

초고속 인터넷 접속망에서 보장형 서비스 제공을 위한 경계 노드의 스케줄링 알고리즘 설계 및 성능 분석

정회원 노대철*, 이재용**, 김병철**

Design and Performance Evaluation of a Scheduling Algorithm for Edge Node supporting Assured Service in High-speed Internet Access Networks

Dae-cheol Roh*, Jae-Yong Lee**, Byung-Chul Kim** *Regular Members*

요 약

최근 초고속 인터넷 접속 서비스에 대한 가입자의 욕구는 단순 인터넷 접속 서비스에서 개인화된 서비스로 변화하고 있다. 초고속 인터넷 접속 서비스 제공 사업자 또한 가입자에게 좀 더 고속의 서비스를 제공하기 위해 ADSL에서 메트로 이더넷 기반의 VDSL로 대체하고 있다. 그러나 이러한 메트로 이더넷 기반의 초고속 인터넷 접속망에 이미 설치된 장비들은 트래픽 특성에 따라 가입자를 구별할 수 없어 개인별로 차별화된 서비스를 제공하는 것은 곤란한 실정이다. 이 논문에서는 접속망이 트리 구조일 경우, 접속망 요소들이 QoS 기능을 제공하지 않아도 경계 노드의 플로우별 트래픽 셰이핑으로 특정 하향 플로우에 대하여 QoS를 제공할 수 있음을 보였다. 본 논문에서는 이를 보이기 위해 트리 구조의 접속망에서 보장형 서비스를 제공하기 위한 경계 노드의 스케줄링 알고리즘을 설계하고, 이것의 성능을 분석하였다. 이 스케줄링 알고리즘은 최대 허용 대역폭으로 각 플로우를 셰이핑하고, 플로우의 SLA에 따라 최소 대역폭을 보장하도록 DRR을 개량하여 설계되었다. 설계된 알고리즘의 성능을 분석하기 위해 다양한 시뮬레이션을 수행하여, 경계 노드에서의 플로우별 트래픽 셰이핑에 의해 접속망에서의 혼잡 및 손실이 줄어들고, TCP 트래픽 플로우의 성능이 크게 향상되며, CBR 보장형 서비스에 대한 손실의 큰 감소와 더불어 지연 및 지터 특성이 향상되었음을 보였다.

Key Words : Assured service; per-flow shaping; scheduling algorithm; Internet access networks.

ABSTRACT

Recently, subscribers have strong desire to get QoS based personalized services in high-speed Internet access. Service providers have been rapidly replacing ADSL, cable broadband access networks with Metro-Ethernet based VDSL. But, it is difficult for Metro-Ethernet based broadband access networks to provide QoS based personalized services, because already deployed network elements cannot distinguish subscribers by specific traffic characteristics. In this paper, when the access network has tree topology, we show that it is possible to provide QoS for each downstream flow with only per flow traffic shaping at the edge node without QoS functions in access networks. In order to show that our suggested scheduling algorithm at the edge node can support the assured service in tree topology access networks, we evaluated its performance by simulation. The suggested scheduling algorithm can shape per-flow traffic based on the maximum bandwidth, and guarantees minimum bandwidth per flow by modifying the DRR scheduler. Simulation results show that congestion and loss in the access network elements are greatly reduced, TCP performance is highly enhanced and loss for assured CBR service flows is reduced by only shaping per-flow traffic at the edge node using our proposed scheduling algorithm.

* KT 기술연구소(phantom@kt.co.kr)

**충남대학교 정보통신공학과

논문번호 : 030322-0801, 접수일자 : 2004년 1월 5일

1. 서론

초고속 인터넷 접속 서비스에 대한 가입자의 욕구가 단순 인터넷 접속 서비스에서 개인화 된 서비스로 변화하고 있다. 네트워크 게임을 주로 하는 가입자는 전달할 데이터의 양은 적지만 손실 없이 빠르게 전달되기를 원하며, 실시간으로 뮤직 비디오를 보거나 음악을 듣는 가입자들은 적절한 대역폭을 확보하고 지연의 변위가 작기를 바란다. 또한 인터넷폰을 이용하여 음성통화를 하는 가입자들은 적절한 대역폭과 적은 지연 및 지연 변위를 요구한다. 이와 같은 QoS 요구사항을 IP망에서 수용하기 위한 방법으로 Intserv[1], Diffserv[2] 등이 제시되고 있으며, 대규모의 ISP도 시험을 통하여 그 가능성을 증명하고 있다. 그 결과에 의하면 대규모의 플로우를 처리하는 백본에서는 노드마다 독립적으로 구동되는 Diffserv가 현실적인 방법이며 적절한 트래픽량 및 경로를 예측함으로써 QoS 보장이 가능함을 보여주고 있다[3].

그러나 초고속 인터넷 접속 서비스에서 가입자의 서비스 욕구는 다양하다. 동일한 서비스(VoIP, VoD 등)에도 품질의 보장을 원하는 가입자도 있고 저렴한 가격으로 최선형 서비스로 처리되기를 원하는 가입자도 있다. 따라서 가입자들의 트래픽이 동일 목적지 IP 주소, UDP 또는 TCP 포트를 가질지라도 가입자의 서비스 요청에 따라 QoS 적용 여부가 결정되어야 한다. 백본에서 가입자의 서비스 등급을 신뢰하고 가입자 트래픽을 Diffserv에 의해 처리하기 위해서는 접속망에서 가입자의 Service Level Agreement(SLA)를 확인하고 이에 따라 트래픽을 처리해주어야 한다[4].

ATM 기반의 ADSL 서비스와 같이 전통적인 방식에서는 가입자 관리 및 접속 인증이 백본의 접속점(Access POP)인 집중국(COC, Central Office Center)에서 이루어져왔다. 실제 망에 QoS 적용을 하기 위해서는 가입자의 SLA 관리가 필요한데, 이것은 가입자 관리 및 인증 정보를 이용하여 수행한다. 메트로 이더넷(Metro-Ethernet)을 접속망으로 이용하는 VDSL 서비스의 경우에도 가입자에게 QoS 등의 좀더 다양한 서비스를 제공하기 위한 가입자 관리 및 인증 기능 적용 등의 접속망 고도화 사업이 대규모 ISP들에 의해 추진되고 있다. 그러나 초고속 인터넷 접속망은 전국의 전역에 폭 넓게 분포하고 있어 QoS 적용이 가능한 망으로 고도화시키

는 데는 많은 비용과 시간을 요하게 된다. 따라서 초고속 인터넷 접속망에 QoS를 적용하기 위한 초기 모델로 접속망의 over-provisioning 및 경계 노드(edge node)에서의 트래픽 제어 방안이 제시되고 있다[5].

