

# MPLS에서 차등화 서비스를 지원하기 위한 E-LSP의 확장

정회원 박기범\*, 정회원 정재일\*\*

## Extension of E-LSP for Supporting Differentiated Service in MPLS

Gi-Bum Park\* Regular Members, Jae-Il Jung\*\* Regular Members

### 요 약

차세대 인터넷은 QoS(Quality of Service)의 보장과 초고속 전송을 특징으로 하는데 그 대표적인 구조가 DiffServ(Differentiated Service)와 MPLS(Multi-Protocol Label Switching)이다. 차세대 인터넷에서 종단간 QoS를 보장하기 위해서 DiffServ와 MPLS를 연동하기 위한 기술에 관한 연구가 필수적이고, 현재 이에 관한 연구가 활발하게 진행되고 있다. IETF(Internet Engineering Task Force)의 MPLS WG(Working Group)에서는 이를 위하여 E-LSP(EXP inferred-PSC LSPs)와 L-LSP(Label-Only-Inferred-PSC LSPs) 방안을 제안했는데, 두 가지 방법 모두 MPLS에서 DiffServ의 서비스 클래스를 지원하는데 문제점을 가지고 있다. 본 논문에서는 기존의 E-LSP를 확장하여 DiffServ의 이론적인 모든 클래스와 ECN(Explicit Congestion Notification)을 비롯한 실험적인 기능을 지원하는 확장 E-LSP를 제안한다. 제안된 확장 E-LSP를 이용한 MPLS망에서 보다 세분화된 DiffServ 클래스를 지원할 경우 서비스 품질을 시뮬레이션을 통하여 검증하였다.

**Key Words** : MPLS; DiffServ; E-LSP; L-LSP; QoS; ECN

### ABSTRACT

NGI(Next Generation Internet) is characterized by QoS(Quality of Service) and high speed transmission. Recently, DiffServ and MPLS become most influential NGI architecture. To guarantee end-to-end QoS, it is essential for NGI to interwork MPLS with DiffServ. Here, MPLS WG(Working Group) in IETF(Internet Engineering Task Force) proposed the method of E-LSP(EXP inferred-PSC LSPs) and L-LSP(Label-Only-Inferred PSC LSPs), but both of them have serious problems to satisfy perfect interworking. In this paper, we proposed an extended E-LSP architecture that supports perfect DiffServ class and experimental function in MPLS such as ECN(Explicit Congestion Notification) capability. We verify that the proposed E-LSP architecture improves QoS in NGI by using ns2 simulator.

\*삼성전자 통신연구소 (jefflin.park@samsung.com), \*\*한양대학교 전자통신전파공학과 멀티미디어 네트워크 연구실 (jjung@hanyang.ac.kr)

논문번호 : 020260-0603, 접수일자 : 2002년 6월 3일  
이 연구는 한국 과학재단 지정 최적설계 신기술 연구 센터의 지원에 의해 수행되었습니다.

### I. 서론

최근 인터넷 환경은 매우 급변하고 있는데, 그 주요한 원인은 크게 두 가지로 나누어 생각할 수 있다. 첫 번째는 인터넷 사용자와 트래픽의 양적인 증가이고, 또 다른 측면은 멀티미디어 서비스와 같은 실시간, 높은 우선순위를 요구하는 트래픽이 증가함에 따라서 서비스 품질의 보장에 관한 요구가 커지고 있다. 차세대 인터넷에서는 기존의 인터넷에서 제공하지 못하는 이러한 두 가지 요구사항을 만족하기 위하여 하위 계층에서는 보다 광대역과 높은 전송속도로 전송 가능한 DWDM(Dense Wavelength Division Multiplexing)과 같은 기술이 개발되고 있고, 전송 및 교환 계층에서는 QoS의 보장과 초고속 전송을 위한 여러 가지 구조들이 제시되고 있는데, 대표적인 것으로는 IntServ(Integrated Service)[1], DiffServ(Differentiated Service)[2], MPLS(Multi-Protocol Label Switching)[3], 트래픽 엔지니어링(traffic engineering), Constraint-Based Routing 등이 있다. 이 중에서 RSVP(Resource reSerVation Protocol)에 기반을 둔 IntServ 모델은 확장성에 문제가 있어서 실제로 구현되지는 못하였고, 트래픽 엔지니어링과 Constraint-Based Routing은 현재 MPLS 기술과 통합되고 있다. 결론적으로 차세대 인터넷의 가장 확실한 대안은 DiffServ와 MPLS라고 할 수 있는데, 차세대 인터넷에서 종단간 서비스품질을 보장하기 위해서는 서로 상이한 계층에 존재하는 이 두 가지 모델을 연동하는 기술이 필수적이다[4].

현재 이를 위해서 IETF의 MPLS WG(Working Group)에서는 E-LSP와 L-LSP 방안을 제안하였는데, E-LSP 방식의 경우에는 8개 이하의 DiffServ 서비스 클래스만을 지원하고, L-LSP방식의 경우에는 PHB 당 하나의 LSP를 유지해야 하므로 실제로 필요한 레이블의 수가 너무 많아지는 단점이 있다.

