

논문 2004-41TC-2-4

Differentiated Service에서 공정한 자원 공유를 위한 WFQ 적용 TC 설계

(The Design of TC with WFQ for Effective Resource Sharing on Differentiated Service)

장 경 성*

(KyungSung Jang)

요 약

Diff-Serv는 개별적으로 구분된 다른 트래픽에 대해서 다른 종류의 네트워크 서비스를 제공하기 위한 메커니즘이며 이용자들에게 양질의 서비스를 제공하기 위한 방법을 제시하고 있다. DS는 토큰 버킷 방법을 사용하여 고정된 호스트들에게 제공되기 때문에 이동성을 가지는 서비스나 개별적인 트래픽에 대한 차등적인 서비스를 제공할 방안이 존재하지 않는다.

본 논문에서는 트래픽컨테이너(TC)를 위해서 WFQ 방법과 변방 노드들과 경계 라우터들 내부에서 제어가 가능한 AggF(Aggregate Flow)를 모니터링하여 스케줄링하는 방법을 제시한다. 이 방법은 트래픽 비율(traffic rate)을 유동적으로 제어 가능하며 대역폭의 효율적인 이용성을 제공하고 있다.

Abstract

Diff-Serv(DS) is a mechanism by which network service providers can offer differing levels of network service to different traffic, in so providing quality of service(QoS) to their customers. Because this mechanism has been deployed just for fixed hosts with the Token Bucket mechanism according to AggF(Aggregate Flow) instead of each flow, DS can not suggest effective usability of traffic resources.

In this paper, we use WFQ mechanism for traffic conditioner and scheduling method monitoring the AggF and the results will be used to control the next flows coming in TC. So it will control traffic rate dynamically and suggest efficient usability of bandwidth.

Keywords: Differential services, Quality of service, Optical burst switching, Offset time, Optical networks

I. 서 론

현재 초고속 패킷처리를 위한 노력은 크게 두 가지 흐름으로 나뉠 수 있는데 하나는 현재 인터넷 라우터 기반의 망을 고속화하는 것이고 다른 하나는 링크계층의 스위칭 기술을 사용하여 패킷 전달의 고속화와 QoS 제공 능력 확장을 위한 연구이다.

DS(Differentiated Service)는 확장성을 고려하여 RSVP

와 같은 개별 flow에 대한 QoS 제공이 아닌 flow들의 집합(aggregate)에 대해 차등적인 서비스를 제공하는 방법이다. 확장성을 고려한 이러한 방안은 광역백본망에서와 같이 수천, 수만의 flow들에 대해 서비스를 제공하고자 할 때, flow들을 집합화하여 차등적인 서비스를 제공하고자 하는 것은 매우 큰 의미가 있다고 할 수 있다. DS의 기본 메커니즘은 네트워크 사업자가 제공가능한 서비스를 협약된 사용자 또는 하부네트워크(subnetwork)들의 flow를 차등적으로 제공하는 것이다.

DS 아키텍처에서는 개별 flow에 대한 QoS 제공

* 정회원, 초당대학교 정보통신학과
(Dept. of Information and Telecommunication, Chodang University)
접수일자: 2003년 1월 9일, 수정완료일: 2004년 2월 3일

이 아닌 AggF(Aggregated Flow)에 대한 QoS를 제공하고자하지만, 이들 AggF 대한 협약사항 준수를 측정하기 위한 방법으로 토큰(Token) 버킷(Bucket) 모델을 사용할 경우, 각각의 flow들에 대한 자원사용의 공정성을 제공하지 못한다는 문제점을 가지게 된다. AggF을 이루는 개별 flow는 동일 하부네트워크에서 기인할 수도 있고, 다른 하부네트워크들에서 기인하는 flow일 수도 있는데 이들간에 공정하게 자원을 사용할 수 있는 방법이 없다. 다수의 사용자가 DS 망과 동일한 서비스 레벨의 QoS로 협약되어있더라도, 현재 DS에서 제공하는 메커니즘으로는 불특정 flow에 대한 QoS 손실을 해결할 방법이 없다. 동일 서비스 레벨의 QoS로 협약된 사용자는 차등적인 서비스를 받음으로서 같은 금액의 사용료를 지불해야 하는데 이러한 문제는 반드시 해결해야 할 사항이다.

본 논문에서는 DS의 Edge node 또는 경계라우터(Border router)에서 다음단계에서 처리될 AggF의 상황을 모니터링할 수 있는 TC(Traffic Conditioner)와 스케줄링(scheduling) 방법으로 WFQ(Weighted Fair Queuing)을 사용한다. 즉, 각각의 service class에서 서비스되어질 다음 단계의 결과를 모니터링하여 그 결과에 의해 WFQ에서 현재 링크상의 이용비율을 동적으로 조정하여 잉여 대역을 효율적으로 사용할 수 있는 방안을 제시한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 관련연구로 DS, TC, PHB에 대해 간략히 소개하고, 3장에서는 공정한 자원공유를 위한 TC에 대해 기술하고, TC의 모니터 결과값을 이용하여 WFQ에서 각 class의 가중(weight)값으로 적용하여 동적으로 링크 이용 비율을 결정하는 방안을 제시하고 결론을 맺도록 하겠다.

