

논문 2004-41TC-6-6

QoS를 보장하기 위한 DiffServ-aware-MPLS 버퍼 관리 기법

(DiffServ-aware-MPLS Buffer Management Scheme for QoS Guarantee)

한철민*, 김변곤**, 정광일***, 김남희****, 박현*, 정경택****, 전병실**

(Cheol-min Han, Byeon-gon Kim, Kwang-il Jung, Nam-hee Kim, Hyeon Park, Kyeong-teak Chung, and Byeong-sil Chon)

요약

IP 네트워크에서, 네트워크 리소스를 효과적으로 이용하기 위해 다양한 형태의 트래픽들이 통계적으로 다중처리 되고 있다. DiffServ-aware-MPLS는 서로 다른 QoS(Quality of Service) 요구를 가지는 통신서비스의 대부분을 지원한다. QoS 구조 기반인 DiffServ-aware-MPLS는 IP 네트워크에서 QoS를 보장하는 다중서비스를 가능하게 한다. 그러나 QoS를 보장하기 위한 IP 네트워크 관리 방법은 여전히 큰 이슈가 되고 있다. 이 논문에서는, 정책과 큐잉 그리고 스케줄링을 이용한 DiffServ-aware-MPLS 버퍼 관리 기법을 제안한다.

Abstract

In an IP network, various types of traffics are statistically multiplexed to utilize efficiently the network resources. The DiffServ-aware-MPLS supports a wide variety of communication services with different QoS requirements. The DiffServ-aware-MPLS based on QoS architecture had become one of the most promising ways to guarantee QoS Multi-service IP network. But how to manage IP network with QoS guarantee is still an open issue. In this paper, we propose DiffServ-aware-MPLS buffer management technique using the specified policing, queuing, and scheduling.

Keywords : QoS, DiffServ, MPLS, DiffServ-aware-MPLS

I. 서론

최근 인터넷 서비스가 급속히 발전하면서 더욱더 빠르고 보다 안정적인 서비스 요구가 증가하고 있다. 이러한 서비스를 만족시키기 위해 개발된 기술이 giga

-bit/tera-bit 라우터, ATM, IntServ, MPLS 등이다. 이와 같이 속도 향상에 의존적인 서비스 기술들은 현재의 인터넷 서비스가 고속의 데이터뿐만 아니라 고속의 음성 및 화상 같은 다양한 트래픽 소스(Traffic Source) 들인 실시간 서비스 등을 만족시키지 못하고 있다. 즉, 서비스 종류에 따른 패킷 전달기능을 차별화하지 않고 동등하게 Best Effort 처리를 하는 것이다.

이와 같은 문제점을 해결하기 위해서 개발된 기술이 DiffServ(Differentiated Service)이다. 하지만 서비스 종류에 따른 차별화 서비스(DiffServ)에 대한 장점에도 불구하고 라우터의 데이터 전송 경로가 손실되었을 때 종단 대 종단(End-to-End) QoS(Quality of Service)를 만족시키지 못하는 단점이 있다. 다양한 인터넷 서비스 종류에 따라 차별화된 서비스와 고속의 패킷전송 뿐만

* 학생회원, ** 정회원, 전북대학교 전자정보공학부 (Chonbuk National Univ., Dept. of Electronic and Information Engineering)

*** 정회원, 한국원자력연구소 (Korea Atomic Energy Research Institute)

**** 정회원, 군산대학교 컴퓨터 정보과학과 (Kunsan National Univ., Dept. of Computer Information Science)

***** 정회원, 군산대학교 전자정보공학부 (Kunsan National Univ., Dept. of Electronic and Information Engineering)

접수일자: 2004년2월28일, 수정완료일: 2004년5월25일

아니라 QoS를 만족시키기 위해 IETF(Internet Engineering Task Force)에서 제안한 기술이 바로 DiffServ-aware-MPLS(Multi-Protocol Label Switching) 기술이다.^[1-5]

대부분 현재의 DiffServ-aware-MPLS 서비스 기술은 DiffServ의 차별화된 서비스에 대해서 MPLS의 빠른 Label 스위칭으로 터널링 서비스만을 하고 있다.^[6]

본 논문에서는 DiffServ-aware-MPLS 서비스를 하기 위해서 DiffServ와 MPLS를 PHB(Per-Hop Behavior)에 따라 트래픽을 효율적으로 관리하기 위해 토큰 버킷(Token Bucket), TSW3CM(Time Sliding Window with 3 Color Marking) 정책과 dsRED(Diff-Serv Random Early Detection) 큐, WRR(Weight Round Robin) 스케줄링을 적용하여 같은 시뮬레이션 모델 환경에서 DiffServ-aware-MPLS 네트워크와 일반 TCP 네트워크, DiffServ 네트워크의 QoS를 비교 분석하였다.