본 논문에서는 이에 대해 MPLS에서 DiffServ를 완벽하게 지원하기 위한 새로운 구조인 확장 E-LSP를 제안한다. 제안된 확장 E-LSP 구조는 DiffServ에서 이론적으로 지원 가능한 모든 서비스 클래스를 지원할 수 있도록 설계되었고, 기존의 두 가지 방식의 장점만을 이용하므로 MPLS에서 DiffServ를 지원하는 보다 효과적인 대안이 될 수 있다. 또한, 기존의 E-LSP 방식을 사용하는 MPLS망과 본 논문에서 제안된 확장 E-LSP를 사용하는 MPLS 망과의 서비스 품질을 ns2[5][6]를 이용한 시뮬레이션을 통

하여 비교하였다

### II. DiffServ

#### 1. DiffServ 구조

DiffServ는 인터넷에서 QoS를 보장하기 위한 자원 예약 구조인 IntServ의 대안으로 1997년에 IETF에서 제시되었다. 이전에 제시된 IntServ 모델은 RSVP 프로토콜을 기반으로 하므로 도메인(domain) 내의 모든 라우터는 플로우(flow)의 모든 상태정보를 가지고 있어야만 하고, 또한 soft-state 메커니즘에 의해서 주기적으로 플로우의 상태를 점검하는 복잡한 구조를 가지고 있다. 이로 인해 IntServ 모델은 실제로 라우터에서 구현하기 힘들고, 처리 부하가 많이 걸리고, 확장성에도 심각한 문제가 있었다.

이러한 문제를 해결하기 위하여 제시된 DiffServ 모델은 패킷을 몇 개의 차별화된 PHB(Per Hop Behavior) 특성을 가지는 클래스의 집합으로 구분하여, IPv4/IPv6의 헤더의 DSCP 필드에 표시하여 전송하는 구조이다. 그림 1에서 보는 바와 같이 IPv4의 TOS 필드나 IPv6의 Class 필드는 6비트의 DSCP 필드와 CU(Currently Unused) 필드로 나누어지고, DiffServ의 PHB는 DSCP 필드에 표시되며, CU 부분은 최근 들어서 효과적인 혼잡 제어를 위한 ECN 필드로 사용되고 있다.

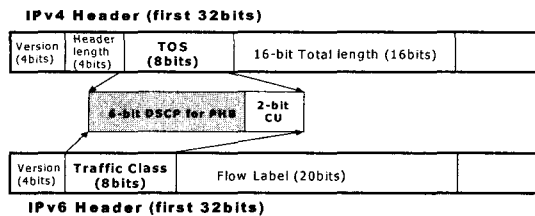


그림 1. DSCP 필드

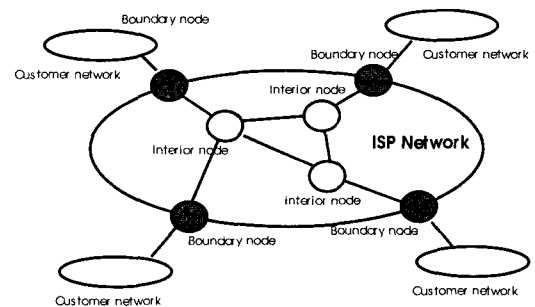


그림 2. DiffServ의 개괄적인 구조

DiffServ에서는 서비스 클래스를 의미하는 몇 개의 PHB 집단으로 패킷을 나누었으므로, 플로우의 복잡도를 줄일 수 있고, 또한 이러한 PHB 정보는 일반적으로 그림 2와 같은 주변 라우터(edge router)에서만 표시하고, 내부 라우터(core router)에서는 표시된 DSCP 필드를 참조하여 단순히 패킷을 전달하는 역할만 한다. 그러므로, DiffServ는 모든 라우터에 RSVP 신호 프로토콜이 필요한 IntServ 구조에 비해서 구조상 간단하고, 확장성이 뛰어나며, 고속으로 전송할 수 있는 장점이 있다.

2. PHB(Per Hop Behavior)

PHB는 DS(DiffServ Domain)내의 라우터 간에 패킷을 어떻게 전송할 것인가에 관한 규정이다. 전송 경로상의 각 노드는 패킷의 DSCP 필드를 참조하여 해당 DSCP에 대해 정의된 PHB로 패킷을 전송한다. 그러므로 각 패킷에 적용할 수 있는 서로 다른 PHB는 결국 서로 다른 서비스 클래스에 해당한다. 일반적으로 IETF의 표준으로 정의된 PHB의 종류는 다음과 같다.[2]

2.1 EF PHB(Expedited Forwarding PHB)

EF PHB[7]는 “101110”으로 DSCP 필드에 표시되며 손실, 지연변이, 전송지연 등을 최소로 갖는 가장 높은 우선순위의 트래픽을 지원하기 위해서 제안된 것이다. 따라서, 실시간 멀티미디어 동영상 전송과 같은 지연에 민감한 응용 트래픽은 EF PHB를 통해 전송된다.

2.2 AF PHB(Assured Forwarding PHB)

AF PHB[8]은 EF PHB와 같은 명확한 서비스 품질을 보장하지는 못하나, 각 트래픽의 서비스 규약에 근접한 수준의 서비스 품질을 제공한다. AF PHB의 기본적인 개념은 DiffServ의 기초가 된 RIO(RED with In and Out) 메커니즘에서 유래하였다. AF PHB는 이러한 RIO 메커니즘을 4개의 전달 클래스와 각각의 클래스에 3개의 폐기 우선순위(drop precedence)를 가진 구조로 확장한 것이다. AF PHB는 다음 표 1과 같이 4개의 클래스와 3개의 우선순위를 갖는 12개의 PHB가 정의되어 있다.

표 1. AF PHB의 DSCP 코드 포인트(code point)

	Class1	Class2	Class3	Class4
Low Drop	001 010	010 010	011 010	100 010
Medium Drop	001 100	010 100	011 100	100 100
High Drop	001 110	010 110	011 110	100 110

2.3 Class Selector PHB

Class Selector PHB는 현재 사용하고 있는 IP 우선순위 필드와의 호환성을 위하여 상대적인 우선순위만 정의하였고, “xxx000”의 코드 포인트를 갖는다.

