MPLS 네트워크에서 DiffServ를 효율적으로 적용하기 위한 동적 흐름 수락 제어

임 지 영† · 채 기 준††

요 약

본 논문에서는 차세대 인터넷으로 진화할 수 있는 새로운 핵심 기술로 평가받고 있는 MPLS(MultiProtocol Label Switching) 네트워크에서 QoS를 제공하기 위하여 DiffServ(Differentiated Services)를 지원하는 동적 흐름 수락 제어를 제안한다. 제안 모델은 아웃 포트의 전송 가능한 용량을 계산하여 이 용량에 맞게 흐름을 받아들이고 패킷을 전송한다. 또한 QoS 라우팅 테이블을 위한 정보 교환으로 혼잡 지역을 알 수 있는 ingress LSR은 혼잡 지역을 배제한 경로로 패킷을 전송하게 한다. 이렇게 함으로써 제안 모델은 DiffServ의 단점으로 인정되는 자원 낭비를 막고 자원을 효율적으로 관리하여 자원의 이용률을 높이고 서비스 받기 위해 네트워크에 진입하는 흐름의 수를 높일 수 있다.

The Dynamic Flow Admission Control for Providing DiffServ Efficiently in MPLS Networks

Ji-Young Lim† · Ki-Joon Chae††

ABSTRACT

MPLS(MultiProtocol Label Switching) is regarded as a core technology for migrating to the next generation Internet. In this paper, we propose an dynamic flow admission control supporting DiffServ(Differentiated Services) to provide QoS in MPLS networks. Our proposed model dynamically adjusts the amount of admissible traffic based on transmittable capacity over one outgoing port. It then transmits the packets while avoiding congested area resulting traffic loss. Ingress LSRs find out the congested area by collecting network state information at QoS state update for QoS routing table. Our proposed model manages the resource efficiently by protecting the waste of resources that is a critical problem of DiffServ and makes much more flows enter the network to be served.

키워드 : MPLS, DiffServ, 흐름수락제어(Flow Admission Control), QoS

1. 서 론

차세대 인터넷으로 진화할 수 있는 새로운 핵심 기술로 평가 받고 있는 MPLS[1]는 스위칭 기술의 접목으로 인한 고속화와 트래픽 엔지니어링을 제공하여 기존의 망에서 제공할 수 없었던 고속 서비스와 다양한 부가 서비스를 지원할 수 있다. 본 논문에서는 MPLS 네트워크에서 QoS를 지원하기 위해 Internet QoS의 한 방법인 DiffServ를 좀 더 효율적으로 지원하기 위한 트래픽 흐름 수락 제어를 제안한다.

정책 기반의 자원 할당이나 흐름 수락에 대한 연구나 표준안이 아직은 저조한 수준이며 이들 방식은 서버의 도움을 필요로 하므로 IPOA(IP over ATM)와 같이 서버 사용 중심이 되어 연결 지향형인 MPLS 네트워크에서 오버헤드를 부과할 수 있다. 본 논문에서는 이와는 다른 측면에서 DiffServ의 트래픽 흐름 수락을 제어하여 서버의 도움 없이 엣지 라우터에서 자체적으로 해결하기 위한 방안을 제안한다. 먼저 하나의 아웃 포트로 전송할 수 있는 용량을 기준으로 트래픽 흐름을 받아들일 수 있는 기본적인 양을 조절하며, 트래픽의 손실을 야기할 수 있는 혼잡 지역을 피하여 트래픽을 네트워크로 보내고자 한다.

두 번째는 네트워크의 혼잡 지역을 반영하여 흐름 수락 제어를 한다. 트래픽의 혼잡 지역을 알기 위해서는 Ingress LSR(Label Switched Router)과 Egress LSR에서 이 정보를 수집할 수 있다. 여기서는 QoS 경로 배정의 링크 상태 조사를 위해 전송하는 제어 메시지를 이용하여 Ingress LSR에서 혼잡 지역을 알아내어 처리한다. 이렇게 함으로써 엣지에서 프로세싱하고 코어에서 스위치하는 MPLS의 특징을 준수하면

† 정 회 원 : 이화여자대학교 컴퓨터학과 교수
†† 종신회원 : 이화여자대학교 컴퓨터학과 교수
논문접수 : 2001년 11월 5일, 심사완료 : 2001년 12월 12일

서 추가의 네트워크 자원 이용 없이 효율적인 자원 이용과 트래픽 흐름 수락을 제어함으로써 보다 높은 수준의 트래픽 엔지니어링을 실현할 수 있으리라고 본다.

2. MPLS의 DiffServ 지원

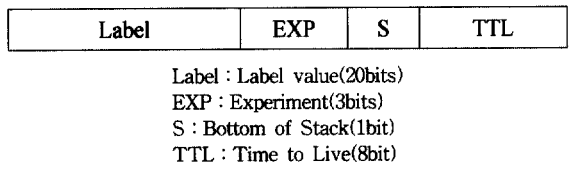
VoIP(Voice over IP)와 VPN(Virtual Private Networks)과 같이 인터넷에서 QoS 보장을 요구하는 새로운 응용 서비스들의 출현으로 IP QoS 가 중요한 과제로 대두되고 있다. 현재의 인터넷은 베스트 에포트 서비스만을 제공하기 때문에 서비스에 따른 패킷의 지연 등과 같은 요구 사항을 보장하지 못하고 있다. 따라서 인터넷에서 서비스의 QoS를 보장하기 위하여 제안된 서비스 모델이 IntServ(Integrated Services)[2]와 DiffServ[3]이다.

또한 MPLS 네트워크에서 QoS는 상위 계층 즉, IP 계층에서의 QoS를 지원하는데 초점이 맞추어져 있다. 즉, MPLS QoS의 목적은 IP QoS보다 다소 월등한 MPLS QoS 방안을 제공하는 것이 아니라 IP QoS와 MPLS QoS사이에 동등성을 확립하는 데 있다[4]. 따라서 IETF에서 MPLS만의 QoS 프레임워크 또는 QoS 지원에 대한 표준화 작업은 미비한 실정이다. 반면에 IP QoS인 IntServ와 DiffServ를 MPLS 네트워크에서 지원하는 방안에 대한 표준화는 완료되었거나 진행되고 있다[5, 6].

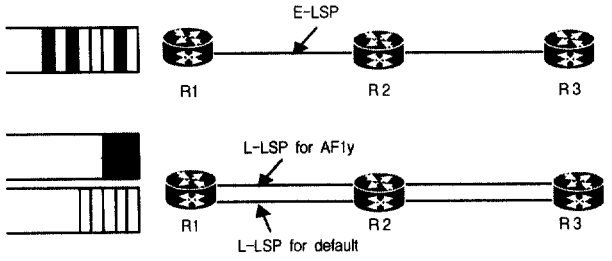
IntServ는 자원을 예약하기 위한 신호처리 프로토콜로서 MPLS처럼 RSVP(Resource ReSerVation Protocol)[7-9]를 사용하기 때문에 MPLS에서 IntServ를 지원하는 것은 어렵지 않다. 반면에 DiffServ는 각각의 흐름 단위로 자원을 예약하는 것이 아니라 트래픽의 성질에 따라 흐름을 클래스로 분류한다. 또한 클래스의 BA(Behavior Aggregation)를 엣지 라우터에서 결정하고 코어 라우터에서는 BA에 따라 패킷을 전송하거나 버린다. 이러한 점에서 DiffServ도 엣지 라우터에서 라우팅을 결정하고 코어 라우터에서는 이에 따라 전송만 하는 MPLS와 잘 어울린다.

DiffServ 모델에서는 특정한 엣지간 서비스가 아닌 홉 단위의 큐 관리 및 스케줄링 기능인 PHB(Per Hop Behavior)로 QoS를 제공한다. DiffServ 모델에서 표준화한 PHB는 다음과 같다. EF(Expected Forwarding)[10]는 경로상의 모든 라우터들이 EF 패킷에 대한 처리를 적어도 패킷들이 도착하는 속도보다는 빠르게 하도록 요구하기 때문에 PDR(Peak Data Rate)로 자원을 예약하여야 한다. AF(Assured Forwarding)[11]는 엣지간 서비스를 위한 하나의 PHB Group으로서, 지연에 민감하지 않으며 드롭 우위를 갖고 있어 CDR(Committed Data Rate)로 예약을 하여도 무방한 클래스이다.

