

Diffserv 네트워크에서 공정성 보장을 위한 능동적 트래픽 Conditioning 메커니즘

정회원 김성철*, 이경희**

An Active Traffic Conditioning Mechanism(ATCM) for Fairness Guarantee in Diffserv Networks

Seong-Cheol Kim*, KyungHee Lee** *Regular Members*

요약

본 논문에서는 Diffserv 영역에서 차별화된 패킷 서비스를 제공함에 있어서 혼잡에 대한 응답 기능을 갖는 TCP-friendly 트래픽이 무응답 트래픽인 UDP 트래픽에 비하여 상대적으로 갖는 불공평성을 완화시켜 보장할 수 있는 새로운 능동적 트래픽 콘디셔닝 메커니즘(Active Traffic Conditioning Mechanism: ATCM)을 제안하였다. 이 방식은 코어 라우터에서 혼잡이 발생할 경우 OUT 패킷의 폐기와 IN 패킷의 마킹, 그리고 혼잡제어 메시지의 피드백을 통하여 능동적 트래픽 콘디셔닝을 제공한다. 본 논문에서 제안한 메커니즘은 이와 같은 상대적인 불공평성을 완화시킬 뿐만 아니라, 어떠한 fair share 기준에도 적용이 가능하고, 특히 MPEG-4 트래픽과 같이 한 트래픽에 여러 특성을 갖는 트래픽의 응용^[1]에서 효율적인 전송을 할 수 있는 장점을 갖는다.

ABSTRACT

In this paper, a new active traffic Conditioning Mechanism for fairness guaranteeing in Diffserv Networks was suggested. The proposed mechanism can be used to improve the fairness between TCP-friendly traffics and unresponsive traffics(e.g. UDP) in Diffserv networks. Active traffic conditioning can be implemented by marking the inprofile packets(IN packets), dropping the outprofile packets(OUT packets), and feedback the congestion notification packet to the edge routers when congestion occurs in the Core routers. Using this mechanism, we can not only improve and guarantee the fairness, but also can adapt this mechanism to any fair share metrics. Also this mechanism can be applied in transmitting the traffics which have various traffic characteristics, like MPEG-4 traffics^[1].

I. 서론

인터넷을 통한 새로운 여러 응용들 음성, 비디오 그리고 월드 와이드 웹 등이 증가함에 따라, 이들 서비스들이 요구하는 서비스들이 다양하며, 이들 다양한 요구 사항들에 대해서 유연하고도 각각의 필요를 구분하여 제공할 수 있는 새로운 네트워크 구조에 대한 요구가 날로 늘어가고 있는 실정이다.

이러한 요구를 만족시키기 위하여 최근에 IETF에서 제안되어 논의되고 있는 Diffserv(Differentiated Service) 구조^[1,2,6]는 바로 전송 트래픽의 간단한 구분 기능을 통하여 통합된 트래픽을 처리하며, 다양한 IP 트래픽의 서비스 질을 제공하는데 목적을 두고 있다. 이 구조에서는 전달되어질 패킷이 Diffserv 네트워크에 들어가기 전에 '패킷 마킹 메커니즘'에 의하여 먼저 여러 클래스로 구분되어지고, 내부 망

* 상명대학교 정보통신학부 (sckim@pine.sangmyung.ac.kr)

** (주)미디어링크 통신기술연구소(o2kiss@medialincs.com)

논문번호: 000010-1229, 접수일자: 2000년 12월 29일

※ 본 연구는 2000년도 상명대학교 자연과학연구소의 연구비 지원하에 수행되었음.