2.4 Default PHB

특정 서비스를 요구하지 않는 경우 기본적으로 제공하는 최선형 서비스(best-effort service)의 경우, “000000”으로 표시하고 Default PHB라고 한다.

III. MPLS

최근 차세대 인터넷에서는 MPLS가 핵심적인 기술로 각광 받고 있는데, MPLS의 등장은 그 동안 데이터 통신의 기반을 이루고 있던 두 가지 기술인 전통적인 데이터그램(datagram) 기반의 IP 망과 ATM이나 프레임 릴레이(frame relay) 와 같은 연결-지향형(connection-oriented) 망과의 통합을 의미한다고 할 수 있다.

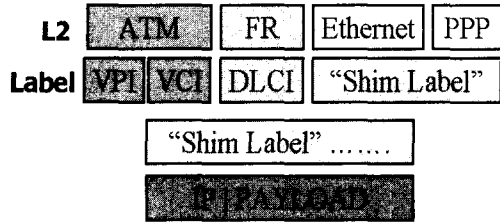
MPLS는 일정한 크기의 레이블을 shim header라는 구조를 사용하여 2계층과 3계층 사이에 삽입, 2계층의 교환 기술과 3계층의 전송 기술을 접목한 기술이다. MPLS 기술은 shim header를 사용하므로 2계층과 3계층의 프로토콜에 관계없이 적용이 가능하며, 강력한 트래픽 엔지니어링 기술과 가상 사설망(Virtual Private Network:VPN)이나 VoMPLS(Voice over MPLS) 등의 다양한 응용 서비스의 창출이 가능하다.

최근에는 IETF를 중심으로 광 네트워크를 포함한 모든 네트워크에 적용할 수 있는 GMPLS(Generalized Multi-Protocol Label Switching)[9]로 발전되어 차세대 인터넷의 기술 표준으로 자리잡아가고 있다.

1. MPLS 구조

그림 3은 MPLS 프로토콜 스택 구조이다. 위의 그림에서 보는 바와 같이 MPLS는 Ethernet이나 PPP(Point-to-Point Protocol) 기반의 망에서는 2계층과 3계층 사이에 레이블 정보를 가지고 있는 shim header를 삽입하여, ATM이나 프레임 릴레이 구조에서는 각각 VPI(Virtual Path Identifier)/VCI(Virtual Circuit Identifier)와 DLCI(Data Link Connection Identifier) 필드에 레이블

값을 할당하여 2계층의 프로토콜에 관계없이 모든 구조에서 적용 가능한 기술이다.



MPLS Encapsulation is specified over various media types. Top labels may use existing format, lower label(s) use a new "shim" label format.

그림 3. MPLS 프로토콜 스택

그림 4는 MPLS 레이블을 포함하고 있는 shim header의 구조이다[10]. Shim header는 32비트로 구성되어 있고 각각의 필드의 내용은 다음과 같다.

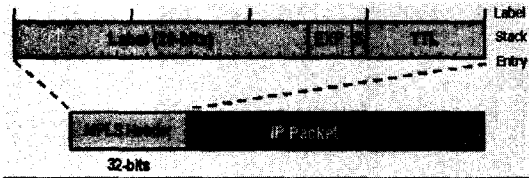


그림 4. MPLS shim header 구조

- \* Label(20bits) : MPLS 레이블 값이 저장되는 필드이다. 0-15번 값은 특수한 레이블 값으로 지정되어 있다.
- \* EXP(3bits) : 실험적인 사용을 위한 필드이었지만, 현재는 주로 서비스 클래스(COS) 필드와의 연동을 위하여 사용되는 필드이다. DiffServ를 지원하는 MPLS에서 가장 중요한 필드이다.
- \* S(1bit) : 레이블 스택의 마지막 항목을 나타내기 위한 비트이다.
- \* TTL(8bits) : 루프 방지 메커니즘 등을 위한 IP 헤더의 TTL(Time To Live) 값이 설정되어 있다

DiffServ를 지원하는 MPLS 망을 설계하기 위해서는 2장에서 언급한 DiffServ의 PHB와 MPLS shim header와의 연동이 필요하다. 본 논문에서는 5장에서 이에 관한 구체적인 방안을 살펴보도록 하겠다.

#### IV. ECN(Explicit Congestion Notification)

#### 1. ECN 개요

1980년대말 이후 Van Jacobson에 의해서 TCP에서의 혼잡 제어 메커니즘이 연구된 후에 여러 가지 IP 상에서의 혼잡 제어 메커니즘이 등장하였다. ECN(Explicit Congestion Notification)[11] 방식은 큐를 모니터하고 하다가 큐가 꽉 차기 전에 미리 패킷을 폐기하여 송신측이 좀 더 빨리 혼잡상황을 인식하도록 하는 기존의 RED(Random Early Detection) [12] 방법을 개선하여 네트워크에서 일어날 가능성이 있는 혼잡을 미리 탐지하여 이를 송신 측에 명시적으로 알려서 전송속도를 조절하는 방법이다.

이러한 명시적인 혼잡제어 방식인 ECN을 사용하면 큐에서 패킷을 미리 폐기하는 것을 방지할 수 있다.

