패킷 버퍼링을 이용한 TCP 처리율 보장 방법
(TCP Throughput Guarantee using Packet Buffering)

최 선 웅 † 김 종 권 ††
(Sunwoong Choi) (Chongkwon Kim)

요 약 본 논문에서는 차별화 서비스(Differentiated Services) 망에서 TCP 플로우(flow)의 처리율을 보장하는 문제를 연구하였다. 확신 서비스는 망의 혼잡 시에도 약속한 대역폭을 보장하기 위하여 패킷 폐기율을 차별화하는 방법을 사용한다. 그러나 토큰 버퍼 표시자는 TCP의 혼잡제어과 잘 동작하지 않아 적절한 성능을 보이지 못한다. 본 논문에서는 토큰 버퍼와 함께 데이타 버퍼를 사용하는 표시자를 제안하였다. 데이타 버퍼를 사용하는 표시자는 TCP 트래픽을 평활화하는 효과가 있어 확신 서비스 메커니즘과 잘 동작한다. 컴퓨터 시뮬레이션 결과 데이타 버퍼를 사용하는 표시자가 토큰 버퍼만을 사용하는 표시자에 비해 목표 대역폭에 보다 근접한 성능을 보이는 것을 확인하였다. 또한 최적 데이타 버퍼 크기는 예약 대역폭과 RTT에 비례하는 것을 관찰하였다.

Abstract This paper deals with the TCP bandwidth guarantee problem in a differentiated services(Diffserv) network. The Diffserv assured service differentiates packet drop probabilities to guarantee the promised bandwidth even under network congestion. However a token buffer marker fails to show adequate performance because TCP generates packets according to the unique TCP congestion control mechanism. We propose a marker that uses a data buffer as well as a token buffer. The marker with a data buffer works well with the assured service mechanism because it smooths TCP traffic. We showed that the marker with a data buffer achieves the target throughput better than a marker with a token buffer only. We also showed that the optimal buffer size is proportional to reserved throughput and RTT.

1. 서 론

기존의 인터넷은 서비스 품질(QoS: Quality of Service)을 보장하지 않았으나 인터넷 사용자의 증가와 실시간 멀티미디어 데이타 전송과 같은 새로운 응용의 등장으로 서비스 품질을 제공해 줄 수 있는 방법이 필요하게 되었다. 인터넷에서 QoS를 제공하기 위해 제안된 첫 방법은 통합 서비스(Integrated Service)[1] 망 구조이다. 통합 서비스 망은 RSVP(Resource Reservation Setup Protocol)[2] 등의 시그널링(Signaling) 프로토콜을 사용하여 각 플로우에 필요한 자원을 미리 종단간에 예약하는 방식을 사용한다. 통합 서비스 망은 플로우가 요구하는 서비스 품질을 보장할 수 있으나 라우터는 모든 플로우의 정보를 유지하여야 하며 패킷을 플로우별로 분류하고 스케쥴링하는 복잡한 패킷 처리 절차를 수행해야 한다. 그러므로 통합서비스 망 구조는 대규모 망에서 구현이 어렵다는 단점을 가지고 있다[3, 4].

차별화 서비스(Differentiated Services) 망 구조는 대규모 망에서도 구현이 가능하도록 규모확장성이 있으면서 서비스 품질을 지원하는 방법이다. 차별화 서비스 망에서도 각 패킷을 구별하여 서비스를 차별화한다. 그러나 차별화의 단위가 플로우가 아니라 DS(Differentiated Services) 코드포인트(codepoint)이고 DS 코드포인트는 8 비트 이내로 표현되므로, 상태 보존의 부담이나 복잡한 패킷 분류 및 스케쥴링 부담을 줄일 수 있다[3]. 현재까지 제안된 차별화 서비스로는 프리미엄 서비스(Premium Service)[5]와 확신 서비스(Assured Service)[6]가 대표적이다. 프리미엄 서비스는 빠른 전송을 보장하는 서비스로서 프리미엄 서비스 패킷을 베스트 에포트 서비스 패킷에 비해 신속히 전송한다. 그리

· 이 논문은 2000년도 두뇌한국21사업에 의하여 지원되었습니다.
† 비 회 원 : 서울대학교 컴퓨터공학부
schoi@popeye.snu.ac.kr
†† 종신회원 : 서울대학교 컴퓨터공학부 교수
ckim@popeye.snu.ac.kr
논문접수 : 2000년 4월 26일
심사완료 : 2001년 1월 8일

고 확신 서비스는 약속한 대역폭을 보장하는 서비스로서 패킷의 폐기 확률에 차별을 둔다. 일정량의 전송대역폭을 예약 없이 보장하는 것은 비교적 새로운 개념으로 확신 서비스의 대역폭 보장 방법과 성능에 대한 많은 연구가 수행되었다[6, 7, 8, 9, 10, 11, 12, 13].

차별화 서비스 망은 규모 확장성을 위한 독특한 구조를 가지고 있다[3, 4, 5, 6]. 약속한 트래픽 프로필을 준수하는지 감시하는 것과 같이 플로우별로 처리해야 하는 복잡한 기능은 패킷이 망에 진입하는 망 가장자리(boundary) 노드에서 수행한다. 그에 반해, 많은 플로우가 집중되는 망 내부에서는 플로우별로 패킷을 처리하는 것이 아니라 복합된(aggregated) 트래픽을 처리하여 내부 노드의 부하를 감소시킨다. 가장자리 노드는 각 플로우가 트래픽 프로필을 준수하는지 감시하는 역할을 하기 위해 프로필 감시자(meter)를 사용한다. 약속한 트래픽 프로필을 준수하는 패킷을 IN 패킷이라고 하며 위반 패킷을 OUT 패킷이라고 한다. 가장자리 노드의 표시자(marker)는 IP 헤더의 DS 코드포인트를 이용하여 IN/OUT 패킷을 표시하고, 내부 노드는 가장자리 노드에서 마킹한 DS 코드포인트에 따라 전송 우선 순위나 폐기 확률 등에 차별을 두어 패킷을 전송한다.

