

대역을 보장하는 망 복구 대체 경로 설정을 위한 방안의 분석

홍석원*, 이황규*, 전병천**, 윤현정**

*명지대학교 전자정보통신공학부

**전자통신연구원

e-mail : swhong@wh.myongji.ac.kr

A Study on network restoration path setup and bandwidth guarantee

Sugwon Hong*, Hwang-Kyu Lee*, Byung-Chun Jeon**, HyunJeong Yun**

*Dev. of Electrical and Information and Comm. Engineering, Myongji University

**Switching & Transmission Lab., Electronic and Telecommunications Research Institute

요 약

망 복구에 있어서 주요한 과제는 복구의 대상으로 하고 있는 작업 경로(working path)에 대한 복구 경로(restoration path) 혹은 대체 경로(backup path)를 설정하는 것이다. 대체 경로를 설정할 때 작업 경로의 대역을 보장할 수 있는 대체 경로를 설정해야 한다. 따라서 이 과제는 대역의 보장을 고려하여 경로를 설정하는 QoS 라우팅의 해결책이 요구된다. 본 논문에서는 동일 링크의 작업 경로가 아닐 경우 대체 경로의 대역을 공유함으로써 최대한 대역의 소비를 피하면서, 작업 경로의 대역을 보장하는 대체 경로를 설정하는 disjoint 경로를 계산하는 알고리즘을 제시하고 시뮬레이션을 통해서 대체 경로로 인한 대역의 소비(bandwidth consumption) 효과를 분석하였다.

1. 서론

최근 인터넷에서 트래픽 엔지니어링을 제공하는 방안은 주요한 과제로 연구되어 왔다[1,2]. 트래픽 엔지니어링을 제공하는 한 방안으로 관심을 모으고 있는 것은 MPLS의 constraint-based routing을 위한 신호 방식을 사용하는 것이다[3,4]. 이러한 라우팅 방식은 QoS 라우팅, 그리고 명시적 라우팅(explicit routing)을 사용한 트래픽의 분산(load balancing)과 망 복구를 위한 대체 경로의 설정에 활용될 수 있다.

망 복구에 있어서 주요한 과제는 복구의 대상으로 하고 있는 작업 경로(working path)에 대한 복구 경로(restoration path) 혹은 대체 경로(backup path)를 설정하는 것이다. 대체 경로를 설정할 때 원래의 작업 경로의 서비스 요구 조건을 충족시키기 위해서는 작업 경로에서 보장한 대역을 역시 보장할 수 있는 대체 경로를 설정해야 한다. 따라서 이 과제는 대역의 보장을 고려하여 경로를 설정하는 QoS 라우팅의 해결책이 요구된다.

대역을 보장하는 대체 경로를 설정할 때 모든 작업 경로에 대한 대체 경로의 대역을 별도로 할당한다면 대체 경로의 설정으로 인한 대역의 낭비가 심해지게 된다. 따라서 대체 경로의 설정에 있어서 작업 경로의 대역을 공유하여 경로를 설정하게 되면 위의 경우 보다 훨씬 대역의 낭비를 감소시킬 수 있다[5].

본 논문에서는 작업 경로를 설정하면서 동시에 망 복구를 위한 대체 경로를 설정할 때 다음과 같은 점을 고려한 알고리즘을 제시하였다.

첫째로 작업 경로의 대역을 보장할 수 있는 대체 경로를 설정한다. 이때 동일한 링크를 공유하는 작업 경로에 대해서는 대체 경로의 대역은 공유되지 않는다. 하지만 동일한 링크를 공유하지 않는 모든 작업 경로에 대한 대체 경로는 대역을 공유하도록 하여 대역의 낭비를 최소한으로 하도록 하였다.

두번째로 작업 경로와 대체 경로는 동일한 시작 노드에서 시작하여 동일한 종단 노드에서 경로가 끝나며 경로 상에서는 어떠한 링크도 서로 공유하지 않

도록 하였다. 이것은 링크의 장애가 발생하였을 때 안전한 대체 경로를 보장하기 위한 필요 조건이라고 할 수 있다.

