: R_+ \rightarrow [1, \infty]$ 를 정의하고  $L_k = L(V_k)$ 로 매핑시킨다. 즉,  $L_k \approx 1$ 은 입력율이 낮은 것을 의미하며  $L_k \approx s$ 는 입력율이 높다는 것을 의미하며  $\Pr\{V_2 = b | V_1 = a\}$ 를 구하는 문제는  $\Pr\{L_2 = l_2 | L_1 = l_1\}$ 을 구하는 문제로 다시 쓸 수 있다. 이제 향후 평균 입력율은 다음과 같이 예측된다.

$$\hat{X}_m = E[\bar{X}_m | \bar{X}_n] = L^{-1}\left(\sum_{i=1}^m i \Pr\{L_2 = i | L_1 = L(\bar{X}_n)\}\right) \quad (4)$$

#### IV. 제안기법

제안하는 큐 관리 기법은 평균 트래픽 입력을 예측 기능과 혼잡 검출에 따른 패킷 폐기 확률 결정 기능으로 구성된다. 평균 트래픽 입력율은 큰 타임스케일에서 LRD 프로세스의 특성을 이용하여 III장에서 정의한 예측기를 통해 예측되며, 혼잡 검출을 통한 패킷 폐기 확률은 예측된 입력율과 서비스율의 비 및 평균 큐 길이 변화를 통해 결정된다. 혼잡 검출 기능은 일정 길이의 혼잡 검출 주기마다 수행된다. 매 혼잡 검출 주기의 끝에서 다음 주기 동안의 평균 입력율이 예측된다. 예측된 평균 입력율이 서비스율보다 크면 첫 번째 패킷 폐기 확률( $p_{mL}$ )이 예측되는 혼잡 정도에 따라 결정된다. 예측 과정에는 오류가 수반되며 예측 오류를 보정하기 위해 두 번째 패킷 폐기 확률( $p_{mS}$ )을 고안하였다. 두 번째 패킷 폐기 확률은 평균 큐 길이에 따라 결정되며 입력 패킷은 확률  $p_{mL} + p_{mS}$ 에 따라 수용 여부가 결정된다.

##### 1. 평균 트래픽 입력을 예측 알고리즘

다음 혼잡 검출 구간 동안의 평균 트래픽 입력율을 예측하기 위해 라우터는 크기  $L \times (L+1)$ 인 2차원 배열 CPDA(Conditional Probability Density Array)를 관리한다. 현재 혼잡 검출 구간( $T$ )의 끝에서 라우터는 현재 구간 동안의 트래픽 입력율  $V_1$ 을 계산하여 트래픽 레벨( $l_1 = L(V_1)$ )을 구한 후 배열 CPDA [ $l_1$ ] [ $L+1$ ]의 값을 하나 증가시킨다. 다음 구간의 끝에서 구간의 트래픽 입력율  $V_2$ 를 구하여 트래픽 레벨( $l_2 = L(V_2)$ )로 변환한 후 배열의 원소 CPDA [ $l_2$ ] [ $L+1$ ]의 값과 CPDA [ $l_1$ ] [ $l_2$ ]의 값을 하나씩 증가시키고  $l_1 = l_2$ 로 치환한다. 이와 같은 과정이 매 혼잡 검출 구간 끝에서 이루어진다. 이처럼 관리되는 CPDA에 따라 다음 구간에서의 트래픽 레벨은 다음과 같이 결정된다. 현재 트래픽 레벨이  $l$ 이고 구간에서의 트래픽 레벨이  $i$ 이 될 조건부 확률은 및 다음 구간에서의 입력 레벨은 다음과 같이 예측된다.

$$\Pr\{L_2 = i | L_1 = l\} = \frac{CPDA[i][l]}{CPDA[l][L+1]} \quad (5)$$

$$E[L] = \sum_{i=1}^L i \cdot \Pr\{L_2 = i | L_1 = l\} \quad (6)$$

그림 1은 조건부 확률 배열을 관리하고 다음 구간에서의 트래픽 레벨을 예측하기 위한 알고리즘을 보여준다. 이처럼 예측된 트래픽 레벨은 역변환 ( $\gamma_f = L^{-1}(E[L])$ )을 통해 예상 평균 트래픽 입력률 ( $\gamma_f$ )로 변환된다.

