

Multiple DiffServ 환경에서 종단간 QoS를 지원하는 Marking Mechanism

김대회^o 박우진 정영환 김경혜 안순신
고려대학교 전자컴퓨터공학과

{ninanoo^o, progress, youngh, rabear, sunshin}@dsys.korea.ac.kr

Marking Mechanism for End-to-end QoS in Multiple DiffServ Environment

DaeHee Kim^o Woojin Park Younghwan Jung Kyunghye Kim Sunshin Ahn
Computer Network Lab. Dept. of Electronics and Computer Eng., Korea University

요 약

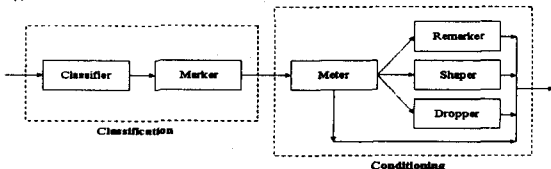
본 논문은 다수의 DiffServ(Differentiated Service) 도메인을 지나가는 Assured Forwarding(AF) 서비스 클래스의 종단간 QoS(Quality of Service)를 향상시키는 새로운 Marking Mechanism을 제안한다. 기본적인 개념은 패킷들이 최초 DiffServ 도메인으로 들어갈 때 부여받은 Marking 정보를 종단까지 유지하고, 이 정보를 근간으로 Marking을 수행하여 종단간 QoS를 보장하도록 해주는 것이다.

1. 서 론

최근 몇 년간 인터넷의 급성장과 함께 다양한 멀티미디어 및 실시간 애플리케이션의 등장으로 양적으로나 질적으로 네트워크 트래픽은 기하급수적으로 증가하였다. 하지만 기존의 Best Effort 방식의 서비스 모델로는 지연(delay)이나 패킷 손실에 민감한 멀티미디어 및 실시간 애플리케이션에 대한 서비스 보장을 해줄 수 없게 되었고, 그 결과 QoS의 필요성이 생겨났다. 이를 위해 IETF에 의해 제안된 것 중 하나가 DiffServ(Differentiated Service)[1]이다. DiffServ에서 서비스 차별(Service Differentiation)은 IP 헤더에 있는 TOS필드를 재정의한 DiffServ Code Point(DSCP)[2] 필드에 의해 제공되고, 같은 DSCP를 가진 패킷은 모두 해당되는 PHB에 의해 동일하게 서비스 받는다. 현재 두개의 기본적인 PHB, EF(Expedited Forwarding)[3]과 AF(Assured Forwarding)[4]가 정의되어 있다. EF PHB는 프리미엄 서비스(low delay, low latency, assured bandwidth)를 필요로 하는 애플리케이션에 적합하고, AF PHB는 각 패킷의 DSCP 필드에 적절한 우선순위를 Marking함으로써 최소한의 대역폭 보장을 제공한다. AF 클래스는 내부적으로 세 개의 드랍 우선순위를 가진다. 드랍 우선순위가 낮은 것부터 높은 순서로 Green, Yellow, Red로 Marking되고 이후에 혼잡이 발생할 경우 이 우선순위에 따라 패킷이 드랍된다. AF 클래스 패킷들이 여러 도메인을 지나가는 경우에는 각 도메인의 Ingress Edge 라우터에서 Marking이 수행된다. 도메인 간의 계약 등 여러 가지 이유로 종단 도메인에서 초기에 Green으로 Marking된 패킷이 Yellow나 Red로 Marking될 수 있고, 이 때문에 드랍이 발생하여 종단간 QoS를 보장해 주지 못하게 된다. 본 논문에서는 이러한 문제점을 해결하고자 다수의 DiffServ 도메인을 지나가는 AF 서비스 클래스의 QoS를 향상시키는 새로운 Marking Mechanism을 제안한다.

