

Differentiated Service에서 효율적인 자원 공유에 관한 연구

이진호*, 장경성**, 김용수*, 김병기*

*전남대학교 정보통신연구소

**초당대학교 정보통신공학과

e-mail:jinho@superse.chonnam.ac.kr

A Study on Effective Resource Share for Differentiated Service

Jin-Ho Lee*, Kyung-Sung Jang**, Young-Su Kim*, Byung-Ki Kim*

*Dept of Computer Science, Chonnam-Nam National University

**Dept of Inform. & Telecomm., Chodang University

요약

DS(Differentiated Service)에서는 RSVP와 같이 개별 flow에 대한 QoS제공이 아닌 flow들의 집합(aggregate)으로 차등적인 서비스를 제공하는 방법이다. 확장성을 고려한 이러한 방안은 광역백본망 에서와 같이 수천, 수만의 flow들에 대해 서비스를 제공하고자 할 때 flow들을 집합(aggregate)화 하여 차등적인 서비스를 제공하고자 하는 것은 매우 큰 의미가 있다고 할 수 있다. 하지만 DS에서는 Token Bucket 방법을 사용하여 트래픽 스트림을 측정하는 이유로 불특정 flow에 대해 심각한 QoS손실을 야기할 수 있다. 본 논문에서는 자원 공유의 공정성을 기여하기 위해 Traffic Conditioner에서 사전 모니터링 방법을 사용하고 이 모니터링 결과를 Weighted Fair Queuing의 파라미터로 사용하여 잉여 자원을 효율적으로 사용하는 방안을 제시한다.

1. 서론

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. 초고속으로 데이터를 전송하려는 각 차세대 인터넷 프로그램들의 노력으로는 IP over ATM, IP over ATM over SONET/SDH, DWDM등을 이용한 초고속 광통신기술등을 들 수 있다[15].

본 논문에서는 DS의 Edge node 또는 Border router에서 다음단계에서 처리될 AggF(aggregate flow)의 상황을 모니터링할 수 있는 Traffic Conditioner와 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(Traffic Conditioner), PHB에 대해 간략히 소개하고, 3장에서는 공정한 자원공유를 위한 TC에 대해 기술하고, TC의 모니터 결과값을 이용하여 WFQ에서 각 class의 weight값으로 적용하여 동적으로 링크 이용 비율을 결정하는 방안을 제시하고 결론을 맺도록 하겠다.

2. 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 network provider가 그들 customer에게 provider network내에서 받아들여 질 수 있는 레벨의 서비스를 제공하는 메커니즘이다. 망으로 들어오는 traffic을 망 경계(network

boundary)에서 분류하고, PHB(Per Hop Behavior) 가능하게 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[3].

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 처리(Traffic Conditioning)하는 것 등을 들 수 있다[2].

2.2 TC(Traffic Conditioner) 구성 컴포넌트

원하는 서비스 level을 충족시키기 위해 네트워크 edge에 있는 지능적인 전송제어 메커니즘을 이용해서 패킷을 마크하고 우선적으로 처리하고자 하는데 이러한 지능적인 전송제어 메커니즘이 TC(Traffic Conditioner)이다[4].

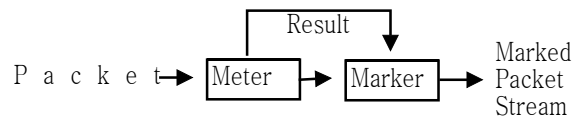
TC의 구성요소들은 TC의 다른 컴포넌트에서 처리하기 이전에 패킷 헤더의 특정영역을 참조하여 트래픽 스트림에서 특정 패킷을 분류하는 Classifier, Classifier에 의해 분류된 패킷들의 트래픽 특성을 TCA의 traffic profile을 기반으로 측정하여 in-profile/out-of-profile을 판별하는 Meter, 패킷 헤더의 DS field 영역에 특정 codepoint값을 지정하는 Marker, 트래픽 스트림을 profile에 일치시키기 위해 하나 이상의 패킷을 지연시키거나 폐기시키는 Shaper, Dropper등의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

2.3 Token Bucket을 사용하는 기존 모델

IETF의 Integrated Services Internet Model에서는 token bucket 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 traffic parameter에 의해 traffic stream을 측정하여 green, yellow, red로 패킷을 마크한다. 이 방법은 두 가지 모드(Color-Blind, Color-Aware)와 세 가지 트래픽 파라미터인 CIR(Committed Information Rate), CBS(Committed Burst Size), EBS(Excess Burst Size)로 Traffic rate를 측정한다.