에서는 그 구분되어진 클래스에 의해서만 처리되어진다. 즉, 패킷의 클래스 구분과 마킹 및 폴리싱 그리고 셰이핑 기능을 포함하는 트래픽 컨디션닝이 네트워크의 주변 라우터에서 이루어지고 Diffserv 영역 내의 코어 라우터에서는 패킷 헤더에 있는 클래스 구분 값에 따라 패킷을 순방향 전송한다. 이러한 새로운 서비스 구조의 구현을 위하여 여러 마킹 방식 및 라우터에서의 버퍼 관리 메커니즘들이 제안되었다. 이들 라우터에서의 버퍼관리 메커니즘 중의 하나가 바로 IETF에서 권장하고 있는 RED (random Early Detection)이다^[8]. 이 방식에서는 Taildrop FIFO 버퍼관리 방식에서 일어날 수 있는 전체 동기화(global synchronization) 현상 회피와 버스티 트래픽에 대하여 유연하게 동작할 수 있는 등의 장점을 가진다^[13]. 즉, FIFO 버퍼 관리에서는 제공되는 버퍼가 수용할 수 있을 정도까지의 패킷만이 저장되고 이후에 들어오는 패킷들이 폐기된다. 폐기된 패킷이 TCP 전송 트래픽인 경우에 패킷 폐기로 인해 TCP 혼잡제어 메커니즘이 slow start 상태로 변해 전송 윈도우의 크기가 동시에 줄어들기 때문에 네트워크의 효율이 떨어지게 된다. 이러한 Taildrop FIFO 버퍼관리 메커니즘의 단점을 보완할 수 있는 RED 메커니즘에서는 버퍼의 평균 크기에 따라 패킷을 폐기함으로써 이와 같은 문제점들을 해결할 수 있다. 그러나 이러한 RED 메커니즘도 몇 가지 단점을 가진다. 즉, RED 메커니즘의 설계는 모든 트래픽이 패킷의 손실을 네트워크에서의 혼잡에 대한 척도로서 사용한다는데 기반을 두기 때문에 만약 혼잡에 대한 무응답 트래픽이 혼잡이 발생할 수 있는 병목 링크에서 많은 대역폭을 차지하게 된다면 RED는 올바르게 동작할 수 없다. 즉, 혼잡에 대한 응답 메커니즘을 가지는 TCP 트래픽은 혼잡이 발생하면 혼잡에 대응하여 자신의 전송률을 낮추지만, 무응답 트래픽은 혼잡 시에도 계속 많은 양의 트래픽을 전송하게 된다. 따라서 이들 두 트래픽 간의 심한 불공평성이 나타나게 된다. 관련 연구로 [9]에서는 이들 두 트래픽에 서로 다른 'color'를 할당하여 폐기 순위를 다르게 함으로 공정성을 높이는 방안이 제시됐으나, 네트워크의 부하가 커지면 공정성을 이룰 수 없는 단점을 가진다. 또한 Seddigh 등은 [3.4]에서 TCP와 UDP 트래픽을 각각 다른 큐를 할당하거나 다른 트래픽 클래스에 넣으므로 공정성을 높이려 하였으나, 코어 라우터에서는 현재의 TCP나 UDP 흐름에 대한 예약된 대역폭을 알 수가 없고, 또한 트래픽 상태가 자주 바뀌

로 TCP/UDP 각각에게 대역폭을 할당할 수 없는 단점을 가진다. 이 외에도 여러 관련 연구들이 있으나^[6,7] 본 논문에서는 무응답 트래픽과 TCP-friendly 흐름간의 공정성을 높이기 위하여 DS 내에 이들 무응답 트래픽을 직접 제어하기 위한 새로운 메커니즘을 제안하였다. 이 메커니즘의 구현에 있어서 코어 라우터에서는 간단한 기능만 수행하고, 복잡한 기능들은 에지 라우터로 일임하는 원래의 Diffserv 설계 원칙은 바뀌지 않는다.

본 논문의 구성은 다음과 같다. 제 2장에서는 능동적 큐 관리 기법인 RED 알고리즘에 대해 살펴보고, 제 3장에서는 Diffserv 영역 내에서 발생한 혼잡제어 정보를 코어 라우터와 주변 라우터들간의 효율적인 전송을 지원하는 메커니즘과 각 라우터에서의 역할 및 전송 메시지 형태에 대하여 살펴보았다. 그리고 이러한 제안된 메커니즘에 대한 성능 비교를 ns-2 시뮬레이터를 이용하여 수행된 결과가 제 4 장에서 보여지고, 마지막으로 본 논문의 결론과 차후 연구 방향에 대한 고찰이 제 5 장에 기술되어진다.