MPLS 네트워크에서 DiffServ를 지원하기 위해서는 IP 헤더에 있는 DSCP(Differentiated Service Code Point) 코드에 따라 DiffServ 클래스의 BA를 지원할 수 있도록 다른 기능을 추가하여야 한다. MPLS에서는 (그림 1)과 같은 헤더 포맷을 가지며 헤더의 EXP 필드가 DSCP와 같은 목적으로 사용할 수 있으나 EXP 필드의 길이는 3비트이다. 이로 인해 6비트 DSCP의 모든 값을 EXP 필드로 일대일 사상시키는 것이 불가능하므로 IETF에서는 MPLS에서 DiffServ를 지원하는 두 가지 방법을 드래프트로서 제안하고 있다[5].



(그림 1) MPLS 헤더



(그림 2) E-LSP and L-LSP

- **E-LSP(EXP-Inferred-PSC(PHB Scheduling Class) LSPs)**

(그림 2)에서와 같이 단일 LSP를 위한 단일 FEC에 8개의 BA를 사상시켜 DiffServ를 지원하는 방법으로, MPLS 헤더의 EXP 필드 3비트를 이용하여 패킷에 적용할 PHB를 결정하는 데 사용한다. LSP에 대한 EXP 필드로부터 PHB로의 사상은 레이블 설정 시 명시적으로 나타내거나 미리 구성해 놓은 사상을 사용한다. 이 방법은 MPLS 헤더의 EXP 필드를 사용하므로 추가로 신호 처리 프로토콜이 필요 없고, 어떠한 레이블 분배 프로토콜과도 잘 어울린다는 장점이 있다.

- **L-LSP(Label-Only-Inferred PSC LSPs)**

E-LSP를 사용한 방법이 8개의 PHB만 지원하기 때문에 그 이상의 PHB를 지원하기 위해 제안된 방법이 L-LSP이다. 이 방법은 (그림 2)에서와 같이 디폴트 패킷과 프리미엄 패킷을 위한 LSP와 단일 AF의 클래스, 예를 들면 AF1y를 처리하기 위한 LSP를 각각 사용한다. 여기에 AF 클래스인 경우에는 EXP 필드를 추가로 사용하여 드롭 우위를 표시한다. 어느 PHB가 어느 레이블과 결합되는지를 결정하기 위해서 FEC로 주로 사용되는 프리픽스와 PHB가 함께 레이블을 결정하도록 레이블 분배 프로토콜을 확장하여야 한다는 단점이 있다.

3. 동적 흐름 수락 제어

흐름 수락 제어는 ATM(Asynchronous Transfer Mode)과 같이 연결형 서비스를 제공하는 네트워크에서는 자원을 예

약하고 연결을 설정하기 위한 트래픽 제어와 함께 필수적인 요소이다. 그러나 인터넷은 베스트 에포트 서비스를 주로 하였기 때문에 흐름 수락 제어와 트래픽 제어를 지원하지 않았지만 점차 인터넷에서 QoS를 요구함에 따라 흐름 수락 제어나 트래픽 제어가 필요하게 되었다.

IntServ는 흐름에 대한 자원을 예약할 수 있는 경우에 흐름을 네트워크로 받아들이고 흐름의 트래픽 프로파일에 따라 패킷을 전송하는 흐름 수락 제어와 트래픽 제어를 하고 있다. 반면에 DiffServ는 트래픽 프로파일에 따라 트래픽을 전송하거나 지연 또는 드롭하는 트래픽 제어의 형태를 띄고 있다. DiffServ는 EF에 대하여 PDR(Peak Data Rate)로 자원을 예약하고 AF에 대하여는 CDR(Committed Data Rate)로 자원을 예약하는 것으로 흐름 수락 제어를 하고 있다. 그러나 EF에 대한 과도한 트래픽 예약으로 인한 자원의 낮은 이용률에 따라 EF의 PDR에 따른 자원 예약 낭비가 DiffServ의 최대 취약점으로 지적되고 있다.

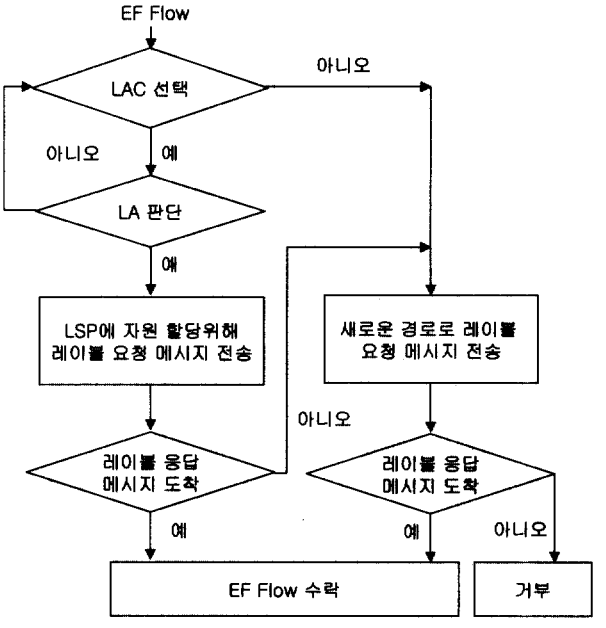
따라서 본 논문에서는 MPLS에서 DiffServ를 효율적으로 지원하기 위해 네트워크의 혼잡을 회피하는 방안으로 유연하면서 동적인 흐름 수락 제어를 제안하고자 한다. 특히 제안하는 방안은 EF 트래픽에 의해 예약되고 사용되지 않는 자원을 사용하여 AF 트래픽을 전송하여 자원의 낭비를 막고자 한다. 또한 이미 설정된 LSP의 수정[12]을 통하여 효율적인 LSP 집합 스킴을 제안한다.

3.1 EF 클래스의 흐름 수락 제어

EF 서비스 클래스는 지연과 손실에 민감한 프리미엄 서비스를 제공하므로 최대 전송률로서 대역폭을 할당하도록 하고 있다[10]. (그림 3)과 같이 EF 트래픽 흐름에 대한 수락 제어는 LSP 집합이 가능한 LSP를 찾아내는 모듈, 선정된 LSP에 추가로 흐름 할당이 가능한지 판단하는 모듈과 이에 따라 새로운 레이블을 요청할 지와 기존 레이블에 추가로 자원을 할당할 지를 결정하여 메시지를 전송하고 이에 따른 처리를 하는 부분으로 크게 나누어 볼 수 있다.

먼저 첫 번째 모듈인 LAC(LSP Aggregation Candidate) 선택에 대해 기술하면 다음과 같다. EF 흐름이 MPLS 네트워크로 들어갈 때 먼저 기존 LSP를 조사한다. 이는 LSP 집합을 도모하기 위함으로 기존에 이미 설정된 LSP 중 같은 곳으로 가면서 각 E-LSP, L-LSP가 제안하는 바대로 같은 LSP를 사용할 수 있는 LSP가 있는지 살핀다. 여기서는 목적지, DiffServ의 DSCP 코드 정보가 필요하며 E-LSP인지 L-LSP인지 혹은 다른 방법이든 연결 설정할 방법에 따라 검토 대상이 될 LSP 후보가 달라진다. 선택된 LSP 후보 중 하나에 대하여 다음 모듈을 적용한다.

LA(LSP Aggregation) 판단 모듈에서는 EF 흐름인 경우에 선정된 LSP로 EF 흐름의 최대 전송률로 할당이 가능한지를 판단한다. 가능하다면 일단 선정된 LSP의 출구 LSR로 Label Request 메시지를 보내고 이에 대한 응답으로 Label Mapping 메시지를 받으면 선정된 LSP로 새로운 EF의 패킷을 기존에 전송되던 흐름과 함께 전송하게 된다[12]. 자원 할당 모듈에서 선정된 LSP로 더 이상 할당할 수 없다면 다른 LSP 후보에 대하여 이를 반복하고 더 이상의 LSP 후보가 없으면 언급한 LDP[13]와 CR-LDP[14]의 방식으로 새로운 LSP를 설정하게 된다. 새로이 설정된 LSP를 통하여 EF 흐름을 전송하게 된다. 만약 새로운 LSP 설정이 불가능하다면 EF 흐름은 네트워크로 트래픽 전송이 거부된다.