약속한 만큼의 대역폭을 예약 없이 보장해 주는 확신 서비스는 크게 표시자 알고리즘과 버퍼관리 알고리즘으로 구성된다. 표시자 알고리즘으로는 리키버킷(Leaky Bucket)을 사용하는 방법과 평균 전송률을 사용하는 TSW(Time Sliding Window)[6], ETSW(Enhanced TSW)[13]와 같은 방법이 제안되었다. 그리고 표시자에서 결정한 패킷의 DS 코드포인트에 따라 패킷의 폐기 확률을 차별화하는 버퍼 관리 방법으로는 RIO(RED with IN and OUT)[6]가 대표적이고, ERED (Enhanced RED)[7], (r, RTT)-적응 알고리즘[13], DRIO(Dynamic RIO)[13] 등이 제안되었다. 확신 서비스의 버퍼 관리 메커니즘들은 망에서 혼잡이 발생하는 경우에 OUT 패킷을 우선 폐기하여 혼잡 시에도 IN 패킷은 높은 확률로 수신자에게 전송될 수 있도록 한다.

토큰 버퍼 표시자를 사용하는 확신 서비스 메커니즘은 TCP 플로우에 적절한 대역폭을 보장하지 못한다는 것이 보고되었다[7, 8, 9, 13, 14]. TCP는 혼잡 제어를 위하여 선형 증가와 지수 감소(Linear Increase Multiplicative Decrease)의 과정을 반복하여 혼잡 윈도우(congestion window)를 톱니 모양으로 변화시킨다. 혼잡 윈도우가 작은 경우에는 예약한 대역폭을 모두 사용하지 못하는 저이용 상태가 된다. 또한 혼잡 윈도우가 큰 경우에는 OUT 패킷이 발생하고 이중 하나라도 폐기되면 윈도우 크기가 급격히 감소하므로 저이용 상태에서 받은 성능 손실을 충분히 보상받기 어렵다. 결과적으로 확신 서비스를 사용하는 TCP 플로우는 베스트 에포트 플로우보다 적은 잉여 대역폭을 제공받게 된다.

확신 서비스를 사용하는 TCP 플로우에 적절한 대역폭을 제공하기 위한 다양한 방법이 제안되었다[7, 8, 10, 13]. 이 방법들은 두 종류로 분류할 수 있다. 하나는 TCP의 혼잡 제어 메커니즘을 변경하거나 새로운 TCP 전송 메커니즘을 제안한 방법들이다. 다른 하나는 전송률이 역동적으로 변하는 TCP와 잘 동작하는 새로운 확신 서비스 메커니즘을 제안한 방법들이다. 두 가지 방법 중 TCP를 변경하거나 새로운 TCP를 사용하는 방법은 기존의 TCP를 대체해야 하는 막대한 노력이 필요하다는 단점이 있다.

본 논문에서는 확신 서비스를 사용하는 TCP 플로우에 적절한 대역폭을 제공하기 위하여 데이타 버퍼를 사용하는 표시자를 제안한다. 제안하는 방법은 TCP를 수정하지 않으므로 비교적 적은 노력으로 적용할 수 있다는 장점이 있다. 데이타 버퍼를 사용하면 TCP의 역동적인 전송률 변화에 적절히 대응할 수 있다. 데이타 버퍼는 트래픽을 평활화(smoothing)하는 효과가 있으므로 TCP 트래픽의 군집성을 줄여 표시자에서의 토큰 손실을 감소시킬 수 있다. 그러나 패킷 버퍼링은 부가적인 지연을 발생시킨다. TCP의 성능은 RTT에 반비례하기[15] 때문에 데이타 버퍼의 크기가 클수록 패킷 버퍼링으로 인한 지연이 증가하여 TCP의 성능을 저하시킨다. 그러므로 TCP 확신 서비스 플로우에 적절한 대역폭을 제공하기 위해서는 적당한 크기의 데이타 버퍼를 사용하는 것이 중요하다.

본 논문에서 제안한 데이타 버퍼 표시자의 성능을 컴퓨터 시뮬레이션을 이용하여 분석하였다. 성능 분석 결과 제안한 방법이 토큰 버퍼 표시자에 비해 보다 목표 대역폭에 근접한 대역폭을 TCP 플로우에 제공하는 것을 관찰하였다. 또한 예약 대역폭, RTT가 데이타 버퍼 표시자에 미치는 영향을 분석하여 데이타 버퍼의 크기가 성능에 큰 영향을 미치는 것을 발견하였다. 성능 분석 결과에서 최적의 데이타 버퍼 크기는 예약 대역폭과 RTT에 비례하는 것을 관찰하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 확신 서비스의 목표 대역폭을 정의하고, TCP 혼잡 제어 메커니즘과 확신 서비스 메커니즘 사이에 발생하는 문제를 설명하였다. 3장에서는 본 논문과 관련된 기존 연구에 대하여 살펴보았다. 4장에서는 토큰 버퍼 표시자와 TCP 사이에 발생하는 문제점을 살펴보고, 그러한 문제

를 해결하기 위하여 본 논문에서 제안한 패킷 버퍼링의 효과를 고찰하였다. 5장에서는 제안한 데이타 버퍼 표시자의 성능을 분석하였다. 실험을 통해 데이타 버퍼의 크기에 따라 성능이 어떻게 변하는지 살펴보고, 예약 대역폭과 RTT가 최적의 데이타 버퍼 크기에 어떻게 영향을 미치는지 분석하였다. 마지막으로 6장에서는 우리의 연구 결과를 요약하고 향후 연구 과제를 제시하였다.

2. 배경지식

확신 서비스는 사용자에게 약속한 대역폭을 보장해 주는 것을 목표로 하는 서비스이다. 그러므로 확신 서비스 메커니즘은 각 플로우와 사전에 맺은 계약에 따라 약속한 최소한의 대역폭을 보장할 수 있어야 한다. 그리고 잔여 대역폭을 확신 서비스 플로우와 베스트 에포트 플로우에 균등하게 제공하여야 한다. 확신 서비스 메커니즘의 요구 조건에 따라 본 논문에서는 다음과 같이 대역폭을 제공하는 것을 목표로 하였다[7, 8, 9, 13].