II. DiffServ-aware-MPLS

초고속 통신망을 위한 DiffServ-aware-MPLS 네트워크는 인터넷 기술 관련 국제 표준화 단체인 IETF의 표준화 문서 RFC 2475, 3031, 3270을 중심으로 표준화가 추진 중이다. 그리고 트래픽 제어 및 관리기술에 따른 차별화 서비스와 MPLS 기반의 트래픽 엔지니어링 기법으로 완전한 QoS를 보장한다.^[2-9]

DiffServ-aware-MPLS 네트워크는 그림 1에 보인 것과 같다. 그림에서처럼 차별화된 서비스를 MPLS 네트워크에서 서비스하기 위해서는, MPLS의 입구노드와 출구노드가 DiffServ의 패킷을 서비스 할 수 있도록 변경되어야 한다. 즉, DiffServ-aware-MPLS 네트워크에서는 DiffServ와 MPLS를 효과적으로 매핑 시키고, MPLS 망 내에서는 다양한 차별화 서비스를 제공하는 것이 가장 중요한 문제점이다.

이러한 망 구조를 구현하기 위해서는 아래와 같은 문제들을 해결해야만 한다.

- ① DiffServ에 정의되어 있는 PHB들을 MPLS LSP(Label Switching Path)에 매핑 시키는 방법
- ② DiffServ 패킷 전달을 위한 MPLS 라우터의 패킷 포워딩 엔진의 구조 변경
- ③ MPLS망에서 DiffServ를 지원하는 LSP를 포함하기 위한 LSP 설정 방법
- ④ PHP(Per-Hop Behavior)의 LSP 매핑 문제점

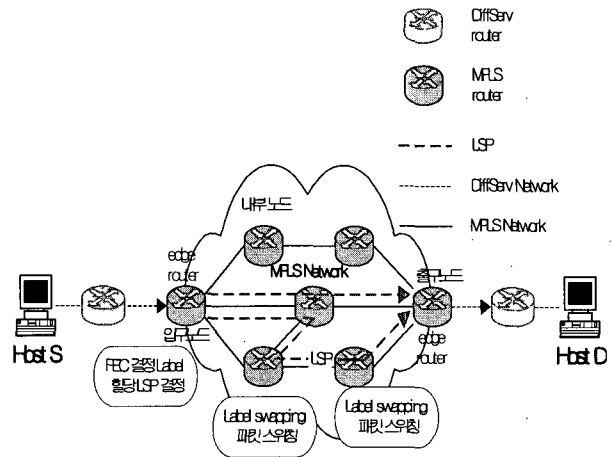


그림 1. DiffServ-aware-MPLS 네트워크
Fig. 1. DiffServ-aware-MPLS network.

DiffServ의 BA(Behavior Aggregate)들을 MPLS LSP에 매핑 시키는 방법에는 다음의 두 가지 방법이 있다.

- ① 첫번째 : BA(1) = LSP(1), 즉 1:1 매핑 방법
- ② 두번째 : 여러 BA = LSP(1), 즉 다 : 1 매핑 방법

위에서 제시한 방법들을 구현하기 위해서는 E(Exp-Inferred-PSC)-LSP와 L(Level-only-Inferred-PSC)-LSP에 따라서 매핑 방법이 차별화 된다. DiffServ와 MPLS를 매핑시키기 위해서는 추가적인 기능으로 OA(Ordered Aggregate)와 PSC(PHB Scheduling Class)가 필요하다. DiffServ 영역의 입구노드(ingress node)는 도착된 패킷의 IP 헤더 내의 TOS(Type of Service)를 검사하여 그 영역내의 모든 노드에서 PHB에 따라서 클래스를 분류한다. 이때 동일한 패킷 전달 서비스를 받게 될 패킷들의 집합을 BA(Behavior Aggregate)라고 하고, 한 BA에 속한 모든 패킷은 동일한 DSCP(DiffServ Codepoint)를 부여받게(Marking)된다. 그리고 네트워크의 내부노드(interior node)들은 패킷을 포워딩할 때 DSCP 필드만을 참조하여 해당 패킷에 대한 적절한 스케줄링(Scheduling)과 패킷 폐기(dropping)를 수행한다. 그렇기 때문에 플로우마다 상태나 정보를 유지할 필요가 없으며, 복잡한 트래픽 조절(Traffic conditioning) 기능을 망의 경계노드(Boundary node)에서 모두 수행하므로 확장성도 좋다.^[10-12]

하지만 망의 효율성과 차별화된 클래스에 따른 서비스를 제공하기 위해서는 경계노드에서 발생하는 트래픽의 효율적인 관리가 필요하다. 이러한 문제점을 극복하기 위해서는 클래스별로 적합한 정책 수행과 큐 관

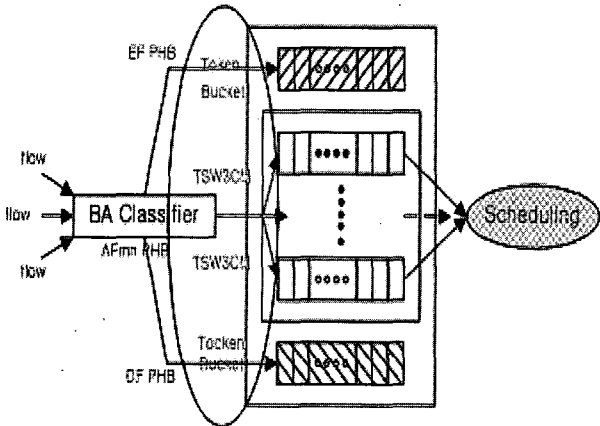


그림 2. 폴리싱과 스케줄링
Fig. 2. Policing and scheduling.