대규모 ISP의 망은 일반적으로 백본은 메쉬(mesh) 구조를 취하며 접속망은 그림 1과 같이 넓은 지역을 커버하기 위해 트리(tree) 구조로 구성한다. 이러한 트리 구조의 망은 가입자 단으로 내려갈수록 링크 당 수용 가입자 대비 대역폭이 넓어지는 특성을 가지고 있다. 상향 트래픽은 그 양이 하향의 1/3 ~ 1/4 정도이므로 충분한 대역폭을 확보하게 되어 정상적인 망 상태에서는 일정 수준의 QoS 보장이 가능하다. 그러나 하향 트래픽은 백본의 서버에서 전달하는 트래픽이 버스트하여 각 트래픽 플로우의 가용 링크 대역폭이 심하게 변화하는(작아지는) 메트로 스위치나 DSLAM에서 혼잡이 발생할 수 있어 패킷의 지연이 심화되거나 손실이 발생할 수 있다. 만약 링크의 대역폭이 작아지는 노드의 혼잡을 완화시킬 수만 있다면 링크 당 수용 가입자 대비 대역폭이 넓어지는 트리 구조의 망에서는 경계 노드에서만 트래픽 제어로도 QoS 보장이 가능하다.

본 논문에서는 이를 보이기 위해 트리 구조의 접속망에서 보장형 서비스를 제공하기 위한 경계 노드의 스케줄링 알고리즘을 설계하고 성능을 분석하였다. 이 스케줄링 알고리즘은 최대 허용 대역폭으로 각 플로우를 셰이핑(shaping)하고, 플로우의 SLA에 따라 최소 대역폭을 보장하도록 Deficit Round Robin(DRR)[6]을 개량하여 설계되었다. 설계된 알고리즘의 성능을 분석하기 위해 다양한 시뮬레이션을 수행한 결과에 따르면 경계 노드에서의 플로우별 트래픽 셰이핑에 의해 인터넷 접속망에서의 혼잡 및 손실이 줄어들고, TCP 트래픽 플로우의 성능이 크게 향상되며, CBR 보장형 서비스에 대한 손실의 큰 감소와 더불어 지연 및 지터 특성이 향상되었음

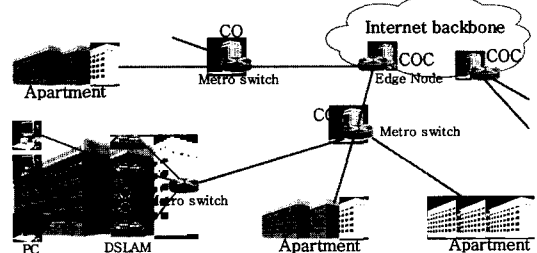


그림 1. 대규모 ISP의 메트로 이더넷 기반의 초고속 인터넷 접속망 구조

을 알 수 있었다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 트리 구조의 접속망 환경에서 보장형 서비스 제공을 위해 설계한 경계 노드의 스케줄링 알고리즘을 살펴본다. 제 3장에서는 시뮬레이션 환경 및 망 모델을 설명한다. 제 4장에서 다양한 시뮬레이션 환경에 따른 성능 분석 결과를 제시한다. 그리고 마지막으로 제 5장에서 결론 및 향후 연구 계획을 제시한다.

II. 보장형 서비스 제공을 위한 경계 노드의 스케줄링 알고리즘 설계

본 장에서는 트리 구조의 접속망 환경에서 보장형 서비스 제공을 위해 설계한 경계 노드의 스케줄링 알고리즘을 살펴본다.

보장형 서비스(assured service)는 망의 혼잡 발생 시에도 신뢰성 있는 서비스를 제공하기 위한 서비스 분류이다[7]. 따라서 보장형 서비스가 사용할 대역폭은 경계 노드에서 보장되어야 하며, 또한 경계 노드는 보장형 서비스가 전달되는 경로상에 있는 각 노드에서 혼잡을 적게 겪도록 다른 플로우의 버스트 정도를 감소시킬 필요가 있다. 실제 망 환경에서 가입자의 단말이 확보할 수 있는 대역폭은 최종 액세스 링크에서 주어지는 10Mbps 정도(ADSL, VDSL 등 접속 방식에 따라 차이가 있음)이므로, 경계 노드에서 특정 가입자에게 100Mbps 또는 1Gbps의 링크 대역폭이 주어진다 할지라도 가입자 단말이 확보한 대역폭 10Mbps 이상은 의미가 없다. 따라서 경계 노드에서 버스트한 트래픽을 가입자 단말로 그대로 전달하여 가입자 단말 앞단의 DSLAM이나 메트로 스위치에서 혼잡을 발생시키는 것 보다는, 사전에 경계 노드에서 트래픽 셰이핑을 수행하는 것이 접속망의 부하를 줄이고 보장형 서비스를 전달하는데 도움이 된다. 즉 경계 노드는 플로우별로 최소 보장 대역폭 및 최대 허용 대역폭(혼잡을 적게 겪도록 레귤레이션하는 대역폭의 상한 값)에 따라 트래픽을 제어할 수 있어야 한다.

또한 경계 노드는 모든 플로우가 동일한 조건의 SLA를 갖는다면 그 SLA에 따라 공평하게 서비스하여야 한다. 특히 모든 플로우가 최대 허용 대역폭 이내로 제한되어 있을 지라도 그 범위(최대 허용 대역폭) 내에서는 플로우간 공평성이 보장되지 않을 수 있다. 따라서 스케줄러는 버퍼에서 플로우별로 패킷을 구분하여 저장하고, 각 플로우별로 서비스할

표 1. 여러 가지 스케줄링 알고리즘의 성능 비교^[6]
* Max = maximum packet size

| Scheduling Algorithm | Fairness | Complexity |
|---|----------|-----------------|
| Round-Robin ^[9] | ∞ | $O(1)$ expected |
| Weighted Fair Queuing ^[10] | Max* | $O(\log(n))$ |
| Self Clocked Fair Queuing ^[11] | 2Max* | $O(\log(n))$ |
| Deficit Round-Robin ^[6] | 3Max* | $O(1)$ expected |

양을 공평하게 할당해 주어야 한다^[8].

이 논문에서 보장형 서비스 제공을 위한 경계 노드의 스케줄링 알고리즘은 구현이 용이하고 가중치 공평성과 셰이핑 기능을 지원하도록 설계되었다. 따라서 경계 노드의 스케줄링 알고리즘으로 기존의 DRR과 shaper를 결합한 Deficit Round Robin with Shaper(DRR-S)를 제안하고 성능을 분석하였다. 표 1은 기존에 제안된 여러 가지 스케줄링 알고리즘의 성능을 비교한 것으로, DRR은 기본적으로 라운드 로빈으로 서비스하기 때문에 $O(1)$ 의 낮은 복잡도를 갖으며 최대 패킷 길이의 3배 이내에서 플로우간의 공평성을 보장한다. DRR 스케줄러는 활성 플로우(패킷이 큐에서 대기하고 있는 플로우)에 대해서 각 큐에 가중치(weight)에 비례하는 quantum을 할당하고, 한 라운드에 각 큐를 순차적으로 방문하여 quantum만큼 패킷을 전송한 다음, 사용되지 않은 quantum은 deficit counter에 저장하여 다음 라운드에 할당되는 quantum과 함께 사용하고, 플로우가 비활성화되면 deficit counter를 0으로 한다. 이와 같이 DRR 스케줄러는 큐에 패킷이 존재하고 링크에 여유가 있는 한 계속 서비스를 하는 work-conserving 스케줄러이다. 따라서 DRR만으로 각 플로우의 버스트 정도를 완화시키거나 전송 대역폭을 조절하기 위한 셰이핑 기능을 구현할 수 없다.