다음 절에서는 이와 같은 두 가지 조건을 만족하는 경로 설정 알고리즘에 대해서 설명하도록 한다. 그리고 3 절에서는 이 알고리즘을 이용하여 대체 경로 설정에 의해 사용되는 대역을 임의의 망에 대해서 시뮬레이션을 통해서 보여 주었다. 그리고 이 결과를 대체 경로를 사용하지 않을 경우와 비교하여 대역의 소비가 얼마나 발생하는지를 보여 주었다.

2. Disjoint 경로 계산 알고리즘

작업 경로(working path)와 링크 장애가 발생할 경우 이를 대체할 수 있는 복구 경로 혹은 대체 경로는 작업 경로의 대역을 보장할 수 있는 대체 경로로서 작업 경로와 대체 경로는 시작 노드에서 종점 노드까지 두 경로 간에는 링크를 전혀 공유하지 않는 경로가 된다. 이러한 대체 경로를 설정할 때 대체 경로의 대역은 동일한 링크를 경유하는 작업 경로가 아닌 경우 대체 경로의 링크의 대역을 공유하게 된다.

이를 위해서는 모든 노드는 현재 링크 상에서 각각의 작업 경로가 차지하고 있는 대역, 각각의 대체 경로가 차지하고 있는 대역, 어떤 작업 경로에 대한 대체 경로인지, 그리고 각 링크에서 작업 경로와 대체 경로에 할당하고 남아 있는 대역의 정보를 필요로 한다.

이러한 조건을 만족하면서 작업 경로와 대체 경로의 disjoint 경로를 계산하는 알고리즘은 다음과 같다. 먼저, Dijkstra의 알고리즘을 이용해서 작업 경로를 구한다. 각 링크에서 필요한 대역은 작업 경로의 요구 대역이 된다. 대체 경로는 일단 작업 경로의 링크를 모두 제외한 링크들 가운데서 Dijkstra 알고리즘을 이용해서 구한다.

대체 경로에 대한 각 링크의 요구 대역은 현재 링크에서 공유된 양에 따라서 달라지게 된다. 만약 요구 대역이 대체 경로 상에 이미 설정되어 있는 대체 경로의 대역보다 작다면 다른 대체 경로와 대역을 공유하기 때문에 더 이상 대역을 할당할 필요가 없다. 요구 대역이 대체 경로 상에 이미 설정되어 있는 대체 경로의 대역보다 크다면, 요구 대역에서 이미 설정되어 있는 대체 경로의 대역을 뺀 만큼의 대역이 필요하다.

어떤 작업 경로에 대한 대체 경로의 대역 설정 시 동일한 링크를 지나가고 있는 다른 작업 경로에 대한 대체 경로가 같은 대체 경로 상의 링크에 존재한다면, 두 대체 경로의 대역은 서로 공유하지 않는다. 만약 이러한 두 개 이상의 대체 경로가 서로 대역을 공유할 경우 두 대체 경로가 동시에 지나가는 링크에 장애가 발생한다면, 원래의 작업 경로에 대한 대역을 보장할 수 없기 때문이다. 식(1),(2)와 (3)은 이러한 대역 할당의 관계를 보여 주고 있다.