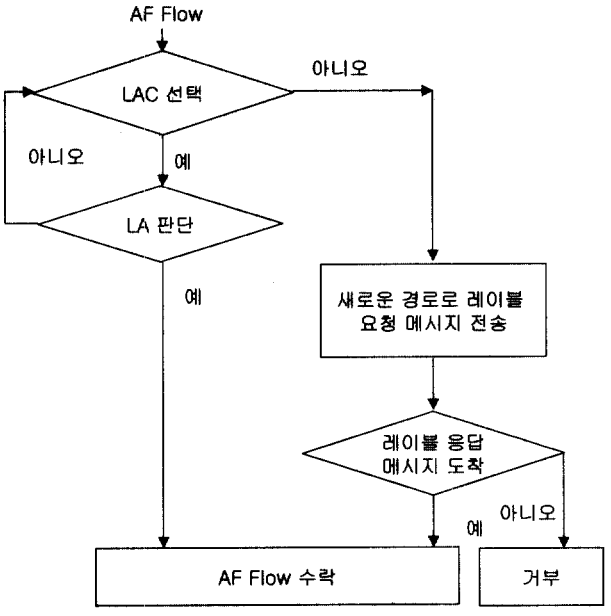
(그림 3) EF 흐름 수락 제어

3.2 AF 클래스의 흐름 수락 제어

AF 클래스는 지연에 민감하지 않고 다만 원하는 드롭률을 만족시키면 되는 클래스이다. AF 흐름에 대하여 평균 데이터 전송률로서 예약을 해도 무방한 클래스이다. 그러나 연결형 서비스를 제공하는 MPLS에서는 자원을 예약하는 것보다는 AF 클래스가 전송할 수 있는 풀을 묵시적으로 정하여 원하는 서비스를 받을 정도로 네트워크에 트래픽을 전송한다면 더 나은 효과를 얻을 수 있을 것으로 보고 AF 클래스에 대하여는 자원을 예약하지 않고 네트워크로 들어가는 흐름의 양을 조절하고자 한다.

(그림 4)는 AF 클래스의 흐름 수락 제어에 대한 그림이다. 먼저 AF 흐름이 도착하면 앞서 기술한 EF의 흐름 수락 제어에서와 같이 함께 전송이 가능한 LSP 후보를 선정하는 작업을 수행한다. 선정된 LSP에 대하여 LSP 집합을 하여도 무방한지를 보고 가능하면 이 LSP를 통하여 패킷을 전송하고 그렇지 않은 경우 다른 LSP 후보에 대하여 이를 반복한다. 여기서 가능한 후보가 없다면 다른 경로로 LDP 방식에 의한 Label Request 메시지를 전송하고 새로운 LSP를 설정한다.

만약 설정 가능한 LSP가 없다면 이 흐름은 네트워크로 전송이 거부된다.



(그림 4) AF 흐름 수락 제어

3.3 LAC(LSP Aggregation Candidate) 선택

이 절에서는 앞에서 기술한 흐름 수락 제어의 첫 번째 모듈인 LAC 선택에 대하여 기술한다. LSP 집합의 목적은 한 LSP에 가능한 여러 트래픽 흐름을 전송하도록 하여 경로 배정 수행 횟수를 줄이고 대역폭의 이용률을 높이고자 한다. LAC 선택은 DiffServ의 DSCP를 어떻게 LSP에 사상시키는 가에 따라 기본적으로 달라진다. 첫 번째로 L-LSP에 대한 LAC 선택, 다음으로는 E-LSP에 대한 LAC 선택 순으로 기술한다.

- **L-LSP의 LAC 선택**

〈표 1〉 L-LSP의 EXP 필드의 값과 PHB의 사상

EXP 필드	PHB
000	AF×1
001	AF×2
010	AF×3

L-LSP는 레이블로 서비스 클래스를 구별하고 AF의 드롭 순위 정보 등은 EXP 필드를 통해서 나타내고 있다. 따라서 DiffServ의 모든 서비스 클래스를 전송하고자 한다면 한 쌍의 입출 LSR의 입장에서 보면 EF 서비스와 베스트 에포트 서비스 지원에 필요한 하나의 LSP와 AF 서비스 클래스를 각각 지원할 LSPs가 필요하다. 따라서 L-LSP의 LSP 집합은 EF 또는 AF 구분 없이 트래픽 흐름이 들어오면 Flow-ID와 목적지 정보, 트래픽 프로파일을 받아 EF 또는 AFx의 동일한 서비스 클래스 중에서 목적지 정보가 같은 LSP를 찾는다. 이 때 찾아진 LSP는 다음 절에서 기술할 판단 모듈을 적용할 대상이 된다. 본 논문에서 사용한 EXP 필드와 PHB의 사상은 <표 1>과 같다.

- **E-LSP의 LAC 선택**

E-LSP는 EXP 필드로 서비스 필드를 구분하며 미리 정해 놓은 서비스 클래스에 대하여 서비스를 진행한다. 또한 AF 클래스를 모두 다 지원하기 위해서는 EF, 베스트 에포트와 AFx를 짝을 지어 전송하면 된다. 따라서 E-LSP의 LSP 집합은 EF는 목적지 정보만 같은 LSP를 찾아 LSP 집합 가능 후보를 선정하면 되고 AF는 AFx의 같은 서비스 클래스를 찾아 L-LSP의 AF 서비스 클래스와 동일하게 LSP 집합 가능 후보를 선정하면 된다. 본 논문에서 사용한 EXP 필드와 PHB의 사상은 <표 2>와 같다.

〈표 2〉 E-LSP의 EXP 필드의 값과 PHB 사상

EXP 필드	PHB
000	EF
001	AF×1
010	AF×2
011	AF×3
111	DF

3.4 LA(LSP Aggregation) 판단

LSP에 함께 전송할 서비스 클래스의 종류를 달리 하는 두 가지 방법에서 선정한 LSP 집합 가능 후보를 대상으로 LSP 집합이 실제로 네트워크에 미치는 영향을 판별하여 추가로 LSP에 트래픽 흐름을 할당해도 좋은 지를 결정한다. 본 논문에서는 트래픽 흐름을 LSP에 추가로 할당할 때 필요한 기준이 되는 정보 수집에 대해 다음과 같은 두 가지 방법을 제안한다.

3.4.1 평균 전송 총량과 링크의 용량에 따른 판단(Decision-A)

DiffServ의 최대 단점인 EF에 대한 자원 낭비를 해소하고자 제안하는 방법이다. EF 트래픽은 흐름의 CDR(Committed Data Rate)이 아닌 PDR(Peak Data Rate)로 자원을 할당하기 때문에 CDR과 PDR의 차이가 클수록 자원을 많이 낭비하게 된다. IETF에서는 EF 트래픽의 자원을 효율적으로 관리하고자 BB(Bandwidth Broker)[15, 16]를 제안하고 있다. 이는 서브넷마다 BB를 두고 서브넷 구성원들의 자원을 관리하고 할당해주며 서브넷간에도 BB 사이에서 정보를 주고 받아 EF에 대한 자원 할당을 해주는 등 일종의 자원 관리 서버이다. 비연결형으로 자원을 할당하지 않는 네트워크에서 DiffServ의 EF 서비스를 효율적으로 제공하기 위해 필요한 것으로, 연결형 서비스를 제공하면서 대역폭도 할당해줄 수 있는 MPLS에서는 역할이 중복되는 서버이다.