II. RED(Random Early Detection) 알고리즘

RED^[5,8,10]는 능동적 큐 관리 알고리즘의 한 예인데, 능동적 큐 관리된 기존의 FIFO 큐에서 인입되는 패킷을 큐에 넣고 또한 서비스를 통하여 큐로부터 패킷을 빼어내는 방식을 확장한 개념이다. 즉, FIFO에서와 같이 큐에 패킷이 차면 그 때부터 들어오는 패킷을 폐기하는 것이 아니라, 큐의 평균 길이가 미리 설정해 놓은 낮은 임계치를 넘어서면 들어오는 패킷을 확률적으로 폐기하고, 만약 큐의 평균 길이가 미리 설정해 놓은 높은 임계치를 넘어서게 되면 들어오는 모든 패킷을 임의로 폐기한다. 따라서 큐의 오버플로우가 일어나기 전에 미리 혼잡정보를 트래픽 소스로 피드백 시킴으로써 패킷의 폐기를 줄일 수 있다.

RED의 동작 원리를 간단히 나타내면 다음과 같다. 두 개의 임계치 min_{th} 과 max_{th} 를 기준으로, 평균 큐의 길이 avg 의 위치에 따라 버퍼에 들어가는 트래픽 양이 조절된다. 즉, 패킷이 들어올 때마다 평균 큐의 길이를 측정하여 만약 $avg < min_{th}$ 일 때는 인입되는 모든 패킷을 받아들이고, $min_{th} \leq avg < max_{th}$ 일 때는 확률적으로 패킷을 폐기한다.

또한 $max_{th} \leq avg$ 일 때는 모든 패킷을 폐기함으로써 혼잡을 막아준다. 이를 위하여 사용되어지는 파라미터가 표 1에서 보여준다^[14].

표 1. RED 제어 파라미터들

avg	평균 큐 크기
count	마지막 drop 된 패킷 이후의 패킷들
w_q	큐의 가중치 (일반적으로 0.002)
min_{th}	최소 임계치
max_{th}	최대 임계치
max_p	p_b 의 최대값
p_a	현재의 패킷 폐기 확률
q	패킷이 도착했을 때 현재의 큐 크기

한편 패킷의 폐기는 평균 큐 크기와 최대 폐기율 등의 함수로 구해진 p_a 에 의해 확률적으로 이루어지는데 평균 큐 크기가 최대 임계치와 최소 임계치 사이에서 변화함에 따라 p_a 는 정해진 최대 폐기확률(max_p)에 대해 선형적으로 증가한다. 이러한 방법에 의해 TCP의 혼잡을 미리 예측할 수 있고, 버스트 트래픽으로 인한 글로벌 동기화를 피할 수 있다. 그림 1은 RED 알고리즘에서 혼잡정도에 따른 폐기 확률을 보여주고 있다^[13]. 그러나 RED의 취약점 중의 하나는 소스로의 혼잡 정보의 알람이 그 혼잡 링크에 다중화 되어지는 연결의 수에 직접 관련이 없다는 것이다. 심하게 혼잡이 되어있는 네트워크에서 버퍼의 오버플로우에 의한 패킷 손실을 피하기 위하여서는 혼잡정보가 충분히 많은 소스들에게 전달되어서 네트워크 부하를 충분히 낮게 줄여야 한다. 그러나 한편으로는 한꺼번에 많은 소스들이 전송률을 낮춤으로써 발생하는 이용률 저하를 고려해야 한다.

만약 N 개의 능동 소스가 혼잡 링크를 공유할 때 하나의 소스가 전송률을 내림으로써 얻을 수 있는 것은 $(1-1/2N)$ 이다^[10]. 여기에서 N이 증가하면 각 흐름의 역할이 그 만큼 감소하게 된다. 따라서 혼잡 정보를 피드백해서 보낼 경우에 그 혼잡 링크에서의 active 플로우의 수를 고려하는 것과 fair share 구하는 것이 중요하다. 이 외에도 TCP 트래픽과 같이 네트워크 혼잡이 발생하면 이에 대응하는 혼잡 제어메커니즘을 통하여 전송률을 제어하는 응답 트래픽과 혼잡 시에도 혼잡회피를 위하여 아무런 조치를 하지 않는 무응답 트래픽 간의 대역폭