$$a_{ij} : \text{link}(i,j) \text{ 상에서 작업 경로의 필요 대역}$$

$$b_{uv} : \text{link}(u,v) \text{ 상에서 대체 경로의 필요 대역}$$

m_{ij} : 논리 채널 (i,j)에 대한 작업 경로의 요구 대역
 n_{uv} : 링크(u,v)에서 대체 경로의 요구 대역
 R_{ij} : link(i,j) 상에 남아 있는 대역
 G_{uv} : link(u,v) 상에 설정되어 있는 대체 경로들 대역의 합

$$n_{uv} = \begin{cases} m_{lm} \\ \sum m_{lm} \quad \text{if } (l,m) \in \text{link}(i,j) \end{cases} \quad (1)$$

$$a_{ij} = \begin{cases} m_{lm} \quad \text{if } R_{ij} > m_{lm} \\ \infty \quad \text{if } R_{ij} < m_{lm} \end{cases} \quad (2)$$

$$b_{uv} = \begin{cases} 0 & \text{if } n_{uv} < G_{uv} \\ n_{uv} - G_{uv} & \text{if } n_{uv} > G_{uv} \text{ and } R_{uv} \geq n_{uv} - G_{uv} \\ \infty & \text{if } R_{uv} \leq n_{uv} - G_{uv} \end{cases} \quad (3)$$

3. 시뮬레이션 분석

먼저 네트워크 토폴로지를 구현하기 위해서 노드와 링크에 대한 구조를 구현하였다. 각 노드들은 연결 노드와 연결된 링크에 대한 정보를 가지고 있다. 각 링크는 연결된 두 노드에 대한 정보, 남은 대역 그리고 대체 경로의 대역에 대한 정보를 가지고 있다. 그리고 경로 설정을 위한 구조에는 요구대역, 지나가는 노드들에 대한 정보를 기록하고 있다. 경로를 설정할 때의 비용(cost)은 각 링크 마다 1/(남은 대역)으로 하였다. 구현 순서는 다음과 같다.

- 1 단계 : 네트워크 초기화
 망 토폴로지를 초기화 하고, 경로의 시작점(source), 종점(destination), 그리고 요구 대역을 랜덤(random) 함수로 발생시킨다.
- 2 단계 : 작업 경로 설정
 현재 링크에 할당된 비용을 고려하여 최단 경로(shortest path)를 구한다. 이 때, 각 링크에서 요구 대역 만큼의 남은 대역이 충분하지 확인한다. 남은 대역이 충분하지 않으면 NULL 을 리턴(return)하고 설정 불가능(blocking)이 발생 했음을 알린다. 남은 대역이 충분하면 남은 대역을 요구 대역 만큼 줄이고, 작업 경로를 설정한다.
- 3 단계 : 대체 경로 설정
 작업 경로와 만나지 않는 경로 설정을 위해서 작업 경로가 지나가는 링크를 배제하고 다시 최단 경로를 구한다. 각각의 링크를 지날 때마다 그 링크를 지나가는 최단 경로의 대체 경로들이 현재 구한 최단 경로의 링크들을 지나가는지 확인한다. 만약 지나가는 대체 경로들이 존재한다면 현재 설정 중인 대체 경로의 요구 대역은 각 대체 경로들의 요구 대역의 총 합이 된다. 식(3)에 따라서 대역을 결정하고 필요 대역을 구한다.
 각 링크 상에 남은 대역이 필요 대역보다 작으면

대체 경로를 구현 할 수 없으므로 지금까지 설정했던 대체 경로를 지우고 NULL을 리턴한다.

망 보호를 위해서는 작업 경로와 대체 경로 두개의 경로를 설정해야 한다. 두개의 경로를 설정 하는 데는 두 가지 방법이 있다. 첫 번째 방법은 작업 경로를 가장 짧은 경로로 설정하고 disjoint를 위해 작업 경로 상의 링크를 제외하고 가장 짧은 경로를 대체 경로로 설정하는 방법이다. 두 번째 방법은 작업 경로와 대체 경로의 홉 수가 가장 작은 것을 구하는 것이다[6]. 그러나 대체 경로는 장애가 발생한 경우가 아니면 사용하는 일이 없고, 대역의 공유를 통해서 대체 경로의 대역을 충분히 줄일 수 있기 때문에 여기서는 첫 번째 방법에 의해서 두 개의 disjoint 경로를 계산하였다.