본 논문에서 제안하는 DRR-S 스케줄링 알고리즘의 구성은 그림 2와 같이 입력된 패킷을 플로우별로 구분하는 분류기(classifier), 각 플로우의 패킷을 링크된 상태로 저장하는 공유 버퍼 풀(shared buffer pool), 각 플로우별 대역폭 제어를 위한 토큰 버킷(token bucket), 그리고 라운드 로빈 스케줄러(scheduler)로 구성된다. 여기에서 $g(i, j)$ ($i = 1, 2, \dots, n$)는 라운드 j 에서 플로우 i 에 할당된 대역폭으로 각 플로우의 버킷에 토큰을 채우는 속도이다. 본 스케줄링 알고리즘은 한 라운드에서 각 활성 플로우에 할당된 대역폭의 합이 링크 대역폭보다 작거나 같게 유지되어 각 플로우의 버킷 길

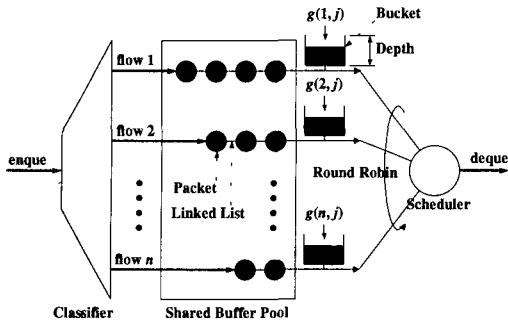


그림 2. DRR-S 스케줄링 알고리즘의 구조

이(depth)가 특별한 제한 값을 갖지 않아도 지속적으로 증가하지 않도록 설계되었다.

입력된 패킷은 분류기에서 플로우 별로 구분되어 공유 버퍼 풀에 링크된 상태로 저장된다. 이때 공유 버퍼 풀이 차있다면 버퍼를 가장 많이 차지하는 활성 플로우의 패킷을 끝에서부터 버린다. 만일 패킷 입력에 의해 새로운 활성 플로우가 생성된다면, 스케줄러가 각 플로우를 순차적으로 서비스할 때 이 신규 활성 플로우를 포함할 수 있도록 서비스 순서의 마지막에 삽입한다.

패킷의 출력은 그림 3의 DRR-S 스케줄러의 동작 흐름도를 이용하여 설명할 수 있다. 여기에서 j 는 라운드의 번호를 의미하며, i 는 플로우의 번호를 의미한다. 스케줄러는 한 라운드를 시작할 때 이 라운드에서 소비할 quantum 총량 Q 를 준비하고, 이 quantum의 총량을 SLA에 의해 결정된 대역폭 G 로 서비스하는데 소요되는 시간을 계산하여 다음 라운드의 예상 시작 시간 r_{j+1} 을 설정한다. 여기에서 각 라운드의 예상 시작 시간은 대역폭 제어를 위해 사용하는 이상적 시간이며 실제 시작 시간 t_{j+1} 과는 다를 수 있다. 스케줄러는 각 활성 플로우를 순차적으로 방문하여, 활성 플로우 i 에게 할당할 대역폭 $g(i, j)$ 을 계산하여 이 대역폭 $g(i, j)$ 과 SLA에 의해 결정된 대역폭 G 의 비율로 quantum의 총량 Q 를 배분하여 활성 플로우 i 에 quantum $q(i, j)$ 를 할당하고, 이전 라운드 $j-1$ 에 사용하고 남아있는 활성 플로우 i 의 deficit counter $s(i, j-1)$ 에 $q(i, j)$ 를 합한 $s(i, j)$ 값을 구한다. 그 후 이 값 범위 내에서 플로우 i 의 패킷들을 서비스하고 그 만큼 $s(i, j)$ 를 감소시킨다. 여기에서 활성 플로우(active flow)란 해당 큐에 전송할 패킷을 가진 플로우를 의미한다. 만일 패킷 서비스중 서비스할 패킷의 길이가 남아있는 $s(i, j)$ 보다 크면

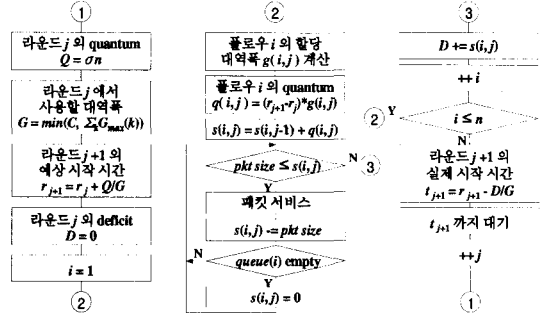


그림 3. DRR-S의 동작 흐름도

남은 $s(i, j)$ 는 다음 라운드에 사용하고, 패킷 서비스로 플로우가 비활성화 되면 남은 deficit counter $s(i, j)$ 를 0으로 한다. D 는 한 라운드에서 각 플로우의 패킷을 서비스하고 남은 deficit counter의 총량으로, 스케줄러는 마지막 활성 플로우 n 에 대한 서비스를 종료한 다음, 현재 라운드에서 발생한 deficit counter의 총량 D 를 소비하는 시간만큼 다음 라운드의 예상 시작 시간 r_{j+1} 을 앞당겨 실제 라운드 시작 시간 t_{j+1} 을 산출한다. 여기에서 deficit 총량 D 는 본래 라운드 j 에 서비스될 양이 라운드 $j+1$ 로 이관되는 것이므로 라운드 j 의 SLA에 의해 결정된 대역폭 G 로 소비되는 것으로 보고 t_{j+1} 를 계산하여야 한다. 결국 스케줄러는 마지막 활성 플로우 n 에 대한 서비스를 완료한 후부터 다음 라운드의 실제 시작 시간 t_{j+1} 까지 대기하게 된다.