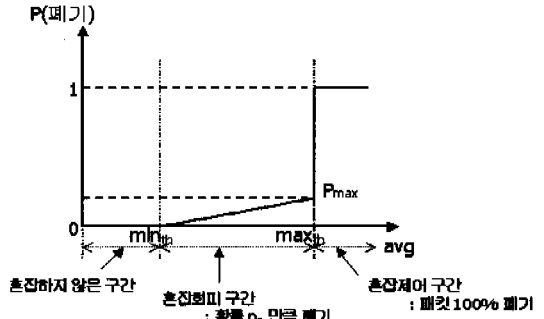


그림 1. 큐의 혼잡정도에 따르는 RED의 폐기확률^[13]

할당에 있어서 불공평성을 가지는 단점이 있다. 이러한 단점을 보완하기 위하여 RED의 변형인 FRED^[5], SPRED^[11] 등의 많은 알고리즘들이 제안되었다. 다음 장에서는 특별히 이러한 트래픽들 간의 불공평성을 완화시키며, 네트워크 전송 효율을 높일 수 있는 능동적 트래픽 콘디션링 메커니즘에 대하여 고찰하였다.

III. 능동적 트래픽 Conditioning 메커니즘(ATCM)

앞장에서 고찰한 바와 같이 Diffserv 내에서 여러 클래스 트래픽의 효율적인 처리를 위하여 RED 알고리즘을 사용하였을 경우에 TCP-friendly 트래픽이 무응답 트래픽의 자원 독점으로 인한 불공평성이 발생한다. 따라서 본 장에서는 네트워크 코어 라우터에서 혼잡이 발생하였을 경우에 혼잡 정보를 Diffserv 인입 라우터에 피드백 함으로써 무응답 트래픽을 보다 많이 폐기시키는 메커니즘에 대하여 고찰하였다. 메커니즘의 설계에 있어서 코어 라우터는 가능한 한 간단하게, 그리고 모든 복잡성은 에지 라우터로 보내는 원칙은 변함이 없다. 그림 2에 Diffserv 네트워크에서 ATCM의 동작을 보여주고 있다. 이 ATCM 메커니즘의 기본 동작은 다음과 같다.

- 파일 전송, 전자 메일, 웹 브라우징 등의 신뢰성 있는 TCP 트래픽들은 인입 라우터에서 이들 트래픽을 IN 패킷으로 마크한다.
- UDP 트래픽은 주로 멀티미디어 응용에서 트래픽이므로 IN/OUT 패킷으로 구분한다. 예를 들어 MPEG4 트래픽의 경우 foreground 비디오 객체 (Video Object)와 시스템 정보를 IN 패킷으로 구분하고^[11], 백그라운드 비디오 객체는 OUT 패킷으로 마킹한다. 일반적으로 인터넷의 최선형(Best-

