# Message Reversing을 이용한 Constraint-based Routed LSP 설정 방법에 관한 연구 *

†김 병 식○ †전 우 직 ‡유 재 호
† 충남대학교 컴퓨터공학과 ‡ ETRI MPLS 소프트웨어팀
kbs@ce.cnu.ac.kr chun@ce.cnu.ac.kr jaeho@etri.re.kr

## Constraint-based Routed LSP Setup by Message Reversing

†Byeongsik Kim○ †Woojik Chun ‡Jae Ho Yoo
† Department of Computer Engineering, Chungnam National University
‡ MPLS Software Team, ETRI

요 약

최근 대두되고 있는 MPLS 기술은 트래픽 엔지니어링에 적절하다고 알려져 있다. 그러나 MPLS를 기반으로 해서 트래픽 엔지니어링을 이루기 위해서는 특정한 제한 조건을 만족하는 경로를 설정하는 것이 기본적인 사항이다. 이를 위해서는 제한 조건 기반의 라우팅의 도움이 필수적이다. 또한, 명시적인 경로의 설정이 Ingress 라우터에서 이루어져야 한다. 그러나 망의 형상이나 이용 가능한 자원양은 동적으로 변동되는 성질을 갖고 있어서 실제 경로의 계산시 사용한 정보는 실제 경로 계산시의 망의 상태를 정확히 반영하지 못하게 된다. 그러므로 이런 정보를 이용해서 계산된 경로를 따라 LSP 설정 작업이 이루어지는 경우에 LSP 설정 실패 확률이 높아 진다. 본 논문에서는 CR-LDP의 메시지들 중 Ingress 라우터로 전달되는 메시지들을 이용해서 LSP 설정 실패 확률을 줄일 수 있는 방법을 기술한다.

## 1. 서론

MultiProtocol Label Switching(MPLS) 는 기존의 IP를 사용하는 응용들에게 여러 가지 부가적인 서비스를 제공함으로써 인터넷 서비스를 획기적으로 향상시킬 수 있는 새로운 기술이다. 인터넷 서비스 제공자들이 제공하고자 하는 대부분의 서비스들은 향후 트래픽 엔지니어링 기술을 전적으로 이용할 계획을 가지고 있다. MPLS는 특성상 트래픽 엔지니어링을 달성하기 위한 적절한 도구로 알려져 있다 [1]. 현재 MPLS 기반의 망에서 트래픽 엔지니어링을 위한 시그날링 프로토콜로 기존의 RSVP [2]를 확장한 RSVP-TE[3]와 CR-LDP [4]를 표준화하는 작업이 진행중이다.

RSVP-TE나 CR-LDP는 트래픽 엔지니어링을 위해서 Constraint-based 라우팅의 도움이 필수적이다. 그러나 경로를 계산하기 위해서 사용되는 각 라우터의 망의 상태에 대한 정보는 경로를 계산하는 시점에서 최신의 정보가 아니기 때문에 이런 정보를 가지고 계산된 경로를 따라서 LSP가 설정되어 가는 과정에서 실패가 발생할 수도 있다.

본 논문에서는 이런 링크 상태 정보의 부정확성으로 인해서 발생하는 LSP 설정의 실패 확률을 줄이는 방법에 대해서 소개한다. 본 논문에서 제시하는 방법은 LSP를 설정하는 과정에서 설정에 실패하거나 성공하는 경우에 Ingress LSR로 전달되는 CR-LDP의 메시지내에 각 링크의 현재 사용가능한 대역폭 정보를 포함시켜 전달함으로써 Ingress LSR의 상태 정보 테이블에 이 정보를 반영시키자는 Feedback 메커니즘 [5] 과 LSP 설정이 실패하는 경우에 이 실패 위치를 Ingress LSR과 Intermediate LSR들에게 전달해서 이 실패 위치를 제외한 다른 경로의 대체 경로를 계산하는 Crankback 라우팅 [6] 방법을 통합한 방법에 해당한다. 본 논문에서는 이런 새로운 메커니즘을 Ingress LSR로 전달되는 메시지들을 이용하기 때문에 "Message Reversing" 메커니즘이라 한다.