이 논문에서 구현한 토폴로지 상에서 홉 수를 비교해 보면 작업 경로의 평균 홉 수가 3 이고 대체 경로의 평균 홉 수가 4 이므로 두개를 더한 다 해도 두 번째 방법에 의한 효과는 거의 없다는 것을 알 수 있다.

4. 결과

본 논문에서는 그림 1 과 같은 27 개의 노드와 37 개의 양방향 링크로 구성된 망 토폴로지에 대해서 시뮬레이션을 수행하였다.

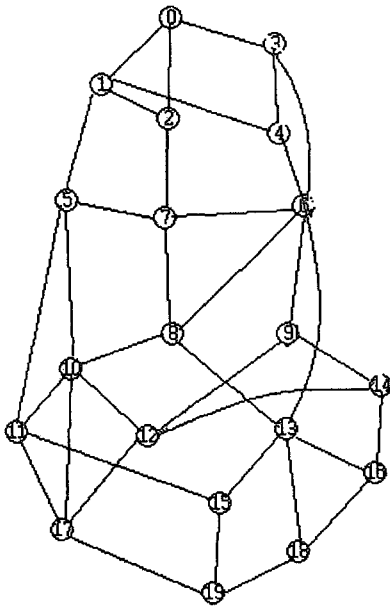


그림 1: 망 토폴로지

시뮬레이션은 다음 3 가지 경우에 대해서 수행하였다. 첫째로 대역 소비 정도를 보기 위해서 각 링크에 대한 대역을 무한대로 하고 작업 경로에 대한 요구를 발생시켰다. 대역이 무한대 임으로 경로 설정 거부(blocking)는 일어나지 않는다.

그림 2 는 작업 경로와 대체 경로의 총 요구 대역과 대체 경로가 없는 경우 작업 경로의 요구 대역을 경로 설정 요구의 수에 대해서 보여 주고 있다. 토폴로지의 각 링크의 대역은 100 에서 300 까지 랜덤하게 할당하였다.

그림 2 에서 보는 바와 같이 한번 경로 설정에 대해서 1 에서 10 크기의 대역을 요구하면 200 번 설정 요구 시에 2680 의 작업 경로 대역과 1206 의 대체 경로 대역이 요구 된다. 시뮬레이션의 결과 작업 경로와 대체 경로의 홉 수(hop count)의 비는 약 3:4 정도이다. 만약 공유를 하지 않는다면 대체 경로의 필요 대역은 작업 경로의 필요 대역과 같아야 한다. 따라서 홉 수를 고려 할 때 대체 경로의 대역 소비는 작업 경로 보다 많은 대역을 소비해야 한다. 그러나 공유를 할 경우 대체 경로의 대역 소비가 작업 경로의 대역 소비에 비해서 훨씬 적은 것을 그림 2 를 통해서 알 수 있다.

두 번째로 경로 설정 시 거부(blocking)의 효과를 보기 위해서 토폴로지의 각 링크에 대해서 유한 대역을 할당하였다. 그림 3 에서는 각 링크의 대역을 200 으로 균등하게 할당하였으며 그림 4 에서는 각 링크의 대역을 100 에서 300 까지 랜덤하게 할당하였다. 그림 3 과 그림 4 에서의 대역 총 합은 같다.

그림 3 에서 보는 바와 같이 대체 경로를 같이 설정 할 경우 설정 요구의 수가 동시에 500 번이 되면 대역 부족으로 인한 설정 거부가 발생하는 반면 대체 경로 없이 작업 경로 만을 설정하는 경우 동시에 600 번의 설정 요구가 있을 경우 설정 거부가 발생하게 된다. 이것은 망 복구를 위해 작업 경로 만큼의 대역을 보장하기 위한 대체 경로를 설정하기 때문에 먼저 거부 현상이 발생하는 것이다

그림 4 의 경우 각 링크의 대역이 일정하지 않은 경우이다. 이 경우 설정 거부 현상이 발생하는 것은 그림 3 의 경우에 비해서 빨리 발생하는 것을 볼 수 있다. 대체 경로 설정의 경우 경로 설정 요구의 수가 450 번이 되면 설정 거부가 발생하여 그림 3 의 500 번 보다 평균 50 번 설정 요구 전에 거부가 발생하는 것을 볼 수 있다. 대체 경로를 설정하지 않는 경우에도 약 600 번 조금 전에 설정 거부가 발생하는 것을 알 수 있다.