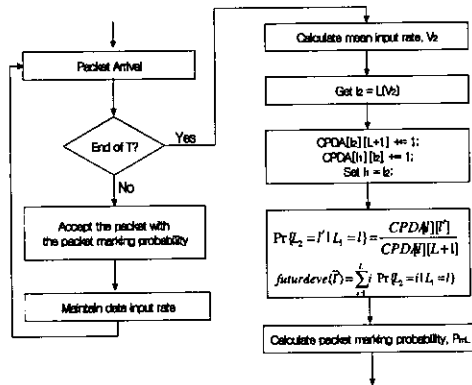


그림 1. 조건부 확률 배열 관리 및 다음 구간 동안의 트래픽 레벨 예측 알고리즘.

트래픽 레벨의 수는 제어의 정확성과 구현 비용과의 상충 관계를 가진다. 즉,  $L$  값이 커질수록 입력 트래픽을 보다 세밀하게 구분할 수 있으나 필요한 메모리 크기는  $L^2$ 에 비례하여 증가한다. 또한  $L$  값이 어느 정도 이상이 되면 세밀한 트래픽 수준 구분을 통해 얻을 수 있는 장점보다는 메모리를 관리하는데 필요한 비용이 더 커지게 된다. 본 논문에서는 이와 같은 정보를 기반으로 트래픽 레벨을 다음과 같이 8개로 구분하였다.

$$L_k=1 \Leftrightarrow V_k \in (-\infty, \mu_X - 3\sigma_X),$$

$$L_k=2 \Leftrightarrow V_k \in [\mu_X - 3\sigma_X, \mu_X - 2\sigma_X),$$

$$\dots$$

$$L_k=8 \Leftrightarrow V_k \in [\mu_X + 3\sigma_X, \infty)$$

## 2. 혼잡 검출을 통한 패킷 폐기 확률 결정 알고리즘

패킷 폐기 확률은 큰 타임 스케일에서 예측된 입력률과 서비스율로 결정되는  $p_{mL}$ 과 예측기의 오류 보정을 위해 작은 타임 스케일에서 평균 큐 길이에 의해 결정되는  $p_{mS}$ 의 합으로 결정된다. 그림 2는 패킷 폐기 확률을 결정하기 위한 알고리즘을 나타낸다.  $p_{mL}$ 은 다음 혼잡 검출 구간 동안의 혼잡 정도를 나타내며, 라우터의 패킷 폐기율을 결정한다.  $\gamma_f$ 를 다음 구간 동안 예측된 평균 입력률,  $c$ 를 서

비스 율이라 하고 이들의 비를  $\rho = \gamma_f/c$ 로 정의한다. 만일  $\gamma_f$ 가  $c$ 보다 작게 되면 다음 구간 동안  $p_{mL}$ 은 0으로 결정되고  $\gamma_f$ 가  $c$ 보다 크면  $p_{mL}$ 은 다음과 같이 결정된다.

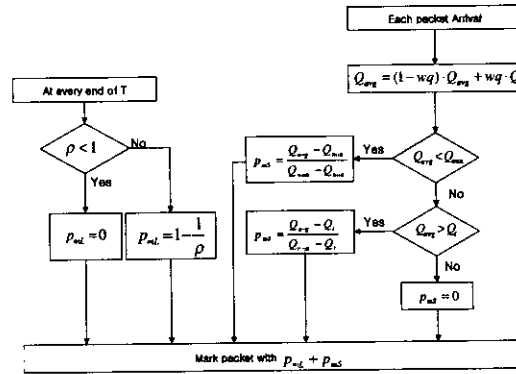


그림 2. 패킷 폐기 확률 결정을 위한 알고리즘.