*본 연구는 한국전자통신연구원의 2000년도 "Constraint-based Routing 기능을 가진 LSR의 기능 및 구조에 관한 연구"과제의 지원하에 이루어 졌음.

## 2. 망 상태 정보의 부정확성

제한 조건 기반 라우팅은 망의 형상정보 이외에도 링크들에서 이용할 수 있는 자원의 양을 기반으로 경로를 계산한다. 그러나 링크의 이용가능한 자원량과 같은 상태 정보는 트래픽의 양이 부단히 변화하는 동적인 망에서는 부정확성을 수반하게 된다.

이런 부정확성은 특히 대규모의 망에서 다음과 같은 이유로 현저하게 발생한다. 첫째, 지역적인 상태의 변화가 다른 전체 노드들에게 전파되기 까지는 무시할 수 없는 전달 지연 시간이 발생한다. 둘째, 링크 상태 프로토콜은 주기적으로 혹은 상태의 변화가 현저하게 발생하는 경우에만 상태 정보 테이블을 갱신한다. 셋째, 대규모 망에서 라우팅의 확장성 문제를 해결하기 위해서 계층적인 라우팅 방법을 사용하는 경우에 상태의 애그리게이션으로 인해서 부정확성을 더욱 악화시킬 수 있다. 이 중에서 본 논문에서 관심의 대상이 되는 경우는 두번째 경우이다. 이 경우의 부정확성은 갱신 주기를 자주 하는것은 이런 갱신을 위한 통신 오버헤드를 피할 수 없기 때문에 실제 각 링크의 메트릭과 라우터에 저장되어 있는 상태 정보 테이블의 차이를 극복할 수 없는 문제점이 있다.

제한 조건 기반 라우팅에서 명시적 경로를 계산하기 위해서는 링크 상태 정보 데이터베이스가 라우팅 영역의 모든 노드에 저장되어 있어야 한다. 이 데이터베이스는 트래픽 엔지니어링의 대상이 되는 망의 링크들에 대한 모든 리스트와 각 링크들이 만족할 수 있는 모든 제한 조건들을 유지하고 있어야 한다. 예를 들어 대역폭은 가장 필수적인 제한 조건의 하나이다. 새로운 LSP들이 설정되고 종료될 때 마다 각 링크의 이용

가능한 대역폭의 양은 수시로 변경되므로 링크 상태 데이터베이스는 실제 망에 대해서 부정확성이 점차 증가하게 된다.

일반적으로 링크 상태를 기반으로 하는 라우팅은 다음과 같은 기본적인 두 가지 작업으로 이루어진다. 첫째 작업은 망의 상태 정보를 수집해서 그 정보를 최신의 정보로 유지하는 작업이고 두번째 작업은 수집된 정보를 기반으로 새로운 LSP를 위한 경로를 계산하는 작업이며 이 작업은 첫 번째 작업이 얼마나 잘 이루어졌는지에 의존한다.

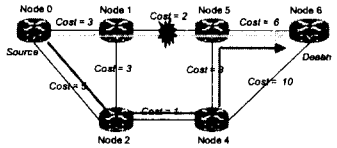


그림 1: 망 형상

OSPF [7] 와 같은 링크 상태 라우팅 프로토콜들은 각 라우터간의 도달 가능성 및 도달하는데 필요한 비용이나 메트릭을 주기적으로 전달한다. 그러나 이런 변경된 정보가 모든 라우터들에게 전달되어서 새로운 경로를 계산하는 동안은 망 상태 정보의 불일치로 전달되는 패킷의 손실을 피할수 없게 된다. 이런 상황을 *ns*-2 [8]를 이용해서 시뮬레이션 하였으며 예제로 사용한 망 형상은 그림 1와 같다.