II. 관련 연구

2.1 Differentiated Service

DiffServ는 인터넷에 유입되는 패킷을 몇 가지 우선 순위로 구분하고, 각 인터넷 라우터에 도달한 패킷을 이 우선 순위에 따라 적절한 서비스를 제공하는 메커니즘을 이용한다. 또한 네트워크 provider가 그들 customer에게 provider 네트워크내에서 받아들여 질 수 있는 레벨(level)의 서비스를 제공하는 메커

니즘이다. 망으로 들어오는 트래픽을 망 경계(network boundary)에서 분류하고, PHB(Per Hop Behavior)가 능하게 조정하여 차등 BA(behavior aggregate)을 할당하고, 각 BA는 단일 DS codepoint에 의해 식별되어 망 내부에서는 단지 DS codepoint에 관련된 PHB에 따라 전송하는 단순한 모델을 기반으로 한다.^[1]

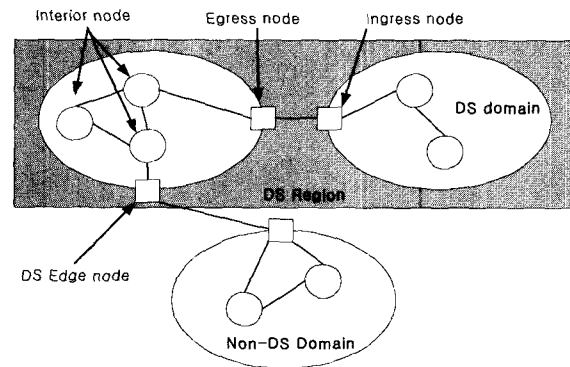


그림 1. Differentiate Service 네트워크 구조

Fig. 1. Network Structure of Differentiate Service.

DS의 기본 정책은 망의 경계노드(edge node)에서 IP 헤더내의 특정영역을 설정(DS codepoint)하고, 특정영역의 값에 따라 입력된 패킷을 어떻게 전송시킬 것인가 결정(PHB)하고, 각 서비스 규격에 맞추어 요구 사항에 일치하는 방법으로 트래픽을 처리하는 것 등을 들 수 있다.^[2]

2.2 TC(Traffic Conditioner) 구성 컴포넌트

원하는 서비스 레벨을 충족시키기 위해 네트워크 가장자리에 있는 지능적인 전송제어 메커니즘을 이용해서 패킷을 마크하고 우선적으로 처리하고자 하는데 이러한 지능적인 전송제어 메커니즘이 TC(Traffic Conditioner)이다[3].

TC의 구성요소들은 TC의 다른 컴포넌트에서 처리하기 이전에 패킷 헤더의 특정영역을 참조하여 트래픽 스트림에서 특정 패킷을 분류하는 Classifier와 Classifier에 의해 분류된 패킷들의 트래픽 특성을 TCA의 트래픽 profile을 기반으로 측정하여 in-profile/out-of-profile을 판별하는 Meter, 패킷 헤더의 DS field 영역에 특정 codepoint값을 지정하는 Marker, 트래픽 스트림을 profile에 일치시키기 위해 하나 이상의 패킷을 지연시키거나 폐기시키는 Shaper, Dropper등의 component로 구성되며 Meter의 세 가지

타입이 존재한다.

첫째, Average Rate Meter는 단순한 meter의 예로 패킷이 들어오는 평균 전송율을 측정하는 방법으로 profile을 고려하여 측정하는데 일치/불일치 중 하나의 level에 상응하는 패킷임을 판단한다.

둘째, Exponential Weighted Moving Average Meter는 다음 계산식과 구현으로 형식화된다.

```

avg(n+1) = (1-Gain) * avg(n) + Gain * actual(n+1)
t(n+1) = t(n) + Delta
if (avg(m) > Arrival Rate)
    non-conforming
else
    conforming
    
```

셋째, Token Bucket Meter는 보다 더 정교한 방법으로 세 개의 파라미터 average rate, packet rate, burst size를 갖는다. 기본적으로 average rate와 패킷의 arrival rate를 비교하는 방법을 사용하며 Token Bucket profile의 일치여부로 측정하는 방법이다.

2.3 토큰 버킷을 사용하는 기존 모델

IETF의 Integrated Services Internet Model에서는 토큰 버킷 모델에 근거를 두고 있는데 이 모델에 근거를 이루고 있는 기존 제안된 방법들에는 다음과 같은 것들이 있다.

srTCM(Single Rate Three Color Marker)은 세 개의 트래픽 파라미터(parameter)에 의해 트래픽 스트림(stream)을 측정하여 green, yellow, red로 패킷을 마크한다. 이 방법은 두가지 모드(Color-Blind, Color-Aware)와 세 가지 트래픽 파라미터인 CIR (Committed Information Rate), CBS(Committed Burst Size), EBS (Excess Burst Size)로 트래픽 전송률을 측정한다. srTCM의 동작방식은 그림 2와 같다.^[4]

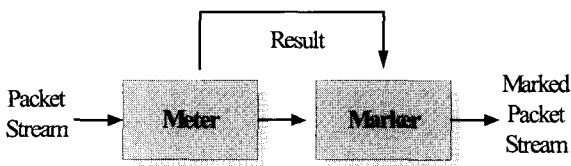


그림 2. srTCM의 Block Diagram
Fig. 2. Block Diagram for srTCM.

trTCM(Two Rate Three Color Marker)은 srTCM과 비슷하게 동작하나 IP 패킷 스트림을 측정하는데 두 개의 rate(PIR, CIR)와 관련된 burst size을 기반으로 하여 green, yellow, red중 하나로 마크하는 방법이다. 이는 패킷이 PIR을 초과하면 red로 마크하고, 그렇지 않으면 CIR를 초과하는지 여부에 따라 yellow, green으로 마크된다. trTCM은 srTCM과 거의 동일한 방법으로 연산된다.^[5]

GTC(Generic Traffic Conditioner)는 아래 그림과 같이 다섯 개의 기능적인 component로 구성된다.

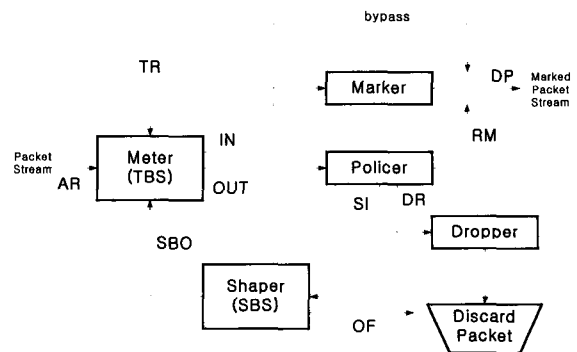


그림 3. GTC의 Block Diagram
Fig. 3. Block Diagram for GTC.