플로우 i의 예약 대역폭을 r_i, 병목 링크의 대역폭을 C, 그리고 해당 병목 링크를 통과하는 플로우가 n개 존재한다고 하자. 각 플로우의 예약 대역폭을 보장하기 위해 전체 대역폭 C 가운데 $\sum_{i=1}^{n} r_i$의 대역폭이 예약된다. 그리고 잔여 대역폭 $(C-\sum_{i=1}^{n} r_i)$는 확신 서비스 플로우와 베스트 에포트 플로우가 공평하게 사용하여야 한다. 즉, 플로우 i가 제공받는 목표 대역폭 t_i는 다음과 같다.

$$t_i = r_i + \frac{(C - \sum_{k=1}^{n} r_k)}{n} \quad (1)$$

총 예약 대역폭이 링크의 대역폭을 넘는 과포화 상태에는 모든 확신 서비스 플로우의 예약 대역폭을 보장할 수 없다. 또한 잔여 대역폭이 존재하지 않기 때문에 베스트 에포트 플로우는 서비스를 받지 못한다. 사용자에게 약속한 최소한의 대역폭을 보장하고 베스트 에포트 플로우의 전송을 가능하게 하기 위해서는, 총 예약 대역폭을 링크의 대역폭 이하로 제한할 필요가 있을 것이다. 그러나 과포화 상태를 막는 방법은 본 논문의 범위를 벗어난다. 과포화 상태가 되는 경우를 상정하면 과포화 시의 목표 대역폭은 다음과 같이 각 플로우의 예약 대역폭에 비례한다.

$$t_i = \frac{r_i}{\sum_{k=1}^{n} r_k} C \quad (2)$$

2.1 TCP와 확신 서비스 메커니즘 사이의 문제점

토큰 버퍼 또는 TSW 표시자와 RIO 버퍼 관리 알고리즘으로 구성된 기존의 확신 서비스 메커니즘은 TCP 플로우의 목표 대역폭을 보장하지 못한다[7, 8, 9, 13, 14]. TCP의 윈도우 기반 혼잡 제어 특성때문에 RTT에 따른 차별 대우가 존재할 뿐만 아니라, 예약 대역폭이 큰 플로우일수록 목표 대역폭에 못 미치는 대역폭을 제공받는 현상이 발생한다. 이는 기존의 확신 서비스 메커니즘이 TCP의 역동적인 전송률 변화에 적절히 대처하지 못하기 때문이다.

TCP는 선형증가와 지수감소의 과정을 반복하며 망의 혼잡에 대처한다. 그러므로 TCP 플로우의 전송률은 그림 1과 같이 톱니 모양으로 변화한다. 확신 서비스를 이용하는 TCP 플로우의 경우 예약 대역폭 이상의 전송률에서 OUT 패킷 마킹으로 인해 패킷 손실이 발생한다. TCP 플로우는 패킷 손실을 감지하면 전송률을 감소시키는데, 혼잡 윈도우가 클수록 전송률이 큰 폭으로 감소한다. 잔여 대역폭은 모든 플로우가 동등하게 사용하기 때문에 예약 대역폭 이상의 구간은 모든 플로우가 동일하다. 그러나 전송률이 클수록 패킷 손실 후의 전송률 감소가 더 크기 때문에 예약 대역폭이 큰 플로우일수록 예약한 대역폭을 모두 사용하지 못하는 저이용 상태에 오래 머무른다. 저이용 상태에 오래 머무를수록 성능 손실이 크기 때문에 예약 대역폭이 큰 플로우일수록 목표 대역폭에 못 미치는 처리율을 얻는 결과를 낳는다.

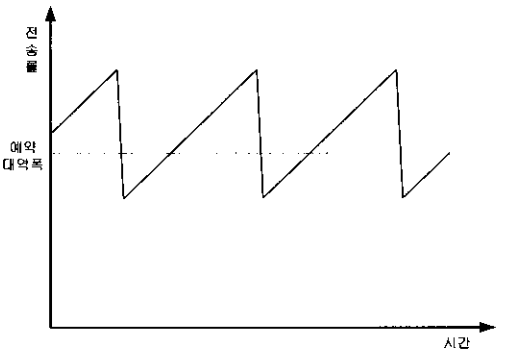

그림 1 TCP의 전송률 변화

예약 대역폭 손실의 또 다른 원인은 TCP 트래픽의 군집성에 기인한다. ack의 압축(ack-compression) 현상[16]으로 알려진 TCP 트래픽의 군집성은 토큰 버퍼 표시자에서 두드러진 성능 손실을 일으킨다. 토큰 버퍼 메커니즘은 토큰 버퍼의 크기만큼 군집을 허용한다. 그러므로 TCP 플로우의 패킷 군집이 토큰 버퍼의 크기를 넘어서면 예약 대역폭 이하의 전송률에서도 토큰 손실이 발생할 수 있다. 본 논문에서는 이와 같이 TCP 트래픽의 군집성으로 인한 성능 저하를 막는 것에 초점을

맞추었다.

토큰 손실을 막기 위하여 단순히 큰 토큰 버퍼를 사용하는 것은 좋은 해결책이 될 수 없다. TCP 플로우의 패킷 군집보다 큰 토큰 버퍼를 사용한다면 토큰의 손실을 막을 수는 있다. 그러나 토큰 버퍼가 클수록 한꺼번에 망에 전송될 수 있는 IN 패킷의 양이 많아지기 때문에 IN 패킷이라도 폐기될 가능성이 커진다. 이것은 확신 서비스의 패킷 폐기 확률 차별화 메커니즘 자체를 위협하는 것으로 토큰 버퍼의 크기에 제한을 둘 필요가 있다.

3. 관련 연구

TCP 혼잡 제어 메커니즘과 확신 서비스 메커니즘 사이의 상호 충돌 현상을 완화하여 TCP 확신 서비스 플로우의 처리율을 보장하기 위해 여러 방안들이 제안되었다. 크게 TCP 혼잡 제어 메커니즘을 변경하는 방법[7, 8, 10]과 TCP에 적합한 확신 서비스 메커니즘을 고안하는 방법[8, 13]으로 나누어 볼 수 있다.