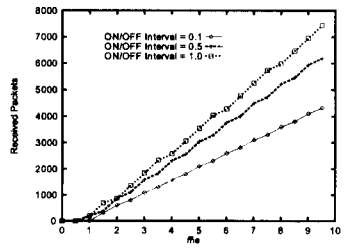


그림 2: 망 형상의 변화에 따른 패킷 손실율

그림 2의 그래프는 그림 1의 망 형상에서 Node 1과 Node 5 사이의 링크를 시간간격 0.1, 0.5, 1.0 으로 각각 ON/OFF 시키는 경우에 동일한 갯수의 패킷을 생성시켜 보았다. 이 경우 망의 형상이 더 자주 변할수록 망의 변화에 대처하기 위해서 재계산하는 동안의 패킷 손실때문에 목적지에 전달되는 패킷의 갯수가 적음을 알 수 있다.

# 3. Message Reversing을 이용한 CR-LSP 설정

## 3.1 Message Reversing 메커니즘

Message Reversing 메커니즘의 목적은 LSP의 설정시 링크 상태 정보의 부정확성을 극복하는데 있다. 또한 이 메커니즘의 기본 개념은 CR-LDP의 시그날링 메시지들중 LabelMapping, Notification, Withdraw 메시지들과 같이 LSP 설정이 성공 혹은 실패 되거나 이미 설정되어 있던 LSP의 실패로 인해서 Ingress LSR 쪽으로 전달되는 메시지들내에 현재의 각 링크에 대한 자원 사용 가능량과 실패가 발생한 위치의 정보를 추가해서 전달하는데 있다. 이런 정보는 최신의 정보이기 때문에 새로운 LSP의 계산을 위해서 유용한 정보로 이용될 수 있으며 새로운 LSP 설정의 성공 확률을 향상시킬 수 있다.

Message Reversing 메커니즘은 다음과 같은 상황에서 수행될 수 있다.

- ER-LSP 설정 요청이 성공했을 때
- ER-LSP 설정 요청이 실패했을 때
- 이미 설정되어있던 ER-LSP가 실패될 때

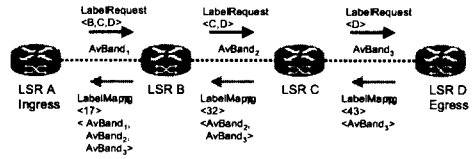


그림 3: LabelMapping 메시지내의 가용 대역폭 정보

현재의 CR-LDP는 LSP 설정에 성공하는 경우에 upstream LSR쪽으로 분배되는 레이블만을 전달한다. 그러나 Message Reversing 메커니즘은 그림 3에서 보는바와 같이 LabelMapping 메시지가 전달되는 각 링크의 현재 이용가능한 대역폭 정보 $<AvBand_1, AvBand_2, AvBand_3>$ 까지도 Ingress LSR 쪽으로 전달한다. 이 최신의 정보를 받은 Ingress LSR은 링크 상태 정보 테이블에 반영해서 다음의 경로를 계산할 때 이용한다. 그러나 이 경우는 LSP 설정이 성공한 경우이므로 LSP 설정이 블록킹 된 위치 정보(LocID)는 포함하지 않는다.

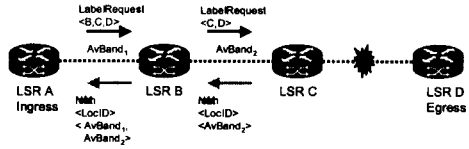


그림 4: Notification 메시지내의 LocID와 가용 대역폭 정보

그림 4에서의 경우처럼, LabelRequest 메시지가 대역폭의 부족이나 링크의 실패로 인해서 블록킹될 때, 생성되는 Notification 메시지내에 추가로 블록킹이 발생한 위치 정보 LocID를 삽입하고 이 메시지가 Ingress LSR로 전달되면서 각 링크의 가용 대역폭 정보를 삽입한다. 이 경우도 위의 경우와 마찬가지로 이 정보를 새로운 경로 계산에 이용한다.