$$p_{mL} = 1 - \frac{1}{\rho} \quad (7)$$

$p_{mS}$ 는 작은 타임 스케일에서 평균 큐 길이를 기반으로 결정된다. 시간  $t$ 에서의 순서 큐 길이를  $q_t$ 라고 하고, 평균 큐 길이를  $q_{avg}$ 라고 하면 평균 큐 길이는  $q_{avg} = (1-wq) \cdot q_{avg} + wq \cdot q_t$ 로 구해지며 이 값은 최소 큐 길이  $Q_{min}$ 과 시스템에서 지정한 논리적 최대 큐 길이 임계값  $Q_t$ 와 비교된다. 만일 평균 큐 길이가  $Q_{min}$ 보다 작게 되면 입력율을 너무 높게 예측하여 과도하게 패킷을 폐기시키는 경우로 판단되고 이를 보정하기 위해  $p_{mS}$ 는 다음과 같이 결정된다.

$$p_{mS} = \frac{Q_{avg} - Q_{min}}{Q_{max} - Q_{min}} \quad (8)$$

만일 큐 길이가 임계값  $Q_t$  이상이 되면 입력율을 너무 낮게 예측한 경우로 판단되고 이를 보정하기 위해  $p_{mS}$ 는 다음과 같이 결정된다.

$$p_{mS} = \frac{Q_{avg} - Q_t}{Q_{max} - Q_t} \quad (9)$$

따라서 라우터는 이와 같이 결정되는 패킷 폐기 확률들의 합 ( $p_{mL} + p_{mS}$ )에 따라서 입력 패킷들을 확률적으로 폐기시킨다.

### V. 성능평가

본 장에서는 모의 실험을 통해 제안 기법의 성능을 평가한다. 성능 평가의 주된 목적은 중단 시스템에서의 혼잡 제어 방식에 무관하게 평균 큐 길이와 큐 길이 변화를 작게 유지할 수 있다는 것과 네트워크로부터의 혼잡 통보에 전송율을 조절하는 플로우와 그렇지 못한 플로우가 병목 라우터를 공유하는 경우 각 플로우별 공평한 대역폭을 할당할 수 있다는 것을 보이는 것이다.

#### 1. 소스 모델

본 논문에서는 자기 유사성 트래픽을 발생시키기 위해 [19][13]에서 검증한 ON/OFF 모델을 사용하였다. ON 구간 혹은 OFF 구간의 지속 시간이 heavy-tailed 분포를 가지는 다수의 소스들을 중첩시키면 다중화된 트래픽은 자기 유사성을 나타낸다. 그림 3은 자기 유사성 트래픽을 발생시키기 위한 소스 모델을 나타낸다. 본 논문에서는 ON 구간과 OFF 구간 모두 heavy-tailed 분포를 나타내는 Pareto 분포를 따르도록 설정하였으며 Pareto 분포의 scaling 파라미터를  $\alpha=1.25$ 로 설정하였고 이와 같은 100개의 소스를 다중화 시켰다.

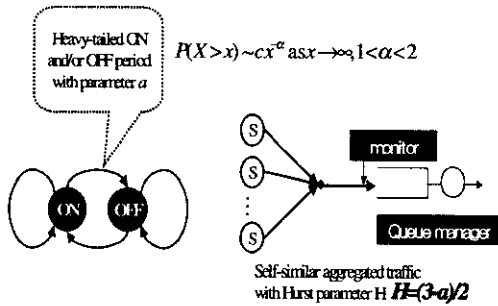


그림 3. 자기 유사성 트래픽 발생기.

#### 2. 혼잡 검출 구간

제안 기법에서 혼잡 검출 구간의 결정은 매우 중요한 요소이다. WAN 트래픽의 자기 유사성은 플로우의 최대 RTT 이상의 타임 스케일에서 나타난다. 따라서 혼자 검출 구간이 너무 작게 되면 LRD 프로세스의 predictability 특성을 이용하여 다음 구간에서의 평균 입력율을 예측할 수 없다. 반면에 혼잡 검출 구간이 너무 길게 되면 CPDA는 네트워크 상황을 반영하지 못하게 된다.