Feng은 [7]에서 주기적 전송(timed transmission)이라는 새로운 TCP 전송 알고리즘을 제안하였다. ack의 압축 현상으로 인하여 발생하는 토큰의 손실을 막기 위하여 주기적으로 데이타 패킷을 전송하는 방법이다. 송신자는 주기적으로 표시자에 토큰이 존재하는지 확인하고, 토큰이 존재하면 수신자 버퍼에 여유 공간이 있는 한 혼잡 윈도우의 크기에 관계없이 패킷을 IN으로 마킹하여 전송한다. 또한 잔여 대역폭을 공평하게 사용할 수 있도록 혼잡 윈도우를 두 부분으로 나누었다. 예약 대역폭에 해당하는 부분과 잔여 대역폭을 위한 부분으로 나누어, 패킷 손실이 발생하였을 때 잔여 대역폭에 해당하는 부분만 절반으로 줄이는 방법을 제안하였다. 혼잡 윈도우의 크기에 관계없이 패킷 손실 시에 감소하는 혼잡 윈도우의 크기가 같기 때문에 여분의 대역폭을 공평하게 사용할 수 있다. 그러나 이 방법은 IN 패킷의 전송률을 망의 상황에 따라 변화시키지 않기 때문에 총 예약 대역폭이 링크의 대역폭보다 큰 상황에서는 과도한 IN 패킷을 생성하는 문제가 있다.

또한 Feng은 [10]에서 최소한의 대역폭 보장이 아닌 평균 대역폭 보장에 목표를 두고 망의 상황에 따라 각 플로우의 IN 마킹률을 조정하는 방법을 제안하였다. 기본적으로 목표 대역폭과 현재 대역폭의 대소관계에 따라 IN 마킹률을 조정하는 것이다. 전송률의 변화폭을 줄이기 위하여 TCP 혼잡 제어 알고리즘과 마킹 알고리즘을 통합한 새로운 TCP 혼잡 제어 메커니즘을 제안하였다. IN 패킷 윈도우와 OUT 패킷 윈도우를 관리하고, 그 크기에 따라 확률적으로 패킷을 마킹한다. 그리고 손실된 패킷의 IN/OUT 여부에 따라 혼잡에 대처하는 방식에 차이가 있다. OUT 패킷이 손실된 경우에는 OUT 패킷 윈도우만 감소시키지만, IN 패킷이 손실된 경우에는 심각한 혼잡으로 판단하여 IN 패킷 윈도우 역시 절반으로 감소시켜 혼잡에 대처한다.

Yeom은 [8]에서 폐기 확률이 높은 OUT 패킷의 양의 제한함으로써 성능 저하를 막는 새로운 표시자 알고리즘을 제안하였다. TCP 메커니즘에 표시자의 피드백에 반응하는 기능을 추가하여 표시자가 패킷을 OUT으로 마킹하면 혼잡 윈도우를 감소시켜 OUT 패킷의 생성을 제한한다. 이 방법은 잔여 대역폭이 적은 경우에는 좋은 성능을 보이나, 잔여 대역폭이 많은 경우에는 확신 서비스 플로우가 잔여 대역폭을 사용하는 것을 제한하기 때문에 잔여 대역폭 사용에 있어서 심각한 불평등을 초래한다.

또한 Yeom은 [8]에서 각 플로우의 예약 대역폭에 따라 패킷 폐기 확률을 결정하는 새로운 패킷 폐기 알고리즘을 제안하였다. TCP의 성능이 패킷의 폐기 확률의 제곱근 값에 반비례[15]하는 것에 근거하여 TCP 플로우의 목표 대역폭을 보장하기 위해서는 패킷의 폐기 확률이 목표 대역폭에 반비례하여야 한다고 주장하였다. 그러나 목표 대역폭을 정확히 알기 어렵기 때문에 예약 대역폭에 따라 패킷을 폐기하는 메커니즘을 제안하였다. 표시자에서 패킷에 예약 대역폭을 기록하여 전송하고, 망에서는 그에 따라 패킷 폐기 확률을 결정한다. 이 패킷 폐기 알고리즘은 RIO에 비해 전반적으로 좋은 성능을 보이지만, 예약 대역폭이 큰 플로우일수록 많은 이득을 얻는 문제점을 보인다.

Rin은 [13]에서 TSW 표시자 알고리즘을 개선한 ETSW(Enhanced TSW) 알고리즘과 RIO 버퍼 관리 메커니즘[6]을 개선한 (r, RTT)-적응 알고리즘, DRIO(Dynamic RIO) 알고리즘을 제안하였다. TSW 알고리즘은 TCP 플로우의 저이용 구간 손실을 보상하기 위해 예약 대역폭이상 일정 수준까지 모든 패킷을 IN으로 마킹한다. Rin은 모든 패킷을 IN으로 마킹하는 기준점이 망의 상황에 따라 달라야 한다고 주장하였고, 순간 전송률과 장기간의 평균 전송률의 차이에 따라 기준점을 조정하는 ETSW 알고리즘을 제안하였다. (r, RTT)-적응 알고리즘과 DRIO 알고리즘은 RIO 알고리즘이 예약 대역폭과 RTT에 따라 TCP 플로우에 다른 성능을 보이는 문제를 개선하였다. 앞서 설명한 바와 같이 RIO 알고리즘을 사용하면 예약 대역폭이 크고 RTT가 큰 TCP 플로우가 차별대우를 받는다. 이 문제를 해

결하기 위하여 (r, RTT)-적응 알고리즘에서는 수학적 분석을 통하여 예약 대역폭과 RTT에 따라 각 플로우의 패킷 폐기 확률을 계산하여 적용한다. 그리고 DRIO 알고리즘은 OUT 패킷의 히스토리 리스트(history list)를 유지하며, 각 플로우의 히트율(hit ratio)에 따라 OUT 패킷의 폐기 여부를 결정한다. 그러나 (r, RTT)-적응 알고리즘과 DRIO 알고리즘은 각 플로우에 대한 정보를 유지하여야 하는 부담이 있다.

4. 패킷 버퍼링의 효과

본 장에서는 패킷 버퍼링의 효과에 대하여 고찰하였다. 토큰 버퍼 표시자 메커니즘이 TCP 혼잡 제어 메커니즘과 상호 충돌하여 발생하는 문제를 기술하고, 패킷 버퍼링의 효과와 그 이유에 대하여 설명하였다.

본 논문에서는 ns[17] 시뮬레이터를 사용하여 성능을 분석하였다. 패킷 버퍼링의 효과에 대한 실험을 수행하기 위하여 리키버킷 표시자 알고리즘과 RIO 알고리즘을 직접 구현하여 ns 시뮬레이터에 추가하였다. 전송 프로토콜은 현재 인터넷에서 널리 사용되고 있는 TCP-Reno 버전을 사용하였다.