그림 3에 사용된 각 컴포넌트는^[9]에서 명시하고 있는 기능들을 대부분 이행한다. 이는 5개의 컴포넌트들과 세 개의 트래픽 파라미터(Token Rate (TR), Token Bucket Size(TBS), Shaping Buffer Size(SBS))에 의해 구성된다. 사용되는 파라미터는 다음과 같다.

표 1. GTC에 사용되는 파라미터
Table. 1. Parameters used in GTC.

Token Bucket Occupancy(TBO)
Shaping Buffer Occupancy(SBO)
SI : unit time 당 shaper에 도착되는 byte의 수
IN : unit time 당 in-profile packet의 byte
OUT : unit time당 out-of-profile packet의 byte
DR : unit time당 drop되는 out-of-profile packet의 byte 수
RM : unit time당 re-mark되는 byte의 수
OF : unit time 당 shaping buffer overflow 수

이 방법의 알고리즘은 크게 두 부분으로 구성되는데 Meter 부분과 Policier에 관한 부분이다.^[6]

FM(Fair Marker)은 동일 가입자(Subscriber) 네

트위크에서 유입되는 flow들이 그들간의 공정성을 갖도록 하기 위해 토큰 버킷으로부터 토큰 분배를 제어한다. FM의 block diagram은 아래와 같다.

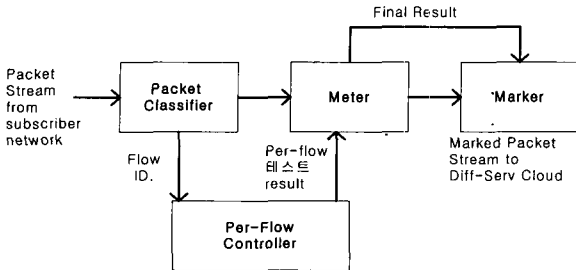


그림 4. FM Block Diagram
Fig. 4. Block Diagram for FM.

그림 4의 meter 컴포넌트는 두 개의 트래픽 파라미터값 할당에 의해 구성된다. PER-FLOW CONTROL 내의 fair 할당 알고리즘은 flow가 다른 flow들과 비교하여 부적절한 토큰 양을 사용하는지 테스트한다[7]. 지금껏 나열한 Token Bucket Meter에 기반한 모델 외에 Average Rate Meter의 방법을 사용하는 TSWTCM(A Time Sliding Window Three Colour Marker)등이 있다.[8]

하지만 이러한 모델들은 여러 서브네트워크에서 기인하는 aggregate flow들을 측정하는 과정에서 토큰 큰 버킷 Burst를 발생시키는 주체가 한 flow에서 기인한 것인지 아니면 여러 flow들에 의해 기인한 것인지를 전혀 고려하지 않고 있는 문제점을 지니고 있다. 즉, classifier를 거친 패킷들(aggregate flow)은 현 시점에서 자신과 같은 서비스를 받고자 하는 패킷들이 많아 Burst가 발생할 경우 예기치 않은 서비스 손실을 입을 수 있다는 것이다.

2.4 PHB 종류와 적용

지금까지 제안된 PHB(Per Hop Behavior)에는 다음과 같은 종류가 있다. IANA에서는 DS영역중 6bit를 사용하여 64개의 codepoint를 사용할 것을 제안하고 있는데 크게 세 가지로 분류하고 있다[9].

Default PHB(codepoint 000000)은 Best-effort 전송특성을 갖는 PHB이다.

트래픽 특성을 여러개의 부류(class)로 구분하고 혼잡 상황이 발생한 경우 drop 레벨을 차별화하여

최소 전송률을 보장하기 위한 AF(Assured Forwarding) PHB Group이 있는데 AF11-AF43까지 12개의 PHB를 제시하고 있다[10].

EF(Expedited Forwarding) PHB에서는 codepoint 101110으로 낮은 loss, delay, jitter가 가능하도록 각 node의 queuing 방법을 최대도착률(maximum arrival rate)이 최소발차률(minimum departure rate)보다 작게 해주자는 서비스가 있다[11].

또한 CBSD(Class Based Service Differentiation)에서는 과거 호환적인(Backward compatibility) 방법으로 상대적인 우선 순위에 기반을 두고 기존 IP precedence bits 값을 이용 가능하도록 제안하는 방법이다[12].

마지막으로 DSWG에서는 DSCP값은 64개이지만 PHB의 수는 제한하지 않고 있다. 즉, 네트워크 domain에서의 DSCP:PHB mapping은 지역적으로 정의할 수 있고, 표준화된 하나의 DSCP:PHB 값 mapping을 권장하지만 네트워크 operator의 다른 방법이 선택 가능하도록 제약을 두지 않고 있다. 이런 이유로 IANA에서 16bit binary field을 사용하여 0-4095범위의 PHB사용을 제안하고 있다[13].

현재 DSWG(Differentiated Service Working Group)에서는 PHB 행위 특성만을 정의하고 있으며 실제 알고리즘이나 메커니즘은 Router vender에 따라 구현하도록 하고 있다. 이는 버퍼관리와 packet scheduling 메커니즘에 의해서 구현 가능한데 Packet 제어를 위해 다양한 파라미터들이 라우터에서 사용 가능하다.

scheduler는 크게 두 종류로 나뉘는데 work conserving disciplines과 Non work conserving disciplines이 있다. 먼저 work conserving disciplines은 큐가 비어있지 않을 때 절대 idle time을 갖지 않는 방법으로 GPS, Delay-EDD, Virtual Clock, WFQ, Self Clocked Fair Queuing, Start-time Fair Queuing의 방법이 있다.

Non-work-conserving disciplines 방법은 end-to-end delay bound에 포커스를 맞추고 있으며 Jitter-EDD, Stop-and-Go, Hierarchical Round Robin 방법 등이 있다.