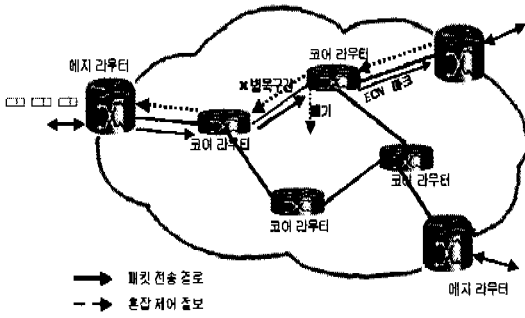


그림 2. Diffserv 네트워크에서의 ATCM

- effort) 트래픽을 OUT 패킷으로 마킹할 수 있다.
- 네트워크에서 혼잡이 발생할 경우 라우터에서 RED를 이용하여 확률적으로 OUT 패킷을 폐기한다.
- 폐기된 OUT 패킷과 같은 플로우의 IN 패킷이 전송되어질 때에 이 IN 패킷에 혼잡 정보를 실려 보낸다. 이 과정에서 RED 메커니즘을 이용하여 측정된 능동 소스(현재 버퍼에 하나 이상의 패킷이 저장되어 있는 소스)의 수를 알아서 같은 메시지 정보에 실려 보낸다.
- Diffserv 망의 출구(egress) 라우터에서 이 혼잡정보를 확인하여 혼잡제어 패킷을 해당 인입(ingress) 라우터로 보낸다.
- 이 혼잡제어 패킷을 받은 인입 라우터에서는 다음과 같은 트래픽 콘디셔닝을 수행한다.
 - * IN 패킷으로 마크되는 패킷의 수를 줄인다
 - * UDP 트래픽에 속한 OUT 패킷을 폐기한다.
 또한 전송률을 줄이는 과정에서 TCP 트래픽의 전체 동기화를 고려하여 UDP 트래픽을 점차 줄이면서 네트워크 전체의 효율을 높일 수 있다. 이 과정에서 공정성을 위하여 능동 소스의 수 등을 고려한 여러 fair share 알고리즘을 사용할 수 있다.

1) 코어 라우터에서의 역할

- 인입되는 패킷을 순방향 전송한다.
- RED를 사용하여 네트워크 부하 모니터링 및 패킷의 폐기 및 마킹을 수행한다. 이 때 모니터링하고 있었던 능동 TCP 수를 혼잡정보 마킹 시에 첨가시킨다.
- 인접 코어로부터 혼잡제어 메시지를 받으면 에지 라우터로 순방향 전송하여야 한다.

2) 에지 라우터에서의 역할

- 정상 상태에서는 정상적인 트래픽 콘디셔닝이 수행된다. 즉 적절한 폴리싱(policing), 셰이핑(shaping), 마킹 등의 기능이 이루어진다.

- 코어로부터 혼잡제어 메시지를 받게되면 해당 플로우에 대하여 트래픽 콘디셔닝을 한다. 즉, 무응답 트래픽에 대한 폐기율과 전송률 등을 조절한다.
- 만약 에지 라우터에서 통합 플로우에서 각각의 플로우를 구분할 수 있는 기능을 갖는다면, 직접 그 플로우를 제어할 수 있게 된다. 즉, 능동 TCP 소스의 수, 네트워크 혼잡 정도 등을 고려하여 무응답 트래픽에 대하여 네트워크의 효율을 저하시키지 않으면서 적절히 전송률을 줄인다. 이 때 플로우 당의 fair share를 알고 있으면 적절히 조절할 수 있게 된다.

ATCM 메커니즘에서 사용되어지는 혼잡제어 메시지 형식을 그림 3에 보여주고 있다. [7]에서 제안된 필드 외에도 인입 라우터에서 전체 네트워크의 급격한 효율저하를 막으며 무응답 트래픽의 효율적인 폐기를 위하여 능동(active) 소스의 수 필드를 첨가한다.

바이트	혼잡이	timestamp/순차번호	granularity	트래픽소스 종류	control power	active 소스의 수
-----	-----	----------------	-------------	----------	---------------	--------------

그림 3. 혼잡제어 메시지 형식

본 논문에서 제안된 메커니즘의 장점은 혼잡 라우터에서 능동 TCP 소스의 수를 계산하여^[10] fair share 만큼의 대역폭으로 점진적으로 혹은 원하는 대역폭으로 감소시키는 방법으로 효율을 증가시킬 수 있는 장점을 가진다. 이를 위하여 Diffserv 영역 내의 코어라우터들과 에지 라우터들의 밀접한 협력이 필요하다.

IV. 시뮬레이션 환경 및 결과

본 논문에서 제안한 메커니즘의 성능 분석을 위하여 ns-2 시뮬레이터^[12]를 사용하였다. 또한 본 시뮬레이션을 위한 네트워크의 구성은 아래 그림 4에 보여준다. 이 시뮬레이션에 사용되어진 RED 파라미터들과 패킷 크기, 그리고 TCP의 윈도우 크기는 표 2에 보여준 값이 사용되었다. 그림 5는 TCP와 UDP소스가 각각 하나씩 동일한 전송률(10Mbps)을 가지며 병목 링크에서 혼잡이 발생하였을 경우 TCP와 UDP 트래픽의 효율을 보여준다. 또한 그림 6은 본 논문에서 제안한 ACRM을 사용

