

MPLS 네트워크 시뮬레이터에 의한 경로 복원 기법의 시뮬레이션 및 성능 평가

안개일^o 전우직
충남대학교 컴퓨터공학과
{fog1, chun}@ce.cnu.ac.kr

Simulation and Performance Evaluation of Path Restoration Mechanism by using MPLS Network Simulator

Gae-il Ahn^o Woo-Jik Chun
Dept. of Computer Engineering, Chungnam University

요 약

본 논문에서는 MPLS 네트워크 시뮬레이터를 이용하여 기존의 경로 복원 기법들을 시뮬레이션 하고 각각의 성능을 비교 분석한다. MPLS 네트워크 시뮬레이터는 레이블 스위칭 기능 및 LDP 와 CR-LDP를 지원하는 시뮬레이터이다. 시뮬레이션할 경로 복원 기법으로써 가장 많이 참조 되고 있는 Haskin에 의해 제안된 기법과 Makam에 의해서 제안된 기법을 선택한다. 시뮬레이션 결과, Haskin의 기법은 노드 에러 시에도 패킷의 손실은 거의 없지만 노드가 복구된 후에는 트래픽이 급격히 증가할 뿐만 아니라 순서가 맞지 않는 패킷의 양도 증가하는 문제가 있다. Makam의 기법은 노드 에러 시에 패킷의 손실이 큰 단점을 가지고 있다.

1. 서론

현재, 인터넷의 폭발적인 증가와 다양한 응용 서비스의 등장으로 인하여 기존의 인터넷 방식에 획기적인 변화가 요구되고 있다. 이런 변화를 수용할 수 있는 기술 중에서 가장 각광 받고 있는 기술이 MPLS(Multi-Protocol Label Switching)[3][5]이다.

본 논문에서는 Network Simulator(NS)[7] 라는 시뮬레이터 상에서 구현된 MPLS 네트워크 시뮬레이터[9]를 사용하여 MPLS에서 중요한 응용 중에 하나인 경로 복원 기법들을 시뮬레이션 한다. MPLS 네트워크 시뮬레이터는 레이블 스위칭 기능 및 LDP(Label Distribution Protocol)[4] 와 CR-LDP(Constraint based Routing-LDP)[1]를 지원하는 시뮬레이터이다. 경로 복원이란 MPLS 망에서 보호해야 할 트래픽의 노드나 링크가 에러이거나 혼잡이 발생했을 때 대체 경로(backup path)로 그 트래픽을 우회 시켜 신뢰성 있는 서비스를 제공하는 기술이다.

본 논문에서는 시뮬레이션할 경로 복원 기법으로써 가장 많이 참조되고 있는 Haskin에 의해 제안된 기법[2]과 Makam에 의해서 제안된 기법[6]을 선택한다. 각각의 기법들에 대한 시뮬레이션 방법을 자세히 서술하고, 그 시뮬레이션 결과를 통하여 각각의 기법들에 대한 성능을 비교 분석한다.

본 논문의 차례는 다음과 같다. 2장에서는 MPLS 네트워크 시뮬레이터에 대하여 간단히 소개하고, 3장에서는 경로복원 기법들을 알아본다. 4장에서는 기존에 제안된 각각의 기법들을 시뮬레이션한 후 5장에서 성능을 평가하고, 6장에서 결론을 맺는다.

2. MPLS 네트워크 시뮬레이터

MPLS 네트워크 시뮬레이터는 다음의 기능들이 구현 되어 있다.

- 레이블 스위칭 기능 -- 레이블 스왑핑 기능, TTL(Time-To-Live), 그리고 Penultimate hop popping
- LDP 메시지 전송 기능 -- Request, Mapping, Withdraw, Release, Notification 메시지 전송
- CR-LDP 메시지 전송 기능

MPLS 네트워크 시뮬레이터에서 다음과 같은 LSP 설정에 관련된 기능을 지원한다.

- LDP에서 정의되어 있는 다양한 LSP 설정 옵션 즉, LSP 트리거 방식, 레이블 할당 방식, 레이블 분배 제어, 레이블 보유 모드에서의 설정 옵션들을 지원한다.
- CR-LDP에 기반한 ER-LSP 설정
- Flow Aggregation 기능 -- fine FEC 를 coarse FEC 에 aggregation할 수 있는 기능 지원

3. 경로 복원 기법

본 논문에서는 기존에 제안된 경로 복원 기법들 중 가장 많이 참조 되고 있는 Makam에 의해 제안된 기법 과 Haskin에 의해 제안된 기법을 소개한다.