2.5 Weighted Fair Queuing(WFQ)

WFQ(Weighted Fair Queuing)는 보장된 전송대

역폭서비스를 허락하는 패킷 스케줄링 기술이다. WFQ의 목적은 동일 link를 몇 개의 session이 공유하게 하는 것이다. 추가적인 과도한 전송대역폭 없이 네트워크 사용자들에게 비슷하게 일관된 응답시간을 제공하는 것이 바람직하는데 이러한 해결책이 바로 WFQ이다. WFQ는 queue가 전송대역폭에 대해 기아상태에 빠지지 않고, 트래픽이 예측가능한 서비스를 얻을 수 있으리라는 것을 보장한다. Low-volumn 트래픽 stream이 우선적으로 서비스를 받고 High-volumn 트래픽 스트림이 그들 사이에 비례적으로 남아있는 용량을 공유하는 메커니즘을 이용하며, IP precedence를 QoS Signalling으로 사용할 경우 동적으로 자원 사용의 비율(link 이용비율)을 Signalling을 통해 변경할 수 있다. 또한 RSVP Signalling을 사용해서 버퍼공간 할당이나, 패킷 스케줄링, 예약된 자원에 대해 guarantee한 bandwidth를 제공하기 위해 WFQ를 사용할 수도 있다^[14].

2.6 Traffic Conditioner(TC)와 DS의 PHB 구현

TC는 결과적으로 PHB 적용에 필요한 제어를 수행하는 곳이다. ds에서 제공되는 서비스는 PHB 적용을 어떻게 제공하느냐에 따라 서비스의 종류와 각 서비스가 제공하는 QoS가 결정이 되는데, TC는 서비스를 제공할 flow를 선별하는 기능을 제공하는 것이라 할 수 있다. 서비스를 제공할 flow 선별을 DS 모델에서는 일반적으로 토큰 버킷 모델을 사용하는 meter를 통해 수행된다.

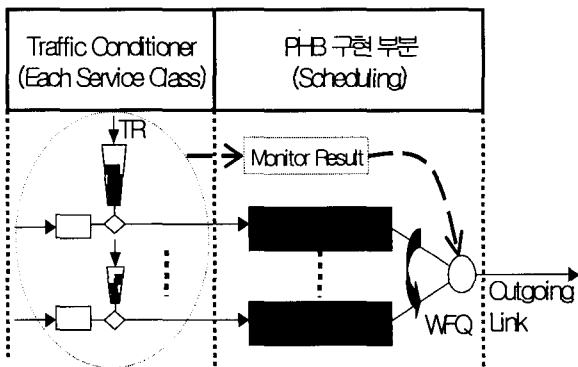


그림 5. DS Node에서의 TC와 PHB구현의 관계
Fig. 5. Relational Model for implementation TC and PHB in a DS node.

그림 5는 각 서비스 class에 대한 모니터링 결과 값을 WFQ에서 링크 이용 비를 산정 하는 값으로 사용하는 것을 보인 것이다.

III. 제안 모델

본 논문에서는 기존의 트래픽 conditioner을 구성하는 meter나 shaper, dropper가 트래픽 profile에 유입된 flow들을 일치시키는 역할만을 수행하고 있는데, 이러한 기능에 자원을 공정하게 사용하고 잉여 자원을 효율적으로 사용할 수 있는 모델을 제시하고 있다.

3.1 제안모델 구성 및 기능

먼저 TC(Traffic Conditioner)에서 각 서비스 class에 대한 트래픽 모니터링은 다음과 같다. DS에서는 개별 flow가 아닌 AggF들 즉, 트래픽 스트림에 대해 전송률과 같은 시간적인 특성을 측정하기 때문에, burst가 발생하는 시점에 도착하는 패킷은 항상 out-of-profile로 판명이 되어 예기치 않은 손실을 야기한다는 문제점이 있다. 극단적인 경우 두 서브네트워크 망에서 진입하는 flow 들간에 경합이 발생할 때 둘 중 하나의 서브네트워크에서 오는 모든 flow가 손실을 입을 수도 있다는 점은 심각한 문제라 할 수 있다.

제안된 트래픽 Conditioner는 classifier를 통해 각 서비스별로 분류된 AggF에 대해 Monitor라는 컴포넌트를 추가하여 버킷의 burst를 사전에 감지하여 burst를 일으키는 AggF중에서 가장 많은 토큰 비율을 사용하는 망의 패킷을 식별하여 토큰 사용의 공정성을 기한다. 이는 개별 AggF(service class)에 대해 자원 공유의 효율을 높이는 것이다.

개별 AggF에 대해 공정한 자원을 공유할 수 있도록 하는 TC를 보이고 있는 그림 6에 사용된 굵은 화살표는 패킷의 흐름을, 가는 화살표는 모니터링을 위해 사용될 패킷양을, 그리고 점선 화살표는 모니터링 결과에 의한 제어 데이터를 나타낸다. 그림 6에 사용된 파라미터는 표 2에서와 같다.

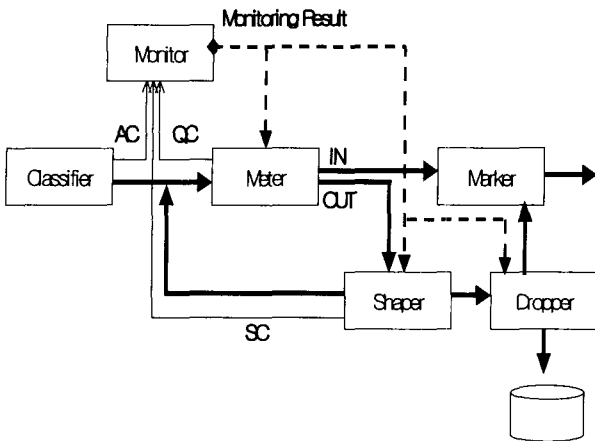


그림 6. 공정한 자원공유를 제공하는 TC
Fig. 6. TC model to suggest efficient resource sharing.

표 2. 제안 모델에 사용되는 파라미터
Table. 2. Parameters used in the suggested model.