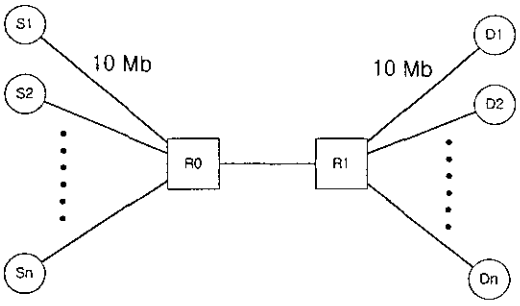

그림 2 망의 구조

본 논문의 실험 결과는 다음과 같은 환경에서 수행한 결과이다. 실험을 수행한 망의 구조는 기존의 논문들[8, 9, 13, 15]에서 사용한 것과 유사한 구조로서 그림 2와 같다. 라우터는 R0와 R1의 두 개가 존재한다. 모든 송신자 노드는 라우터 R0에, 모든 수신자 노드는 라우터 R1에 대역폭 10Mbps인 링크로 연결되어 있다. 라우터 R0와 R1을 연결하는 링크가 혼잡이 발생하는 병목 구간으로 실험에 따라 다른 대역폭을 할당하였다. 표시자의 토큰 버퍼 크기는 (예약 대역폭 * RTT)와 같다. 확신 서비스 플로우의 패킷은 표시자에서 IN 또는 OUT으로 마킹되어 전송되는 반면, 베스트 에포트 플로우의 패킷은 모두 OUT 패킷으로 전송된다. 각 노드에서는 RIO 알고리즘을 사용하여 크기가 100인 버퍼를 관리한다. RIO 알고리즘의 인자는 IN 패킷에 대하여는 20/80/0.02[1], OUT 패킷에 대하여는 20/50/0.1을 사용하였다. 모든 송신자는 FTP를 사용하여 항상 전송할 데이타가 존재한다.

4.1 토큰 버퍼 표시자의 문제점

토큰 버퍼 표시자는 다음과 같이 동작한다. 토큰은 일정한 간격으로 생성되어 토큰 버퍼에 저장되고, 패킷을 IN으로 마킹하여 전송할 때 토큰이 사용된다. 패킷이 도착한 순간에 토큰 버퍼에 토큰이 없으면 OUT 패킷으로 전송한다. 토큰 버퍼 표시자는 토큰 버퍼의 크기만큼 IN 패킷 군집을 생성할 수 있으며 평균 IN 마킹률을 토큰 생성 속도 이하로 제한한다.

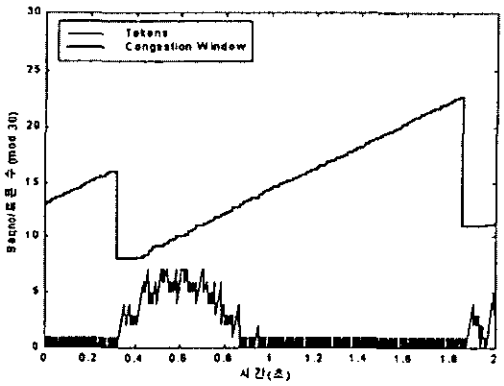

그림 3 토큰 버퍼 표시자와 TCP의 동작

토큰 버퍼 표시자와 TCP 혼잡 제어 메커니즘의 상호 동작을 알아보기 위하여 다음과 같은 실험을 수행하였다.

실험 1은 그림 2의 망에 6개의 TCP 확신 서비스 플로우와 6개의 TCP 베스트 에포트 플로우를 사용하였다. 각 확신 서비스 플로우의 예약 대역폭은 1Mbps이고, 병목 링크의 대역폭은 10Mbps이다.

그림 3은 TCP 혼잡 윈도우의 변화와 표시자의 토큰 버퍼 내에 존재하는 토큰의 개수 변화 및 손실된 토큰의 개수를 나타낸다. TCP 혼잡 제어 메커니즘과 토큰 버퍼 표시자 사이의 상호 충돌 때문에 많은 토큰이 손실되는 것을 볼 수 있다. 이것은 군집성이 큰 TCP 트래픽을 토큰 버퍼 표시자가 제대로 처리하지 못하기 때문이다. TCP 트래픽의 군집성은 ack의 압축 현상[16]

1) min_{th} / max_{th} / max_p

때문에 혼잡 윈도우가 계단 모양으로 증가하는 것에서 확인할 수 있다. ack이 군집을 이루어 도착하면 윈도우에 기반한 TCP 전송 메커니즘에 따라 데이타 패킷 역시 군집을 이루어 전송되는 결과를 가져온다. TCP 송신자는 짧은 시간에 많은 데이타 패킷을 전송하고 그에 비해 훨씬 긴 RTT 동안 ack이 도착하지 않아 패킷을 전송하지 않는 것이다. 이와 같이 TCP 플로우의 군집성 때문에 많은 토큰이 손실되고, 많은 토큰의 손실은 확신 서비스 플로우의 성능을 저하시킨다.

4.2 데이타 버퍼를 포함하는 리키버킷 표시자

본 논문에서는 패킷 버퍼링을 하여 토큰 손실을 막는 방법을 제안한다. 토큰 버퍼 뿐 아니라 데이타 버퍼도 포함하는 리키버킷 표시자를 제안한다. 데이타 버퍼를 포함하는 표시자는 패킷이 도착했을 때 토큰 버퍼에 토큰이 없으면 패킷을 데이타 버퍼에 저장한다. 토큰이 생성되면 데이타 버퍼의 가장 앞에 있는 패킷을 IN으로 마킹하여 전송한다. 만일 데이타 버퍼 오버플로우가 발생하면 데이타 버퍼의 가장 앞에 있는 패킷을 OUT 패킷으로 전송한다.