2. DiffServ에서의 기본적인 Marking Mechanism

그림 1은 DiffServ Ingress Edge 라우터에서 수행하는 기본적인 기능을 보여주고 있다. Marking은 기본적으로 Edge 라우터에서 수행된다. 사용자로부터 들어오는 트래픽을 IP 헤더의 여러 필드(source address, destination address, DS field, protocol ID, source port, destination port, incoming interface 등)를 기준으로 트래픽별로 적절한 클래스로 분류하고, 해당 DSCP로 Marking한다. Marking된 패킷들은 미터링되고, 미리 계약된 traffic profile을 만족하는지에 따라 적절하게 Remarking, Shaping되거나 드랍된다. 일반적으로 Marker라 하면 이 중에서 트래픽을 미터링하고 그에 따라 Marking하는 부분까지를 말한다.



<그림 1> DiffServ Boundary Function

Marker는 송신자의 traffic profile에 따라 트래픽을 Marking한다. traffic profile을 따르는 트래픽은 낮은 드랍 확률을 가지는 Green으로 Marking되고 보다 나은 서비스를 받게 된다. 반면에 traffic profile을 초과하는 트래픽은 높은 드랍 확률을 가지는 Yellow나 Red로 Marking되고 Green에 비해 낮은 서비스를 받게 된다.

traffic profile에 부합되는지 확인하기 위해 사용되는 메커니즘에 따라 Marker는 크게 Token bucket based marker와 Average rate estimator based marker로 분류할 수 있다.

Token bucket based marker는 트래픽을 측정하기 위해 하나 또는 여러개의 토큰 버킷을 사용한다. 패킷이 도착했을 때, 버킷에 충분한 토큰이 있을 경우, 패킷은 in-profile로 Marking되고 그렇지 않을 경우 out-of-profile로 Marking된다. 널리 알려진 srTCM(single rate Three Color Marker)[5]이나 trTCM(two rate Three Color Marker)[6]은 두 개의 토큰 버킷을 사용하여 AF 클래스 패킷들을 Green, Yellow, Red로 분류한다.

Average rate estimator based marker는 트래픽을 측정하기 위해 토큰 버킷 대신 average rate estimating algorithm을 사용한다. 가장 널리 알려진 rate estimating algorithm은 TSW(Time Sliding Window)[7] 알고리즘이다. TSW를 사용하여 트래픽을 세가지 색깔로 Marking하는 Marker가 tswTCM(time sliding window Three Color Marker)[8]이다. tswTCM에서는 패킷이 도착할 때마다 estimated arrival rate를 계산하고 이것을 PIR, CIR과 비교하여 확률에 따라 Green, Yellow, Red로 Marking한다. 그림 2는 tswTCM의 알고리즘을 보여준다.

본 논문에서는 tswTCM를 근간으로 하여 종단간 QoS를 보장하기 위한 새로운 Marking Mechanism을 제안한다.

avg_rate = Estimated Avg Traffic Sending Rate
CIR = Committed Information Rate
PIR = Peak Information Rate

```

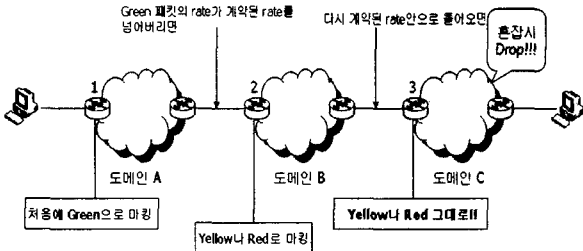
if (avg_rate <= CIR)
    the packet is marked as green;
else if (avg_rate <= PIR) AND (avg_rate > CIR)
    calculate P0 = (avg_rate - CIR) / avg_rate
    with probability P0 the packet is marked as yellow;
    with probability (1-P0) the packet is marked as green;
else
    calculate P1 = (avg_rate - PIR) / avg_rate
    calculate P2 = (PIR - CIR) / avg_rate
    with probability P1 the packet is marked as red;
    with probability P2 the packet is marked as yellow;
    with probability (1-P1-P2) the packet is marked as green;
    
```