NAME	설 명
AC	Classifier에 의해 분류된 aggregated flow양
SC	재측정을 위해 되돌려질 flow양
QC	TB의 처리를 대기하는 flow양
IN	측정후 in-profile로 판명된 packet
OUT	측정후 out-of-profile로 판명된 packet
BS	bucket의 size

제안하는 Traffic Conditioner는 6개의 컴포넌트로 구성된다.

① Classifier 컴포넌트는 MF(Multi-Field) Classifier를 사용한다. 유입되는 flow들을 다수개의 필드를 통해 식별하는 기능을 수행한다. flow가 Classifier의 처리를 거치고 나면, DS Domain과 협약된 service class가 무엇인가를 알 수 있고 할당되는 트래픽 profile에 따라 다음 컴포넌트들에 의해 트래픽 조절을 받는다. 여기서 트래픽 profile에는 협약된 SLS나 TCS등의 정보가 저장되어있다. 무엇보다 중요한 classifier 컴포넌트의 기능은 Monitor, Meter, Shaper, Dropper 컴포넌트에서 사용될 파라미터인 (flow 식별자, flow 양-패킷크기) 값을 식별이 끝난 후 패킷 전송 시 함께 보내준다는 것이다. 즉, Monitor 컴포넌트로 AC를 보낼 때는 물론 Meter 컴포넌트로 패킷을 전달할 때 위의 파라미터 정보

를 함께 보낸다.

② Meter 컴포넌트는 토큰 버킷 모델을 이용한다. Meter의 기본적인 동작은 트래픽 profile에 준하여 트래픽스트림을 측정하고 트래픽 stream이 in-profile인지 out-of-profile인지를 판별한다. in-profile인 경우는 Marker 컴포넌트로 traffic stream을 전송하고 out-of-profile인 경우는 Shaper 컴포넌트로 traffic stream을 전송한다. 이러한 기능에 추가적으로 Monitor 컴포넌트에 의해 모니터링 된 결과를 이용하여 공정한 자원사용을 어긴 flow에 대해 traffic stream의 측정을 거치지 않고 Shaper로 바로 전송한다.

③ Shaper 컴포넌트는 out-of-profile로 판정된 flow를 traffic profile에 일치시키기 위해 flow의 일부 또는 전체를 지연시키는 역할을 한다. 이러한 기능에 추가적으로 모니터링 결과를 이용하여 다음과 같은 작업을 수행한다.

- ㉠ Meter 컴포넌트로 재 측정을 위해 보낼 packet을 선별하여 처리
- ㉡ 재 측정을 위해 보냈을 때 부하만을 발생시킬 경우 바로 Dropper로 전달

이러한 방법을 통해 불필요한 작업을 반복하지 않음으로서 처리속도를 향상시킬 수 있다

④ Monitor 컴포넌트는 TC의 다음 단계에서 처리되어질 상황을 미리 예측하는 일을 담당한다. Monitor 컴포넌트로의 입력은 AC, QC, SC를 기반으로 Meter와 Shaper, Dropper 컴포넌트에서 처리할 수 있는 파라미터를 제공하는 역할을 한다.

Monitor 컴포넌트는 하나이상의 (flow 식별자, flow 양) 값을 가지는 AC, QC, SC 값을 입력으로 취하여 다음과 같은 결과를 산출한다.

- ㉢ aggregated flow를 구성하는 flow중에서 공정정을 어기는 flow 식별자와 flow양
- ㉣ Shaper에서 재측정(Remetering)을 위해 보내질 flow양
- ㉤ 다음 단계에서 이용 가능한 잉여 자원에 대한 flow양 => Dropper에서 바로 Marker로 전송될 flow양
- ㉥ Shaper에서 Dropper 컴포넌트로 보내질 flow양

Monitor 컴포넌트 동작의 알고리즘은 아래와 같다.

표 3. Monitor 컴포넌트의 알고리즘

Table. 3. Algorithm for Monitor Components.

- ㉑ 모든 파라미터 초기화
- ㉒ AC, QC, SC 입력(또는 다른 컴포넌트에 의해 resetting됨)
- ㉓ (flow 식별자, flow 양)의 값 갱신
- ㉔ if $AC+SC+QC > S_Bytes$ then
 - ㉕ 만약 Burst가 SC에 의해 기인된다면
 - ㉖ F_Num 만큼 반복
 - (1) 각 flow에서 평균을 초과하는 flow 양을 산정 ($R_Fnn = \text{개별 flow 양} - \text{Temp}$)
 - ㉗ 재추정을 위해 전송될 flow양을 0으로 지정
 - ㉘ SC flow를 dropper로 전달
 - ㉙ 그렇지 않으면
 - ㉚ $\text{Temp} = S_Bytes - (AC+QC)$
 - ㉛ 재추정을 위해 전송되는 flow양을 Temp로 지정
 - ㉜ else
 - ㉝ $S_Bytes - (AC + SC + QC)$ bytes를 Dropper에서 Marker로 전송
- ㉞ 종료

㉕ Dropper 컴포넌트는 모니터 컴포넌트의 결과에 의해 잉여자원을 효율적으로 사용할 수 있는 기능을 수행한다. 그 외 Marker 컴포넌트는 DS 아키텍처에서 정의하고 있는 사항을 처리하는 컴포넌트이다.

Meter, Shaper, Dropper 컴포넌트로 패킷이 전달되면 각각의 컴포넌트는 패킷유실을 방지하기 위하여 queue에 먼저 보관을 하게 되는데 본 논문에서 사용되는 컴포넌트들의 Queue 구조는 그림 7과 같다. flow 식별자와 flow 양은 패킷 전달 시 함께 대상 컴포넌트로 전달되는 파라미터로 전체적인 모니터를 위해 사용되는 값이다.