기본 경로(working path)는 보호할 트래픽에 설정된 경로를 말한다. 그림 1에서 LSR1과 LSR11사이의 실선이 기본 경로이다. 기본 경로는 대체 경로에 의하여 보호 받는다.

Makam 기법에서 대체 경로는 다음과 같이 설정된다.

- 기본 경로를 거치지 않도록 하면서 대체 경로를 설정한다. 그림 1에서 LSR1-2-4-6-8-10-11이 대체 경로이다.
- 노드 에러가 발생했을 때 그 에러를 탐지한 노드가 대체 경

로가 설정된 노드까지 여러 알림(notification) 메시지를 전달하여 트래픽이 대체 경로로 전송되도록 한다. 그림 1에서 LSR7이 에러일 때 점선은 에러 알림 메시지의 경로를 나타낸다.

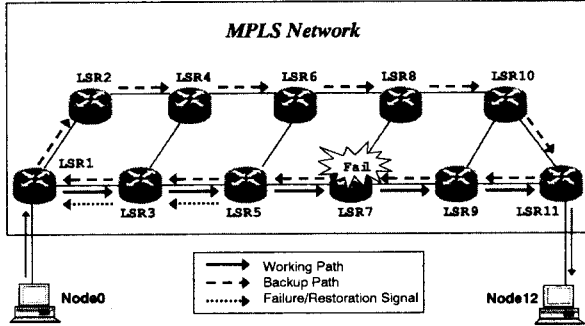


그림 1: Makam 과 Haskin 의 경로 복원 기법들

Haskin 기법에서 대체 경로는 다음과 같이 설정된다.

- 기본 경로와 중복되지 않도록 하면서 대체경로의 일부를 설정한다. 그림 1에서 LSR1-2-4-6-8-10-11이 대체 경로의 일부이다.
- 기본 경로와 반대방향으로 대체 경로의 일부를 설정한다. 그림 1에서 LSR11-9-7-5-3-1이 대체 경로의 일부이다.
- 두 대체 경로를 일부를 연결하여 하나의 완벽한 대체 경로를 만든다.
- 에러가 탐지되면 대체 경로로 트래픽을 보낸다. 예를 들어 그림 1에서 LSR7이 에러일 때 대체경로는 LSR5-3-1-2-4-6-8-10-11이 된다.

4. 경로 복원 기법 시뮬레이션

시뮬레이션 환경은 그림 1과 같다. 각 링크는 1Mbps의 대역폭과 10ms의 지연, 그리고 DropTail 큐를 가지고 있다. Node0에서는 200byte 길이의 패킷을 500Kbps인 CBR(Constant Bit Rate)로 생성한다.

```
# make nodes
set node0 [$sns node]
set LSR1 [$sns MPLSnode]
.....
# make links
$sns duplex-link $node0 $LSR1 1Mb \
    10ms DropTail
.....
# configure ldp agents on all mpls nodes
$sns configure-ldp-on-all-mpls-nodes
# in Haskin's mechanism
$sns enable-reroute drop
# in Makam:enable-reroute notify
```

그림 2: 그림 1의 환경 구축을 위한 코드

그림 2는 그림 1의 시뮬레이션 환경을 구축하기 위하여 Tcl 언어로 작성된 코드이다. 그림 2에서 'MPLSnode'는 새로운 MPLS 노드를 생성하기 위해 사용되며, 'configure-ldp-on-all-mpls-nodes'는 생성된 모든 MPLS 노드에 LDP 에이전트를 탑재하기 위해 사용된다. 'enable-reroute'는 에러를 탐지한 노드가 해야 할 행동을 명시하기 위해 사용되며, 'set-

protection-lsp'는 보호해야 할 트래픽 및 에러를 탐지하는 시간 간격을 명시하기 위해 사용된다. 그림 2에서 보호해야 할 트래픽은 LSPID가 2000인 LSP이며, 0.01초마다 에러가 발생했는지를 조사한다. 'notify-er-lsp-fail'은 콜백(callback) 함수로써 노드가 에러일 때 호출된다.

그림 3-(a)는 Haskin 기법에 대한 이벤트 스케줄링 코드이다. 0.0초에 ER인 LSR3-5-7-9-11을 따라서 LSPID가 2000인 ER-LSP가 생성된다. 이것은 기본 경로로써 사용된다. 0.0초에 또한 LSPID가 1000인 ER-LSP가 생성되고, 0.1초에 LSPID가 1005인 ER-LSP가 생성된다. 이 두 ER-LSP는 대체 경로로 사용되며, 두 LSP를 결합하기 위하여 LSPID가 1005인 ER-LSP를 설정할 때 LSPID가 ER 홈으로써 사용되었다. 0.3초에 Node0는 패킷을 생성하기 시작하고, 그 트래픽은 LSPID가 2000인 ER-LSP로 바인딩 된다. 0.4초에 LSPID가 2000인 ER-LSP를 보호한다. 0.5초부터 0.7초 LSR5와 LSR7사이 링크에러가 발생한다. 본 논문에서는 이 링크에러는 노드에러로 간주 하겠다.