표 1. 클래스에 따른 폴리싱과 스케줄링
Table 1. Policing and scheduling of each class.

Class	Policing	Parameter	RED VQ (CP)	VQ parameter	E-drop Probability
EF	Token	CIR 2400000	VQ1 (101110)	L 20 H 60	0.1
		CBS 3000	VQ2 (111110)	L 20 H 60	
AF1	TSW3	CIR 1500000	VQ1 (001101)	L 20 H 50	0.2
			VQ2 (010100)	L 20 H 50	
		PIR 3500000	VQ3 (011100)	L 20 H 50	0.8
			VQ3 (011100)	L 20 H 50	
AF2	TSW3	CIR 1500000	VQ1 (100000)	L 20 H 40	0.4
			VQ2 (101000)	L 20 H 40	
		PIR 2800000	VQ3 (110000)	L 20 H 40	1.6
			VQ3 (110000)	L 20 H 40	
DF	Token	CIR 1500000	VQ1 (001000)	L 20 H 30	3
		PIR 5000	VQ2 (000000)	L 20 H 30	

리, 그리고 스케줄링을 적용시키는 방법이 하나의 대안으로 제시 될 수 있다.

III. 제안된 버퍼관리 기법

본 논문에서는 서비스 클래스의 우선순위에 따라 파라미터 특성 값을 달리 줌으로써, QoS에 따라 차별화된 서비스를 효율적으로 제공하기 위해, IETF에서 제안한 RFC 3270 표준화 문서를 만족시키는 PHB와 PSC에 대

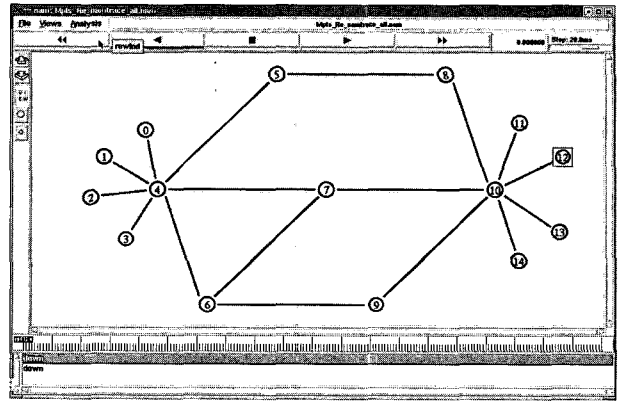


그림 3. 시뮬레이션 컴포넌트 모델
Fig. 3. Simulation component model.

표 2. 시뮬레이션 구성 요소
Table 2. Configuration component for simulation.

시뮬레이션 구성 요소	종류
Simulation Model	TCP, DiffServ DiffServ-aware-MPLS
Class Type	EFn, AF1n, AF2n, DFn (n=1,2,3)
Traffic Source Number	Three of Each Class Type
Policing	Token-buket, TSW3CM
Queueing	dsRED queue, Drop-tail
Scheduling	WRR
MPLS Routing	CR-LDP(E-LSP, L-LSP)
Node number	14
Link Bandwidth	10 Mbps
Simulation Time	20 sec
Drop Down Time	10 ~ 15 sec
Total Source Rate	9.6 Mbps

해서, 그림 2와 같이 EFn과 DFn 클래스에 토큰 버킷 정책을, AF1n과 AF2n에는 TSW3CM 정책을 적용시켰다. 그리고 dsRED와 Drop-tail 큐를 적용시키고, 스케줄링 기법으로는 WRR을 사용함으로써, DiffServ와 MPLS를 매핑 시킬 때 문제되는 버퍼의 오버플로우로 인한 패킷 손실을 최소화하도록 하였다.

표 1에서는 클래스의 우선순위에 각각 적용된 정책 파라미터 값들에 차등을 두었고, Codepoint에 따른 가상 큐 파라미터 역시 클래스 우선순위에 따라 EF에는 High일 때 60, Low일 때 20, 그리고 AF1과 AF2의 가상 큐의 값은 각각 (50, 20), (40, 20)을 주었다. 마지막으로 DF의 경우는 (30, 20)을 주었다.

그리고 적용된 WRR 스케줄링 파라미터들도 클래스 우선순위가 높을수록 E-drop 확률을 작게 줌으로써, DiffServ-aware-MPLS 네트워크에서 QoS에 따라 보다 차별적인 서비스뿐만 아니라, 트래픽을 효율적으로 관리 할 수 있게 하였다.