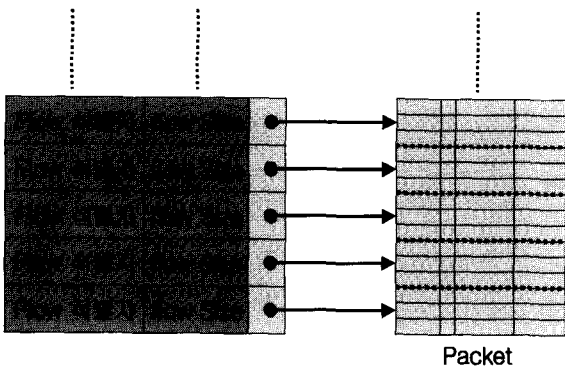


그림 7. 구성 컴포넌트에서 사용되는 queue
Fig. 7. Queue structure used in components.

Meter와 Shaper 컴포넌트에서는 그림 7과 같이 Queue을 사용한다. 우측의 모양은 배열과 같은 자

료구조를 사용한다. 화살표는 실제 패킷이 위치한 위치를 가리킨다. Dropper 컴포넌트에서는 Flow 식별자를 관리하는 대신에 패킷이 도착된 시간 값을 사용한다. Meter, Shaper, Dropper 컴포넌트 모두 queue의 overflow를 처리하기 위해 flow size의 총합이 항상 정해진 Queue size의 범위를 넘지 않도록 처리된다.

3.2 성능평가를 위한 시뮬레이션

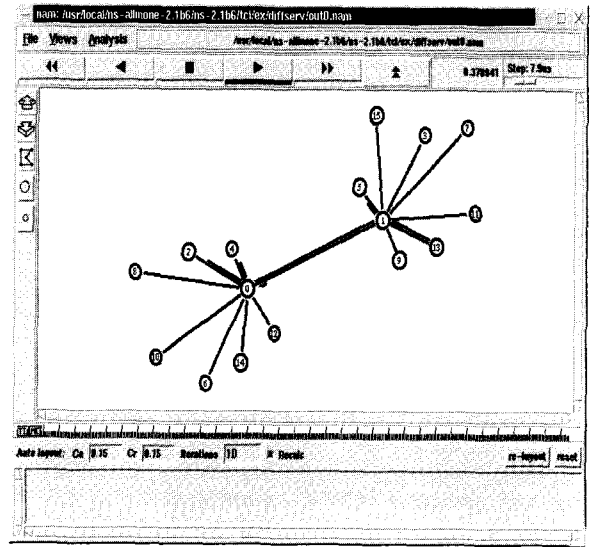


그림 8. 시뮬레이션을 위한 토폴로지
Fig. 8. Topology for simulation.

실험을 하기위해서 ns-2 시뮬레이터를 사용하였으며, 그림 8의 토폴로지는 NAM (Network Animator) 유틸리티를 이용하여 보여지는 그림으로 노드번호 0과 1이 DS 서비스를 제공하는 노드이고, 양쪽 7개 쌍의 노드들이 DS 망을 이용하는 근원지와 목적지 노드를 가리킨다. 7개의 근원지 노드는 두 개의 EF 서비스로 협약된 노드(source 0,1), 3개의 AF 서비스로 협약된 노드(source 2,3,4)와 2개의 BE 서비스(source 5,6)로 협약된 노드로 구성하였다.

근원지에서 DS Edge 노드로 연결된 노드(노드 0, 1 과 연결된 노드)들 사이의 링크 대역폭은 10Mbps이고 0.1ms의 지연 값을 갖도록 설정되었고, 노드 0에서 1간의 링크대역폭은 5Mbps이고 1ms의 지연 값과 1Mbps의 대역폭과 1ms의 지연 값을 각각 적용하였다.

기존 DS 모델을 1Mbps의 대역폭 1ms의 지연과, 5Mbps의 대역폭으로 실행했을 때 결과는 아래 표

4과 같다. 먼저 대역폭을 1Mbps로 설정한 경우 각 노드들이 DS 망을 사용할 때 서비스를 제공받은 대역폭을 보이고 있다. 세 개의 AF11 서비스로 협약된 소스 2, 3, 4는 각각 114, 127, 148Kbps의 서비스를 제공받았음을 의미하고 있다. 세 개의 동일한 서비스레벨에 속하는 flow들 중 제공받은 서비스의 최고 차가 34Kbps임을 보이고 있다.

표 4. 1Mbps 대역폭에서의 자원사용량
Table. 4. The resource usability on the 1 Mbps bandwidth.

Goodput per source:
Goodput for source 0 (EF): 99.372 kb/s
Goodput for source 1 (EF): 0.63 kb/s
Goodput for source 2 (AF11): 114.0 kb/s
Goodput for source 3 (AF11): 127.90000000000001 kb/s
Goodput for source 4 (AF11): 148.09999999999999 kb/s
Goodput for source 5 (BE): 250.0 kb/s
Goodput for source 6 (BE): 250.0 kb/s

표 5는 동일 환경에서 대역폭만을 5Mbps로 변경하여 실행한 시뮬레이션 결과를 보이고 있다.

표 5. 5Mbps 대역폭에서의 자원사용량
Table. 5. The resource usability on the 5 Mbps bandwidth.

Goodput per source:
Goodput for source 0 (EF): 199.75200000000001 kb/s
Goodput for source 1 (EF): 199.773 kb/s
Goodput for source 2 (AF11): 638.5 kb/s
Goodput for source 3 (AF11): 707.5 kb/s
Goodput for source 4 (AF11): 691.0 kb/s
Goodput for source 5 (BE): 1277.90000000000001 kb/s
Goodput for source 6 (BE): 1278.0 kb/s

표 5의 결과에서도 표 4에서의 결과와 마찬가지로 AF 서비스로 협약된 flow들이 자원 사용량의 차이가 69Kbps로서 공정하게 사용되지 못함을 보이고 있으며, 네트워크의 대역폭이 적을수록 자원의 불공정 서비스가 더욱 심화되고 있다.

제안모델 적용전의 AF 서비스에 해당되는 flow들의 자원사용의 최대 차가 34Kbps의 결과였는데, Monitor 컴포넌트를 통한 사전의 모니터링 결과를 적용하였을 경우는 표 6과 같이 동일 서비스 레벨에 해당하는 flow가 대략 5Kbps의 차를 보임을 알 수 있다.