한 라운드의 예상 소요 시간은 패킷 서비스의 버스트(burst) 정도와 구현의 용이성을 동시에 고려하여 결정하여야 한다. 라운드의 예상 소요 시간을 짧게 하면 패킷 서비스의 버스트 정도는 감소할 수 있으나 구현에 있어서 높은 프로세싱 성능을 요구하게 되며, 길게 하면 구현은 용이해 지지만 패킷 서비스는 버스트해져 스케줄링 효과가 감소한다[12]. 따라서 라운드의 소요시간을 고정하는 것보다는 한 라운드에 서비스할 양을 정해놓고 소요시간을 가변시키는 것이 바람직하다. DRR-S는 한 라운드에서 서비스할 양, 즉 quantum의 총량을 해당 라운드의 시작 시점에 존재하는 활성 플로우 수 n 과 라운드 당 모든 플로우 서비스 양의 평균 σ 의 곱으로 계산한다. 이 σ 는 스케줄러의 프로세싱 성능에 따라 조정 가능한 상수이다.

이에 따라 다음 라운드의 예상 시작 시간(현재 라운드의 예상 종료 시간)은 다음 식 (1)과 같이 표

현할 수 있다. 여기에서 r_{j+1} 은 다음 라운드 $j+1$ 의 예상 시작 시간이며, r_j 는 이전 라운드 $j-1$ 에서 예상했었던 라운드 j 의 예상 시작 시간으로 라운드 j 의 실제 시작 시간은 이와 다를 수 있다. C 는 출력 링크의 대역폭이고, n 은 라운드의 실제 시작 시점에 생성되어있는 활성 플로우의 개수이며, $G_{\max}(i)$ 는 각 활성 플로우 i ($i=1, 2, \dots, n$)의 최대 허용 대역폭이고, σ 는 모든 플로우의 서비스 양에 대한 평균값이다.

$$r_{j+1} = r_j + \sigma \frac{n}{\min\left(C, \sum_{i=1}^n G_{\max}(i)\right)} \quad (1)$$

결국 한 라운드의 quantum 총량은 σn 이 되고, $\sum_{i=1}^n G_{\max}(i) < C$ 일 경우에는 대기시간이 발생하며 대기시간동안 패킷 서비스 없이 소비되는 양은 $(r_{j+1} - r_j)(C - \sum_{i=1}^n G_{\max}(i))$ 이 된다.

각 플로우별로 대역폭의 할당에 있어서 각 플로우간 공평성을 유지하여야 한다. 각 활성 플로우 별로 공평하게 대역폭을 할당하는 메커니즘으로 Max-Min fair share 스케줄링 알고리즘[13]을 고려해 볼 수 있다. 그러나 Max-Min fair share 스케줄링 알고리즘은 각 플로우를 요구 대역폭 기준으로 오름차순으로 정렬하여 서비스를 해야 하므로 구현에 있어서 복잡도가 증가하는 문제가 있다. 이러한 구현상의 복잡도를 줄이기 위해 본 스케줄링 알고리즘에서는 ATM Forum TM 4.1[14]에서 제안한 General Weighted Fairness(GWF)[15]를 이용하였다. 이 GWF는 최소 대역폭을 보장하면서 남은 여유 대역폭에 대하여 각 플로우별로 가중치를 주어 공평성을 조정한다.

우선 각 활성 플로우별로 quantum을 할당하기에 앞서 각 플로우에 대한 대역폭 할당 조건을 살펴본다. 라운드 j 에서 활성 플로우 i 에게 주어지는 대역폭 $g(i, j)$ 는 최대 허용 대역폭 $G_{\max}(i)$ 와 최소 보장 대역폭 $G_{\min}(i)$ 사이의 값으로, 기본적으로 $G_{\min}(i)$ 을 포함하고 나머지 부분은 다른 활성 플로우와 공유하게 되므로 각 라운드마다 값이 달라질 수 있다. 이것은 다음 식 (2)와 같이 표현할 수 있다.

$$G_{\min}(i) \leq g(i, j) \leq G_{\max}(i) \quad (2)$$

큐에서 패킷 서비스를 대기하고 있는 활성 플로우들은 다음 식 (3)과 같이 각 활성 플로우의 할당

된 대역폭의 합은 링크 대역폭보다 작거나 같도록 제어된다. 여기에서 n 은 해당 라운드 j 에서 활성화되어 있는 플로우의 개수이다.

$$\sum_{i=1}^n G_{\min}(i) \leq \sum_{i=1}^n g(i, j) \leq C \quad (3)$$

위와 같은 조건 하에서 라운드 j 에서 활성 플로우 i 에 주어지는 대역폭 $g(i, j)$ 는 각 활성 플로우의 최대 허용 대역폭의 합이 링크 대역폭보다 작을 때와 그렇지 않을 때로 구분하여 계산한다. 전자는 모든 플로우가 최대 허용 대역폭으로 서비스 가능함을 의미하며, 후자는 최소 보장 대역폭을 우선 보장하고 나머지 대역폭은 다른 플로우들과 공유해야 함을 의미한다. $g(i, j)$ 는 다음 식 (4)와 같이 표현할 수 있다. 여기에서 $g_{share}(i, j)$ 는 링크 대역폭에서 모든 활성 플로우에게 최소 보장 대역폭을 할당하고 남은 대역폭을 각 활성 플로우 별로 공유하는 양으로 라운드 j 에 활성 플로우 i 에게 주어진 여분의 대역폭이다. 이때 $w(i)$ 는 남은 대역폭을 활성 플로우 별로 공유할 때 플로우 i 에 적용되는 공평성에 대한 가중치이다.

$$g(i, j) = \begin{cases} G_{\max}(i), & \text{if } \sum_{k=1}^n G_{\max}(k) \leq C \\ \min(G_{\max}(i), G_{\min}(i) + g_{share}(i, j)), & \text{otherwise} \end{cases} \quad (4)$$

$$g_{share}(i, j) = \frac{w(i) \left(C - \sum_{k=1}^n G_{\min}(k) \right)}{\sum_{k=1}^n w(k)}$$

스케줄러는 순차적으로 활성 플로우를 방문할 때마다 해당 플로우의 quantum을 계산하고, quantum 만큼 패킷을 서비스하면서 패킷 서비스에 사용된 quantum은 폐기한다. 만일 사용하고 남은 quantum이 서비스할 패킷의 길이보다 작으면 deficit counter에 저장하여 다음 라운드에 사용하고, 패킷의 서비스로 플로우가 비활성화되면 남은 quantum을 모두 폐기한다. 라운드 j 에 플로우 i 가 패킷을 서비스할 수 있는 최대 서비스 가능량 $s(i, j)$ 는 식 (5)과 같이 계산할 수 있으며, 이때 라운드 j 에서 플로우 i 에 할당하는 quantum $q(i, j)$ 는 다음 식 (6)과 같이 계산할 수 있다. 여기에서 $d(i, j-1)$ 은 이전 라운드 $j-1$ 에 플로우 i 가 최대 서비스 가능량 $s(i, j-1)$ 에서 패킷을 서비스하면서 소비하고 남긴 deficit counter이다.

$$s(i, j) = q(i, j) + d(i, j - 1) \quad (5)$$

$$q(i, j) = g(i, j)(r_{j+1} - r_j) = \sigma n \frac{g(i, j)}{\min\left(C, \sum_{k=1}^n G_{\min}(k)\right)} \quad (6)$$