IV. 시뮬레이션 모델 및 파라미터

그림 2와 표 1을 DiffServ-aware-MPLS 네트워크에서 시뮬레이션 하기 위해 그림 4.1처럼 입구노드, 4번 노드, 와 출구 노드, 10번 노드, 에 각각 4개의 소스 노드들로 구성하였고, 각 소스 노드는 서로 다른 클래스 (EFn, AF1n, AF2n, DFn)를 갖게 하였으며 각각의 소스 노드는, 동일하지만 우선순위가 다른, 3개의 트래픽 소스(n=1,2,3)를 갖도록 모델링 하였다.

표 2는 그림 3의 시뮬레이션 모델의 구성 요소를 표로 작성한 것이다.

인터넷 서비스 망에서 QoS의 중요한 성능평가 기준에는 종단 대 종단의 전송 지연(Delay)과 지연 변화(Jitter)의 평균, 그리고 비트/패킷/프레임 손실률이 있다.

- 전송지연 = 정보의 양/전송용량
- 최소 대역폭 = WRR 스케줄링 알고리즘에 의해 보장됨

① 서비스 번호 : 파라미터와 관련된 서비스를 표시한다.

- ◇ 1 : 기본 값
- ◇ 2 - 254 : 개별적인 서비스에 할당

② 파라미터 번호 : 파라미터 자체를 표시

- ◇ 1 - 127 : 모든 서비스에 적용하는 파라미터
- ◇ 128 - 254 : 특별한 서비스 위한 파라미터

- 지연 Jitter = 패킷이 갖는 가장 큰 지연과 가장 작은 지연의 차
- 패킷 손실률 = 손실 패킷/전체 전송 패킷 (Congestion에 의해 자주 발생)
- 파라미터 ID = <서비스 번호, 파라미터 번호>
- TSpec and RSpec : 보장된 서비스는 토큰버킷 TSpec 파라미터 사용

① TSpec(Traffic Descriptor)

- ◇ r : Bucket rate(bytes/sec) : 토큰이 토큰 버킷에 도달하는 율
- ◇ p : Peak rate(bytes/sec) : 패킷이 전송할 수 있는 최대 율
- ◇ b : Bucket depth(bytes) : 토큰 버킷의 사이즈
- ◇ m : Minimum policed unit(bytes) : m보다 작은 사이즈의 패킷은 m바이트로 카운트
- ◇ M : Maximum packet size(bytes) : 수용될 수 있는 최대 패킷 사이즈

② RSpec(Service Specification)

- ◇ R : Service Rate(bytes/sec) : 대역폭 또는 서비스 율
- ◇ S : Slack Term(microseconds) : 한 노드가 종단 대 종단 지연을 만족하는 범위 안에서 보장 될 수 있는 지연

- 정책(Policing) : 동의된 TSpec 파라미터와 트래픽 율 네트워크의 경계에서 비교함
- Shaping : TSpec이 인접 상향 링크를 위해 예약된 TSpec보다 작을 때 필요하고, 패킷이 토큰 버킷 파라미터에 맞을 때까지 버퍼링 되면서 수행함
- 지연 계산

① ETS worst Queuing Delay (Do not consider error term)

$$= \frac{b(p-R)}{R(p-r)} \quad (p \geq R \geq r)$$

② ETS worst Queuing Delay (Consider error term)

$$= \frac{(b-M)(p-R)}{R(p-R)} + \frac{M+Ctot}{R} + Dtot \quad (p \geq R \geq r)$$

$$= \frac{M+Ctot}{R} + Dtot \quad (R \geq p \geq r)$$

V. 시뮬레이션 결과 및 분석

가. 시뮬레이션 결과

서론에서 언급한 차별화 서비스와 DiffServ-aware-MPLS 서비스를 2장에서 제안된 정책과 스케줄링을 3장에서 제시한 시뮬레이션 모델에 적용시켜 ns-2 소프트웨어 툴을 이용한 컴퓨터 시뮬레이션의 결과를 보이도록 한다. 그림 4는 Drop-Down 없는 네트워크 모델

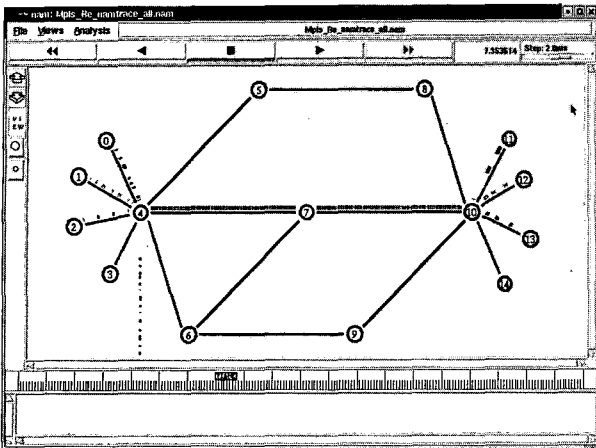


그림 4. None drop-down 네트워크 시뮬레이션 모델
Fig 4. None drop-down network simulation model.