AF 서비스의 경우, 자원을 예약한다면 CDR로 예약하도록 하고 있으며 AF 서비스의 특징이 정해진 드롭 제한이 지켜지는 한 지연에는 무관하기 때문에 EF 서비스처럼 반드시 예약하여 전송할 필요는 없다. 그러나 AF 서비스 클래스를 제한 없이 무조건 다 받아들인다면 클래스의 요구 사항을 만족시키지 못할 뿐 아니라 네트워크의 폭주를 야기하여 다른 서비스 클래스에도 지장을 초래할 수 있다. 따라서 본 논문에서는 AF의 자원을 예약하지는 않지만 네트워크로 들어오는 흐름을 제한하여 원하는 서비스 요구 사항을 만족시키는 동시에 네트워크의 혼잡을 가능한 피하고자 한다.

제안 모듈, Decision-A는 LSR에서 특정 아웃 링크를 통해 동시에 전송할 수 있는 트래픽의 양은 아웃 링크를 통해 전송될 수 있는 총량, 즉, 링크의 용량보다 클 수 없다는 기본적인 사실을 바탕으로 한다. 특정 아웃 링크의 용량보다 더 많은 양의 트래픽이 도착하면 전송되지 못한 트래픽은 큐에 남게 될 것이다. 전송되지 못하는 트래픽의 양이 많아지면 큐에 트래픽이 가득 쌓여 더 이상 큐에 남지 못하고 일부를 드롭하여야 한다. 따라서 LSP 집합 후보 중에서 다음을 만족하지 못하는 LSP는 LSP 집합 후보에서 제거된다.

$$b_{tr\_o_i} + b_{con\_o_i} + b_{new\_o_i} < B_{o_i}$$

위 식에서 B_{o_i}는 LSR i의 아웃 링크, o_i의 링크 총량이고, $b_{tr_o_i}$는 LSR i를 경유하는 트래픽 중 아웃 링크, o_i로 향하는 트래픽이 사용하는 대역폭량이며 $b_{con_o_i}$는 LSR i에서 생성된 트래픽 중 아웃 링크, o_i로 향하는 트래픽의 대역폭 사용량이고 $b_{new_o_i}$는 새로이 생성되어 네트워크로 들어가려는 트래픽 중 아웃 링크, o_i로 향하는 트래픽의 대역폭 사용량이다.

$b_{tr_o_i}$는 LSR이 단지 코어 라우터의 역할만을 수행하는지 아니면 엣지 라우터의 역할과 코어 라우터의 역할을 동시에 수행하는 지에 따라 이 항목은 무시될 수 있다. $b_{tr_o_i}$는 EF와 AF의 서비스 구분 없이 실제 전송 측정치를 이 값으로 사용한다. 먼저 지나온 LSR에서 이와 같은 작업을 거쳐 EF와 AF 사이에 전송되는 트래픽량을 조절했기 때문이다.

그러나 $b_{con_o_i}$는 LSR i에서 LSP를 생성한 트래픽에 대한 값이므로 EF와 AF 트래픽에 따라 $b_{con_o_i}$에 합산되는 값이 다음과 같이 결정된다.

$$b_{con\_o_i} = b_{conEF\_o_i} + b_{conAF\_o_i}$$

AF 트래픽 흐름의 대역폭 사용량, $b_{conAF_o_i}$는 아웃 링크, o_i로 향하는 트래픽 중 AF 트래픽 흐름의 CDR의 합으로 결정한다. 또한 EF 트래픽 흐름의 대역폭 사용량, $b_{conEF_o_i}$는 실제 자원을 할당한 PDR이 아닌 CDR로 점유된 대역폭의 양을 결정한다. EF 트래픽 흐름이 항상 일정한 양을 생성하는 ATM의 CBR과 같은 트래픽이 아닌 경우 실제 전송되는 트래픽 양은 평균 값인 CDR을 기준으로 최대 PDR, 최소 0으로 트래픽을 생성할 수 있기 때문이다. 그러나 EF 트래픽 흐름의 PDR과 CDR의 차이가 크면 AF 트래픽을 제때 전송하지 못하고 큐에 쌓일 수 있다. 따라서 큐의 상태에 따라 $b_{conEF_o_i}$에 α (>1)를 곱하여 LSP 집합을 제한하도록 한다.

3.4.2 입구 LSR에서의 혼잡 링크 정보 반영(Decision-B)

네트워크 상태를 알기 위해 QoS 경로 배정[17]에서 링크의 QoS 상태를 알기 위해 사용하는 제어 메시지를 이용한다. Decision-B의 작동은 다음과 같다. 자신의 큐가 임계치를 넘으면 혼잡을 예방하기 위하여 OSPF[18]에서와 같이 제어 메시지를 전송할때 혼잡 경고 비트를 추가하여 이를 1로 하여 전송할 수 있다. 이를 받은 LSR은 이 LSR을 혼잡 링크로 분류하고 이 링크를 경유하는 LSP 집합이나 새로운 경로 설정에는 이 링크를 제외한다. 다음 제어 메시지가 도착할 때 혼잡 경고 비트가 0이면 혼잡 링크를 정상 링크로 상태를 변경한다. 본 논문에서 사용한 경로 배정[17]은 제어 메시지를 자주 보내지 않으므로 혼잡이 발생한 연후에 이를 알리기 보다는 그 이전에 임계치 값에 다다르면 미리 보내도록 하여 혼잡을 예방할 수 있다. 링크 상태 전파 시간이 너무 긴 경우 오랜 시간 경로 배정이나 LSP 집합을 할 수 없을 수 있으므로 상태 전파를 위한 제어 패킷을 혼잡이 해제되면 미리 보내도록 할 수 있다.

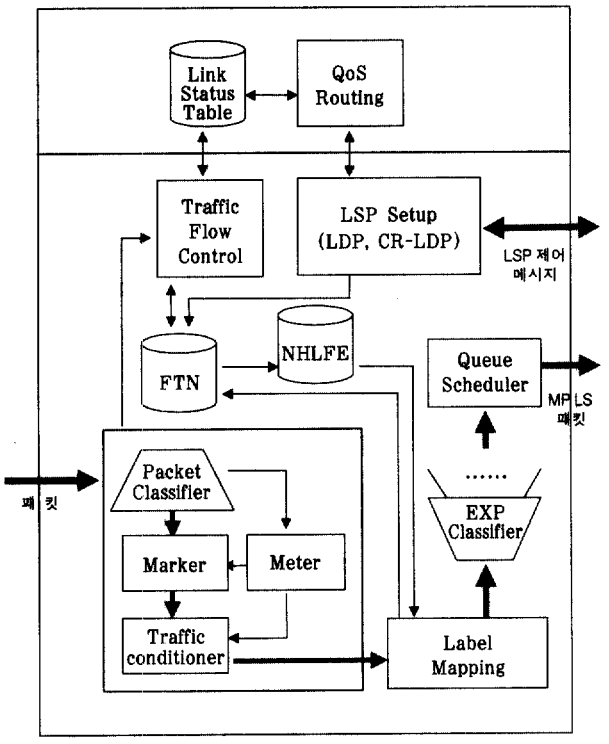
4. 제안 모듈과 DiffServ, MPLS와의 관계 구조

이 장에서는 3장에서 제안한 트래픽 흐름 수락 제어 모듈과 MPLS, DiffServ와의 관계 구조에 대하여 기술한다. (그림 5)는 입구 LSR에서의 제안한 트래픽 흐름 수락 제어 모듈과 MPLS, DiffServ와의 관계 구조를 도식화한 것이며, (그림 6)은 이에 따른 코어 LSR에서의 관계 구조를 도식화한 것이다. 먼저 수행 흐름에 따라 입구 LSR에 대하여 기술하면 다음과 같다.