남겨진 quantum의 총량, 즉 각 플로우 deficit counter의 총합은 다음 라운드의 quantum의 총량에 더해져 다음 라운드에 소비된다. 따라서 이 deficit counter의 총합을 소비하는 시간만큼 다음 라운드의 실제 시작 시간은 앞당겨진다. 각 플로우의 서비스 후에 발생하는 라운드 j 의 deficit counter의 총량 $\sum_{i=1}^n d(i, j)$ 은 SLA에 의해 결정된 대역폭 $\min(C, \sum_{i=1}^n G_{\max}(i))$ 을 기준으로 다음 라운드에서 소비되어야 한다. 이에 따라 다음 라운드 $j+1$ 의 실제 시작 시간 t_{j+1} 은 다음 식 (7)과 같이 표현할 수 있다. 즉 다음 라운드의 실제 시작 시간은 예상 시작 시간에서 현재 라운드에서 발생한 deficit counter의 총량이 소비되는 시간을 뺀 값이다.

$$t_{j+1} = r_{j+1} - \frac{\sum_{i=1}^n d(i, j)}{\min\left(C, \sum_{i=1}^n G_{\max}(i)\right)} \quad (7)$$

한 라운드의 quantum의 총량은 라운드의 시작 시점에 확인한 활성 플로우 수에 의해 결정된다. 따라서 라운드 수행 중에 새로이 생성되는 활성 플로우가 사용할 quantum은 현재 라운드의 quantum 총량에서 할당하지 않고 다음 라운드의 quantum 총량에서 할당한다.

III. 초고속 인터넷 접속망 시뮬레이션 모델

본 장에서는 초고속 인터넷 접속망 시뮬레이션을 위해 구성한 시뮬레이션 환경에 대해 살펴본다.

본 논문에서 제안한 DRR-S 스케줄링 알고리즘을 이용하여 시뮬레이션하고자 하는 목적은 트리 구조의 초고속 인터넷 접속망에서 각각의 모든 플로우가 적정 수준으로 셰이핑될 때 접속망 각 노드에서 손실이 작아지는 환경을 제공함으로써 보장형 서비스가 가능해짐을 보이는데 있다. 이를 검증하기 위해 NS-2^[6]를 이용하여 시뮬레이션을 수행하였다.

시뮬레이션 망 모델은 그림 4와 같이 현재 대규모 ISP들이 구축하여 운영하고 있는 초고속 인터넷 접속망 환경과 유사하게 구성하였다. 32개의 소스 단말 S와 목적지 단말 D가 있고, 노드 A 및 AS는

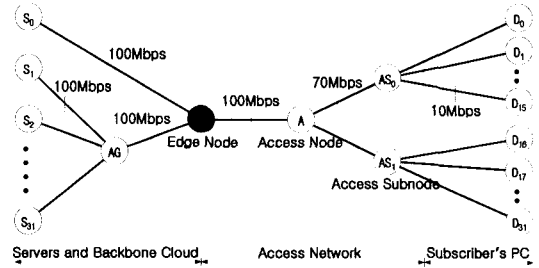


그림 4. 초고속 인터넷 접속망 적용 시뮬레이션 망 모델

트리 구조를 구성하기 위한 접속망 노드로 경계 노드 E부터 목적지 단말 D까지 트리 구조로 구성된다. 각 소스는 최대 100Mbps까지 전송할 수 있지만 목적지 단말은 최대 10Mbps까지만 받을 수 있도록 소스와 목적지의 링크 대역폭을 설정하였으며, 노드 A와 AS 간의 링크 대역폭은 70Mbps로, 경계 노드 E와 노드 A 간의 링크 대역폭은 소스와 동일한 100Mbps로 설정함으로써 전체 망 구조상 병목 구간이 되도록 실제 인터넷 접속망처럼 시뮬레이션 망 모델을 설계하였다.

소스 단말 S와 목적지 단말 D는 각각 1대 1로 독립적으로 통신한다. S0와 D0는 트래픽 성능 평가를 하기 위한 소스와 목적지이고, S1 ~ S31은 경계 노드 E에 백그라운드 트래픽을 부과하기 위한 소스이다. 백그라운드 트래픽을 발생시키는 각 소스의 트래픽은 on 구간과 off 구간의 평균이 50ms 및 3,000ms인 Pareto 확률 변수 분포를 가지는 Pareto on/off 트래픽이다. 노드 AG는 이 트래픽을 다중화시켜 자기 유사성을 갖게 하였다. 트래픽 성능 평가를 위한 트래픽은 적응형 트래픽인 TCP와 비적응형 트래픽인 UDP를 사용하였다. 표 2는 각 트래픽 소스의 설정 파라미터 및 SLA를 보여준다. 여기에서 B1은 백그라운드 트래픽 소스이고 A1 및 A2는 모니터링을 위한 트래픽 소스이다.

경계 노드 E에는 스케줄링 알고리즘간의 성능 비교를 위해 DRR-S, DRR 및 Drop Tail 스케줄링 알고리즘을 각각 적용하고, 큐 길이는 6Mbps로 설정하여 패킷 손실이 거의 발생하지 않도록 하였으며, DRR-S 적용 시에는 평균 서비스 길이 σ 는 6,000Byte로 설정하고 각 플로우별 공평성 가중치는 모든 플로우에 동일하게 1로 설정하였으며, DRR 적용 시에는 quantum을 6,000Byte로 설정하였다. 백그라운드 트래픽을 다중화하는 노드 AG에는 1MByte의 큐 길이를 갖는 Drop Tail 스케줄링 알고리즘을 적용하였으며, 나머지 노드 A 및 AS에는 100KByte의 큐 길이를 갖는 Drop Tail 스케줄

링 알고리즘을 적용하였다.

표 2. 트래픽 소스의 설정 파라미터 및 SLA

| 구분 | B1 | A1 | A2 |
|--------------------|-------------------|-------|-----|
| Source | $S_1 \sim S_{31}$ | S_0 | |
| Type | Pareto | TCP | CBR |
| Packet size (Byte) | 700 | 1,400 | 900 |
| Burst time (ms) | 50 | . | . |
| Idle time (ms) | 3,000 | . | . |
| Shape | 1.5 | . | . |
| Interval (ms) | . | . | 2.4 |
| Rate (Mbps) | 100(burst) | . | 3 |
| G_{max} (Mbps) | 10 | 10 | 10 |
| G_{min} (Mbps) | 0 | 0 | 3 |

IV. 시뮬레이션 및 성능 분석

본 장에서는 초고속 인터넷 접속망 환경에서의 시뮬레이션 결과를 살펴본다.