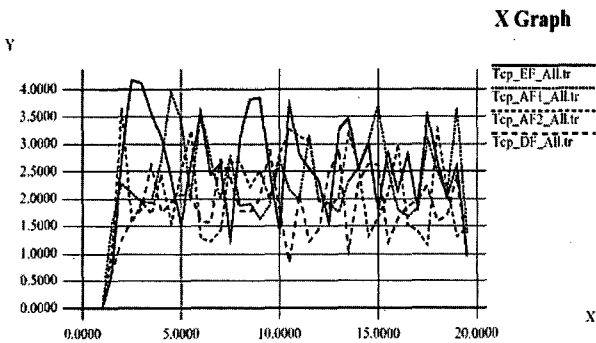


그림 5. TCP 네트워크의 클래스별 수신율
Fig 5. Arrival rate of each class on TCP network.

을 보여주고 있고, 그림 5는 TCP 네트워크에서 폴리싱과 스케줄링을 적용하지 않고 서비스 클래스에 따른 표 1의 E-drop 확률만을 적용시켜 각 클래스에 대한 패킷 수신율을 나타낸 것이다. 그림 6과 그림 7은 폴리싱과 스케줄링을 적용시킨 각 클래스에 대한 패킷 수신율을 나타낸 그래프이다.

그림 8은 DiffServ의 단점인 라우팅 경로가 Drop 되었을 경우에 경로를 재설정하지 못하는 문제점을 DiffServ-aware-MPLS와 비교하기 위한 것이고, DiffServ-aware-MPLS 네트워크에서 노드 7번과 노드 10번 사이의 경로가 Drop Down 되었을 때 라우팅 경로를 재설정하는 과정을 보여주고 있다.

그림 9, 10, 11은 그림 8의 Drop-Down (Rerouting) 시뮬레이션 모델링 조건에 따른 결과 그래프이고, 그림 9와 10의 그래프를 보면 Drop-Down된 10~15초 동안 패킷 수신율이 0%에 가까움을 알 수 있지만, DiffServ-aware-MPLS 네트워크의 시뮬레이션 결과인 그림 11은 경로를 재설정 후 바로 패킷을 수신하기 때문에 그림 7과 비교하였을 때 차이가 없음을 알 수가 있다.

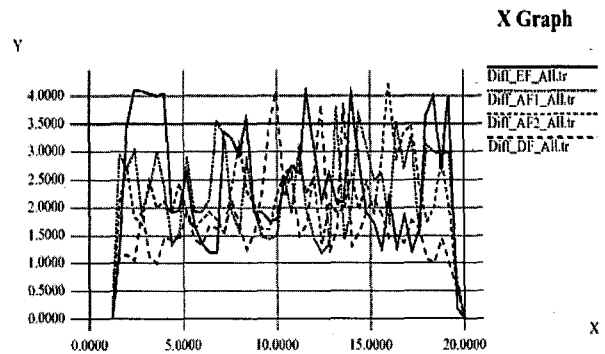


그림 6. None Drop-down DiffServ 네트워크의 클래스 수신율

Fig 6. Arrival rate of each class on none drop-down DiffServ network.

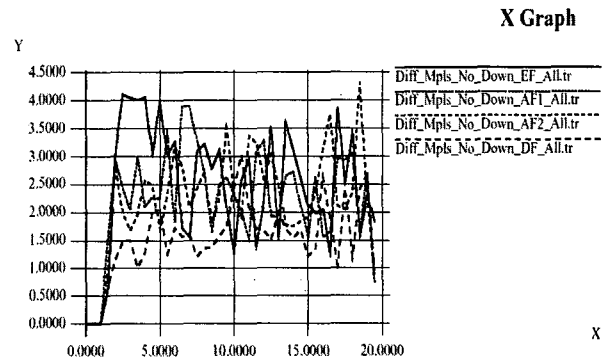


그림 7. None drop-down DiffServ-aware-MPLS 네트워크의 클래스별 수신율

Fig 7. Arrival rate of each class on none drop-down DiffServ-aware-MPLS Network

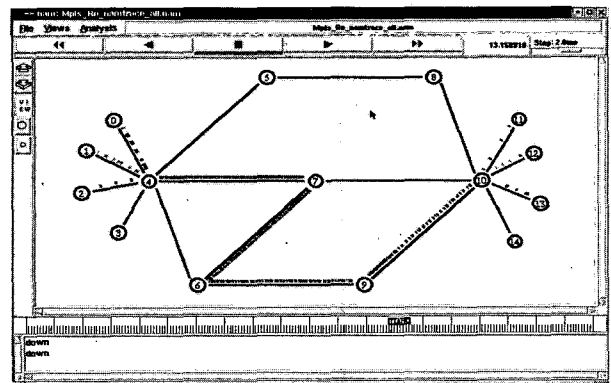


그림 8. Drop-down 네트워크 시뮬레이션 모델
Fig 8. Drop-down(Rerouting) Network Simulation Model.