<그림 2> tswTCM 알고리즘

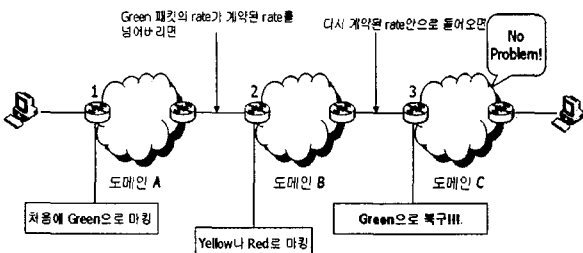
3. 다수의 DiffServ 도메인에서의 Marking Mechanism

일반적으로 전체 네트워크는 다수의 DiffServ 도메인으로 이루어져 있다. 이웃 도메인들은 서로 얼마만큼의 트래픽이 한 도메인에서 다른 도메인으로 통과될 수 있는지에 대한 SLA를 협상한다. 각 도메인의 Ingress Edge 라우터에 있는 Marker는 SLA에 명시된 traffic profile에 따라 적절하게 Marking을 수행한다. 이 때 여러가지 이유(SLA의 차이나 여러 도메인에서 온 트래픽이 합쳐지는 경우)로 traffic profile을 초과할 수 있는데, 기존의 Marking Mechanism에서는 원래의 Marking값에서 demotion(Green에서 Yellow, Yellow에서 Red)이 발생하여 사용자와 계약한 서비스를 제공해주지 못하는 경우가 생길 수 있다. 기존의 srTCM이나 trTCM은 demotion이 발생했을 경우, 그 다음 도메인에서 traffic profile을 만족시켜도 promotion(Yellow에서 Green, Red에서 Yellow)을 시켜주지 못한다. 따라서, 적절한 서비스를 받아야 할 패킷이 드랍되어 서비스의 보장이 되지 않을 수 있다. 본 논문에서는 처음에 Green으로 Marking되었던 패킷이 중간 도메인에서 Yellow나 Red로 demotion되었을 경우, 그 이후의 도메인에서 traffic profile을 만족시키면 보다 나은 서비스를 받을 수 있도록 promotion을 해주는 Marking Mechanism을 제안한다.

그림 3은 기존의 Marker들의 동작을 보여준다. 최초의 DiffServ 도메인 A로 들어오는 패킷이 Green으로 Marking되었다고 가정하자. 도메인 A와 B 사이에서 traffic profile을 초과하면, Yellow나 Red로 Marking된다. 다음 도메인 B와 C 사이에서 다시 traffic profile을 만족시키다 할지라도 기존의 Marker들은 다시 Green으로 promotion 시켜주지 못한다. 그래서 도메인 C안에서 Yellow나 Red의 값을 가지게 되므로 혼잡이 발생할 경우 드랍될 확률이 높다. 그림 4는 본 논문에서 제안하는 Marking Mechanism이 이러한 문제점을 해결하는 한 예를 보여주고 있다. 도메인 A에서 Green으로 Marking된 패킷이 도메인 B에서 traffic profile을 만족시키지 못하면 Yellow나 Red로 demotion되고 도메인 C에서 다시 traffic profile을 만족시킬 경우 Green으로 promotion된다.



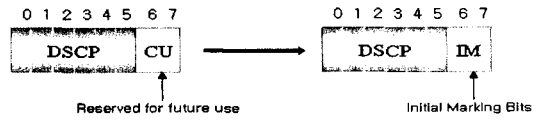
<그림 3> 기존 Marker의 동작



<그림 4> 새로운 Marking Mechanism의 동작

4. 새로운 Marking Mechanism

우리가 제안하는 Marking Mechanism의 기본적인 개념은 초기의 Marking 정보를 패킷 헤더에 유지해 가면서 이 정보를 근간으로 하여 효율적인 End-to-End QoS를 보장해 주는 것이다. 이러한 정보를 유지하기 위한 방법으로, DSCP 필드에서 현재 사용되지 않는 2개의 CU(Currently Unused) 비트를 IM(Initial Marking) 비트로 정의하여 사용한다. 그림 5는 원래 DSCP 필드와 우리가 사용할 IM 비트를 정의한 필드를 보여주고 있다.