그림 3-(b)는 Makam 기법을 위한 이벤트 스케줄링 코드이다.

```
$sns at 0.0 "$LSR1 make-explicit-route 11 3_5_7_9_11 2000 -1"
$sns at 0.0 "$LSR1 make-explicit-route 11 2_4_6_8_10_11 1000 -1"
$sns at 0.1 "$LSR11 make-explicit-route 11 9_7_5_3_1_1000 1005 -1"
$sns at 0.3 "$src0 start"
$sns at 0.3 "$LSR1 flow-erlsp-install 12 -1 2000"
$sns at 0.4 "$LSR1 reroute-lsp-binding 2000 1000"
$sns at 0.4 "$LSR3 reroute-lsp-binding 2000 1005"
$sns at 0.4 "$LSR5 reroute-lsp-binding 2000 1005"
$sns at 0.4 "$LSR7 reroute-lsp-binding 2000 1005"
$sns rtmodel-at 0.5 down $LSR7 $LSR9
$sns rtmodel-at 0.7 up $LSR7 $LSR9
$sns at 0.9 "$src0 stop"
```

(a) Haskin 기법

```
$sns at 0.0 "$LSR1 make-explicit-route 11 3_5_7_9_11 2000 -1"
$sns at 0.0 "$LSR1 make-explicit-route 11 2_4_6_8_10_11 1000 -1"
$sns at 0.1 "$LSR11 make-explicit-route 11 9_7_5_3_1_1000 1005 -1"
$sns at 0.3 "$src0 start"
$sns at 0.3 "$LSR1 flow-erlsp-install 12 -1 2000"
$sns at 0.4 "$LSR1 reroute-lsp-binding 2000 1000"
$sns rtmodel-at 0.5 down $LSR7 $LSR9
$sns rtmodel-at 0.7 up $LSR7 $LSR9
$sns at 0.9 "$src0 stop"
```

(b) Makam 기법

그림 3: 이벤트 스케줄링 코드

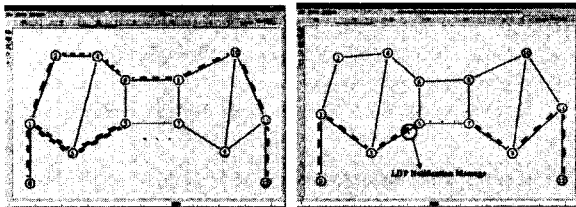
5. 시뮬레이션 결과 및 분석

MPLS 네트워크 시뮬레이터를 통하여 생성된 결과를 Network Animator(NAM)[8]으로 실행 시키면 그림 4와 같은 결과를 볼 수 있다. 그림 4에서 보듯이 Haskin의 기법은 에러를 탐지한 LSR5에서 즉시 트래픽이 대체 경로로 우회되지만, Makam 기법은 LSR1을 향하여 Notification 메시지를 보내고 있는 동안 트래픽들은 보호 받고 있지 못하다.

그림 5는 패킷 손실 및 순서가 어긋난 패킷 수에 따라서 두 기법의 성능을 보여주고 있다. Haskin 기법은 에러가 탐지됨과 거의 동시에 트래픽을 대체 경로로 전달할 수 있으므로 데이터 손실이 거의 없지만, 에러를 발생하는 노드가 목적지에 가까울수록 순서가 맞지 않는 패킷의 양도 함께 증가하는 문제가 있

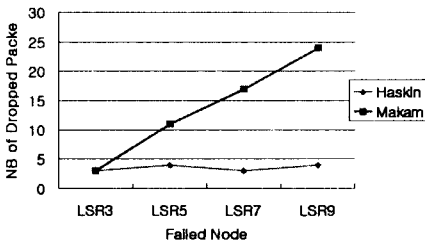
다. Makam 기법은 순서가 맞지 않는 패킷의 양은 거의 없지만, 에러 알림 메시지를 보내는 시간이 길리기 때문에 에러를 발생하는 노드가 복귀지에 가까울 수록 손실되는 패킷의 수는 비례하여 증가 한다.