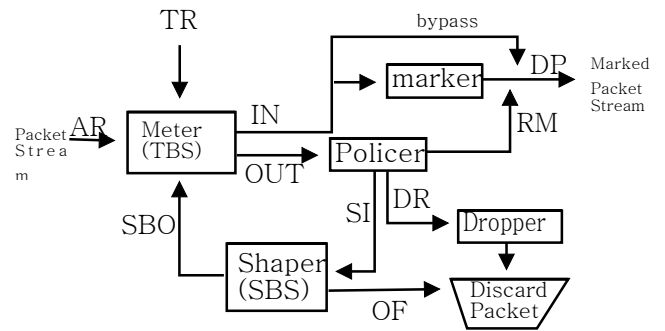
srTCM의 동작방식은 아래 그림과 같다[5].



[그림1] srTCM의 Block Diagram

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[6].

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



[그림2] GTC의 Block Diagram

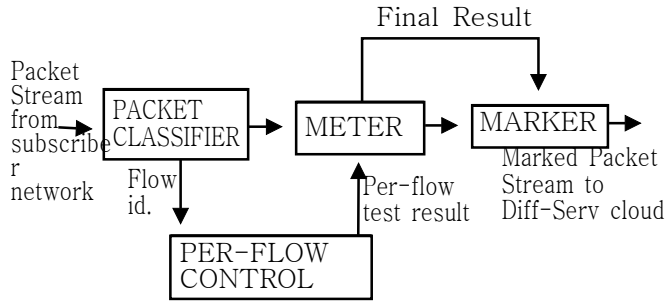
[그림2]에 사용된 각 컴포넌트는 [1]에서 명시하고 있는 기능들을 대부분 이행한다. 이는 구성되는 컴포넌트와 세 개의 traffic parameter(Token Rate (TR), Token Bucket Size(TBS), Shaping Buffer Size(SBS))에 의해 구성된다. 사용되는 parameter는 다음과 같다.

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 time 당 drop되는 out-of-profile packet의 byte 수
RM : unit time당 re-mark되는 byte의 수
OF : unit time 당 shaping buffer overflow 수

[표1] GTC에 사용되는 파라미터

이 방법의 알고리즘은 크게 두 부분으로 구성되는데 Meter 부분과 Policer에 관한 부분이다[7].

FM(Fair Marker)은 동일 subscriber network에서 유입되는 flow들이 그들간의 공정성을 갖도록 하기 위해 token bucket으로부터 토큰 분배를 제어한다. FM의 block diagram은 아래와 같다.



[그림3] FM Block Diagram

[그림3]의 meter 컴포넌트는 두 개의 traffic parameter값 할당에 의해 구성된다. PER-FLOW CONTROL 내의 fair 할당 알고리즘은 flow가 다른 flow들과 비교하여 부적절한 token 양을 사용하는지 test한다[8]. 지금껏 나열한 Token Bucket Meter에 기반한 모델 외에 Average Rate Meter의 방법을 사용하는 TSWTCM(A Time Sliding Window Three Colour Marker)등이 있다[10].

하지만 이러한 모델들은 여러 subnetwork에서 기인하는 aggregate flow들을 측정하는 과정에서 Token Bucket 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[1].

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

트래픽 특성을 여러개의 부류(class)로 구분하고

congestion 상황시 drop level을 차별화하여 최소 전송률을 보장하기 위한 AF(Assured Forwarding) PHB Group이 있는데 AF11-AF43까지 12개의 PHB를 제시하고 있다[12].

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[13].

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

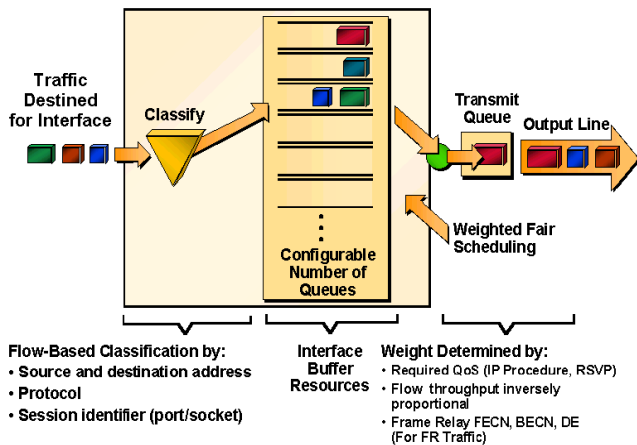
현재 DSWG(Differentiated Service Working Group)에서는 PHB 행위 특성만을 정의하고 있으며 실제 알고리즘이나 메커니즘은 Router vender에 따라 구현하도록 하고 있다. 이는 버퍼관리와 packet scheduling 메커니즘에 의해서 구현 가능한데 Packet 제어를 위해 다양한 parameter들이 라우터에서 사용 가능하다.