<그림 5> DSCP 필드의 재정의

<Initial Marking 비트>

No initialization	00
Green	01
Yellow	10
Red	11

<표 1> Initial Marking 비트의 정의

본 논문에서 제안하는 Marking Mechanism은 임의의 DiffServ 도메인으로 패킷이 들어올 때, IM 비트가 "00"이면 아직 Marking이 되지 않은 것을 의미하며 tswTCM에 따라 Marking을 수행한다. 그리고 그 Marking 정보를 위의 표 1에 따라 IM 비트를 세팅한다. 이 정보는 여러개의 도메인을 거치는 동안 변하지 않는다. 그러면 중간에 다른 도메인에서는 IM 비트를 확인해 보고 "00"이 아니면 이미 Marking된 패킷이라는 것을 알고 이 정보를 참조하여, 큐의 길이와 avg_rate(트래픽의 평균 송신 rate)를 가지고 미터링하여 조건이 충족되면 원래의 Marking값이나 여유가 된다면 더 좋은 값으로 promotion 시켜 준다. IM 비트의 Marking은 제일 처음의 경계 라우터에서 한 번만 수행된다. 구체적인 알고리즘에 대한 pseudo code는 그림 6에서 보여진다.

새로운 Marking Mechanism에서는 기본적으로 각각의 Green, Yellow, Red로 Marking된 패킷별로 큐를 가지고 있다고 가정하고, 각 큐의 길이와 avg_rate, IM 비트를 가지고 Marking을 수행한다. 큐의 길이는 라우터마다 다르므로 절대적인 길이가 아니라 전체 큐의 크기에 대한 비율로 정의한다.

그림 6에서 Line 1~3은 IM 비트가 "00"일 때, 즉 아직 Marking이 수행되지 않았을 때의 알고리즘을 보여준다. 이때는 tswTCM(그림 2 참조)과 똑같이 Marking을 수행하고, Marking 정보에 따라 IM 비트를 표1과 같이 세팅한다.

Line 4~19는 IM 비트가 "01"일 때, 즉 초기 Marking 정보가 Green일 때를 보여준다. 초기 Marking 정보가 Green이므로 큐의 길이와 avg_rate를 참조하여 최대한 Green으로 Marking하려고 노력한다. avg_rate가 CIR보다 작거나 같으면 Green으로 Marking하고, avg_rate가 CIR보다 크고 PIR보다 작거나 같을 때에도 초기 Marking 정보가 Green이므로 Green 큐의 길이가 0.5보다 작거나 같을 때에는 Green으로 Marking한다. 0.5보다 클 때에는 기존의 tswTCM과 같다. avg_rate가 PIR보다 클 때에도 초기 Marking 정보가 Green이므로 Green 큐에 여유가 있을 때(여기서는 0.3보다 작거나 같을 때에는 Green으로 Marking한다. 0.3보다 클 때에는 역시 기존의 tswTCM과 같다).

Line 20~38은 IM 비트가 "10", 즉 초기 Marking 정보가 Yellow일 때를 보여준다. 초기 Marking 정보가 Yellow이므로 큐의 길이와 avg_rate를 참조하여 최대한 Yellow를 보장하여 주고, 여유가 되면 Green으로 Marking하기도 한다. avg_rate가 CIR보다 작거나 같을 때, 원래부터 Green이 아니라 Yellow였으므로 무조건 Green으로 Marking하는 것이 아니라 Green 큐의 길이가 0.5보다 작거나 같을 때에만 Green으로 promotion 해주고 그렇지 않을 때에는 Yellow로 Marking한다. avg_rate가 CIR보다 크고 PIR보다 작거나 같을 때, Green 큐의 길이가 0.3보다 작거나 같으면 Green으로 Marking하고, 0.3보다 크면 기존의 tswTCM과 똑같이 Marking한다. avg_rate가 PIR보다 클 때에는 Yellow 큐의 길이가 0.5보다 작거나 같으면 원래 Marking 정보인 Yellow로 Marking하고 그 외에는 기존의 tswTCM과 똑같이 Marking한다.