그림 4는 실험 1에서 크기가 3인 데이타 버퍼를 갖는 표시자를 사용하였을 때 TCP 혼잡 윈도우의 변화와 토큰 버퍼 내의 토큰의 변화를 나타낸다. 데이타 버퍼를 사용함으로써 손실되는 토큰이 크게 감소한 것을 볼 수 있다. 데이타 버퍼를 사용하면 트래픽을 평활화하는 효과가 있어 TCP 트래픽의 역동적인 전송률 변화를 완충할 수 있다. 데이타 버퍼를 사용한 경우 TCP 혼잡 윈도우의 증가 모습을 보면 토큰 버퍼만을 사용하였을 경우 계단 모양으로 증가하던 것과는 달리 매끄러운 직선 형태로 증가하는 것을 볼 수 있다. 이것은 패킷 버퍼링을 함으로써 트래픽이 평활화되어 ack이 고른 간격으로

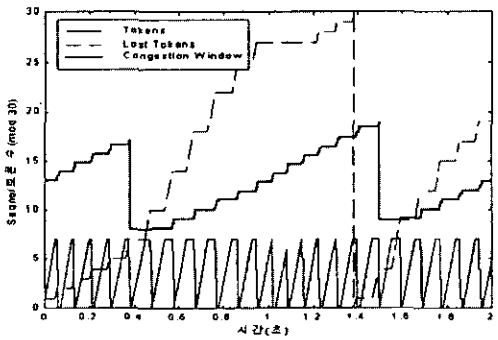

그림 4 크기 3인 데이타 버퍼를 포함하는 표시자와 TCP의 동작

도착하기 때문이다. TCP 트래픽의 군집성이 감소하여 리키버킷 알고리즘과 잘 동작하고 토큰 손실이 적다. 토큰의 손실이 적으므로 예약 대역폭에 가까운 IN 마킹률을 얻을 수 있다.

Aggarwal은 [18]에서 TCP 페이싱(pacing)을 구현하고 다양한 상황에서 성능의 실험하였다. 페이싱된 TCP 플로우는 패킷을 일정 간격으로 분산하여 전송한다. 페이싱된 TCP 플로우의 패킷 전송은 일정한 간격으로 오랫동안 이루어지기 때문에 군집성이 큰 다른 플로우의 영향을 받아 패킷이 손실될 가능성이 높다. 반면에 TCP-Reno와 같이 군집성이 큰 플로우는 상대적으로 패킷 손실 확률이 작다. 이와 같은 이유로 [18]에서는 페이싱된 TCP 플로우의 성능이 TCP-Reno의 성능보다 나쁜 결과를 보인다. 그러나 확신 서비스 망에서는 패킷 버퍼링으로 인한 트래픽 평활화가 TCP의 성능에 악영향을 미치지 않는다. 확신 서비스 망에서는 일반적으로 IN 패킷이 다른 플로우의 영향을 받지 않고 성공적으로 전송된다. 그러므로 확신 서비스 망에서 트래픽 평활화는 오히려 TCP 플로우의 성능을 높일 수 있다.

그러나 과도한 데이타 버퍼의 사용은 오히려 성능을 저하시킬 가능성이 있다. 데이타 버퍼를 사용하면 토큰이 생성되기를 기다리는 부가적인 지연이 발생하므로 필요 이상으로 큰 데이타 버퍼를 사용하면 오히려 TCP 확신 서비스 플로우의 성능을 악화시킨다. 데이타 버퍼가 클수록 IN 마킹률은 증가하지만, 패킷 버퍼링으로 인한 지연이 증가한다. TCP의 성능은 RTT에 반비례하므로[15], 큰 데이타 버퍼를 사용하면 TCP 확신 서비스 플로우의 성능을 저하시킬 수 있다. 그러므로 적절한 크기의 데이타 버퍼를 사용하는 것이 중요하다.

5. 성능 평가 결과

본 논문에서는 시뮬레이션을 통해 패킷 버퍼링의 성능을 고찰하였다. 먼저 데이타 버퍼를 사용함으로써 얻는 성능 향상을 살펴보고, 데이타 버퍼의 크기에 따른 성능 변화를 알아보았다. 다음으로 최적의 데이타 버퍼 크기가 여러 요인들에 어떤 영향을 받는지 고찰하였다. 예약 대역폭의 크기, 예약률, 그리고 RTT에 따른 최적의 데이타 버퍼 크기 변화를 관찰하고, 그 원인을 분석하였다. 최적의 성능은 각 플로우의 목표 대역폭에 얼마나 근접한 처리율을 제공하는지에 따라 평가하였다.

5.1 패킷 버퍼링으로 인한 성능 향상

표 1은 토큰 버퍼만을 가진 표시자와 크기 3인 데이타 버퍼를 추가한 표시자의 성능을 비교한 것이다. 두 경우의 IN 패킷 전송률과 그것에 OUT 패킷 전송률을

표 1 패킷 버퍼링의 효과 (단위: Mbps)

플로우 번호	토큰 버퍼만 사용한 경우		데이타 버퍼를 사용한 경우	
	총 처리율	IN 패킷 전송률	총 처리율	IN 패킷 전송률
0	1.13	0.80	1.16	0.94
1	1.07	0.80	1.18	0.96
2	1.04	0.79	1.22	0.95
3	1.10	0.80	1.17	0.94
4	1.11	0.81	1.21	0.96
5	1.12	0.81	1.14	0.94
6	0.54	0	0.43	0
7	0.57	0	0.52	0
8	0.52	0	0.49	0
9	0.63	0	0.49	0
10	0.56	0	0.48	0
11	0.57	0	0.47	0

더한 플로우의 총 처리율을 비교하였다. 실험 1에서 이상적인 결과는 예약 대역폭 1Mbps의 IN 패킷을 전송하고 잔여 대역폭 4Mbps를 공평하게 분배받아 1.33Mbps의 처리율을 얻는 것이다. 데이타 버퍼를 사용한 경우에 토큰 버퍼만을 사용한 경우에 비해 예약 대역폭에 가까운 IN 패킷 전송률을 얻고, 목표 대역폭에 보다 근접한 처리율을 얻는 것을 확인할 수 있다.

패킷 버퍼링의 효과를 좀더 명확히 규명하기 위하여 데이타 버퍼의 크기에 따른 성능 변화를 관찰하였다. 데이타 버퍼의 크기에 따른 성능 변화를 알아보기 위하여 실험 1에서 데이타 버퍼의 크기를 변화시키면서 실험을 수행하였다. 데이타 버퍼에 따라 확신 서비스 플로우의 처리율 변화를 나타낸 결과가 그림 5이다. 그림 5에서 아래 점선은 IN 패킷의 전송률을 나타내고, 위의 실선은 IN 패킷과 OUT 패킷을 합한 플로우의 처리율을 나타낸다.