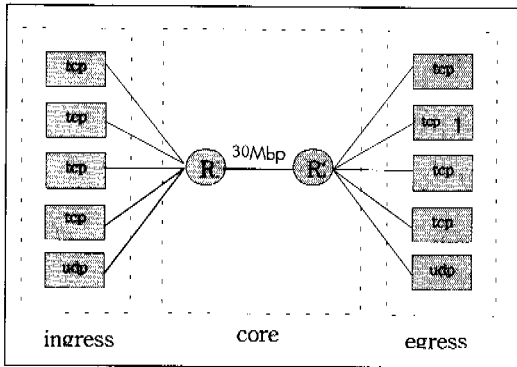


그림 4. 시뮬레이션 네트워크 구성도

하였을 경우의 TCP와 UDP의 효율을 보여주고 있다.

표 2. 시뮬레이션에 사용된 파라미터들

RED minth	5 packets
RED maxth	15 packets
RED wq	0.002
패킷 크기	1000bytes (1Kbyte)
윈도우 크기	25

즉, 그림 5에서 병목 링크에서 혼잡이 발생하였을 경우 TCP 트래픽 소스들은 네트워크 내부에서 일어난 패킷 폐기로 인하여 slow start 단계로 들어가게 되어 전송률을 급격히 줄이나 UDP 트래픽 소스들은 오히려 TCP 소스들의 대역폭까지도 점유하는 현상을 볼 수 있다. 그러나 그림 6에서와 같이 혼잡 정보 메시지가 인입 라우터로 피드백 되어지면, 인입 라우터에서 UDP 트래픽들에 대한 전송률을 단계적으로 내림으로 TCP의 효율이 증가하는 것을 볼 수 있다.

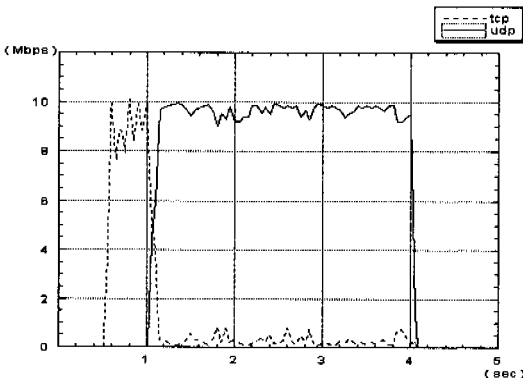


그림 5. 병목 링크에서의 TCP와 UDP의 효율

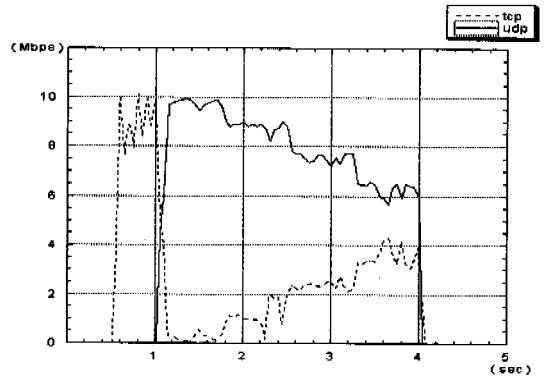


그림 6. ACRM을 사용하였을 경우의 TCP의 효율 변화

그림 7에서는 여러 소스가 병목 링크를 공유할 경우에 RED 및 Taildrop, 그리고 본 논문에서 제안된 능동적 혼잡제어 메커니즘에 대한 TCP의 효율을 보여주고 있다. 위의 그림들에서 보여준 것과 같이 ACRM을 이용하여 혼잡이 발생하였을 경우 UDP의 트래픽을 단계적으로 내려줌으로써 TCP 트래픽과의 공평성을 높이고 또한 네트워크 전체의 효율을 높일 수 있음을 알 수 있다.