본 논문에서는 predictability가 최대가 되는 타임 스케일을 결정하기 위해 조건부 확률 밀도의 엔트로피를 계산하였다. 트래픽 입력 레벨을 8 단계로 구분한 경우 조건부 확률 밀도 함수의 엔트로피는 다음과 같이 정의된다.

$$entropy(E_{l_i}) = \sum_{i=1}^8 \frac{\Pr(L_2=l_2|L_1=l_1)}{\log(\Pr(L_2=l_2|L_1=l_1))} \tag{10}$$

조건부 확률이 균일 분포를 가지는 경우 엔트로피는 최대가 되고 분포가 단일 지점으로 편중될수록 엔트로피는 작아진다. 트래픽 레벨을 8단계로 나누었으므로 평균 엔트로피는 다음과 같이 결정된다.

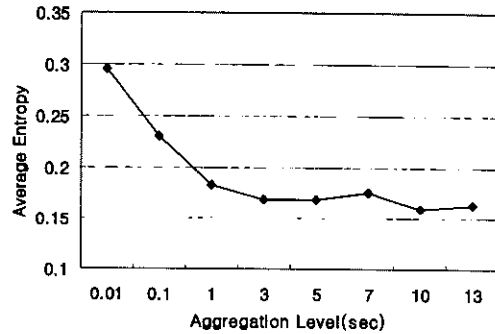


그림 4. 자기 유사성 트래픽의 평균 엔트로피.

$$\bar{E} = \sum_{i=1}^8 E_{l_i} / 8 \tag{11}$$

그림 4는 다중화된 소스의 평균 엔트로피를 나타낸다. 그림에서 보는 바와 타임 스케일이 커짐에 따라 평균 엔트로피는 감소하며 일정 시간 이후에는 거의 일정하게 유지되는 것을 볼 수 있다. 혼잡 검출 구간이 길어 질수록 네트워크 상황이 변화할 확률은 커지게 된다. 따라서 본 논문에서는 평균 엔트로피 곡선에 근거하여 모의 실험을 위한 혼잡 검출 구간을 1초로 정하였다.

#### 3. 평균 큐 길이 제어 기능

그림 5는 모의 실험 네트워크를 나타낸다. 모의 실험에서는 100개의 TCP Reno 소스가 단일 병목 라우터를 공유하는 경우를 설정하였다. 소스를 TCP Reno로 설정한 이유는 Reno 버전의 TCP가 현재 가장 많이 사용되기 때문이다. 모의 실험에서 모든 소스들은 항상 전송할 패킷이 있으며 패킷의 크기는 512 바이트로 가정하였다. 패킷 크기 512 바이트

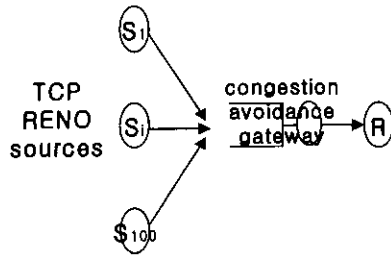


그림 5. 모의 실험 네트워크 I.

트는 PATH MTU Discovery 기능을 사용하지 않는 TCP의 기본 MSS (Maximum Segment Size)로서 대부분의 데이터 TCP 패킷의 크기는 512 바이트를 가진다<sup>[20][21]</sup>.

라우터의 버퍼 크기는 4000 패킷, 큐 길이의 임계값은 3200 패킷,  $Q_{min}$ 은 0 패킷으로 정하였으며 소스의 RTT는 100msec로 같게 설정하였다. RED 파라미터들(*minth*, *maxth*, *wq*, *map*)는 모의 실험에 최적이 되도록 설정하였다<sup>[6][7]</sup>.