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 Robin방법 등이 있다.

2.5 Weighted Fair Queuing(WFQ)

WFQ(Weighted Fair Queuing)는 보장된 bandwidth 서비스를 허락하는 패킷 스케줄링 기술이다. WFQ의 목적은 동일 link를 몇 개의 session이 공유하게 하는 것이다. 추가적인 과다한 bandwidth 없이 네트워크 사용자들에게 비슷하게 일관된 응답시간을 제공하는 것이 바람직한데 이러한 해결책이 바로 WFQ이다. WFQ는 queue가 bandwidth에 대해 기아 상태에 빠지지 않고, traffic이 예측가능한 서비스를 얻을 수 있으리라는 것을 보장한다. Low-volumn traffic stream이 우선적으로 서비스를 받고 High-volumn traffic stream이 그들 사이에 비례적으로 남아있는 용량을 공유하는 메커니즘을 이용한다[16].



[그림4] Weighted Fair Queuing

[그림4]는 일반적인 WFQ의 수행과정을 보이고 있다. WFQ의 이점 중 가장 중요한 것은 QoS signalling과의 접목으로 더 향상된 기능을 수행할 수 있다는 것이다. 예를들어 IP precedence값을 Signalling으로 사용할 경우, 한 인터페이스 상에서 각각의 precedence level이 하나의 flow를 가진다면 각 flow는 링크의 precedence+1 부분을 사용하게 된다.

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$$

즉 각각의 flow들은 링크의 8/36, 7/36, 6/36,...,1/36을 얻을 것이다. 그런가 하면, precedence-1번째 flow가 18개이고 다른 flow들은 각각 하나의 flow를 가진다고 가정하면,

$$1 + 18 * 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36 - 2 + 18 * 2 = 70$$

flow들은 링크의 8/70, 7/70, 6/70,..., 1/70을 얻고

18개의 flow 들은 대략 링크의 2/70을 얻을 것이다.

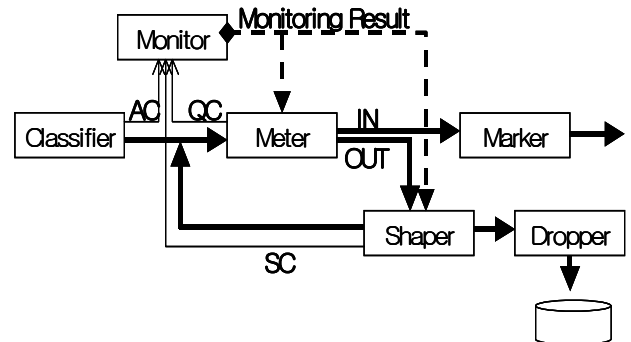
즉, IP precedence를 QoS Signalling으로 사용할 경우 동적으로 자원 사용의 비율(link 이용비율)을 Signalling을 통해 변경할 수 있다는 것이다. 또한 RSVP Signalling을 사용해서 버퍼공간 할당이나, 패킷 스케줄링, 예약된 자원에 대해 guarantee한 bandwidth를 제공하기 위해 WFQ를 사용할 수도 있다 [17].

3. 제안내용

먼저 TC(Traffic Conditioner)에서 각 서비스 class에 대한 traffic 모니터링은 다음과 같다. DS에서는 개별 flow가 아닌 AggF(aggregate flow)들 즉, traffic stream에 대해 시간적인 특성(rate, ..)을 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 서브네트워크 망에서 오는 모든 flow가 손실을 입을 수도 있다는 점은 심각한 문제라 할 수 있다.

사용되는 Traffic Conditioner는 classifier를 통해 각 서비스별로 분류된 AggF(aggregate flow)에 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.

아래그림은 개별 AggF에 대해 공정한 자원을 공유할 수 있도록하는 TC를 보이고 있다.



[그림5] 공정한 자원공유를 제공하는 TC

[그림5]에 사용된 굵은 화살표는 패킷의 흐름을, 가는 화살표는 모니터링을 위해 사용될 패킷양을, 그리고 점선 화살표는 모니터링 결과에 의한 제어 데이터를 나타낸다.

[그림5]에 사용된 파라미터는 다음과 같다.