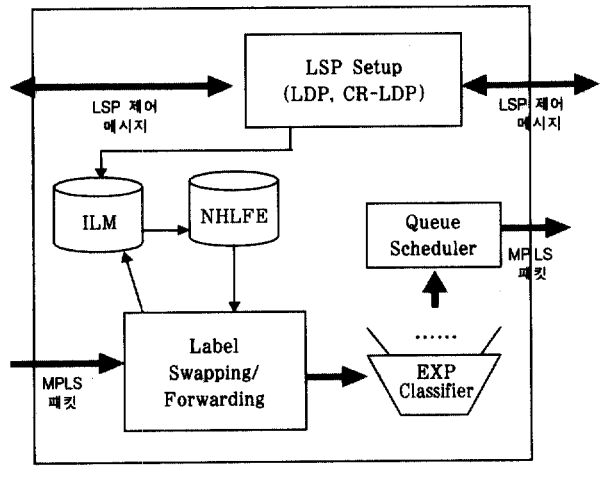
패킷이 DiffServ 모듈에 들어오면 먼저 흐름의 시작인지 아닌지를 구분한다. 이 패킷이 흐름의 시작이면 트래픽 흐름 제어 모듈에 이 패킷의 프로파일을 보낸다. 이를 받은 흐름 수락 제어 모듈은 FTN(FEC(Forward Equivalence Class)-to-NHLFE(Next Hop label Forwarding Entry)) MAP을 통해서 기존 LSP를 대상으로 3.3에서 기술한 LAC 선택 모듈을 수행한다. 여기서 선택된 LSP에 대하여 3.4에서 기술한 LA 판단 모듈이 LSP 집합을 결정한다. 이에 따라 FTN을 수정한다. 만약 LSP 집합 대상이 없으면 LSP Setup 모듈에 트래픽 프로파일에 따라 새로운 LSP를 설정하도록 요청한다. 새로운 LSP가 설정되면 이에 대한 정보 역시 FTN과 NHLFE에 기재되게 된다.

한편 들어온 패킷은 DiffServ 모듈을 통과한 후에 Label

Mapping 모듈에서 FTN을 통해 NHLFE의 엔트리 내용에 따라 레이블을 사상하고 트래픽의 프로파일과 패킷의 DSCP 코드에 따라 EXP 필드를 결정한다. 다음 홉으로 전송되기 전에 먼저 EXP 필드에 따라 DiffServ의 요구 조건에 맞는 큐로 들어가고 Queue Scheduler에 따라 다음 홉으로 전송된다.



(그림 5) 입구 LSR의 구조



(그림 6) 코어 LSR의 구조

3.4.2에서 기술한 Decision-B는 QoS 경로 배정에 따른 링크 상태 전파에 의해 수정된 Link Status Table의 내용을 참조하여 Traffic Flow Admission Control 모듈을 수행한다. LSP Setup 모듈에서 경로 배정을 통한 새로운 LSP 설정을 위하여 트래픽 특성에 따른 차등화된 경로 설정 모듈[17]에 경로를 요청하면 QoS Routing 모듈은 변경된 Link Status Table에 기초하여 경로를 결정한다.

다음으로는 코어 LSR의 작동 상황에 맞추어 (그림 6)에 대하여 설명하면 다음과 같다. MPLS 패킷은 Label Swapping/Forwarding 모듈을 통과하면서 ILM(Incoming Label Mapping)과 NHLFE를 통해 다음 홉으로 가기 위한 레이블을 사상하고 EXP 필드의 값에 따라 트래픽 프로파일을 만족시켜 줄 수 있는 큐로 각각 들어가고 Queue Scheduler에 따라 다음 홉으로 전송된다. 또한 LSP 제어 메시지에 의해 FTN과 NHLFE는 엔트리 내용이 변경된다.

5. 실험 및 성능 분석

이 장에서는 L-LSP와 E-LSP에 따른 LAC 선택과 Decision-A와 Decision-B에 따른 LA 판단으로 구성된 제안 흐름 수락 제어에 대한 성능을 분석한다. <표 3>은 우의 사항의 다양한 조합으로 구성된 실험할 리스트를 보여준다. 5.1에서는 실험 모델에 대하여 기술하고 5.2와 5.3, 5.4에서는 성능 평가 결과를 보여준다.

〈표 3〉 시뮬레이션 리스트

	Resource Reservation		LSP Types	LA
	EF	AF		
llsp + res	○	○	L-LSP	If resource is reserved
elsp + res	○	○	E-LSP	
llsp + nolmt	○	×	L-LSP	unlimited
elsp + nolmt	○	×	E-LSP	
llsp + A	○	×	L-LSP	Decision-A
elsp + A	○	×	E-LSP	
llsp + A&B	○	×	L-LSP	Decision-A + Decision-B
elsp + A&B	○	×	E-LSP	

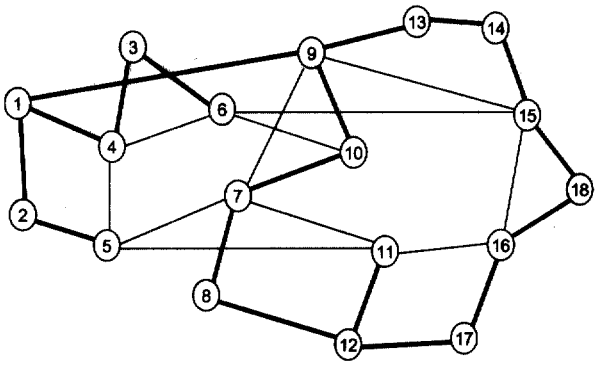
〈표 4〉 트래픽 모델

	EF	AF1y	AF2y	AF3y
Required bandwidth	3	1		
Flow duration	60	30		
PDR / CDR	3 / 1	1 / 1		

5.1 실험 모델

(그림 7)은 본 논문에서 사용하는 ISP 백본 네트워크[18]의 예이다. 실험을 간단하게 하기 위하여 모든 링크는 각 방향마다 동일한 대역폭 용량, C를 갖는 양방향 링크이다. 흐름은 도착율 λ의 포아송 프로세스에 따라 토폴로지의 모든 노드 중 하나에 도착하고 도착지는 무작위로 근원지를 제외한 노드 중 하나가 선택된다. 흐름의 지속 시간은 평균 $1/\mu$의 지수 분포를 따른다.

네트워크에 제공되는 로드는 $\rho = \lambda Nh/\mu LC$[19]이며, 여기서 N은 흐름 발생 노드의 수이고 L은 링크의 수이고 h는 흐름의 평균 홉 수이다. k개의 흐름을 고려하고 각 흐름 i가 $1/\mu_i$의 평균 지속 시간을 갖으며 B_i의 대역폭을 요구한다면 네트워크에 제공되는 로드는 $\rho = \sum_{i=1}^{k} \rho_i$라 할 수 있다. 우리는 <표 4>와 같이 두 가지의 트래픽을 고려하며 EF에 대한 트래픽 로드, ρ_1/ρ는 AF에 대한 트래픽 로드, ρ_2/ρ와 동일하다. 이에 따른 실험에 사용되는 매개변수의 값은 $C=20$, $N=18$, $L=60$, $h=2.58$이다. AF_{xy}의 요구 드롭률은 모두 0.05로 하였다.