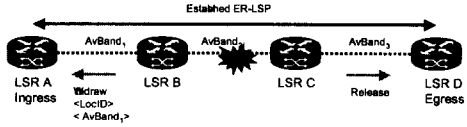


그림 5: Withdraw 메시지내의 LocID와 가용 대역폭 정보

그림 5의 경우는 이미 설정되어 있던 LSP가 중간의 링크에 실패가 발생하는 경우에 발생하는 Withdraw 메시지내에 LocID와 각 링크의 가용 대역폭 정보를 삽입해서 전달하는 예에 해당한다.

위의 세가지 경우에 Ingress LSR은 LabelMapping, Notification, Withdraw 메시지를 받으면 최신의 정보인 LocID와 이 메시지들이 전달되어 온 각 링크의 가용 대역폭 정보를 링크 상태 데이터베이스에 반영한다. 이런 최신의 정보를 가지고 Ingress LSR이 새로운 LSP를 계산해서 설정하는 경우에는 이전에 발생한 실패 확률을 줄일 수 있게 된다.

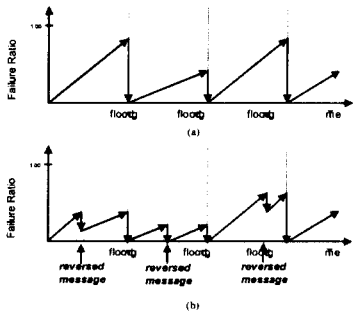


그림 6: ER-LSP 설정 실패율

### 3.2 Flooding System과의 비교

그림 6은 일반적으로 Flooding만 사용하는 시스템과 Flooding과 Message Reversing을 같이 사용하는 시스템에서 LSP 설정의 실패율을 비교한 그래프이다. 그림 6.(a)처럼 Flooding만을 사용하는 경우에 ER-LSP의 실패율이 Ingress LSR이 Flooding 메시지를 받은 순간에만 0 으로 되지만 그림 6.(b)처럼 Message Reversing 까지 사용하는 경우에는 Flooding 메시지가 도달하기 전에도 ER-LSP 설정 실패율을 줄일 수 있음을 보여준다.

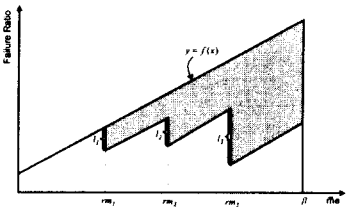


그림 7: Flooding과 Message Reversing System의 ER-LSP 설정 실패율

그림 7은 그림 6의 두 그래프를 하나로 합쳐놓은 그래프이다. 이 그래프에서 상위 곡선은 LSP 설정 요청의 실패율이 다음 Flooding 메시지를 받을 때까지 선형적으로 증가한다는 가정하에 주어진 것이고 하위 곡선은 Message Reversing 메커니즘을 사용하는 경우의 LSP 설정 요청의 실패율을 나타낸 것이다. 이 예에서 Flooding 메시지를 받기전에 시간 $rm_1$, $rm_2$, $rm_3$에서 Message Reversing에 의해서 세개의 메시지를 받고 시간 $fl$에서 Flooding 메시지를 받는다고 가정하면, Flooding 메커니즘만을 사용하는 경우의 LSP 설정의 평균 실패 확률은

$$Prob_{fail}(fl) = \int_0^{fl} f(x)dx \qquad (1)$$

가 된다. 그러나 Message Reversing이 사용되는 경우에는 LSP 설정의 평균 실패 확률은

$$\begin{aligned} Prob_{fail}(mr) &= \int_0^{rm_1} f(x)dx \\ &+ \int_{rm_1}^{rm_2} (f(x) - l_1)dx \\ &+ \int_{rm_2}^{rm_3} (f(x) - l_1 - l_2)dx \\ &+ \int_{rm_3}^{fl} (f(x) - l_1 - l_2 - l_3)dx \qquad (2) \end{aligned}$$