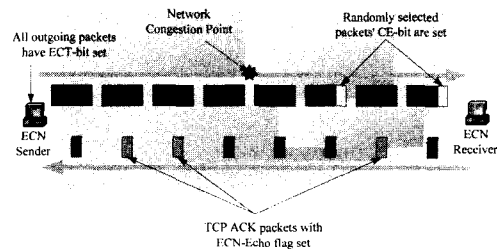


그림 5. 중단간 ECN 동작도

그림 5는 중단간 ECN의 동작을 나타낸 것이다. ECN 알고리즘은 네트워크에서 혼잡이 예상되면 IP 헤더의 CE(Congestion Experienced) 비트를 1로 표시하고 이 패킷이 수신측에 도착하면 이를 송신 측에 알리기 위하여 TCP 헤더에 존재하는 ECN-Echo 플래그를 표시하여 전송한다. ECN-Echo 플래그가 표시된 패킷을 송신 측에서 받았을 때에 송신 측에서는 전송하는 윈도우 사이즈를 절반으로 줄여서 네트워크의 혼잡 상황을 개선한다.

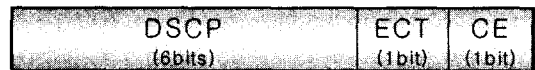


그림 6. TOS 필드 내의 ECN 지원 비트

ECN 기능은 그림 6에서와 같이 IP TOS 필드의 마지막 2비트를 사용한다. IP TOS 필드의 상위 6비트는 DSCP 필드로 DiffServ의 PHB의 정보를 저장한다고 2장에서 이미 언급하였다. IP TOS 필드의

7번째 비트는 ECT(ECN Capable Transport) 비트이고, 8번째 비트는 위에서 설명한 CE 비트이다.

현재 ECN 프로토콜의 기능은 모든 IP 라우터에서 지원하는 것이 아니므로 해당 라우터가 ECN의 기능을 지원하는지의 여부를 표시하는 비트가 필요하다. ECT 비트는 이러한 기능을 수행하는데, 만일 송신측과 수신측에서 ECN의 기능을 모두 지원한다면 송신자는 ECT 비트를 설정한 후에 패킷을 전송한다. 하지만, ECT 비트가 설정이 되지 않았다면 ECN 기능을 사용하지 못하므로 RED 알고리즘을 사용하여 패킷을 단순히 폐기할 수밖에 없다.

## 2. ECN을 지원하는 MPLS

위에서 설명한 IP 망에서의 ECN의 기능을 MPLS에서도 구현이 가능하다면 MPLS에서 최적의 LSP(Label Switched Path)를 찾기 위한 또 하나의 효과적이고 신속한 정보가 될 수 있을 것이다.

하지만 MPLS에서 ECN을 지원하기 위해서는 심각한 제한 요소가 있다. Shim header를 사용하는 MPLS 망인 경우에는 EXP 필드 중 한 비트를 ECN에 할당한다면 DiffServ를 위한 EXP 필드는 2비트로 줄어들게 되어서 최대 지원 가능한 PHB는 4개로 줄어들어 된다.

본 논문에서는 이러한 문제를 해결하기 위하여 새로운 형태의 확장 E-LSP 구조를 사용하여 MPLS에서 DiffServ의 가능한 모든 PHB와 함께 ECN을 지원하도록 하였다.

## V. DiffServ를 지원하는 MPLS

### 1. DiffServ를 지원하는 MPLS 구조

서론에서 언급한 바와 같이 차세대 인터넷에서 중단 간 서비스품질을 보장하기 위해서는 MPLS 망에서 DiffServ를 지원하는 기술(DiffServ over MPLS)이 매우 중요하다. 이를 위해서는 MPLS의 각각의 LSR이 DiffServ의 DSCP 값을 인식해야 하는데, DSCP 필드는 IP 헤더에 들어있으므로 PHB에 관한 정보를 LSR이 직접 참조하지 못한다. 그러므로 MPLS에서 DiffServ를 제대로 지원하기 위해서는 LSR이 패킷을 전송할 때 참조하는 shim header에 PHB에 관한 정보가 있어야 한다. 하지만, MPLS의 shim header 설계 당시에 DiffServ의 PHB는 IP의 우선순위 필드에 해당하는 Class Selector PHB의 수준만을 제공하고 있었으므로,

EXP 필드는 MPLS의 헤더 사이즈를 최소화하기 위해서 3비트로 설계되었다. 이러한 역사적인 배경으로 인하여 최근에 DiffServ에서 사용가능한 PHB의 숫자가 계속 증가함에 따라 6비트의 의미 있는 정보를 담고 있는 DSCP 필드의 값을 MPLS에 적용하기 위하여 3비트의 EXP 필드에 매핑 함으로서 생기는 문제가 발생한다.

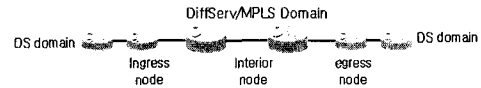


그림 7. DiffServ를 지원하는 MPLS의 구조

그림 7은 DiffServ를 지원하는 MPLS의 구조이다. 일반적으로 MPLS에서 DiffServ를 지원하는 기술은 QoS 결정은 DiffServ 부분에 의해 이루어지고, 패킷의 전송은 MPLS에 의해서 이루어진다[4].

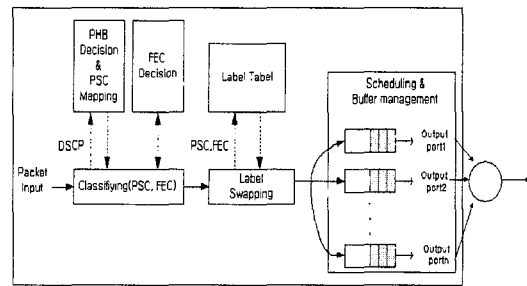


그림 8. DiffServ를 지원하는 MPLS의 동작

### 2. E-LSP(EXP inferred-PSC LSPs)

그림 8은 일반적인 MPLS를 지원하는 DiffServ의 동작을 나타낸 것이다. DiffServ는 일반적으로 DSCP를 참조하여 PHB를 결정하게 되는데, DiffServ를 지원하는 MPLS망은 패킷이 MPLS망으로 진입할 때에 이러한 PHB에 관한 정보를 shim header에도 유지시켜 주어야 한다. 이 경우에 가장 쉽게 적용할 수 있는 방법이 E-LSP[13]이다.