나. 시뮬레이션 결과

본 논문에서 제안한 기법에 대하여 DiffServ-aware-MPLS 네트워크에서 QoS를 만족하는지 평균 패킷 지연(Delay)과 평균 패킷 지터(Jitter), 평균 패킷 Drop 율을 분석하고, DiffServ-aware-MPLS의 결과 값이 경로를 재설정 후의 결과 값과 차이가 없음을 측정된 결과를 바탕으로 증명하도록 한다.

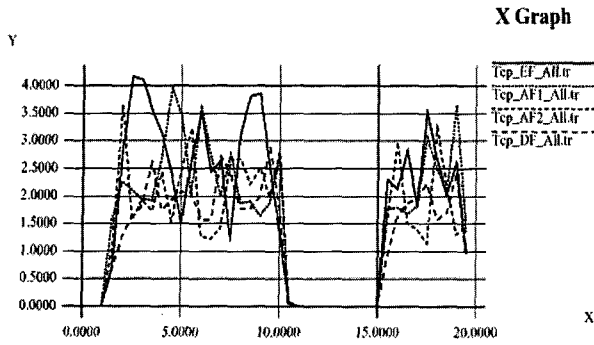


그림 9. Drop-down(Rerouting) TCP 네트워크의 클래스별 수신율

Fig 9. Arrival rate of each class on Drop-down TCP, Network

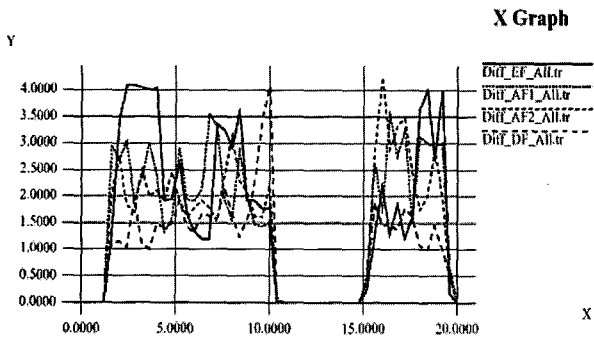


그림 10. Drop-down(Rerouting) DiffServ 네트워크의 클래스별 수신율

Fig 10. Arrival rate of each class on drop-down DiffServ Network

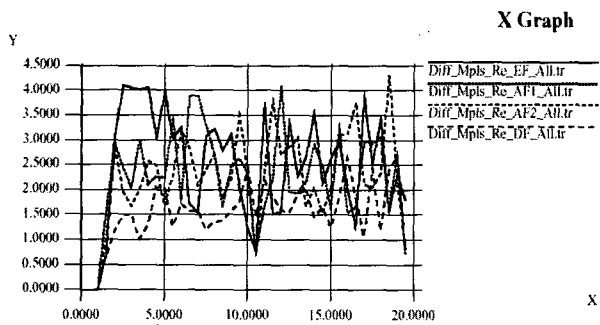


그림 11. Drop-down(Rerouting) DiffServ-aware-MPLS 네트워크의 클래스별 수신율

Fig 11. Arrival rate of each class on drop-down DiffServ-aware-MPLS network.

(1) Delay 성능평가

표 3은 각 서비스에 대해서 클래스 별 평균 지연 시간을 나타내고 있고, 그림 12를 그래프로 나타낸 것이다. 그림 12를 보면 TCP 네트워크에서의 평균 지연 시간은 클래스에 따라 차등이 없었고, DiffServ의 지연 시간은 클래스에 따라 약간의 차등적인 값을 나타냈다. 그러나 None Drop-Down 네트워크일 경우의 DiffServ-aware-MPLS의 결과 값은 DiffServ 네트워크의 결

표 3. 서비스 클래스에 따른 네트워크별 평균 지연 Table 3. Mean delay of each class on each network

	TCP	DiffServ	DiffServ-aware-MPLS	DiffServ-aware-MPLS Rerouting
EF	0.23	0.22	0.22	0.28
AF1	0.23	0.23	0.23	0.29
AF2	0.23	0.24	0.25	0.30
DF	0.23	0.26	0.27	0.33

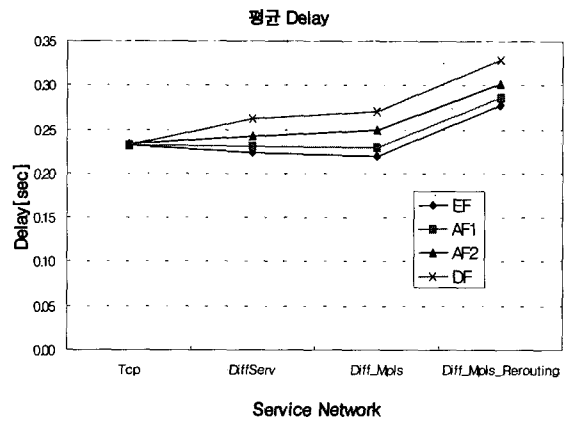


그림 12. 서비스 클래스에 따른 네트워크별 평균 지연 Fig. 12. Mean delay of each class on each network

과 값보다 좀더 차별화된 서비스가 이루어지고 있음을 알 수 있었고, Drop-Down(Rerouting) DiffServ-aware-MPLS 네트워크일 경우, 경로를 재설정하는 시간만큼의 지연이 있을 뿐, 차별화 서비스는 None Drop-Down DiffServ-aware-MPLS의 경우와 비교할 때 차이가 거의 없음을 알 수 있다.