Line 39~55는 IM 비트가 "11", 즉 초기 Marking 정보가 Red일 때를 보여준다. 초기 Marking 정보가 Red이긴 하지만, Green 큐나 Yellow 큐에 여유가 있을 경우, Green이나 Yellow로 Marking한다. avg_rate가 CIR보다 작거나 같을 때, Green 큐의 길이가 0.3보다 작거나

나 같으면 Green으로 Marking하고, Green 큐가 0.3보다 크고 Yellow 큐가 0.5보다 작으면 Yellow로 Marking한다. 그렇지 않으면 Red로 Marking한다. avg_rate가 CIR보다 크고 PIR보다 작거나 같을 때에는 Green 큐의 길이가 0.1보다 작으면 Green으로 Marking하고, Green 큐가 0.1보다 크고 Yellow 큐가 0.3보다 작으면 Yellow로 Marking한다. 그 외에는 모두 Red로 Marking한다.

이 알고리즘을 tswTCM과 비교해보면, tswTCM은 이전의 Marking 정보를 고려하지 않고 오직 현재의 avg_rate만을 가지고 Marking을 수행하는 color-blind 모드로 동작하고, 제안한 Marking Mechanism은 최초의 Marking 정보를 기억하고 그것을 고려하여 Marking을 수행하는 color-aware 모드로 동작한다는 차이점을 가지고 있다. color-blind 모드의 장점은 이전 Marking 값과는 상관없이 현재의 rate 만 가지고 Marking을 수행하므로 현재 자원이 넉넉하면 Yellow나 Red 로 Marking된 패킷도 Green으로 Marking되어 더 나은 서비스를 받을 수 있다는 것이다. 하지만 이렇게 될 경우, 이 패킷들 때문에 다음에 들어오는 패킷들의 avg_rate가 커지고 그렇게 되면 다음에 들어오는 Green 패킷들이 CIR을 넘어 Green으로 Marking되지 못할 수가 있다. 제안한 Marking Mechanism은 이러한 문제를 큐의 길이를 이용하여 해결한다. Yellow나 Red인 패킷들이 avg-rate가 CIR보다 작다고 하여도, 무조건 Green으로 promotion 시켜주는 것이 아니라 큐의 길이가 일정 값보다 작을 경우에만 Green으로 올려준다.

color-aware 모드로 동작하는 srTCM이나 trTCM과 비교해보면 srTCM이나 trTCM은 바로 이전 도메인의 Marking값만을 참조하여 여유의 자원이 있을 때에도 이전의 Marking값 이상으로 promotion시켜 주지 않는다. 하지만 제안한 Marking Mechanism은 바로 이전의 도메인의 Marking 정보가 아니라 사용자와 SLA를 맺은 도메인과의 Marking 정보를 IM 비트에 기록함으로써 정확히 사용자의 요구 사항을 알 수 있고, 또한 그 값을 참조하여 여유의 자원이 있을 때에는, 원래의 Marking값으로 promotion 시켜 주어 사용자가 요구하는 서비스를 받을 수 있게 해준다. 이렇게 초기의 패킷 우선순위(Marking정보)를 복원시켜 줌으로써 다른 DiffServ 도메인에서 높은 우선순위를 가지는 패킷들을 보호해 줄 수 있고, 따라서 효율적인 종단간 QoS를 보장해 줄 수 있다.

요약하면, 제안한 Marking 메커니즘은 두 가지 방식으로 동작한다. 첫 번째, IM 비트가 "00"인 경우, Marking이 되어 있지 않은 것이므로 기존의 tswTCM에 따라 DSCP에 Marking을 하고, 그 Marking값에 따라 IM 비트를 세팅한다. 두 번째, IM 비트가 "00"이 아닌 경우, IM 비트를 참조하여 주어진 알고리즘에 따라 DSCP에 Marking을 수행한다.