표 6. 제안모델의 1Mbps 대역폭에서의 실행 결과
Table. 6. The resource usability of the suggested model on the 1 Mbps bandwidth.

Goodput per source:
Goodput for source 0 (EF): 99.372 kb/s
Goodput for source 1 (EF): 0.63 kb/s
Goodput for source 2 (AF11): 136.0 kb/s
Goodput for source 3 (AF11): 131.30000000000001 kb/s
Goodput for source 4 (AF11): 131.19999999999999 kb/s
Goodput for source 5 (BE): 250.0 kb/s
Goodput for source 6 (BE): 250.0 kb/s

표 7에서는 링크 대역폭을 5Mbps로 설정한 경우의 결과를 보인다. 이 결과에서도 동일 서비스레벨의 flow 들간의 최대 차가 7Kbps임을 확인할 수 있다.

표 7. 제안모델의 5Mbps 대역폭에서의 실행결과
Table. 7. The resource usability of the suggested model on the 5 Mbps bandwidth.

Goodput per source:
Goodput for source 0 (EF): 199.75200000000001 kb/s
Goodput for source 1 (EF): 199.75200000000001 kb/s
Goodput for source 2 (AF11): 678.89999999999998 kb/s
Goodput for source 3 (AF11): 684.29999999999995 kb/s
Goodput for source 4 (AF11): 677.20000000000005 kb/s
Goodput for source 5 (BE): 1277.8 kb/s
Goodput for source 6 (BE): 1278.0 kb/s

그림 9에서 그림 12까지는 DS node에서 이를 수행하는 전체적인 수행절차를 개괄적으로 보여주고 있으며, 시간(x축, sec)흐름에 대해서 전송대역폭(y축, kbps)을 나타낸다. 또한 r2, r3, r4은 각각 AF11 서비스의 source 2, source 3, source 4를 나타내고 있다.

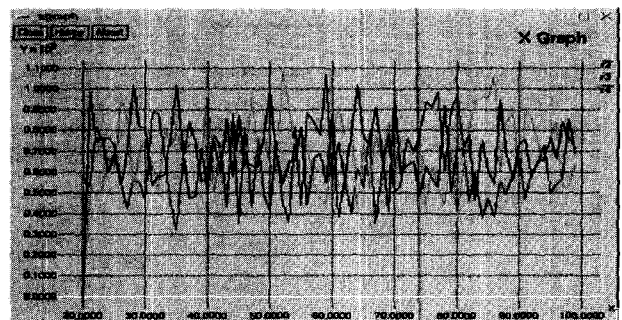


그림 9. 기존 모델의 AF PHB에 해당되는 세 개의 flow가 받는 서비스
Fig. 9. The result of usability for 3 flows according to AF PHB in old model.

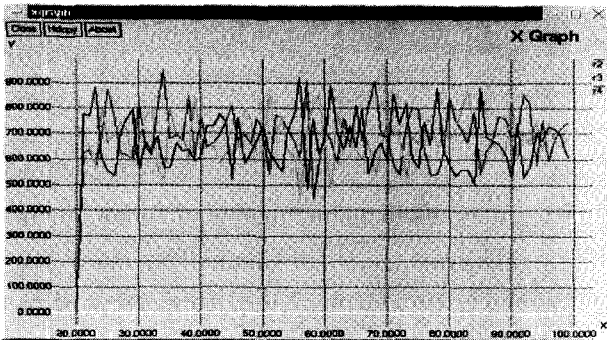


그림 10. 제안모델 적용 후의 AF11의 자원 사용량
Fig. 10. The result of usability for AF PHB after implementing the suggested model.

그림 10에서 시간대별 각 flow가 사용한 자원의 양(대역폭)을 보면 불특정 flow의 자원사용의 상한은 900Kbps를 가리키고, 자원 사용의 하한은 450Kbps 이상을 사용하고 있음을 알 수 있다. 이 결과는 다수개의 flow로 구성되는 AggF에 대한 서비스의 하한을 예상할 수 있다는 것이다. 즉, DS에서 제공하는 서비스가 어떠한 경우라도 결과로 보여지는 450Kbps이상의 서비스를 제공할 수 있다는 것이다.

그림 9와 그림 10의 결과를 통하여 최소 자원만을 할당하여 서비스를 제공하고 있는 AF 서비스에서는 언제든지 네트워크가 혼잡상황을 발생시킬 수 있는데 기존모델에서는 이러한 혼잡상황에서의 불특정 flow의 QoS 손실로 서비스를 예측할 수 없었다. 하지만 제안 모델에서는 어떠한 경우라도 네트워크 사업자와 사용자가 협약한 서비스레벨의 예측된 하한의 서비스를 제공할 수 있음을 알 수 있다.

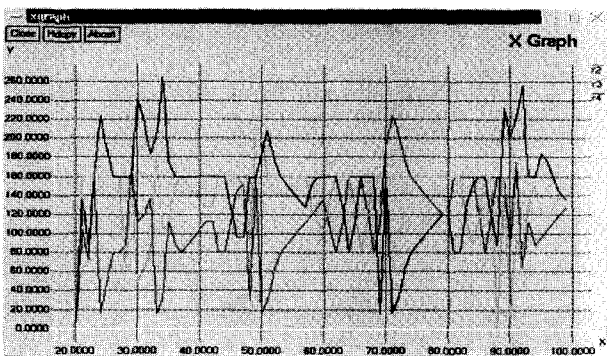


그림 11. 1Mbps에서 기존 모델의 AF11의 자원 사용량
Fig. 11. The result of usability for AF PHB on 1 Mbps in old model.