(2) Jitter 성능평가

표 4와 그림 13은 평균 지터의 결과 값을 보이고 있다. 평균 지연과 마찬가지로 DiffServ 네트워크에 대한 평균 지터는 클래스에 따라 적었으나, TCP 네트워크의 경우 평균 지터는, 클래스에 따른 패킷 Drop율에 영향을 받은 결과일 뿐, 큰 차이는 없었다.

그리고 None Drop-Down DiffServ-aware-MPLS의 평균 지터는 패킷 폐기에 따른 재전송 때문에 평균 지연 시간의 성능보다 좀더 차별화 되었고, Drop-Down DiffServ-aware-MPLS의 평균 지터 역시, 경로가 재설정되는 지연 시간 만큼의 차이를 보였을 뿐 None Drop-Down DiffServ-aware-MPLS의 평균 지터와의 차이는 없었다.

표 4. 서비스 클래스에 따른 네트워크별 평균 지터
Table 4. Mean jitter of each class on each network

	TCP	DiffServ	DiffServ-aware-MPLS	DiffServ-aware-MPLS Rerouting
EF	0.77	0.00	0.00	0.09
AF1	0.71	0.08	0.14	0.25
AF2	1.02	0.55	0.79	0.88
DF	1.44	3.01	3.43	3.66

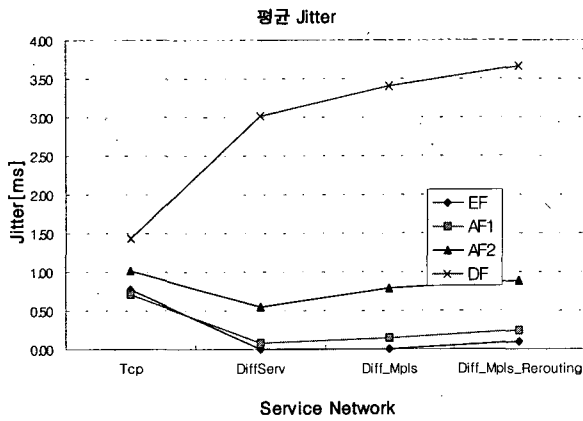


그림 13. 서비스 클래스에 따른 네트워크별 평균 지터
Fig. 13. Mean jitter of each class on each network

(3) Drop 율 성능평가

표 5는 평균 패킷 drop 율의 결과 값을 나타낸 것이고, 그림 14는 결과 값을 그래프로 나타낸 것이다. 표 5에서 DiffServ-aware-MPLS의 패킷 drop 율을 보면 DiffServ의 결과 값보다 클래스 DF의 drop 율은 작아진 반면, 클래스 AF1과 AF2의 drop 율은 커짐에 따라 상대적으로 안정적인 차별화 서비스가 이루어짐을 알 수 있었다. 그리고 Drop-Down DiffServ-aware MPLS 경우에는 손실률이 약간 증가함을 알 수 있었다.

IV. 결 론

지금까지 보인 평균 지연, 지터, 그리고 패킷 Drop 율의 결과 값들을 통해, DiffServ와 비교해 볼 때 제안된 알고리즘을 적용한 DiffServ-aware-MPLS의 차별화 서비스가 우수하였으며, None Drop-Down DiffServ-aware-MPLS와 비교할 때, Drop-Down DiffServ-aware-MPLS의 Rerouting된 결과 값이 경로가 재설정되는 지연 시간만큼의 차이만 있을 뿐 차별화 서비스에는 큰 차이가 나지 않음을 검증하였다. 또한 DiffServ 네트워크가 차별화 서비스를 효과적으로 제공하고 있음은 알 수 있었으나, 라우팅 경로가 Drop 될 경우 그림

표 5. 서비스 클래스에 따른 네트워크별 평균 패킷 Drop 율

Table 5. Mean drop rate of each class on each network

	TCP	DiffServ	DiffServ-aware-MPLS	DiffServ-aware-MPLS Rerouting
EF	200	0	0	11
AF1	200	22	38	48
AF2	230	103	230	240
DF	400	478	403	423