E-LSP는 MPLS shim header의 EXP 필드에 DSCP 정보를 매핑하여 DiffServ의 PHB 정보를 유지하는 것이다. 이러한 E-LSP를 이용하면 DiffServ망의 모든 PHB를 하나의 MPLS LSP(Label

Switched Path)로 할당 가능하다는 장점이 있다.

하지만, EXP 비트는 원래 3비트의 크기로 설계 되었으므로 E-LSP 방법을 통해서도 최대 8가지의 PHB만 지원 가능하므로 현재의 DiffServ 표준에서 정의하고 있는 기본적인 PHB의 개수도 지원할 수 없다는 단점이 있다. (DiffServ에서 BE(1) + AF(12) + EF(1)의 PHB만 지원한다고 가정해도 14개의 PHB가 필요하다.)

또한, E-LSP 방식은 shim header를 사용할 수 없는 ATM, 프레임 릴레이(frame relay)망에서는 적용이 불가능하다.

### 3. L-LSP(Label-Only-Inferred-PSC LSPs)

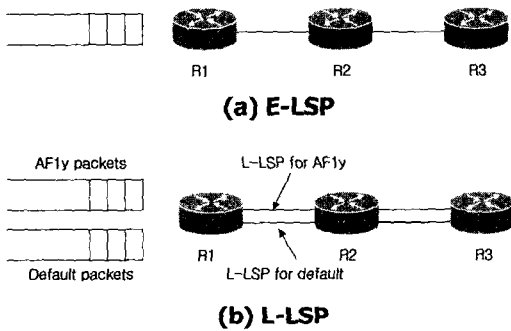


그림 9. E-LSP와 L-LSP의 동작

L-LSP는 E-LSP와는 달리 기본적으로 EXP 필드에 관계없이, AF의 한 클래스에 해당하는 OA(Ordered Aggregate)와 FEC(Forward Equivalence Class)의 쌍에 대해 하나의 LSP를 매핑하는 방법이다. 그림 9는 E-LSP와 L-LSP의 동작을 비교하여 설명한 것이다.

L-LSP를 사용하면 EXP 필드의 값을 참조하지 않고 8개 이상의 PHB를 지원할 수 있고, 이 경우에 EXP 필드의 값은 패킷 페기 우선순위 필드로 사용될 수도 있다. 또한, L-LSP는 E-LSP를 적용할 수 없는 shim header를 사용하지 않는 ATM이나 프레임 릴레이(frame relay) 망에서도 적용 가능한 방식이다. 하지만, L-LSP를 사용할 경우에는 OA당 하나의 LSP를 할당해야 하므로, 레이블 정보 테이블을 유지하기 위해서 메모리의 공간이 많이 필요하다는 단점이 있다.

이는 MPLS를 지원하는 라우터의 성능의 가장 중요한 요소인 최대 지원 LSP(Maximum number of

LSP)의 개수에 직접적인 영향을 준다. 일반적으로 MPLS를 지원하는 라우터의 NP(Network Processor) 모듈은 직접 제어할 수 있는 한정된 메모리 공간을 가지고 있고, 이 한정된 공간에 L3의 FIB(Forwarding Information Base)와 MPLS의 LFIB(Label Forwarding Information Base) 정보를 모두 담고 있어야 하므로 보다 많은 수의 LSP를 지원하기 위해서는 L-LSP를 사용하지 않는 방법이 바람직하다.

### 4. 확장 E-LSP(Extended E-LSP)

#### 4.1 이상적인 DiffServ를 지원하는 MPLS 모델의 조건

일반적으로 MPLS 망에서 DiffServ를 완벽하게 지원하는 가장 이상적인 모델은 표 2와 같은 요구 조건을 가진다.

표 2. 이상적인 DiffServ를 지원하는 MPLS 모델의 조건

- (a) DiffServ의 DSCP에서 표현가능 한 모든 서비스 클래스(26 bits= 64 PHBs)를 지원해야 한다.
- (b) 하나의 LSP로 모든 DiffServ의 서비스 클래스를 표현할 수 있어야 한다.
- (c) 2계층의 인터페이스에 관계없이 동작 가능해야 한다

현재 IETF의 MPLS WG에서 제안하고 있는 E-LSP와 L-LSP 는 어떤 방식도 표 2의 모든 조건을 만족하지 못하고 있다. E-LSP의 경우에는 표 2의 (a)(c)항을 위반하고, L-LSP의 경우에는 (b)항을 위반한다. 그러므로 현재 IETF의 초안에서는 E-LSP와 L-LSP의 조합을 통해서 DiffServ를 지원하는 MPLS망을 설계하도록 권고하고 있다.

하지만, 이러한 방법은 근본적인 문제 해결 방법이 아닐 뿐만 아니라 실제 구현에 있어서 복잡도가 높고 많은 문제점을 지니고 있다. 이러한 문제점을 해결하기 위해서 본 논문에서는 MPLS망에서 하나의 LSP를 가지고 DiffServ의 이론적으로 존재할 수 있는 서비스 클래스를 완벽하게 지원하기 위하여 확장 E-LSP 구조를 제안한다.