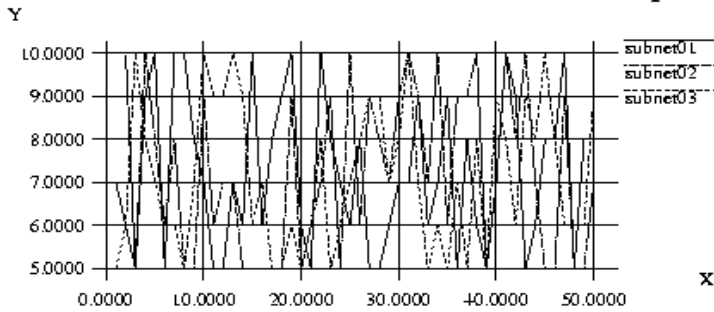
NAME	설 명
AC	Classifier에 의해 분류된 aggregated flow양
SC	remetering을 위해 되돌려질 flow양
QC	TB의 처리를 대기하는 flow양
IN	측정후 in-profile로 판명된 packet
OUT	측정후 out-of-profile로 판명된 packet
BS	bucket의 size

[표2] 제안 모델에 사용되는 파라미터

모니터 컴포넌트는 classifier에 의해 AggF로 분류된 packet의 flow양(AC)과 대기 큐에 대기중인 flow양, 그리고 shaper에서 다음 단계에서 재 측정을 받을 flow양을 기반으로 하여 monitoring을 한다. 그리고 이 모니터링 결과를 참조하여 Token Bucket 처리기와 shaper 처리기에서 처리를 한다.

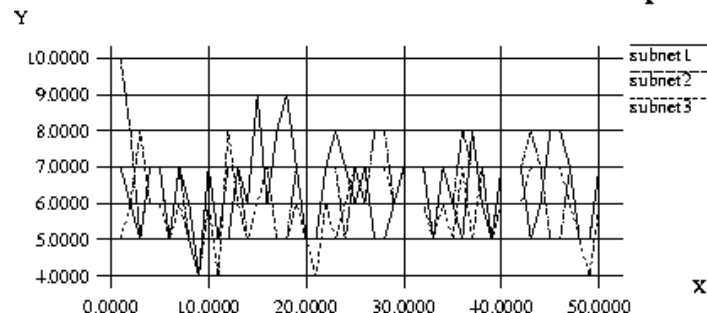
이러한 TC 사용의 결과를 비교해 보면 다음과 같은 결과를 얻을 수 있다.

X Graph



[그림6] 개별 aggregate flow를 구성하는 각 flow량

X Graph

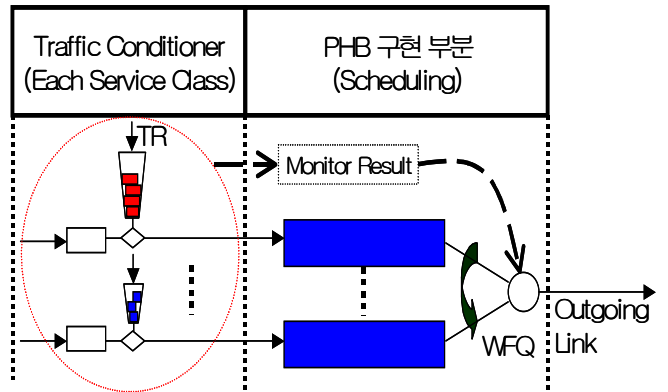


[그림7] 공정한 자원공유 TC 적용후의 각 flow량

[그림6]와 [그림7]을 비교해 보면 적용하는 TC 사용으로 공정한 자원공유를 제공할 수 있다.

개별 AggF를 위와 같이 처리할 때 다수의 AggF를 적용하는 방법을 제시하고자 한다. 이 방법의 핵심은 다수의 AggF의 모니터링 결과를 WFQ의 weight값 적용의 Signal로 사용한다는 것이다.

아래 그림은 DS node에서 이를 수행하는 전체적인 내용을 보여준다.



[그림8] DS Node에서의 TC와 PHB구현의 관계

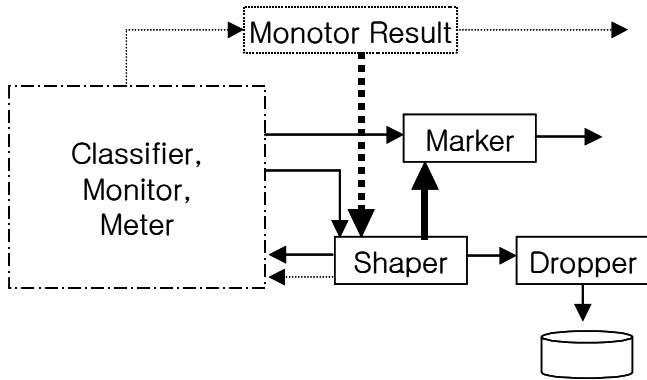
[그림8]은 각 서비스 class에 대한 모니터링 결과값을 WFQ에서 링크 이용 비를 산정 하는 값으로 사용하는 것을 보인 것이다. 물론 [그림8]의 Traffic Condition의 내용은 [그림5]의 모든 컴포넌트를 대신 하는 traffic stream 측정의 단편을 보인 것이다.