첫 번째 시뮬레이션은 31개의 백그라운드 트래픽 B1을 $S_1 \sim S_{31}$ 에 적용하고 S_0 에 모니터링 트래픽 A1을 적용한 환경에서 경계 노드에 DRR-S, DRR 및 Drop Tail을 각각 적용할 때, TCP가 어느 정도 유희 대역폭을 활용하는지 알아보기 위한 것이다. 이를 위해 30초 동안 시뮬레이션을 수행하고 노드 A 및 AS에서의 점유 큐 길이의 변화 및 손실율을 경계 노드에 적용한 스케줄링 알고리즘별로 비교 관찰하였다. 표 3은 이 결과를 보여준다. 이 결과는 각 플로우별로 셰이핑을 수행할 때 그렇지 않을 때보다 노드 A 및 AS0에서 패킷 입력 시 점유하는 큐 길이의 평균 및 편차가 작아져서 손실이 줄어드는 환경을 제공함을 보여준다. 또한 트래픽 유형이 TCP 일 경우 DRR-S를 적용할 때 DRR이나 Drop Tail을 적용할 때보다 더 높은 throughput을 제공할 수 있다.

그림 5는 경계 노드 E에 여러 가지 스케줄링 알고리즘을 적용하였을 때 노드 AS0 연결 링크에 대한 노드 A의 점유 큐 길이에 대한 도수 분포를 비교한 것이다. DRR 또는 Drop Tail을 적용하였을 경우 점유 큐 길이는 주로 최대 큐 길이 근처에 분포함을 볼 수 있다. 따라서 노드 A의 버퍼량이 조금만 작아지거나 트래픽의 버스트한 상황이 좀더 오래 지속된다면 많은 손실이 발생할 가능성을 가지고 있다. 그러나 DRR-S를 적용하였을 경우에는

표 3. 시뮬레이션 1의 결과
단위 : 대역폭(Mbps), 큐 길이(Byte), throughput(Mbps)

| 경계 노드 적용 알고리즘 | DRR-S | DRR | Drop Tail | |
|----------------|------------|-----------------------|-----------------------|-----------------------|
| E~A의 평균 사용 대역폭 | 56.27 | 55.03 | 55.36 | |
| A~AS0 링크 | 큐 길이 평균 | 31,202 | 86,497 | 89,290 |
| | 큐 길이 편차 | 16,510 | 37,764 | 37,336 |
| | 손실율 | 1.08×10^{-3} | 0.166 | 0.182 |
| AS0~D0 링크 | 큐 길이 평균 | 21,786 | 20,730 | 23,459 |
| | 큐 길이 편차 | 3,835 | 23,205 | 22,123 |
| | 손실율 | 0 | 0 | 0 |
| 모니터링 트래픽 A1 | Throughput | 9.9 | 5.5 | 5.8 |
| | 손실율 | 0 | 3.08×10^{-3} | 2.18×10^{-3} |

평균 큐 점유 길이 21,786Byte 근처에 주로 분포함을 볼 수 있다. 따라서 좀더 버스트한 상황이 발생하여도 손실이 많이 발생하지 않는다. 이것은 모든 트래픽 소스들이 경계 노드에 의해 균일화(regulation)되어 있기 때문으로 추가적인 트래픽이 유입되어도 거의 손실 없이 서비스가 가능함을 의미한다.

그림 6은 경계 노드 E에 적용한 스케줄링 알고리즘별로 모니터링 트래픽 A1(TCP)의 대역폭 변화 및 누적 데이터량을 비교한 것이다. 그림 6의 (a)에서 DRR-S를 적용하였을 경우에는 항상 일정한 대역폭을 유지하지만 DRR 및 Drop Tail을 적용하였을 경우에는 노드 A에서의 혼잡 및 손실로 인해 대역폭이 심하게 변화하는 것을 볼 수 있다. 또한 그림 6의 (b)는 누적 데이터량의 변화는 DRR 적용 시와 Drop Tail 적용 시에 TCP 성능 상 큰 차이가 없음을 보여주며, 이때 TCP 성능에 영향을 준 손실은 모두 노드 A에서 발생하였다. 이때 모니터링 트

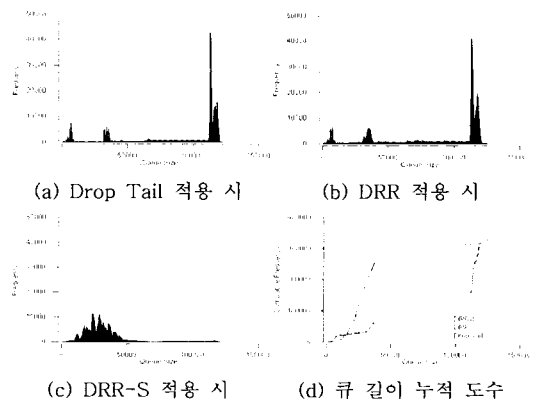


그림 5. 노드 A의 점유 큐 길이에 대한 도수분포

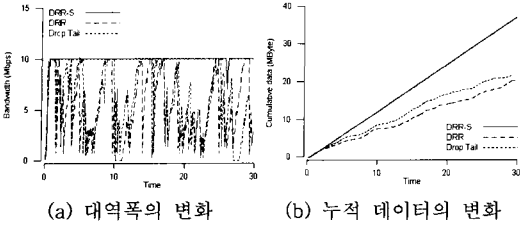


그림 6. 모니터링 트래픽 A1의 누적 데이터 변화

래픽 A1의 평균 throughput은 DRR-S 적용 시 9.9Mbps, DRR 적용 시 5.5Mbps, Drop Tail 적용 시 5.8Mbps로, DRR-S를 적용할 경우 혼잡이 거의 발생하지 않아 TCP 성능이 매우 우수하게 개선되는 것을 알 수 있다.

표 4. 시뮬레이션 2의 결과
단위 : 대역폭(Mbps), 큐 길이(Byte), throughput(Mbps), 지연 및 지터(ms)

| 경계 노드 적용 알고리즘 | DRR-S | DRR | Drop Tail | |
|------------------------------------|------------|------------|------------|------------|
| E~A의 평균 사용 대역폭 | 51.27 | 52.86 | 52.86 | |
| A~AS ₀ 링크 | 큐 길이 평균 | 24,084 | 85,958 | 88,043 |
| | 큐 길이 편차 | 9,886 | 37,271 | 36,908 |
| | 손실율 | 0 | 0.179 | 0.194 |
| AS ₀ ~D ₀ 링크 | 큐 길이 평균 | 4,057 | 4,072 | 4,165 |
| | 큐 길이 편차 | 495 | 793 | 1,129 |
| | 손실율 | 0 | 0 | 0 |
| 모니터링 트래픽 A2 | 수신량 | 14,992,200 | 14,787,000 | 14,761,800 |
| | 손실율 | 0 | 0.0137 | 0.0154 |
| | 지연 범위(99%) | 24.8 | 31.1 | 42.5 |
| | 지터 범위 | 2.8 | 7.7 | 7.4 |

두 번째 시뮬레이션은 31개의 백그라운드 트래픽 B1을 S1~S31에 적용하고 S0에 모니터링 트래픽 A2를 적용한 환경에서 경계 노드에 DRR-S, DRR 및 Drop Tail을 각각 적용할 때, 소스 트래픽이 CBR로 균일화(regulation)되어 있을 때 대역폭에 대한 보장성, 지연 및 지터(jitter)를 확인하기 위한 것이다. 이를 위해 40초 동안 시뮬레이션을 수행하고 노드 A 및 AS에서의 점유 큐 길이의 변화 및 손실율을 경계 노드에 적용한 스케줄링 알고리즘별로 비교 관찰하였다. 표 4는 이 결과를 보여준다. 노드 A의 큐 길이 변화는 시뮬레이션 1의 결과와 대동소이하지만 노드 AS0의 큐 길이 변화는 전체적인 평균 및 편차가 줄어들었다. 이것은 소스 트래픽이 CBR로 소스에서 이미 균일화되어 있기 때문이다.