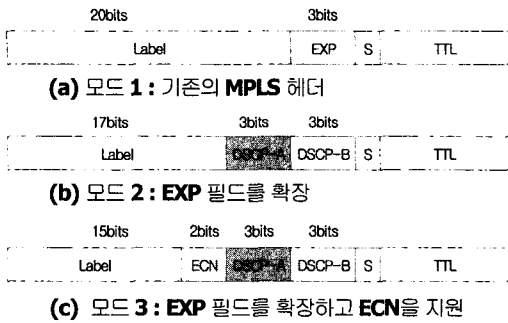


그림 10. 확장 E-LSP 구조

4.2 확장 E-LSP(Extended E-LSP) 구조

그림 10은 본 논문에서 제안하는 확장 E-LSP 구조이다. 확장 E-LSP 구조의 기본적인 개념은 기존의 E-LSP가 EXP 필드만을 참조하여 DiffServ의 클래스를 매핑함으로써 DSCP 필드와 EXP 필드와의 왜곡이 생기는 단점을 보완하는 것이다.

확장 E-LSP 구조는 기존의 MPLS shim header와의 호환성을 위하여 세 가지 동작 모드를 가지고 있다. 모드는 기존의 MPLS와의 호환성을 위하여 존재하는 모드이고, 모드2는 기존의 MPLS 레이블 영역을 분할하여 EXP 필드를 6비트로 확장하여, DiffServ의 DSCP 값을 확장된 EXP 필드에 그대로 매핑하여, PHB의 왜곡 없이 모든 DiffServ의 서비스 클래스를 지원하는 구조를 가지고 있다. 모드2는 확장 E-LSP를 지원하는 MPLS 라우터의 핵심적인 모드이다.

그림 10-(c)의 모드3은 모드2의 기능에 보다 실험적인 기능을 추가한 모드이다. 모드 3에서는 레이블을 15비트로 축소하여 모드2의 모든 기능을 지원하면서 여분의 2비트를 다음과 같은 용도로 사용할 수 있다.

- (1) MPLS에서 ECN을 지원하기 위한 비트
- (2) MPLS를 지원하는 분산 라우터 구조에서 내부적인 포트 정보를 알려주기 위한 비트

ECN을 지원하기 위한 2비트 필드를 생성하여, MPLS 상에서 ECN의 정보를 전달할 수 있도록 하였다. 모드3을 통해서 앞으로 MPLS에서 직접 ECN 필드를 지원할 수 있도록 하여 MPLS 수준에서의 혼잡 제어의 기능성과 함께 constraint-based routing에 ECN의 정보제공이 가능하게 하였다. 모드3의 비트는 또한 완전 분산구조 라우터에서 내부적인

인터페이스 정보로도 사용될 수 있다. 완전 분산 구조 라우터에서는 내부적으로 MPLS 패킷이 지나갈 인터페이스 정보가 필요하고 모드 3은 이런 정보를 위한 비트로 할당될 수 있다.

VI. 확장 E-LSP의 장·단점

본 논문에서 제안된 확장 E-LSP는 MPLS의 레이블 필드를 3비트 혹은 5비트까지 줄여서 이를 DSCP의 상위 3비트와 ECN을 비롯한 실험적인 비트로 활용하는 방법이다. 이러한 방법을 사용할 경우 가장 문제가 될 수 있는 부분은 사용할 수 있는 레이블의 수가 감소한다는 점이지만, 모드 2의 경우에도 실제 지원 가능한 레이블의 수는,  $2^{17} - 16 (= 131056)$ 개로 일반적으로 단위 라우터가 지원하는 최대 레이블의 수로서 충분하고, 모드 3의 경우에도  $2^{15} - 16 (= 32752)$ 보다 현재의 수십 기가급의 상용 라우터가 지원하는 최대 LSP의 개수를 지원할 수 있고, 코어 망이 아닌 라우터에서는 충분히 적용할 수 있는 레이블 공간을 지원한다. 또한 확장 E-LSP 방법은 확장 E-LSP를 지원하는 라우터 망에서 할당하고 관리하는 레이블 공간을 조절하는 방법이므로, 일반적인 MPLS를 지원하는 망에서의 레이블 공간에는 영향을 미치지 않는다.

확장 E-LSP 방법은 다음 [표 3]과 같은 장점을 갖는다.

표 3. 확장 E-LSP의 장점

최대 PHB의 개수	DiffServ에서 이론적으로 가능한 PHB 모두 지원 *최대 64개의 PHB 지원 * DiffServ의 DSCP 필드와 1:1 mapping 가능
MPLS 라우터에서 최대 LSP의 개수	Label 정보를 유지하기 위해 필요한 공간을 줄여주므로, 동일한 H/W 상에서 보다 많은 LSP의 개수 지원하여 MPLS 라우터의 성능 향상
QoS Policy	L3와 MPLS 사이에 하나의 QoS policy 적용 가능 * L3와 MPLS 간의 QoS translation이 필요 없다. * Data plane에서 L3와 MPLS를 동일한 QoS engine으로 구현 가능하다.

### VII. 시뮬레이션

본 논문에서 제안하고 있는 확장 E-LSP 구조의 성능을 검증하기 위하여 linux 환경에서 Network Simulator ns-2(version 2.1b8)[5][6]를 사용하여 시뮬레이션을 하였다. 시뮬레이션의 기본적인 방향은 DiffServ over MPLS망에서 PHB의 숫자를 증가함에 따라서 서비스 품질에 어떠한 영향을 미치는지 분석하였고, ns2에 내장되어 있는 Nortel의 DiffServ 모듈[14]을 이용하여 실험하였다.