그림 6 은 에러인 노드의 위치별 대역폭의 변화를 보여주는 그림이다. 노드 에러가 발생한 시점인 0.5 초부터 약 0.2 초 동안은 두 기법은 대역폭 변화는 비슷하지만, 노드가 복구된 시점인 약 0.7 초부터는 Haskin의 기법의 대역폭이 크게 늘어난다. 이것은 곧 패킷이 거의 손실되지 않는다는 것을 의미한다. 그러나 QoS를 고려한 경우라면, 즉 500Kbit 만을 예약한 경우에 500Kbit 이상인 대역폭은 모두 삭제될 것이다. 그래서 그 경우에는 두 기법은 거의 성능에서 차이가 나지 않을 것이다.

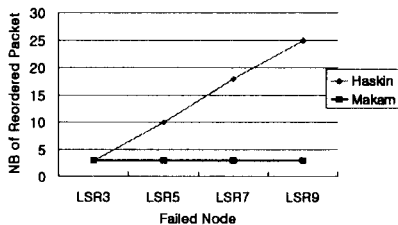


(a) Haskin 기법 (b) Makam 기법

그림 4: NAM에 의해 출력된 결과

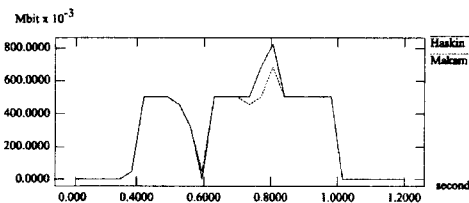


(a) 패킷 손실 비교

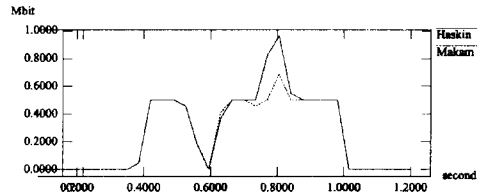


(b) 순서가 어긋난 패킷 수 비교

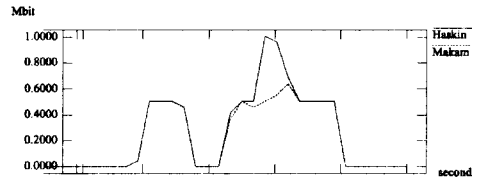
그림 5: 패킷 손실 및 순서가 어긋난 패킷수 에서 비교



(a) LSR5에서 에러



(a) LSR7에서 에러



(a) LSR9에서 에러

그림 6: 에러인 노드의 위치별 대역폭 비교

6. 결론 및 향후 연구 과제

본 논문에서는 MPLS 네트워크 시뮬레이터를 사용하여 기존에 제안된 경로 복원 기법들에 대한 시뮬레이션 방법을 기술하였고 각각의 기법들에 대한 성능을 비교 분석 하였다.

시뮬레이션 결과, Haskin의 기법은 노드 에러 시에도 패킷의 손실은 거의 없지만 노드가 복구된 후에는 트래픽이 급격히 증가할 뿐만 아니라 순서가 맞지 않는 패킷의 양도 증가하는 문제가 있다. Makam의 기법은 노드 에러 시에 패킷의 손실이 큰 단점을 가지고 있다.

향후 연구 과제로써는 실제로 운영되고 있는 대규모 MPLS 망에서 ER-LSP가 아닌 CR-LSP를 사용하여 각각의 기법들을 시뮬레이션하는 것이다.

7. 참고 문헌

- [1] Bilel Jamoussi, "Constraint -Based LSP Setup using LDP," Internet Draft, Oct. 1999.
- [2] D. Haskin, R. Krishnan, "A Method for Setting an Alternative Label Switched Paths to Handle Fast Reroute," Internet Draft, May 2000.
- [3] Eric C. Rosen, Arun Viswanathan, Ross Callon, "Multiprotocol Label Switching Architecture," Internet Draft, April 1999.
- [4] Loa Andersson et al., "LDP Specification," Internet Draft, June 1999.
- [5] R. Callon et al., "A Framework for Multiprotocol Label Switching," Internet Draft, Sep. 1999.
- [6] Srinivas Makam, Ken Owens, Vishal Sharma, Changcheng Huang, "A Path Protection/Restoration Mechanism for MPLS Networks," Internet Draft. Oct. 1999.
- [7] UCB/LBNL/VINT, "ns Notes and Documentation," July 1998, URL: <http://www-mash.cs.berkeley.edu/ns>.
- [8] UCB/LBNL/VINT, "Network Animator (nam)," URL: <http://www-mash.cs.berkeley.edu/nam>.
- [9] 안개일, 천우적, "MPLS Network Simulator," <http://www.raonet.com>, 2000.