그림 6은 제안 기법과 RED의 시간에 따른 평균 큐 길이 변화를 나타내며 표 1은 평균 큐 길이와 큐 길이 변화율을 나타낸다. 그림 6과 표 1에서 보는 바와 같이 라우터의 자원 용량이 같은 경우 제안 기법은 RED 보다 평균 큐 길이와 큐 길이의 변화 폭을 작게 유지시킨다. TCP 소스는 패킷 손실을 기반으로 네트워크 혼잡 발생을 통보 받는다. 따라서 네트워크에서 패킷 손실이 발생하고 이것을 해당 TCP 소스가 검출하여 전송율을 조절하기까지는 어느 정도 시간이 소요된다. RED는 매 패킷 입력시마다 평균 큐 길이 변화를 이용하여 패킷 폐기 확률을 결정한다. RED 큐는 혼잡 검출을 작은 타임 스케일에서 수행하므로 혼잡 검출 후 패킷을 폐기하고 해당 TCP 소스들이 전송율을 조절하는 동안 입력

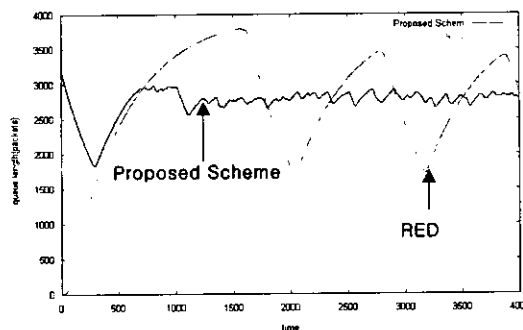


그림 6. 평균 큐 길이 비교

표 1. 평균 큐 길이와 큐 길이 변화도의 비교표

	제안 기법	RED
평균 큐 길이(패킷)	2802	2927
큐 길이 변화도(패킷)	78	580

되는 패킷들을 수용할 수 있어야 한다. 따라서 패킷을 폐기하고 송신단이 전송율을 감소시키지까지 RED 큐 길이는 증가하게 되며, 송신단이 전송율을 감소시키면 RED 큐는 감소하게 되고, 송신단이 TCP의 전송율 제어 기법에 따라 또 다시 전송율을 점진적으로 증가시키게 되면 이와 같은 현상이 반복된다. 따라서 RED 큐는 혼잡을 검출하고 송신단이 전송율을 감소시키기는 동안 입력되는 패킷을 수용할만큼 충분한 버퍼를 가지고 있어야 하며, 위와 같은 현상으로 인해 평균 큐 길이의 변화 폭이 커지게 된다. 반면 제안 기법은 작은 타임 스케일에서 향후 예측되는 평균 입력율에 따라 입력 패킷을 확률적으로 폐기시키므로 실제 네트워크 심각한 혼잡 상황에 처하기 전에 송신단이 전송율을 미리 조절할 수 있게 해준다. 따라서 평균 입력율은 항상 서비스율과 거의 비슷한 수준으로 유지되며 이로 인해 평균 큐 길이의 변화 폭은 작아지게 된다. 이와 같은 결과는 제안 기법은 지연과 지터를 작게 유지할 수 있다는 것을 의미한다. 따라서 제안 기법은 중단간 지연과 지터에 엄격한 요구 사항을 가지는 실시간 멀티미디어 서비스 지원에 적합하다.

#### 4. 대역폭 할당의 공정성

제안 기법이 TCP와 같이 네트워크 혼잡 신호에 따라 전송율을 조절하는 플로우들과 네트워크 혼잡 신호를 무시하는 플로우들이 혼합되는 경우 각 플로우별로 대역폭을 공정하게 할당해 주는 것을 평가하기 위해 그림 7과 같은 모의 실험 네트워크를 구성하였다. 모의 실험 네트워크에서는 5개의 TCP Reno 소스와 하나의 CBR 소스가 동일 라우터를 공유하고 있다. 라우터의 버퍼 용량은 1000 패킷이며 링크 용량은 초당 400 패킷을 처리한다. TCP 패킷 크기는 전과 동일한 이유로 512 바이트로 고정하였으며 RTT를 50msec에서 200msec로 변화시켜 가면서 실험하였다. 제안 기법이 각 플로우에게 얼마나 공평하게 자원을 할당하는가를 정량화 하기 위해 다음과 같이 정의되는 공정도 색인을 구하였다<sup>[22]</sup>.