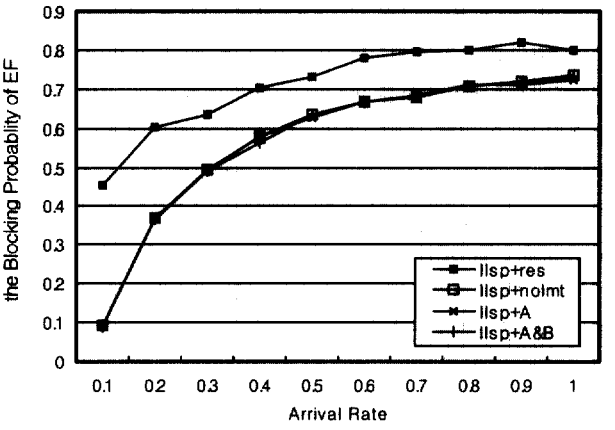


(그림 7) ISP 백본 네트워크 토폴로지

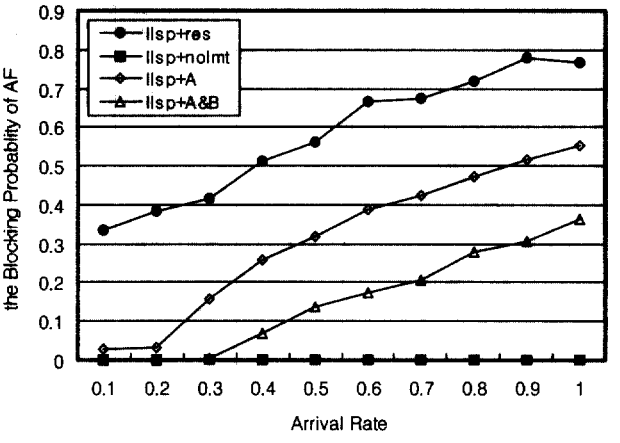
5.2 차단률과 대역폭 이용률

먼저 L-LSP 연결 방식에 대한 비교 모델사이의 EF 흐름의 차단률을 비교하고 L-LSP와 E-LSP 연결 방식에 대한 AF 흐름의 차단률을 비교한다. 마지막으로 전체적인 대역폭 이용률을 비교한다. (그림 8)은 AF와 EF 흐름의 요구 자원을 예약하는 *llsp+res*가 가장 높은 차단률을 보여주고 그 외의 다른 모델은 거의 비슷한 차단률을 보인다. (그림 9)는 AF 흐름에 대한 차단률에 대한 그래프로서 여기서도 가장 높은 차단률은 *llsp+res*이다. 여기서 AF와 EF 흐름의 요구 자원을 예약하면 EF에 의한 사용되지 않는 자원이 많아 전체적인 흐름 차단률도 높아지는 것을 알 수 있다.

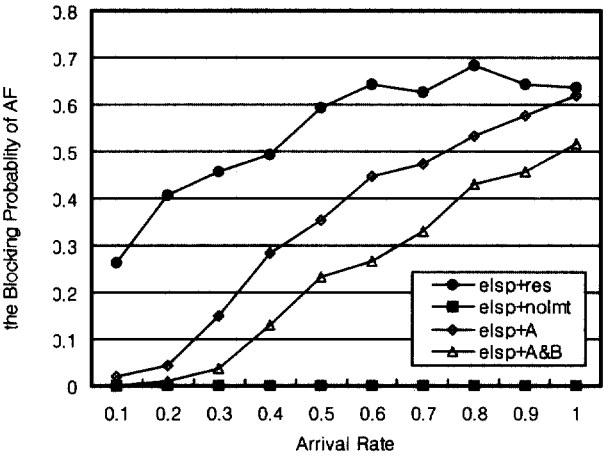
(그림 9)에서 *llsp+nolmt*는 AF 흐름의 네트워크 진입을 제한하지 않기 때문에 당연하게도 가장 낮은 차단률을 보인다. *llsp+A*와 *llsp+A&B*의 차단률을 비교하면 Decision-A와 Decision-B를 함께 사용한 방법이 혼잡 지역을 피하여 경로를 정하기 때문에 차단률이 더 낮음을 알 수 있다. E-LSP의 경우도 L-LSP의 경우가 비슷한 결과를 보인다. 단지 (그림 9)와 (그림 10)을 비교하여 보면 *elsp+A*와 *elsp+A&B*의 차단률이 *llsp+A*와 *llsp+A&B*의 차단률에 비해 좀 더 높음을 알 수 있는 데 이는 L-LSP 방식이 각 AF 그룹에 대하여 LSP를 설정하여 AF 트래픽에 대하여 더 많은 LSP 집합을 허용하는 한편 각 AF 그룹을 위해 설정한 LSP는 같은 그룹의 흐름만을 전송하기 때문이라고 볼 수 있다.



(그림 8) L-LSP 연결 방식에 대한 EF 흐름의 차단률



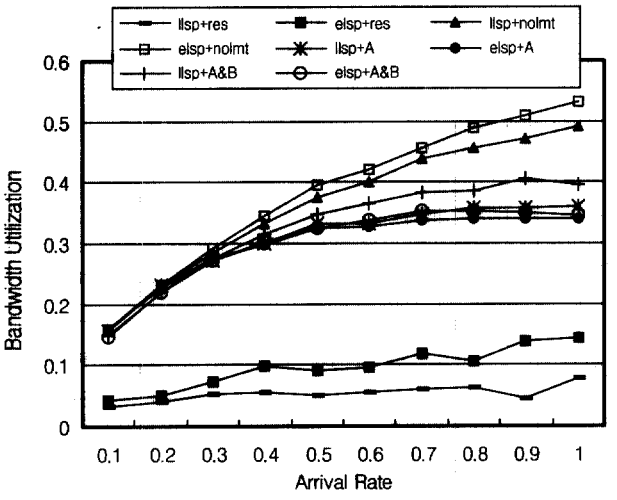
(그림 9) L-LSP 연결 방식에 대한 AF 흐름의 차단률



(그림 10) E-LSP 연결 방식에 대한 AF 흐름의 차단률

(그림 11)은 대역폭 이용률에 대한 그래프로 요구 자원을 예약하는 *llsp+res*와 *elsp+res*의 대역폭 이용률이 가장 낮아 MPLS에서 DiffServ를 지원하기 위하여 EF와 AF가 요구하는 자원을 모두 예약하는 것이 심각한 자원 낭비와 성능 저하를 일으킬 수 있음을 보여준다. 또한 무조건 AF 트래픽을 받아들이기 때문에 *llsp+nolmt*와 *elsp+nolmt*의 대역폭 이용률

은 가장 높다. *llsp+A*와 *llsp+A&B*의 대역폭 이용률을 비교하면 Decision-A와 Decision-B를 함께 사용한 방법이 혼잡 지역을 포함하는 경로를 배제하기 때문에 대역폭 이용률이 더 높음을 알 수 있다. *elsp+A*와 *elsp+A&B*의 대역폭 이용률에 대한 비교는 L-LSP의 결과와 유사하다. 그러나 차단률에서와 같이 *elsp+A*와 *elsp+A&B*의 대역폭 이용률이 *llsp+A*와 *llsp+A&B*의 대역폭 이용률에 비해 좀 더 낮음을 알 수 있는데 이는 연결 설정 방식의 차이에 의한 것으로 차단률에서와 같은 이유로 볼 수 있다.