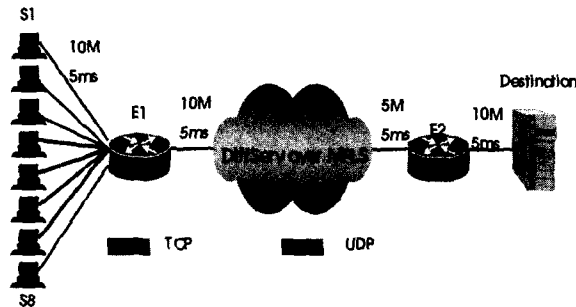


그림 11. 시뮬레이션의 전체적인 구성도

그림 11은 본 논문에서 제안하고 있는 확장 E-LSP의 시뮬레이션 구성도이다. 시뮬레이션에서는 그림 11과 같이 s1 - s8까지의 8개 소스 노드에서도 도착지까지 AF와 BE 스타일의 TCP와 UDP 트래픽을 1:1의 비율로 흘려보낸 후에 최대 PHB를 각각 6개, 9개, 12로 증가시키면서 DiffServ망의 경계 라우터인 E2 라우터에서 전송률을 측정하였다.

PHB6의 경우에는 기존의 E-LSP를 통해서 구현할 수 있지만, PHB9, PHB12의 경우는 본 논문에서 제안한 확장 E-LSP를 통해서만 구현 가능한 실험적인 모델이다. 표 4, 그림 12, 그림 13은 시뮬레이션의 결과를 나타낸 것이다. 표 4에서 보이는 바와 같이 같은 양의 트래픽을 발생시켜 측정하였으므로, 최대 PHB의 개수에 따른 평균 전송률은 거의 비슷한 수준을 보인다. 하지만, 지원하는 최대 PHB의 개수가 증가함에 따라서 높은 서비스 품질을 요구하는 AFX0 계열의 평균 전송률은 올라가게 된다[표 4][그림 13]. 특히 가장 중요한 서비스 품질 요소인 같은 AFX0 계열의 fairness는 PHB6, PHB9에서 보다 PHB12에서 현격히 개선됨을 알 수 있다. 이는 DiffServ를 지원하는 MPLS 망에서

보다 많은 서비스 클래스를 지원하게 된다면, 보다 다양하고 세밀한 큐 파라미터의 조절을 통하여 보다 좋은 QoS 특성을 얻을 수 있다는 것을 반증한다.

표 4. DiffServ를 지원하는 MPLS에서 지원 가능한 PHB의 개수에 따른 전송률의 변화

	AF10	AF20	AF30	AF40	AFX0	평균전송률
PHB6	90.95	82.84			86.90	50.25
PHB9	99.80	99.65	68.54		89.33	52.04
PHB12	99.02	97.60	90.41	84.24	92.89	50.79

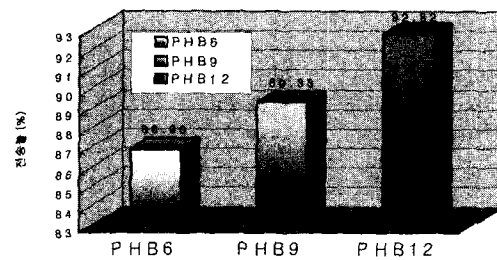


그림 13. 최대 지원 PHB의 개수에 따른 AFX0 트래픽의 평균 전송률

### VIII. 확장 E-LSP의 구현

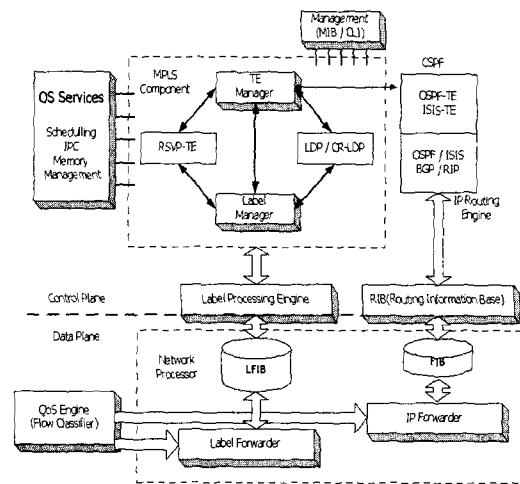


그림 15. DiffServ over MPLS를 지원하는 라우터의 Block Diagram



그림 14는 확장 E-LSP를 적용하기 위한 DiffServ over MPLS를 지원하는 라우터의 일반화된 block diagram을 나타낸 것이다. 확장 E-LSP의 인식은 CLI(Command Line Interface)를 통해서 라우터 내부에서 설정이 가능하도록 하는 지원이다. 또한, 확장 E-LSP 모델의 세 가지 모드는 표 5와 같은 MPLS의 특수 레이블 값을 이용하여 전환될 수 있다.

표 5. 확장 E-LSP 구조에서 정의한 특수 레이블

Label Value	내 용
15	Mode1에서 mode3으로 전환
14	Mode1에서 mode2로 전환
13	Mode3에서 mode1으로 전환
12	Mode2에서 mode1으로 전환

3.2절에서 언급한 바와 같이 32비트의 shim header 중에 MPLS 레이블은 20바이트를 차지하고 있고, 레이블 값 중 0 - 15는 특수한 목적으로 예약되어 있다. 확장 E-LSP 구조는 레이블 스택의 첫 번째 레이블이 표 3에서 정의한 특수한 값의 레이블일 경우에, 그림 10에서 제시한 세 가지 모드 사이에서 전환을 한다.