가 된다. 이때, $l_1, l_2, l_3$는 각각 $rm_1$, $rm_2$, $rm_3$ 시간에 Message Reversing에 의해서 Ingress LSR로 전달된 링크들의 가용 대역폭 정보의 갯수에 해당한다. 이 값들이 클수록 링트 상태 정보 데이터베이스의 정확성이 증가하고 다음에 설정될 LSP의 성공 확률이 증가한다. 그림 7의 그늘진 부분 즉, $Prob_{fail}(fl) - Prob_{fail}(mr)$부분이 Message Reversing 메커니즘을 사용함으로써 얻을 수 있는 이득에 해당한다.

## 4. 결론 및 향후 연구

본 논문에서는 망의 가용 자원 정보의 부정확성이 LSP 설정에 미치는 영향에 대해서 알아보았으며 이에 대처할 수 있는 LSP 설정 방법에 대해서 기술하였다. 본 논문에서 제시하는 Message Reversing 방법은 LSP를 설정하는 과정에서 실패 혹은 성공하였을때와 기존에 이미 설정되어 있던 LSP가 실패하는 경우에 생성되는 메시지들내에 실패가 발생한 위치에 대한 정보를 삽입하고 이 메시지가 Ingress LSR로 전달되는 과정에서 각 링크의 최신 대역폭 정보를 삽입하게 한다. 이런 정보들을 받은 Ingress LSR은 Flooding 메시지가 도착하기 전에 전체 망의 일부분에 대해서는 최신의 정보를 알 수 있으므로 다음 LSP 설정시에는 단지 Flooding 만을 사용하는 방법에 비해서 LSP 설정 실패 확률을 줄일 수 있다. 또한 실패가 발생한 위치 정보를 이용해서 이 위치를 제외한 새로운 경로 계산에 이용할 수 있으므로 LSP의 복구 및 대체 LSP 설정에도 적용이 가능하다.

향후 연구 과제로는 실질적인 LSP 설정의 실패에 대한 확률 분포 함수를 도출하는데 있다. 현재 본 연구실에서 개발한 MPLS Network Simulator인 *mns* [9]로 Message Reversing 메커니즘을 구현중에 있다.

## 참고 문헌

[1] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. *Requirements for Traffic Engineering Over MPLS.* RFC2702, September 1999.

[2] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. *Resource ReSerVation Protocol(RSVP) - Version 1 Functional Specification.* RFC2205, September 1997.

[3] D. Awduche, L.Berger, D. Gan, T. Li, G. Swallow, and V. Srinivasan. *RSVP-TE: Extensions to RSVP for LSP Tunnels.* draft-ietf-mpls-rsvp-lsp-tunnel-06.txt, July 2000.

[4] B. Jamoussi. *Constraint-Based LSP Setup using LDP.* draft-ietf-mpls-cr-ldp-04.txt, July 2000.

[5] P. Ashwood-Smith, B. Jamoussi, D. Fedyk, and D. Skalecki. *Improving Topology Data Base Accuracy with LSP Feedback via CR-LDP.* draft-ietf-mpls-te-feed-00.txt, February 2000.

[6] A. Iwata, N. Fujita, and G. Ash. *Crankback Routing Extensions for CR-LDP.* draft-fujita-mpls-crldp-crankback-01.txt, July 2000.

[7] J. Moy. *OSPF Version 2.* RFC2328, April 1998.

[8] `http://www-mash.cs.berkeley.edu/ns/`. *UCB/LBNL/VINT Network Simulator - ns (Version 2).*

[9] `http://www.raonet.com/mns/`. *MPLS Network Simulator (mns).*