그러나 DRR 및 Drop Tail을 적용하였을 경우에는 노드 A에서의 혼잡 발생으로 모니터링 트래픽 A2에 대한 손실이 발생하였고, DRR-S를 적용하였을 경우에는 접속망에서 어떤 손실도 발생하지 않았으며 단대단 지연 및 지터의 성능도 상당히 개선되었음을 확인할 수 있다. 이 결과는 플로우별로 셰이핑을 수행할 경우 특정 플로우의 품질 보장에 대한 신뢰성을 높일 수 있음을 보여준다.

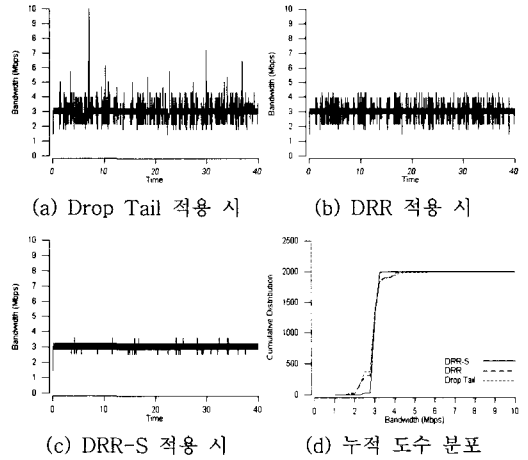


그림 7. 노드 A와 AS0 연결 링크에서 모니터링 트래픽 A2의 대역폭 변화. (측정 주기 : 20ms)

그림 7은 노드 A와 AS0를 연결하는 링크에서 모니터링 트래픽의 대역폭 변화를 적용 스케줄링 알고리즘 별로 확인한 것이다. 이때 측정 주기는 20ms로 놓았다. 소스 트래픽은 대역폭이 3Mbps로 제한되어 균일화되어 있지만 Drop Tail을 적용하였을 경우에는 노드 E부터 버스트해졌고 DRR을 적용하였을 경우에는 노드 A를 지나면서 버스트해졌으나, DRR-S를 적용하였을 경우에는 노드 A 및 AS0를 경유해도 항상 일정한 대역폭을 유지하였다. 이것은 앞서 첫 번째 시뮬레이션을 통해 확인한 결과와 동일하게 경계 노드 E에서 모든 플로우에 대한 셰이핑으로 트래픽이 균일화되어 있어 하부의 액세스 노드 A 및 AS에서 다른 플로우의 영향을 거의 받지 않기 때문이다.

그림 8은 S0에서 D0까지 모니터링 트래픽 A2의 전달시 발생하는 단대단 지연의 변화를 측정하는 것이다. 단대단 지연의 변화는 경계 노드 E에 Drop Tail 또는 DRR 적용하였을 경우 DRR-S 적용한 경우에 비해 넓게 분포하는 것을 볼 수 있다. 여기에서 단대단 지연의 99%는 Drop Tail 적용 시에는 42.5ms, DRR 적용 시에는 31.1ms, DRR-S 적용

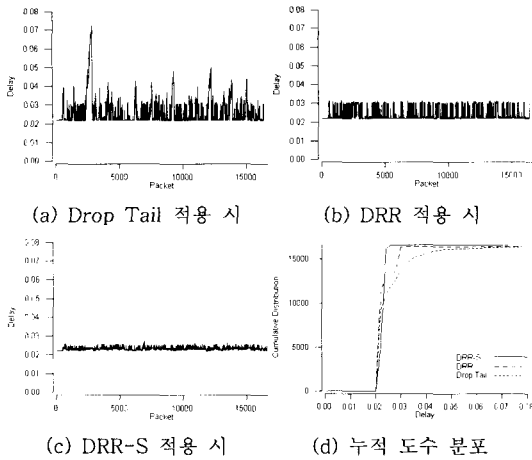


그림 8. 모니터링 트래픽 A2의 단대단 지연 시간

시에는 24.8ms 이내에 있다. 이것은 경계 노드 E에 DRR-S를 적용할 경우 단대단 지연이 작아져 VoIP와 같은 단대단 지연에 민감한 응용에 더 우수한 성능을 제공할 수 있음을 보여준다.

그림 9는 모니터링 트래픽 A2의 지터를 측정하는 것이다. 여기에서 지터는 패킷 간 단대단 지연 시간의 차에 절대값을 취한 것이다[17]. 이 지터 또한 경계 노드 E에 Drop Tail 또는 DRR 적용하였을 경우가 DRR-S 적용한 경우에 비해 넓게 분포하는 것을 볼 수 있다. 여기에서 지터의 범위는 Drop Tail 적용 시에는 7.4ms, DRR 적용 시에는 7.7ms, DRR-S 적용시에는 2.8ms 이내이다. 이것은 경계 노드 E에 DRR-S를 적용할 경우 Drop Tail 또는

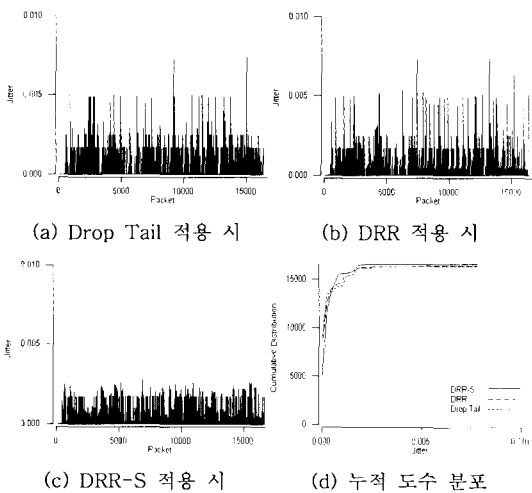


그림 9. 모니터링 트래픽 A2의 지터

DRR 적용하였을 경우에 비해 노드 A에서 혼잡이 적게 발생하여 지터가 감소하는 것으로, 목적지에서는 트래픽을 좀더 빠르게 재생(playback)할 수 있음을 보여준다.