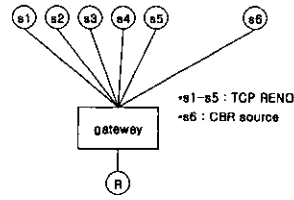


그림 7. 모의 실험 네트워크 II.

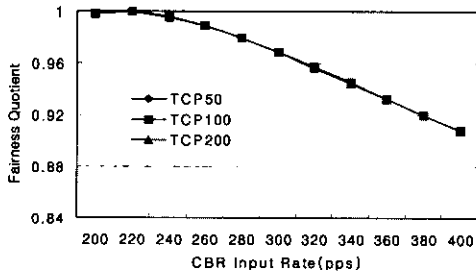


그림 8. CBR 입력율에 따른 공정도 색인의 변화.

$$F = \frac{(\sum_{i=1}^N b_i)^2}{N \sum_{i=1}^N b_i^2} \quad (12)$$

여기서  $N$ 은 플로우의 수를 나타내며  $b_i$ 는 소스  $i$ 가 할당받은 대역폭을 나타낸다. 위 공정도 색인은 0 과 1 사이의 값을 가지며 값이 1에 가까울수록 공정성이 더 커진다는 것을 의미한다.

그림 8은 CBR 소스의 입력율에 따른 공정도 색인의 변화를 나타낸다. 그림에서 보는 바와 같이 CBR 소스의 입력율이 증가하더라도 공정도 색인의 감소율은 작으며, 이와 같은 현상은 TCP 소스의 RTT와 무관하다는 것을 볼 수 있다. 또한 극단적인 경우 CBR 소스의 입력율이 링크 대역폭과 같게되어 CBR 소스로 인해 망이 심하게 혼잡한 경우에도 공정도 색인은 0.9 이상으로서 각 플로우별로 거의 공정하게 대역폭을 할당한다는 것을 볼 수 있다. 이와 같은 현상이 나타나는 이유는 제안 기법이 예상되는 패킷 입력율을 기반으로 패킷을 폐기시키기 때문이다. 즉, 라우터가 패킷을 폐기시키게 되면 TCP 소스는 네트워크가 혼잡 상태라고 판단하고 전송율을 감소시키지만 CBR 소스는 패킷 손실과 무관하게 패킷 전송율을 동일하게 유지한다. 따라서 라우터의 패킷 폐기 후 입력 트래픽에는 CBR 소스로부터 전송된 패킷의 양이 많아지게 된다. 제안 기법은 입력율을 예측하여 패킷 폐기 확률을 결정하

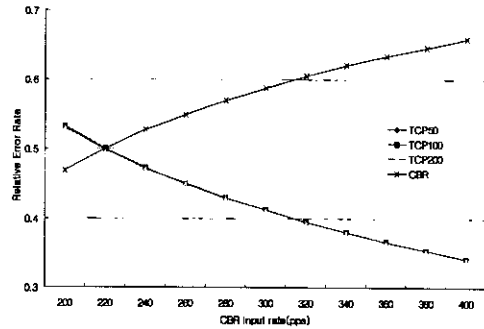


그림 9. CBR 입력율에 따른 플로우별 상대적 오류율 변화도

므로 CBR 소스의 패킷들이 보다 많이 폐기된다(그림 9). 즉, 제안 기법은 네트워크로부터의 혼잡 통보에 전송율을 조절하지 않는 소스들의 패킷을 보다 많이 폐기시킴으로써 TCP 플로우들을 보호하고 플로우들이 중단간 혼잡 제어 기능을 사용하도록 유도한다.