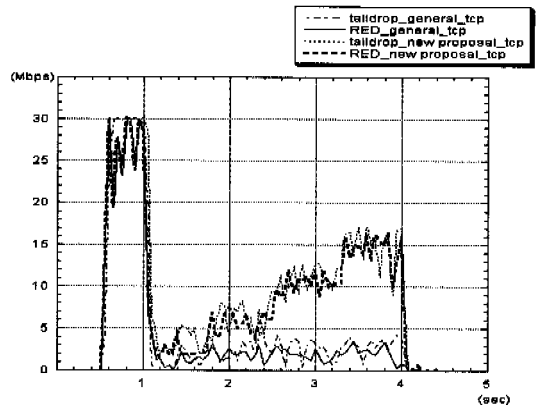


그림 7. 여러 알고리즘에서의 TCP 효율 비교

V. 결론 및 향후 연구

본 논문에서는 라우터에 RED 버퍼 관리 알고리즘을 사용하여 서비스의 차별화를 하는 Diffserv 망에서 일어날 수 있는 응답 트래픽과 무응답 트래픽 간의 대역폭 할당에서의 불공평성을 완화시키며, 서비스 보장할 수 있는 능동적 혼잡제어 메커니즘을 제안하였다. 제안된 메커니즘을 사용하여 무응답 트래픽을 제어함에 있어서 급격한 네트워크의 효율 감소를 막을 수 있을 뿐 만 아니라, 여러 공평성 기

준에도 쉽게 적용할 수 있다. 다만 피드백 되어지는 혼잡제어 메시지로 인한 트래픽의 증가와 보다 효율적인 제어를 위한 메시지 형식, 그리고 효율적인 피드백 메커니즘에 대한 연구는 계속되어질 예정이다.

참 고 문 헌

[1] K. Nichols, B. Carpenter, "Differentiated Services Operational Model and Definitions," Internet Draft, Feb., 1998

[2] Ibanez J. and Nichols K, "Preliminary Simulation Evaluation of an Assured Service", Internet Draft, Aug., 1998.

[3] N. Seddigh, B. Nandy, and P. Piedad, "Study of TCP and UDP Interaction for the AF PHB", Internet Draft 1999.

[4] N. Seddigh, B. Nandy, and P. Piedad, "Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network", *GLOBECOM 99*, Rio De Janeiro, Dec., 1999.

[5] D. Lin and R. Morris, "Dynamics of Random Early Detection," *ACM SIGCOMM*, 1977.

[6] H. Chow, A. Leon-Garcia, "A Feedback Control Extension to Differentiated Services", Internet Draft, Mar., 1999.

[7] Haitao Wu, Keping Long and Shiduan Cheng, "A Direct Congestion Control Scheme for Non-responsive Flow Control in Diff-Serv IP Networks", Internet Draft, Aug., 2000

[8] S. Floyd, and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *ACM/IEEE Transactions on Networking*, 1(4):397-413, Aug., 1993.

[9] Mukul Goyal, et al, "Effect of Number of Drop Precedences in Assured Forward -ing," *GlobeCom 99*, Dec. 1999.

[10] W. Feng, D. Kandlur, D. Saha, K. Shin, "A Self-Configuring RED Gateway," *INFOCOM '99*, , March 1999.

[11] Y. T. Hou D. Wu, et al, " A differentiated services architecture for multimedia streaming in next generation Internet," *Computer Networks* vol. 32, pp 185-209, 2000.

[12] <http://www-mash.cs.berkeley.edu/ns/>

[13] Martin May, Thomas Bonald and Jean -Chrysostome Bolot, "Analytic Evaluation of RED Performance," *INFOCOM 2000*

[14] M. Christiansen, Kevin Jeffay, D. Ott, F. D. Smith, "Tuning for Web Traffic," *Proc. of ACM SIGCOMM 2000*, pp. 139-150.

김 성 철(Seong-Cheol Kim)

정회원



1981년: 인하대학교 전자공학과 (학사)
 1991년: 뉴욕폴리텍 대 Electrical Eng. (석사)
 1995년: 뉴욕폴리텍 대 Electrical Eng. (박사)

1994년 7월~1997년 2월: (주) 삼성전자 수석연구원
 1997년 3월~현재: 상명대학교 정보통신학부
 <주관심 분야> 초고속통신망, 광통신망, ATM, 네트워크 성능분석 등.

이 경 희(Kyung-hui Lee)



1999년 2월: 상명대학교 전자계산학과 졸업
 2001년 2월: 상명대학교대학원 전자계산학과(석사)
 2001년 2월~현재: (주)미디어 링크 통신기술연구소 연구원

<주관심 분야> Internet QoS, TCP/IP Protocol