MPLS 라우터에서 확장 E-LSP를 지원하기 위해서는 label manager를 통해서 각각의 모드에 맞는 레이블 범위를 설정하도록 지원해야 하고, NP(Network Processor)안에서 비트 보충(bit padding)과 비트 이동(bit shift) 연산을 통해서 확장 E-LSP를 지원하도록 설계된 LFIB를 참조하여 Label Forwarding을 수행할 수 있도록 마이크로 코드를 수정해야 한다. 하지만, 확장 E-LSP를 사용하면 L3 패킷과 MPLS 패킷에 대해서 라우터 내부에서 통일된 QoS 정책을 수행할 수 있으므로, data plane의 QoS 엔진을 L3 QoS와 MPLS QoS 간의 QoS translation 등의 절차가 필요없는 간단한 구조로 설계할 수 있다.

### IX. 결 론

본 논문에서는 현재 MPLS에서 DiffServ를 지원하기 위하여 제안된 E-LSP와 L-LSP의 문제점을 지적하고, MPLS에서 DiffServ의 모든 서비스 클래스를 지원하고 ECN 기능을 갖는 확장 E-LSP 구조를 제안한다. 또한, 제안된 확장 E-LSP 구조를 사용하

였을 경우에 기존의 E-LSP 구조에 비해 서비스 품질이 향상됨을 시뮬레이션을 통하여 검증하였다.

확장 E-LSP 모델은 기존의 E-LSP와 L-LSP의 단점을 보완하여, 하나의 LSP를 가지고 이론적으로 가능한 모든 DiffServ의 서비스 클래스를 지원할 수 있고, 불필요한 레이블 공간을 줄여서 동일한 하드웨어 상에서 라우터의 성능을 높일 수 있다. 또한, 라우터 내부에서는 동일한 QoS 정책을 가질 수 있다는 구조의 통일성 있는 구조를 지원하고, 기존의 MPLS 소프트웨어를 조금만 변경하면 구현이 가능하고 기존의 구조와의 호환성도 매우 뛰어난 구조라고 할 수 있다.

본 논문에서는 기본적인 확장 E-LSP를 지원하는 모드 2 이외에 실험적인 모드 3를 제안하였다. 모드 3를 통한 ECN의 기능은 구현방안과 성능향상은 아직 검증하지 못하였지만, 모드 3를 통한 정보를 분산 서버에서의 내부적인 포트 정보의 필드로 사용할 수도 있을 것이다. 또한, 이러한 레이블 필드 분할 기법은 본 논문에서 제안하고 있는 두 가지 모드 이외에도 Martini draft[15]나 MPLS VPN등과 같은 stacked label을 사용하는 망에서 많은 응용이 적용될 수 있을 것이라 생각하며, 추후에 이런 부분에 대한 보다 추가적인 연구가 필요하다고 생각한다.

### 참 고 문 헌

- [1] R. Braden et al., "Integrated Services in the Internet Architecture : an Overview", RFC 1633, June 1994.
- [2] S. Blake et al., "An Architecture for Differentiated Services", RFC 2475, Dec 1998.
- [3] E. Rosen et al., "Multi-Protocol Label Switching Architecture", RFC 3031, Jan 2001.
- [4] Francois Le Faucheur, et al., "MPLS Support of Differentiated Service", RFC 3270, May 2002.
- [5] McCanne and S.Floyd, ns-Network Simulator, <http://www.isi.edu/nsnam/ns>.
- [6] Kevin Fall, Kannan Varadhan, *The ns Manual*, UC Berkeley and Xerox, June 20, 2001.
- [7] V. Jabcoson et al., "An Expedited

- Forwarding PHB”, RFC 2598, June 1999.
- [8] J. Heinanen et al., “Assured Forwarding PHB Group”, RFC 2597, June 1999.
- [9] Peter Ashwood-Smith et al., “Generalized Multi-Protocol Label Switching Architecture”, draft-ietf-ccamp-gmpls-architecture-07.txt, May 2003.
- [10] E. Rosen et al., “MPLS Label Stack Encoding”, RFC 3032, January 2001.
- [11] K. Ramakrishnan, S. Floyd, “A Proposal to Add Explicit Congestion Notification(ECN) to IP”, RFC 2481(Experimental RFC), January 1999.
- [12] S. Floyd, and V. Jacobson “Random Early Detection Gateways for Congestion Avoidance”, *IEEE/ACM Transactions on Networking*, Vol. 1, No.4, pp 397-413, August 1993.
- [13] S.Ganti, N.Seddigh, B.Nandy, “MPLS Support of Differentiated Service using E-LSP”, Internet-Draft, draft-ganti-mpls-diffs-erv-elsp-02.txt, June 2002.
- [14] Peter Peda, Jeremy Ethridge, Mandeep Baines, and Farhan Shallwani, A Network Simulator, Differentiated Services Implementation. Open IP, Nortel Networks, 2000.
- [15] Luca Martini et al., “Encapsulation Methods for Transport of Ethernet Frames Over IP/MPLS Networks”, Internet-Draft, draft-ietf-pwe3-ethernet-encap-04.txt, October 2003
- 박 기 범(Gi-Bum Park)                                   정회원  
2000년 2월 : 한양대학교 전자/전자통신/전파공학과 학사졸업  
2002년 2월 : 한양대학교 전자/전자통신/전파공학과 석사졸업  
2002년 3월 ~ 현재 : 삼성전자 근무  
  
<주관심분야> 광대역 통신망, QoS
- 정 재 일(Jae-Il Jung)                                   정회원  
03-28-4B-12 참고