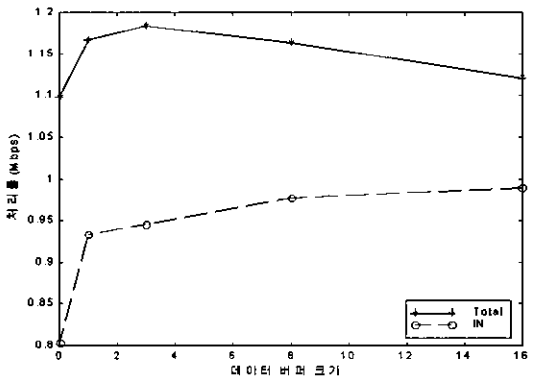

그림 5 데이타 버퍼의 크기에 따른 성능 변화

그림 5를 통해 데이타 버퍼의 크기에 따른 성능 변화가 직관적인 예상과 일치하는 것을 확인할 수 있다. 데이타 버퍼의 크기가 증가함에 따라 IN 패킷의 전송률은 예약 대역폭인 1Mbps에 점점 가까워지지만, OUT 패킷의 전송률을 합한 총 처리율은 어느 정도까지 증가한 후 점차 감소하는 모습을 보인다. 패킷 버퍼링을 하면 IN 패킷이 증가하여 TCP의 성능이 향상되지만, 데이타 버퍼가 어느 이상으로 커지면 패킷 버퍼링으로 인한 부가적인 지연이 증가하여 성능이 저하되는 것이다. 토큰 버퍼의 크기를 변화시키면서 실험을 수행한 결과 데이타 버퍼의 크기에 따른 경향성에는 큰 차이가 없었다.

5.2 예약 대역폭의 영향

데이타 버퍼가 차별화 서비스 망에서 확신 서비스에 미치는 영향이 어떻게 달라지는지 분석하기 위하여 예약 대역폭이 서로 다른 여러 개의 플로우가 공존하는 환경에서 성능을 평가하였다. 실험 2는 8개의 TCP 확신 서비스 플로우와 2개의 TCP 베스트 에포트 플로우가 있으며 8개의 확신 서비스 플로우의 예약 대역폭은 2Mbps, 1Mbps, 0.5Mbps, 0.1Mbps로 서로 다르다. 2개 플로우씩 같은 크기의 예약 대역폭을 할당하여 총 7.2Mbps의 대역폭을 예약하였고, 병목 링크의 대역폭은 12Mbps으로 설정하였다.

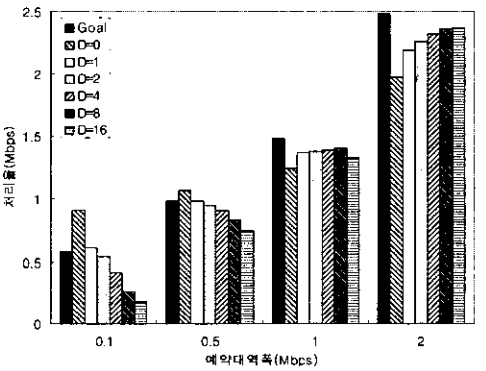

그림 6 예약 대역폭과 실제 처리율 사이의 관계

그림 6은 예약 대역폭과 데이타 버퍼의 크기에 따라 실제 처리율이 어떻게 변하는지 나타낸다. 데이타 버퍼가 없는 토큰 버퍼 표시자를 사용하면 예약 대역폭이 작은 플로우는 목표 대역폭보다 큰 처리율을 얻고, 예약 대역폭이 큰 플로우는 목표 대역폭보다 낮은 처리율을 얻는 것을 볼 수 있다. 그에 비해 데이타 버퍼를 사용한 경우에는 비슷한 경향성을 나타내며 목표 대역폭에 보

다 근접한 처리율을 제공하고 있다.

목표 대역폭에 근접한 처리율을 제공하는 최적의 데이타 버퍼 크기가 예약 대역폭에 따라 커지는 것을 관찰할 수 있다. 예약 대역폭은 토큰의 생성 속도와 일치한다. 패킷 버퍼링으로 인한 지연은 토큰 생성을 기다리는 시간이므로(데이타 버퍼 내에 있는 패킷은 버퍼 오버플로우 시 이외에는 토큰이 생성될 때 전송된다.), 예약 대역폭이 크면 상대적으로 패킷 버퍼링으로 인한 지연이 감소한다. 그러므로 예약 대역폭이 클수록 최적의 데이타 버퍼가 큰 결과를 보인다. 그리고 0.1Mbps를 예약한 플로우와 같이 목표 대역폭보다 더 많은 처리율을 얻는 플로우의 경우에는 IN 마킹률을 높여서 얻는 이득보다 패킷 버퍼링으로 인한 지연의 부작용이 커서 오히려 성능이 저하된다. 그러나 결과적으로 목표 대역폭에 보다 근접한 처리율을 제공받는 것을 볼 수 있다.

다음에는 확신 서비스 플로우의 총 예약 대역폭에 따른 데이타 버퍼의 효과를 분석하였다. 최적의 데이타 버퍼 크기는 예약률에도 영향을 받는다. 총 예약 대역폭이 병목 링크의 대역폭을 넘지 않는 상태를 불포화 상태, 총 예약 대역폭이 병목 링크의 대역폭을 넘는 상태를 과포화 상태라고 하자. 실험 2에서 병목 링크의 대역폭을 12, 8, 6Mbps로 변화시키면서 실험을 수행하였다. 총 예약 대역폭이 7.2Mbps이므로 예약률은 각각 60, 90, 120%이다. 그림 7은 예약 대역폭이 2Mbps인 플로우가 예약률에 따라 얻는 처리율의 변화를 나타낸 것이다. 불포화 상태(60%, 90%)에서는 서로 간에 유사한 경향성을 보이지만, 과포화 상태(120%)에서는 다른 경향성을 보이고 있다. 과포화 상태에서는 불포화 상태에 비해 최적의 데이타 버퍼 크기가 작은 것을 볼 수 있는데, 이것은 불포화 상태에서는 IN 패킷의 폐기 확률이 매우 작지만 과포화 상태에서는 IN 패킷의 폐기 확률이 무시할 수 없는 정도가 되기 때문이다. 과포화 상태에서는 IN 마킹률을 높여서 얻는 이득에 비해 패킷 버퍼링으로 인해 생기는 지연의 효과가 부각되어 작은 데이타 버퍼를 사용할 때 좋은 성능을 보이는 것이다.