5. 결론 및 향후 연구

본 논문에서는 tswTCM을 확장하여 여러 개의 DiffServ 도메인을 지나가는 AF 서비스 클래스의 종단간 QoS를 효율적으로 보장해주는 Marking Mechanism을 제안하였다. 제안한 Marking Mechanism은 다른 Marker들과 호환 가능하며, 매우 scalable하다. 향후 우리는 IXP 2400 Network Processor[9][10] 상에서 DiffServ 모듈 구현을 통해 이 메커니즘의 성능을 측정하고 향상시켜 나갈 것이다. 그리고 최고의 성능을 보이는 큐의 길이(여기서는 0.5, 0.3, 0.1을 사용)를 찾아내야 할 것이다. 그리고 여기서는 IM 비트의 값만을 참고로 Marking을 수행하는 Mechanism을 제안하였는데, 보다 나은 성능 향상을 위해서는 IM 비트 뿐만 아니라 바로 이전 도메인의 DSCP를 참고하는 부분을 추가 시켜야 할 것이다.

6. 참고 문헌

- [1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss.: An architecture for differentiated services. RFC 2475 (1998)
- [2] K. Nichol, S. Blake, F. Baker, D. Black.: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474 (1998)
- [3] V. Jacobson, K. Nichols, and K. Poduri. : An expedited forwarding PHB. RFC 2598 (1999)
- [4] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski.: Assured forwarding PHB group. RFC 2597 (1999)
- [5] Heinanen, J. and R. Guerin.: A Single Rate Three Color Marker. RFC 2697 (1999)
- [6] Heinanen, J. and R. Guerin.: A Two Rate Three Color Marker. RFC 2698 (1999)
- [7] D.D. Clark, W. Fang.: Explicit Allocation of Best Effort Packet Delivery Service". IEEE/ACM Transactions on Networking, August 1998, Vol 6, No. 4, pp. 362-373
- [8] W. Fang, N. Seddigh, and B. Nandy.: A Time Sliding Window Three Color Marker (tswTCM). RFC 2859 (2000)
- [9] Intel IXP2400 Network Processor Programmer's Reference Manual (2003)
- [10] Intel IXP2400 Network Processor Hardware Reference Manual (2003)

```

avg_rate = Estimated Avg Traffic Sending Rate
qlen_Green = Estimated Queue Length of Green packet
qlen_Yellow = Estimated Queue Length of Yellow packet
qlen_Red = Estimated Queue Length of Red packet
CIR = Committed Information Rate
PIR = Peak Information Rate
IM = Initial Marking bits
P0 = (avg_rate - CIR) / avg_rate
P1 = (avg_rate - PIR) / avg_rate
P2 = (PIR - CIR) / avg_rate

01 if (IM == "00") // 아직 Marking이 되지 않은 패킷
02 Equal to tswTCM
03 Marks IM bits (Green, Yellow or Red);

04 if (IM == "01") // 초기에 Green으로 Marking된 패킷
05 if (avg_rate <= CIR)
06   Green;
07 else if (avg_rate > CIR) AND (avg_rate <= PIR)
08   if (qlen_Green <= 0.5)
09     Green;
10   else
11     with P0, Yellow;
12     with (1-P0), Green;
13   else
14     if (qlen_Green <= 0.3)
15       Green;
16     else
17       with P1, Red;
18       with P2, Yellow;
19       with (1-P1-P2), Green;

20 if (IM == "10") // 초기에 Yellow로 Marking된 패킷
21 if (avg_rate <= CIR)
22   if (qlen_Green <= 0.5)
23     Green;
24   else
25     Yellow;
26 else if (avg_rate > CIR) AND (avg_rate <= PIR)
27   if (qlen_Green < 0.3)
28     Green;
29   else
30     with P0, Yellow;
31     with (1-P0), Green;
32   else
33     if (qlen_Yellow <= 0.5)
34       Yellow;
35     else
36       with P1, Red;
37       with P2, Yellow;
38       with (1-P1-P2), Green;

39 if (IM == "11") // 초기에 Red로 Marking된 패킷
40 if (avg_rate <= CIR)
41   if (qlen_Green <= 0.3)
42     Green;
43   else if (qlen_Yellow <= 0.5)
44     Yellow;
45   else
46     Red;
47 else if (avg_rate > CIR) AND (avg_rate <= PIR)
48   if (qlen_Green <= 0.1)
49     Green;
50   else if (qlen_Yellow <= 0.3)
51     Yellow;
52   else
53     Red;
54   else
55     Red;
    
```

<그림 6> 제안한 Marking Mechanism의 pseudo code