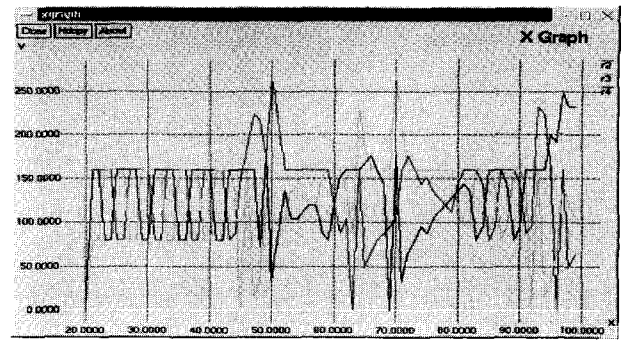


그림 12. 1Mbps에서 제안 모델의 AF11의 자원 사용량
Fig. 12. The result of usability for AF PHB on 1 Mbps in the suggested model.

3.3 시뮬레이션 결과 분석

시뮬레이션의 결과들은 각 서비스 class에 대한 모니터링 결과값을 WFQ에서 링크 이용 비를 산정하는 값으로 사용한다. 물론 결과의 트래픽 Condition의 내용은 그림 6의 모든 컴포넌트를 대신하는 트래픽 스트림 측정의 단편을 보인 것이다.

WFQ의 가중값은 각 AggF를 측정하는 TBS로 간주한다. 즉, 단위 시간당 WFQ에서 처리할 수 있는 트래픽 양은 트래픽 Conditioner에서 단위 시간에 측정할 수 있는 양과 같으며 TC에서 사용되는 TBS의 전체 합이 WFQ에서 링크의 비로 사용되는 전체 합과 같다. WFQ는 이러한 TBS값의 비율로 링크의 비율을 결정한다.

그림 11과 그림 12의 Monitor Result로 제시된 결과값은 WFQ에서 사용할 TBS의 값을 Update한 결과이다. 이는 실제 TBS의 크기를 변경하는 것이 아닌 WFQ에서 가중값 결정을 위해 참조하는 임시 TBS값들이다. TBS값은 토큰 버킷에서 트래픽 스트림을 측정하여 보내지는 트래픽 양은 협약된 양을 초과할 수 없으므로 임의의 AggF의 양이 잉여자원을 갖는다면 burst가 발생한 AggF의 트래픽을 잉여 자원만큼 Shaper에서 재측정을 위해 보내지 않고 Marker로 전송하여 다음단계에서 잉여자원을 효율적으로 사용될 수 있도록 할 수 있다. 제안모델에서는 Monitor 결과값에 따라 임의의 AggF의 잉여자원을 burst가 발생한 AggF의 서비스 class에서 사용하는 방법을 보이고 있다. Shaper에서 다음단계 상황에서 재측정되거나 dropper될 flow양을 Marker로 전송하여 잉여 자원을 사용한다. 이와 같은 방법을 사용하여 다음단계에서 적용될 WFQ의 링크 이용비를 이용하여 서비스 사업자

네트워크에서 제공되는 DS에 관한 트래픽들을 잉여 자원의 낭비 없이 효율적으로 사용할 수 있다. 모니터링의 결과 값을 WFQ의 Signal로 사용하는 단순한 메커니즘을 통해 output 링크의 효율을 높일 수 있다.

VI. 결론 및 추후 연구

본 논문에서는 DS의 Edge node 또는 Border router에서 이용 가능한 효율적인 자원 공유에 대한 방안을 제시하였다. 각 노드에는 공정한 자원공유를 할 수 있는 트래픽 Conditioner와 PHB를 이행할 WFQ를 사용하였고 트래픽 Conditioner에서 모니터링된 결과를 WFQ에서 참조하여 잉여 자원을 사용할 수 있는 방안을 제시하였다.

현재 논문에서는 다양한 트래픽 특성에 대한 추가적인 실험과 더불어 실제 환경에서 적용을 위한 연구가 진행되고 있으며, 향후 연구방향은 트래픽 특성을 고려하여 좀더 정교한 QoS를 만족시킬 수 있는 사항들을 연구되어야 할 것이다.

참 고 문 헌

- [1] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC2474, December 1998
- [2] Davies, E., Keshav, S., Verma, D., Carlson, M., Ohlman, B., Blake, S., Bernet, Y., Binder, J., Wang, Z., Weiss, W., "A Framework for Differentiated Services", Internet Draft, October 1998
- [3] Y. Bernet, A. Smith, S. Black, "A Conceptual Model for Diffserv Routers", Internet Draft, December 1999
- [4] J. Heinanen and R. Guerin, "A Single Rate Three Color Marker," RFC2697, March 1999.
- [5] J. Heinanen and R. Guerin, "A Two Rate Three Color Marker," RFC2698, March 1999.
- [6] Fang-Ching Ou, "A Generic Traffic Conditioner" Internet Draft, October 1999
- [7] Hyogon Kim, "A Fair Marker", INTERNET-DRAFT, October 1999
- [8] Wenjia Fang, "A Time Sliding Window Three Colour Marker(TSWTCM)", INTERNET-DRAFT, October, 1999
- [9] D. Black, S. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, " An Architecture for Differentiated Services", RFC 2475
- [10] J. Heinanen et al., "Assured Forwarding PHG Group," RFC 2587, February 1999.
- [11] Jacobson, V., Nichols, K., "An Expedited Forwarding PHB", RFC 2598, August 1998
- [12] Constantinos Dovrolis, "Class-Based Service Differentiation", Internet Draft, December 1998
- [13] Peter, F., "Integrated Services Over Differentiated Services", Internet Draft, March 1998
- [14] <http://ipsit.bu.edu/sc546/sc441Spring2003/wfq/wfq.htm>

저 자 소 개



장 경 성(정희원)

e-mail : unixhunt@chodang.ac.kr

1986년 전남대학교 물리학과 졸업
(학사)

1991년 호주 타스마니아주립대 응용
컴퓨터공학과 Graduated Dip. 졸업

1996년 전남대학교 전산통계학과(이학석사)

2002년 전남대학교 전산통계학과(이학박사)

1997년 - 현재 초당대학교 정보통신공학과 조교수
<주관심분야: 인터넷, 이동통신, 망관리 및 QoS, 네트워크 보안, 운영체제>