V. 결론

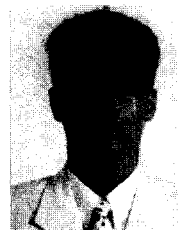
본 논문에서는 트리 구조의 인터넷 접속망에서는 QoS 기능을 제공하지 않아도 경계 노드의 플로우별 트래픽 셰이핑을 수행하는 스케줄링 알고리즘에 의해 특정 하향 플로우에 대하여 보장형 서비스를 제공할 수 있음을 보였다. 또한 이러한 플로우별로 셰이핑을 수행하고 플로우의 SLA에 따라 최소 대역폭을 보장하는 스케줄링 알고리즘을 제안하고 시뮬레이션을 통해 성능을 검증하였다. 특히 경계 노드에서 각 플로우별로 적절한 수준의 셰이핑을 수행할 경우 하위 노드의 큐 길이의 변동이 작아져서 손실을 줄이는 환경을 제공함으로써, TCP 트래픽 플로우의 성능이 크게 향상되고, CBR 보장형 서비스에 대하여 거의 손실 없이 정해진 대역폭으로 서비스가 가능하였으며, 이와 더불어 지연 및 지터 특성이 개선되었다. 그러나 경계 노드에 단순히 DRR이 적용되었을 경우에는 Drop Tail이 적용되었을 경우와 비교하여 TCP 트래픽의 성능상의 차이는 없었으며, 다만 소스 트래픽이 균일화되어있는 경우에 DRR이 적용되어있는 경계 노드에서만 품질이 보장됨을 확인하였다. 본 논문에서 제안한 스케줄링을 다단계로 수행한다면 링크의 대역폭이 작아지는 노드에서의 손실도 완전히 없앨 수 있다. 그림 4의 시뮬레이션 망 모델을 통해 간단하게 설명하자면 셰이핑은 노드 AS와 D간의 링크 대역폭으로 설정했기 때문에 이 구간의 큐 길이가 매우 일정함을 알 수 있다. 그러나 노드 A와 AS간의 링크 대역폭은 노드 E와 A간의 링크 대역폭보다 작으므로 특정 AS로 트래픽이 집중될 때 손실이 발생할 수 있다. 만일 노드 E와 AS간에 VLAN으로 연결되어 있다면, 이 VLAN에 대해 70Mbps로 셰이핑을 하고 해당 VLAN 내의 단말 D에 대해서는 10Mbps로 셰이핑을 한다면 노드 A에서의 손실도 없앨 수 있다. 본 논문에서 제안한 스케줄링 알고리즘은 트리구조를 가지는 초고속 인터넷 접속망에서 단순히 경계 노드만의 트래픽 셰이핑과 스케줄링에 의해 사용자의 서비스 품질을 향상시킬 수 있는 방안으로, 국내 초고속 인터넷 접속망 환경에서 쉽게 구현되어 널리 쓰일 수 있을 것으로 예상된다.

참 고 문 헌

- [1] R. Braden, D. Clark and S. Shenker, "Integrated services in the Internet architecture: An overview", RFC 1633, June 1994.
- [2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for differentiated services", RFC 2475, December 1998.
- [3] 신호정, 임태성, 최양희, 전효섭, 이정수, "ISP 네트워크에 대한 QoS 제공 방안", JCCI, 2003
- [4] T. Li and Y. Rekhter, "A provider architecture for differentiated services and traffic engineering", RFC 2430, October 1998.
- [5] Architecture & Transport Working Group, "DSL evolution - architecture requirements for the support of QoS-enabled IP services", Technical Report DSL Forum TR-059, September 2003.
- [6] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin", *IEEE/ACM Transaction on Networking*, vol. 4, pp.375-385, June 1996.
- [7] X. Xiao and L. Ni. "Internet QoS: A big picture", *IEEE Network*, pp. 8-17, March 1999.
- [8] S. Keshav, 'An engineering approach to computer networking', pp. 209-231, *Addison Westley*, 1997.
- [9] John Nagle, "On packet switches with infinite storage", *IEEE Transactions on Communications*, vol. COM-35, no. 4, April 1987.
- [10] L. Zhang, "Virtual clock: A new traffic control algorithm for packet switched networks", *ACM Transactions on Computer Systems*, vol. 9, no. 2, pp. 101-125, May 1991.
- [11] S. Golestani, "A self clocked fair queueing scheme for broadband applications", *In Proc. IEEE INFOCOMM'94*, 1994.
- [12] A.K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case", *IEEE/ACM Transaction on Networking*, vol. 1, pp. 344-357, 1993.
- [13] L. Kalampoukas, A. Varma and K. K. Ramakrishnan, "An efficient rate allocation algorithm for atm networks providing max-min fairness", *In Proc. of the 6th IFIP International Conference on High Performance Networking*, 1995.
- [14] The ATM Forum Traffic Management Working Group, "Traffic management specification version 4.1", ATM Forum/AF-TM-0121.000, 1999.
- [15] Bobby Vandalore, Sonia Fahmy, Raj Jain, Rohit Goyal and Mukul Goyal "General weighted fairness and its support in explicit rate switch algorithms", *Computer Communications*, vol. 23, No 2, pp. 149-161, January 2000.
- [16] The ns-2 simulator, URL:<http://www.isi.edu/nsnam/ns>
- [17] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A transport protocol for real-time application", RFC 1889, January 1996.

노 대 철(Dae-Cheol Roh)

정회원



1993년 : 인하대학교

전자공학과(공학사)

2004년 : 충남대학교

정보통신공학과(공학석사)

1996년~현재 : KT 기술연구소

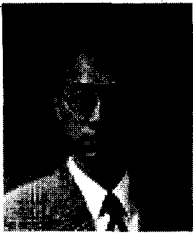
전임연구원

<관심분야> 초고속통신, 인터

넷, 네트워크 설계 및 성능분석

이재 용(Jae-Yong Lee)

정회원



1988년 : 서울대학교
전자공학과(공학사)
1990년 : 한국과학기술원
전기 및 전자공학과
(공학석사)
1995년 : 한국과학기술원
전기 및 전자공학과

(공학박사)

1990년~1995년 : 디지콤 정보통신연구소 선임연구원

1995년~현재 : 충남대학교 정보통신공학부 부교수

<관심분야> 초고속통신, 인터넷, 네트워크 성능분석

김 병 철(Byung-Chul Kim)

정회원



1988년 : 서울대학교
전자공학과(공학사)
1990년 : 한국과학기술원
전기 및 전자공학과
(공학석사)
1996년 : 한국과학기술원
전기 및 전자공학과
(공학박사)

1993년~1999년 : 삼성전자 CDMA 개발팀

1999년~현재 : 충남대학교 정보통신공학부 조교수

<관심 분야> 이동인터넷, 이동통신 네트워크, 데이터통신