WFQ의 weight값은 각 AggF(aggregate flow)를 측정하는 TBS(Token Bucket Size)로 간주한다. 즉, 단위 시간당 WFQ에서 처리할 수 있는 traffic 양은 Traffic Conditioner에서 단위 시간에 측정할 수 있는 양과 같다는 것이다. 바꾸어 말하면 TC에서 사용되는 TBS의 전체합이 WFQ에서 링크의 비로 사용되는 전체합과 같다는 것이다. WFQ는 이러한 TBS값의 비율로 링크의 비율을 결정한다.

Monitor Result로 제시된 결과값은 WFQ에서 사용할 TBS의 값을 Update한 결과이다. 이는 실제 Token Bucket Size의 크기를 변경하는 것이 아닌 WFQ에서 weight 결정을 위해 참조하는 임시 TBS 값들이다. 이런 Token Bucket Size값은 다음과 같은 방법으로 얻을 수 있다.

Token Bucket에서 traffic stream을 측정하여 보내지는 traffic 양은 협약된 양을 초과할 수 없으므로 임의의 AggF(aggregate flow)의 양이 잉여자원

을 갖는다면 burst가 발생한 AggF의 트래픽을 잉여 자원만큼 Shaper에서 remetering을 위해 보내지 않고 Marker로 전송하여 다음단계에서 잉여자원을 효율적으로 사용될 수 있도록 할 수 있다. 아래 그림은 이러한 상황을 보여준다.



[그림9] 잉여자원의 flow량을 Shaper에서 사용

[그림9]에서는 Monitor 결과값에 따라 임의의 AggF(aggregate flow)의 잉여자원을 burst가 발생한 AggF의 서비스 class에서 사용하는 방법을 보이고 있다. Shaper에서 다음단계 상황에서 remetering 되거나 dropper될 flow량을 Marker로 전송하여 잉여 자원을 사용한다.

이와 같은 방법을 사용하여 다음단계에서 적용될 WFQ의 링크 이용비를 이용하여 service provider network에서 제공되는 DS에 관한 traffic들을 잉여 자원의 낭비 없이 효율적으로 사용할 수 있다. 모니터링의 결과 값을 WFQ의 Signal로 사용하는 단순한 메커니즘을 통해 output 링크의 효율을 높일 수 있다.

4. 결론 및 향후 연구

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

현재 논문에서는 traffic 특성을 고려하지 않고 단일 traffic이라 가정하였는데 향후 연구방향은 traffic 특성을 고려하여 좀더 정교한 QoS를 만족시킬 수 있는 사항들을 연구하고자 한다.

참고문헌

- [1] D. Black, S. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, " An Architecture for Differentiated Services", RFC 2475
- [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] 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
- [4] Y. Bernet, A. Smith, S. Black, "A Conceptual Model for Diffserv Routers", Internet Draft, December 1999
- [5] J. Heinanen and R. Guerin, "A Single Rate Three Color Marker," RFC2697, March 1999.
- [6] J. Heinanen and R. Guerin, "A Two Rate Three Color Marker," RFC2698, March 1999.
- [7] Fang-Ching Ou, "A Generic Traffic Conditioner" Internet Draft, October 1999
- [8] Hyogon Kim, "A Fair Marker", INTERNET-DRAFT, October 1999
- [9] Longsong Lin, "A Generic Traffic Conditioner", INTERNET-DRAFT, April 1999
- [10] Wenjia Fang, "A Time Sliding Window Three Colour Marker(TSWTCM)", INTERNET-DRAFT, October, 1999
- [11] Jacobson, V., Nichols, K., "An Expedited Forwarding PHB", RFC 2598, August 1998
- [12] J. Heinanen et al., "Assured Forwarding PHG Group," RFC 2587, February 1999.
- [13] Constantinos Dovrolis, "Class-Based Service Differentiation", Internet Drafte, December 1998
- [14] Peter, F., "Integrated Services Over Differentiated Services", Internet Draft, March 1998
- [15] 한국전자통신연구원, "ATM상의 인터넷 서비스 기술개론", 진한도서, 1999
- [16] <http://www.sics.se/~ianm>
- [17] <http://www.cisco.com>