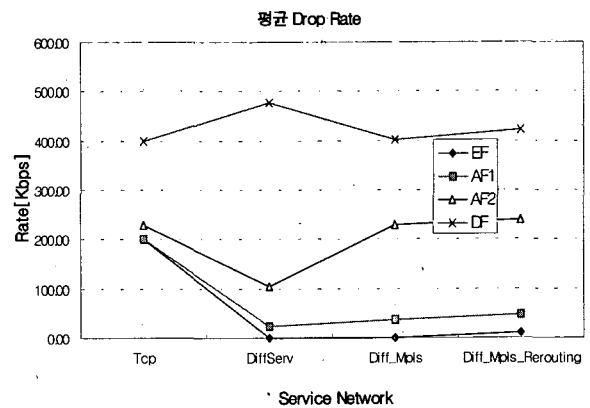


그림 14. 서비스 클래스에 따른 네트워크별 평균 패킷 Drop 율

Fig. 14. Mean drop rate of each class on each network.

10처럼 패킷이 손실되어 이 경우에 QoS를 보장할 수 없음을 알 수 있었다.

본 연구에서는 DiffServ-aware-MPLS 네트워크에서 보다 다양한 차별화 서비스를 보장하기 위한 방법으로, 각 클래스의 특성에 따라 적합한 정책과 큐잉 그리고 스케줄링을 적용함으로써 성능을 향상시켰다. 그리고 제안된 버퍼관리 기법을 검증하기 위해 4장과 5장에서 기존의 TCP, DiffServ 네트워크와 제안된 알고리즘을 적용한 DiffServ-aware-MPLS 네트워크 내에서의 QoS를 비교 분석하여 본 논문에서 제안된 방법이 우수함을 증명하였다.

그러나 MPLS에서 대두되고 있는 TE(Traffic Engineering)인 CR-LDP방식의 E-LSP 서비스만을 시뮬레이션 하였고, 나머지 하나인 L-LSP 서비스는 현재 진행 중이다.

향후 보다 다양한 차별화 서비스로 인해 PHB가 증가할 수 있지만, E-LSP 방식으로는 한정된 EXP 필드는 보다 다양한 PHB를 매핑 시킬 수 없다. 따라서 3비트의 한정된 EXP 필드를 극복하기 위한 L-LSP 방

식을 적용한 TE 기술 개발이 요구되어진다.

참고 문헌

[1] K. Nichols, S. Blake, F. Baker D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[2] S. Blake, D. Black, M. Carlson, Wang and Weiss, "An Architecture for Differentiated Services" , RFC 2475, December 1998.

[3] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB", RFC 2597, June 1999.

[4] V. Jacobson, K. Nichols, Cisco Systems, K. Poduri, Bay Networks, "An Expedited Forwarding PHB", RFC 2598, June 1999.

[5] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[6] E. Rosen, A.Viswanathan, R.Callon. "Multiprotocol Label Switching Architecture", RFC3031, Jan. 2001.

[7] B. Jamoussi (Editor), "Constraint-Based LSP Set up using LDP", RFC 3212, January 2002.

[8] F.E. Faucheur, et. al. "MPLS Support of Differentiated Services" , RFC3270, May. 2002.

[9] Baker, F., Chan, K., and A. Smith, "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.

[10] M. Melody et. al., "Supporting Differentiated Services with Per-Class Traffic Engineering in MPLS" , IEEE ICCCN2001, Oct 2001, pp.354-360.

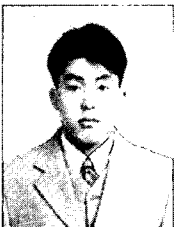
[11] P. Trinintzios, I. Andrikopoulos, et al. "A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks," IEEE Commun. Mag., Special Issue in IP Operations and Management, vol. 39, no. 5, IEEE, May 2001, pp.80-88.

[12] T. Choi, et al. "Wise<TE>: Traffic Engineering Server for a Large-scale MPLS-based IP Network," NOMS 2002, Apr 2002, pp.251-264.

저자 소개



한 철 민(학생회원)
 2002년 한남대학교 전자공학과 졸업(공학사).
 2004년 전북대학교 전자공학과 졸업(공학석사).
 2004년 현재 전북대학교 전자공학과 박사과정.
 <주관심분야: Internet QoS, NGN, SoC 설계, Wireless & Mobile Network>



정 광 일(정회원)
 1997년 전북대학교 전자공학과 졸업(공학사).
 1999년 전북대학교 전자공학과 졸업(공학석사).
 2003년 전북대학교 전자공학과 졸업(공학박사).
 2003년~2004년 한국표준과학연구원 정보전산그룹
 2004년 현재 한국원자력연구소 계측제어인간공학 연구부 선임연구원
 <주관심분야: 인터넷QoS, 초고속통신망구조>

김 변 곤(정회원)
 제39권 TC편 제6호 참조

김 남 희(정회원)
 제41권 TC편 제1호 참조

박 현(학생회원)
 제39권 TC편 제4호 참조

정 경 택(정회원)
 제39권 TC편 제4호 참조

전 병 실(정회원)
 제39권 TC편 제4호 참조