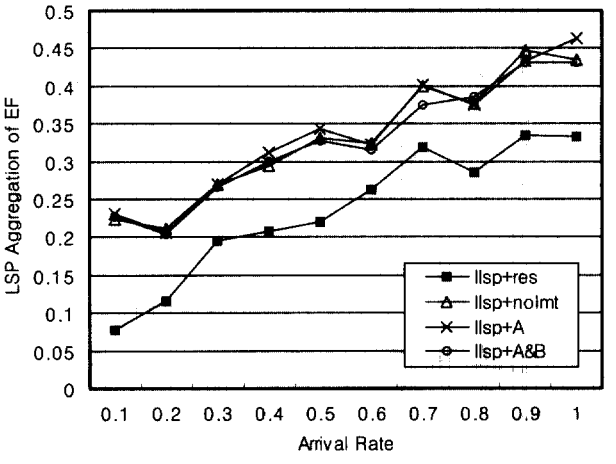


(그림 11) 대역폭 이용률

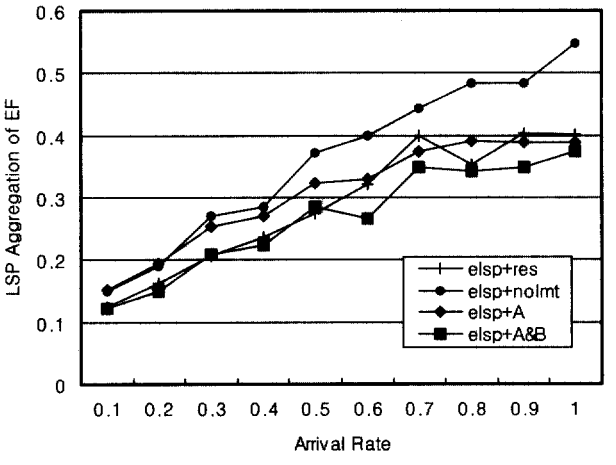
5.3 LSP 집합률

(그림 12)과 (그림 13)은 EF에 대한 두 가지 연결 방식에 대한 LSP 집합률을 보여준다. L-LSP 연결 방식에 대한 그래프인 (그림 12)에서 *llsp+res*의 결과값이 가장 낮고 다른 세 모델은 비슷하지만 *llsp+A&B*의 결과가 그 중 근소한 차이로 낮다. 이는 자원 예약을 해야 하는 *llsp+res*에서 이미 설정해 놓은 LSP로 자원 예약이 어렵기 때문에 다른 세 모델보다 낮은 결과를 보인다. 또한 다른 세 모델은 AF 흐름의 받아들이는 정도만 다르지만 그 값이 비슷하게 나온 것으로 보아 이는 L-LSP에서 가능한 EF 흐름에대한 LSP 집합의 한계를 보여준다고 할 수 있다. 그러나, (그림 13)에서는 *elsp+A&B*의 결과가 가장 낮지만 (그림 12)와는 달리 *elsp+res*의 값과 많은 부분 유사하다. 다른 모델의 값도 (그림 12)에서보다 그 차이가 뚜렷하다. 이렇게 (그림 12)와 (그림 13)의 결과가 상이하게 나타나는 것은 E-LSP 연결 방식은 한 LSP에 EF와 AF 흐름이 공존하기 때문에 EF 흐름의 집합률이 AF 흐름에 대한 제어에 따라 상이한 결과를 보인다고 할 수 있다.

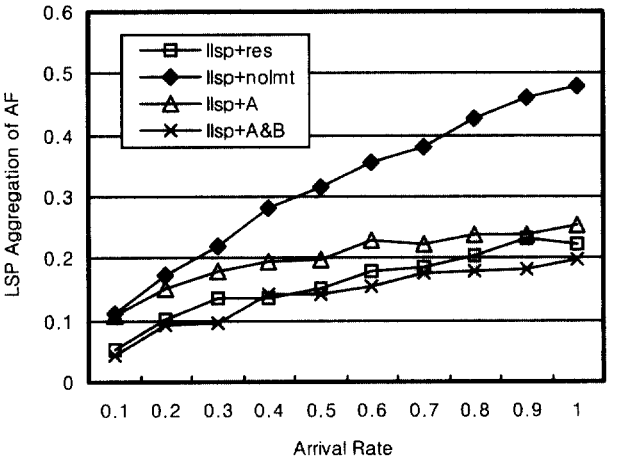
(그림 14)와 (그림 15)는 AF에 대한 두 가지 연결 방식에 대한 LSP 집합률을 보여준다. (그림 14)와 (그림 15)를 보면 연결 방식에 관계없이 Decision-A와 Decision-B를 사용하는 두 방법의 결과가 가장 낮다. 이는 네트워크의 상황을 파악하여 가능하면 혼잡 지역을 피하고 다양한 경로를 이용하는 새로운 LSP를 설정하기 때문이다.



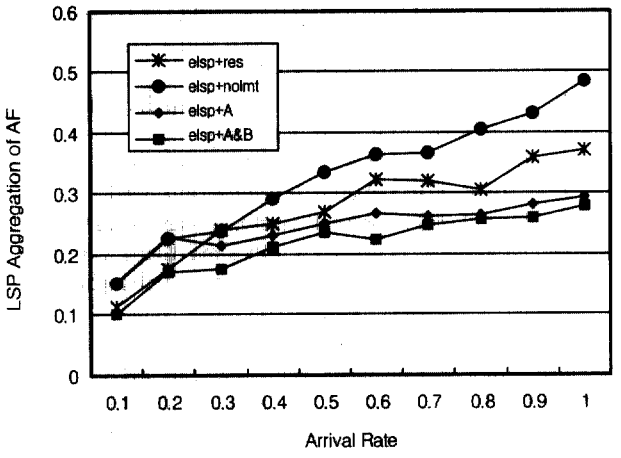
(그림 12) L-LSP 연결 방식에 대한 EF 흐름의 LSP 집합률(%)



(그림 13) E-LSP 연결 방식에 대한 EF 흐름의 LSP 집합률(%)



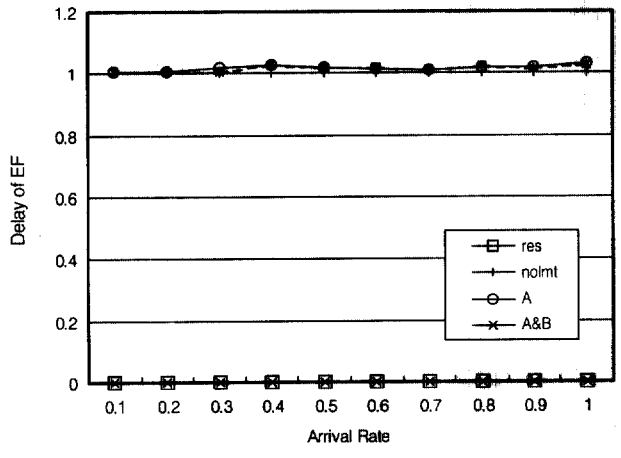
(그림 14) L-LSP 연결 방식에 대한 AF 흐름의 LSP 집합률(%)



(그림 15) E-LSP 연결 방식에 대한 AF 흐름의 LSP 집합률(%)

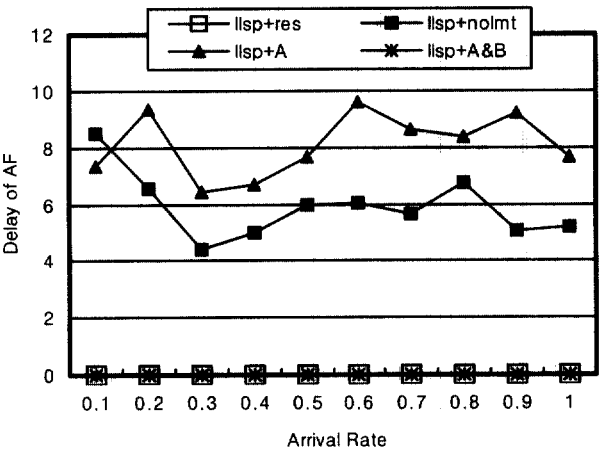
5.4 지연과 드롭률

먼저 EF와 AF 흐름에 대하여 패킷의 평균 지연시간을 기술하면 다음과 같다. (그림 16)에서 모델 *A*와 모델 *nolmt*의 EF에 대한 지연은 약 1 실험 시간 단위인데 반하여 모델 *res*과 모델 *A&B*은 0의 지연 시간을 갖음으로 요구 사항을 만족하는 것을 볼 수 있다. 또한 (그림 17)에서 L-LSP 연결 방식에 대한 AF 흐름의 지연은 요구 사항이 아니지만 *llsp+A*와 *llsp+nolmt*의 AF에 대한 지연은 최소 4 실험 시간 단위 이상인데 반하여 *llsp+res*과 *llsp+A&B*은 0의 지연 시간을 갖는다. 그러나 *llsp+A*의 지연이 *llsp+nolmt*의 지연보다 큰 것은 다음 (그림 18)에서 보면 알 수 있듯이 *llsp+nolmt*의 패킷이 중도에 드롭되는 비율이 높기 때문이다. E-LSP 연결 방식에 대한 결과는 (그림 17)과 유사하므로 생략하였다.

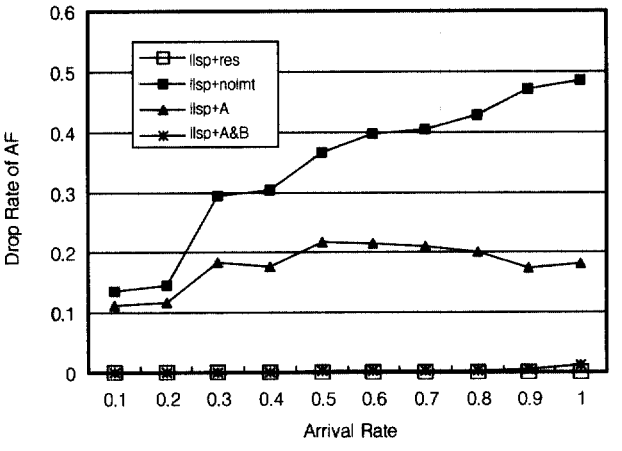


(그림 16) EF 흐름의 지연

EF의 드롭률은 모든 모델에서 PDR로 자원을 예약하였기 때문에 0을 나타내므로 본 논문에서는 그래프를 생략하였다. (그림 18)은 L-LSP 방식에 대한 AF 흐름의 패킷 도롭률로서 무제한으로 AF 흐름을 받아들이는 *llsp+nolmt*의 드롭률이 최대로 AF 흐름의 요구 조건을 만족시키지 못하고 있다. 또한 *llsp+A*의 드롭률은 *llsp+nolmt*의 드롭률보다는 낮지만 AF 흐름의 요구 조건을 만족시키지 못하고 있다. 반면에 *llsp+res*과 *llsp+A&B*의 드롭률은 거의 0에 가까워 AF 흐름의 요구 조건을 충분히 만족시킨다고 볼 수 있다.