그림 10은 CBR 입력율에 따른 평균 큐 길이가 변화도를 나타낸다. 그림에서 보는 바와 같이 CBR 입력율이 증가하게 되면 RED의 평균 큐 길이는 버퍼의 최대 큐 길이 근처로 수렴하게 된다. 이와 같은 사실은 패킷 폐기 확률에 따라 패킷이 폐기되는 것이 아니라 큐의 오버플로우에 의해 패킷이 폐기된다는 것을 의미한다. 이 경우 TCP 플로우들이 전송율을 감소시켰음에도 불구하고 이들이 전송한 패킷들이 모두 폐기되므로 TCP 플로우들은 계속하여 전송율을 감소시키게 되며 이로 인해 얻어진 대역폭을 CBR 플로우가 차지하게 된다. 반면 제안 기법은 CBR 입력율이 증가하더라도 평균 큐 길이를 일정 수준으로 유지할 수 있으므로 입력 패킷을 패킷 폐기 확률에 따라 큐 포화 이전에 폐기시킬 수 있다.

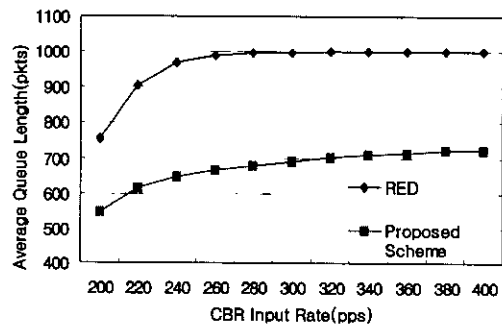


그림 10. CBR 입력율에 따른 평균 큐 길이 비교도



## V. 결론

본 논문에서는 네트워크 혼잡을 회피하고, 평균 큐 길이를 작게 유지하며, 종단 시스템의 혼잡 제어 기법 사용 여부와 무관하게 동일 라우터를 공유하는 플로우들 간에 대역폭을 공정하게 할당해 주는 큐 관리 기법을 제안하고 그 성능을 평가하였다. 제안 기법은 입력 트래픽 프로세스의 자기 유사성을 이용하여 일정 시간마다 향후 평균 입력율을 예측하고 이에 따라 입력 패킷을 확률적으로 폐기 시킴으로써 네트워크 혼잡을 방지한다. 이와 같은 동작 원리로 인하여 입력율은 항상 서비스율 근처에서 변화하게 되며, 평균 큐 길이와 큐 길이 변화폭은 작아지게 된다. 또한 제안 기법은 예상 입력율을 기반으로 패킷을 폐기 시킴으로써 종단간 혼잡 제어 기법을 사용하지 않는 플로우들이 네트워크 대역폭을 독점하는 현상을 방지할 수 있음을 모의 실험을 통해 검증하였다. 제안 기법은 각 플로우별 상태 정보를 관리하지 않으므로 확장성이 있고, 인터넷 설계 원리에 부합하므로 기존 인터넷 구조에 쉽게 적용이 가능하다. 특히 제안 기법은 소수의 PHB에 따라 각 홉별로 차등화된 서비스를 제공하는 Diffserv 구조에서 서비스 품질 제공을 위한 메커니즘으로 적합하다.

## 참 고 문 헌

- [1] J. Nagle, "Congestion Control in IP/TCP Internetworks", RFC 896, January, 1984.
- [2] V. Jacobson, "Congestion Avoidance and Control", in *Proc. ACM SIGCOMM*, Vol. 18, No. 4, August, 1988.
- [3] Y. Bernet et al., "A Framework for Differentiated Services", Internet Draft, dtaft-ietf-diffserv-framework-01.txt, November, 1998.
- [4] S. Floyd and V. Jacobson, "On traffic phase effects in packet switched gateways", *ACM Computer Communications Review*, April, 1991.
- [5] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2039, April, 1998.
- [6] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, August, 1993.
- [7] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A Self-Configuring RED Gateways", in *Proc. IEEE INFOCOM99*, March, 1999.
- [8] F. M. Anjum and L. Tassiulas, "Balanced RED: An Algorithm to Achieve Fairness in the Internet", in *Proc. IEEE INFOCOM 99*, March 1999.
- [9] J. Beran, *Statistics for Long-Memory Processes*, London, U.K, Chapman & Hall, 1994.
- [10] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic(Extended Version)", *IEEE/ACM Transactions on Networking*, Vol. 2, pp. 1-25, 1994.
- [11] M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and Possible Causes", in *Proc. ACM SIGMETRICS*, May, 1996.
- [12] K. Park, G. Kim and M. E. Crovella, "On the relationship between file sizes, transport protocol and self-similar network traffic", in *Proc. IEEE International Conference on Network Protocols*, pp. 171-180, 1996.
- [13] J. S. Park, J. Y. Lee, S. B. Lee, "Internet Traffic Measurement and Analysis in a High Speed Network Environment: Workload and Flow Characteristics", *JCN*, October, 2000.
- [14] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queuing algorithm", in *Proc. ACM SIGCOMM 89*, pp. 3-12, October, 1990.
- [15] L. Zhang, "A New Architecture for Packet Switching Network Protocols", MIT/LCS/TR-455, Laboratories for Computer Science, MIT, August, 1989.
- [16] D. Lin and R. Morris, "Dynamics of random early detection", in *Proc. ACM SIGCOMM 97*, Cannes, France, September, 1997.
- [17] J. Heinanen, F. Baker, W. Weiss, and J.

Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June, 1999.

[18] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June, 1999.

[19] W. Willinger, M. S. Taqqu, R. Sherman and D. V. Wilson, "Self-similarity through high-variability: Statistical Analysis of Ethernet LAN traffic at the source level", *IEEE/ACM Transactions on Networking*, Vol. 5, pp. 71-86, 1997.

[20] W. Stevens, *TCP/IP Illustrated Volume 1. The Protocols*, Addison-Wesley, New York, 1994.

[21] J. Apisodorf, K. Claffy, K. Thompson and R. Wilder, "OC3mon: Flexible, Affordable, High Performance Statistics Collection", in *Proc. INET 97*, Kuala Lumpur, Malaysia, 1997.

[22] D. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks", *Computer Networks and ISDN Systems*, Vol. 17, No. 5, pp. 1-14, 1989.

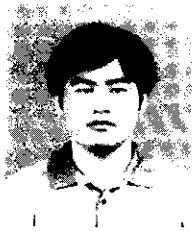
박 재 성(Jaesung Park) 정회원



1995년 2월 : 연세대학교 전자공학과 졸업  
 1997년 2월 : 연세대학교 전자공학과 석사  
 2001년 2월 : 연세대학교 전기, 전자공학과 박사

2001년 4월~현재 : University of Minnesota Postdoc 과정  
 <주관심 분야> Traffic Measurement and Analysis, Congestion Control in Internet, Policy-based Network Management

신 용 철(Yong-chul Shin) 정회원

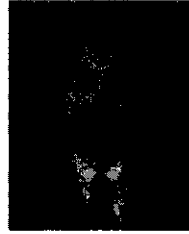


1985년 2월 : 한양대학교 전자공학과 졸업  
 1987년 2월 : 한양대학교 전자공학과 석사  
 1996년 3월~현재 : 연세대학교 전기, 전자공학과 박사과정

1987년 3월~1996년 12월 : 한국 원자력 연구소 선임 연구원

1997년 1월~현재 : 한국전력기술 책임연구원  
 <주관심 분야> Wireless LAN, 초고속 LAN, Network Management System, 계측제어 분야

박 회 윤(Heui-youn Park) 정회원



1984년 2월 : 경북대학교 전자공학과 졸업  
 1990년 2월 : 경북대학교 전자공학과 석사  
 1996년 3월~현재 : 연세대학교 전기, 전자공학과 박사과정

1987년 3월~1996년 12월 : 한국 원자력 연구소 선임 연구원

1997년 1월~현재 : 한국전력기술 책임연구원  
 <주관심 분야> performance analysis of safety-critical communication networks and network management systems

이 재 용(Jai-yong Lee) 정회원



1977년 2월 : 연세대학교 전자공학과 졸업  
 1984년 5월 : IOWA State University 공학 석사  
 1987년 5월 : IOWA State University 공학 박사

1987년 6월~1994년 8월 : 포항공과대학 교수  
 1994년 9월~현재 : 연세대학교 전자공학과 교수  
 <주관심 분야> Protocol design for QoS management, network management, high speed networks and conformance testing