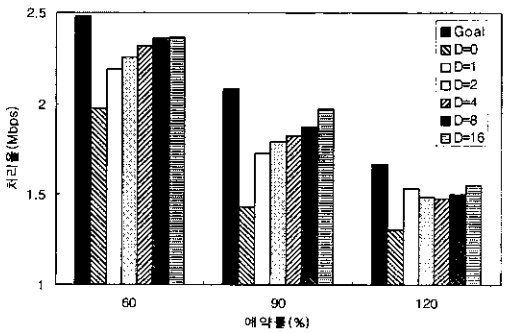

그림 7 예약률에 따른 성능 변화

5.3 RTT의 영향

RTT에 따라 데이타 버퍼가 TCP 확신 서비스 플로우의 성능에 미치는 영향을 분석하였다. 실험 3은 8개의 TCP 확신 서비스 플로우와 8개의 TCP 베스트 에포트 플로우로 구성되었다. 각 플로우의 RTT는 10, 20, 40, 80ms로 서로 다르다. 모든 확신 서비스 플로우의 예약 대역폭은 1Mbps이고, 병목 링크의 대역폭은 14Mbps이다.

그림 8은 실험 3의 결과이다. RTT가 큰 플로우일수록 최적의 데이타 버퍼 크기가 큰 것을 볼 수 있다. 이것은 TCP의 성능이 RTT에 반비례하기 때문이다[15]. 패킷 버퍼링으로 의해 생기는 지연의 크기가 동일하면 RTT가 작을수록 성능 저하가 심해진다. 즉, RTT가 클수록 패킷 버퍼링으로 인하여 생기는 지연의 효과가 성능에 미치는 효과가 감소하므로 최적의 데이타 버퍼 크기가 큰 것이다.

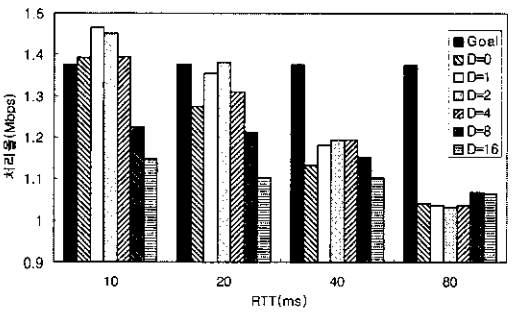

그림 8 RTT에 따른 성능 변화

6. 결론 및 향후과제

본 논문에서는 패킷 버퍼링을 사용하여 TCP 확신 서비스 플로우에 목표 대역폭을 보장하는 방법에 대하여 연구하였다. 표시자에 데이타 버퍼를 추가함으로써 토큰 버퍼만을 가진 리키버킷 표시자에 비해 목표 대역폭에 보다 근접한 처리율을 제공하는 것을 관찰하였다. 데이타 버퍼를 사용하면 토큰의 손실이 줄어들고 IN 마킹률이 증가하면서 TCP의 성능이 향상시킬 수 있지만, 과도하게 큰 데이타 버퍼는 RTT의 증가로 인하여 오히려 TCP의 성능을 저하시킬 수도 있었다.

데이타 버퍼의 최적 크기를 분석하기 위하여 예약 대역폭, 예약률, RTT에 따라 최적의 데이타 버퍼 크기가

어떻게 변하는지 살펴보고, 그 원인을 고찰하였다. 컴퓨터 시뮬레이션 결과 최적의 데이타 버퍼 크기는 예약 대역폭과 RTT에 비례하는 것을 관찰할 수 있었다.

TCP 확신 서비스 플로우에 목표 대역폭을 보장하는 문제와 관련하여 아직도 많은 연구 과제가 남아 있다. 본 논문에서는 시뮬레이션을 통하여 패킷 버퍼링과 관련한 사실들을 고찰하였다. 현재 최적의 데이타 버퍼 크기를 도식화하기 위하여 수학적 모델링 방법을 연구 중이다. 수학적인 방법으로 분석을 한다면 패킷 버퍼링의 효과를 명확히 기술할 수 있을 것이다. 그리고 더 나은 마킹 알고리즘이 존재하는지도 생각해 볼 문제이다.

참 고 문 헌

[1] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: An Overview," RFC 1633, Jun., 1994.

[2] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, "RSVP: A New Resource ReSerVation Protocol," Transaction on Networking, Sep., 1993.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, Weiss, W., "An Architecture for Differentiated Services," RFC 2475, Dec., 1998.

[4] X. Xiao, L. Ni, "Internet QoS: the Big Picture," Mar./Apr., IEEE Network.

[5] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," Internet Draft, Dec., 1997.

[6] D. Clark, W. Fang, "Explicit Allocation of Best Effort Delivery Service," Transactions on Networking, Aug., 1998.

[7] W. Feng, D. Kandlur, D. Saha, K. Shin, "Understanding and Improving TCP Performance over Networks with Minimum Rate Guarantees," Transactions on Networking, Apr., 1999.

[8] I. Yeom, A. Reddy, "Realizing throughput guarantees in a differentiated services network," ICMCS'99.

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

[10] W. Feng, D. Kandlur, D. Saha, K. Shin, "Adaptive Packet Marking for Providing Differentiated Services in the Internet," ICNP '98.

[11] M. May, J. Bolot, A. Jean-Marie, C. Diot, "Simple Performance Models of Differentiated Services Schemes for the Internet," INFOCOM'99.

[12] S. Sahu, D. Towsley, J. Kurose, "A Quantitative Study of Differentiated Services for the Internet," GLOBECOM'99.

[13] W. Lin, R. Zheng, J. C. Hou, "How to Make Assured Services More Assured," ICNP'99.

[14] N. Seddigh, B. Nandy, P. Pieda, "Bandwidth Assurance Issues for TCP flows in a Differentiated Services Network," GLOBECOM'99.

[15] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," SIGCOMM'98.

[16] L. Zhang, S. Shenker, D. Clark, "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic," SIGCOMM'91.

[17] UCB, LBNL, VINT Network Simulator - ns http://www-mash.cs.berkeley.edu/ns/ns.html

[18] A. Aggarwal, S. Savage, T. Anderson, "Understanding the Performance of TCP Pacing," INFOCOM'2000.

최 선 웅

1998년 2월 서울대학교 전산과학과 졸업(학사). 2000년 2월 서울대학교 전산과학과 졸업(석사). 2000년 3월 ~ 현재 서울대학교 컴퓨터 공학부 박사과정 재학중.

김 종 권

정보과학회논문지 : 정보통신
제 28 권 제 1 호 참조