(그림 17) AF 흐름의 지연



(그림 18) AF 패킷의 드롭률

5.5 결과 요약

본 논문에서 제안한 방법(llsp+A&B와 elsp+A&B)을 기존의 방법(nolmt와 res)과 비교하여 차단률과 대역폭 이용률, 집합률, 지연, 드롭률의 성능을 분석하였다. 그 결과 llsp+nolmt와 elsp+nolmt는 차단률과 대역폭 이용률, 집합률에 있어서 좋은 성능을 보임을 알 수 있었다. 그러나 이 방법은 DiffServ의 요구 조건인 지연과 드롭 정도를 만족시키지 못하였으므로 앞으로 MPLS 네트워크에서 사용될 수 있는 방법이 될 수 없다. 따라서 이 모델은 차단률과 대역폭 이용률, 집합률의 상한선을 제시한다고 할 수 있다. 또한 llsp+res와 elsp+res는 지연과 드롭률에 있어서는 DiffServ의 요구 조건을 완전하게 만족시키지만 차단률과 대역폭 이용률, 집합률에 있어서 많은 자원 낭비를 야기한다. 반면에 본 논문에서 제안한 llsp+A&B와 elsp+A&B은 지연과 드롭률에 있어서는 DiffServ의 요구 조

건을 충분히 만족시키는 동시에 차단률과 집합률에 있어서도 크게 나쁘지 않으면서 대역폭 이용률에 있어서도 우수하여 llsp+res와 elsp+res의 자원 낭비를 해결한다. 따라서 llsp+nolmt와 elsp+nolmt, llsp+res, elsp+res의 단점을 해소하고 장점을 지원함으로 인하여 제안한 방법이 MPLS 네트워크에서 DiffServ가 서비스될 때 자원을 효율적으로 관리하기 위하여 충분히 이용될 수 있는 방법이다.

6. 결론 및 향후 계획

본 논문에서는 WAN에 적합한 DiffServ를 자원 예약이 가능한 MPLS에 적용하는 경우에 DiffServ의 단점인 자원 낭비가 심화될 것으로 예측하고 이를 저하시키기 위한 방안으로 동적이고 유연한 흐름 수락 제어를 제안하였다. 본 논문에서 제안한 모델은 아웃 포트의 전송 가능한 용량에 맞추어 받아들일 수 있는 트래픽의 약을 동적으로 조절하고(Decision-A), 트래픽 손실을 야기하는 혼잡 지역을 피하여 트래픽을 전송한다(Decision-B). Ingress LSR은 QoS 라우팅을 하기 위하여 링크의 상태를 제어 메시지를 통하여 교환하므로 이를 조금만 변경하여 이용하면 혼잡 지역을 찾아내는 정보를 얻을 수 있다. Decision-A와 Decision-B를 사용하는 제안 모델은 DiffServ의 최대 약점인 자원 낭비를 극복하고 효율적인 자원 이용률을 보이며 AF 흐름의 네트워크 진입률을 높였다. 또한 MPLS에서처럼 엣지에서만의 작업으로 흐름을 수락 여부를 결정하므로 MPLS 트래픽 엔지니어링을 실현하는데 도움을 줄 것으로 기대한다. 향후 계획으로는 본 논문의 실험은 AF에 대하여 CBR 트래픽에 대하여만 실험하였는데 VBR 트래픽에 대하여 실험한다면 본 모델의 더 나은 실험 결과를 보여 줄 수 있을 것이다.

참 고 문 헌

[1] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001.

[2] R. Braden, et al., "Integrated Services in the Internet Architecture : an Overview," RFC 1633, Jun. 1994.

[3] S. Blake, et al., "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.

[4] Bruce Davie, Yakov Rekhter, *MPLS : Technology and Applications*, Morgan Kaufmann Publishers, 2000.

[5] F. L. Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, J. Heinanen, "MPLS Support of Differentiated Services," Internet Draft (work in progress) draft-ietf-mpls-diff-ext-08.txt, Feb. 2001.

[6] D. O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, "RSVP-TE : Extensions to RSVP for LSP Tunnels," Internet Draft (work in progress) draft-ietf-mpls-rsvp-lsp-tunnel-08.txt, Feb. 2001.

[7] A. Mankin, et al., "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment," RFC 2208, Sep. 1997.

[8] R. Branden et al., "Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification," RFC 2205, Sep. 1997.

[9] J. Wroclaswski, "The Use of RSVP with IETF Integrated Services," RFC 2210, Sep. 1997.

[10] V. Jacobson, K. Nichols, K. Poduri, "An Expected Forwarding PHB," RFC 2598, Jun. 1999.

[11] J. Heinanen, F. Baker, W. Weiss, K. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, Jun. 1999.

[12] J. Ash, Y. Lee, P. Ashwood, Smith, B. Jamoussi, D. Fedyk, D. Skalecki, L. Li, "LSP Modification Using CR-LDP," Internet Draft (work in progress) draft-ietf-mpls-crlsp-modify-03.txt, Mar. 2001.

[13] L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas, "LDP Specification," RFC 3036, Jan. 2001.

[14] B. Jamoussi, "Constraint-Based LSP Setup using LDP," Internet Draft (work in progress) draft-ietf-mpls-cr-ldp-05.txt, Feb. 2001.

[15] R. Neilson, J. Wheeler, F. Reichmeyer, S. Hares, "A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment,"Internet2 Qbone BB Advisory Council, Ver. 0.7, Aug. 1999.

[16] "Ca*net II Differentiated Services Bandwidth Broker System Specification," British Columbia Institute of Technology, Technology Center, Group for Advanced Information Technology, Oct. 1998.

[17] Ji-Young Lim, Ki-Joon Chae, " Differentiated Link based QoS Routing Algorithms for Multimedia Traffic in MPLS Networks," 15th International Conference on Information Networking(ICOIN-15), Japan, pp.587-592, Feb. 2001.

[18] J. Moy, "OSPF Version 2," RFC 2328, Apr. 1998.

[19] A. Shaikh, J. Rexford, K. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," *ACM SIGCOMM*, pp.215-226, Sep. 1999.

임 지 영

e-mail : jylim@ewha.ac.kr

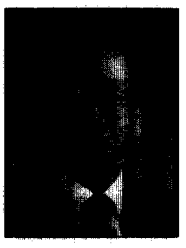
1990년~1994년 이화여자대학교 전자계산학과 이학사

1994년~1996년 이화여자대학교 전자계산학과 이학석사

1996년~2001년 이화여자대학교 컴퓨터학과 공학박사

2001년~현재 이화여자대학교 컴퓨터학과 대우전임강사

관심분야 : MPLS, 트래픽 엔지니어링, DiffServ, QoS 라우팅, 액티브 네트워크



채 기 준

e-mail : kjchae@ewha.ac.kr

1982년 연세대학교 수학과 이학사

1984년 미국 시라큐즈대학교 컴퓨터학과 석사

1990년 미국 노스캐롤라이나 주립대학교 컴퓨터공학과 박사

1992년~현재 이화여자대학교 컴퓨터학과 교수

관심분야 : 네트워크 보안, 액티브 네트워크 보안 및 관리, 인터넷/무선통신망/고속통신망 프로토콜 설계 및 성능 분석