이것은 링크의 대역이 균등할 경우 트래픽 부하 분배(Load balancing)가 되어 여유 대역이 전체적으로 모든 링크에 비슷하게 되기 때문에 랜덤한 경우에 비해서 설정 거부의 가능성이 감소하기 때문이다. 만약 링크의 대역이 랜덤하다면 홉 수가 많아지므로 대역의 소비가 더 많고 여유 대역이 적은 링크를 더 많이 포함하게 되므로 로드 분배(load balancing)가 제대로 이루어 지지 않기 때문이다.

5. 결론

본 논문에서는 작업 경로의 대역을 보장하는 대체 경로를 설정하는 disjoint 경로를 계산하는 알고리즘을 제시하고 시뮬레이션을 통해서 대체 경로로 인한 대역의 소비 효과를 보여 주었다.

대체 경로의 대역을 동일 링크의 작업 경로가 아닐 경우 공유함으로써 최대한 대역의 소비를 피하였다. 이 경우 대체 경로로 인한 대역이 작업 경로에 비해서 어떻게 되는지를 시뮬레이션을 통해 보여주고 또한 대역을 부족으로 인해 설정이 거부되는 효과를 보여 주었다.

이 결과는 대역을 고려하여 대체 경로를 설정할 때 최대한 고려해야 하는 대역이 어느 정도이며 또한 어느 정도의 작업 설정 요구를 수용할 수 있는지를 분석하고 있다.

참고문헌

- [1] D. O. Awduche et al, "A Framework for Internet Traffic Engineering," Internet Draft <draft-ietf-tewg-framework-02.txt>, July 2000
- [2] D. O. Awduche et al, "Requirements for Traffic Engineering Over MPLS," RFC 2702, September 1999
- [3] B. Jamoussi et al, "Constraint-Based LSP Setup using LDP," Internet Draft <draft-ietf-mpls-cr-ldp-04.txt>, July 2000
- [4] D. O. Auduche et al, "RSVP-TE: Extensions to RSVP for LSP Tunnels," Internet Draft <draft-ietf-mpls-rsvp-lsp-tunnel-07.txt>, August 2000
- [5] Murali Kodialam T.V. Lakshman "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration" INFOCOM 2000
- [6] J.W. Suurballe R.E. Tarjan "A Quick Method for Finding Shortest Pairs of Disjoint Paths" Networks, vol. 14 (1984)

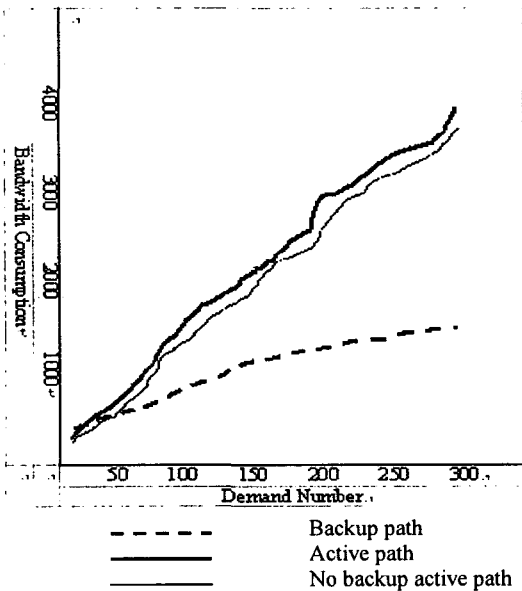


그림 2: 경로 설정 요구에 따른 대역 소비

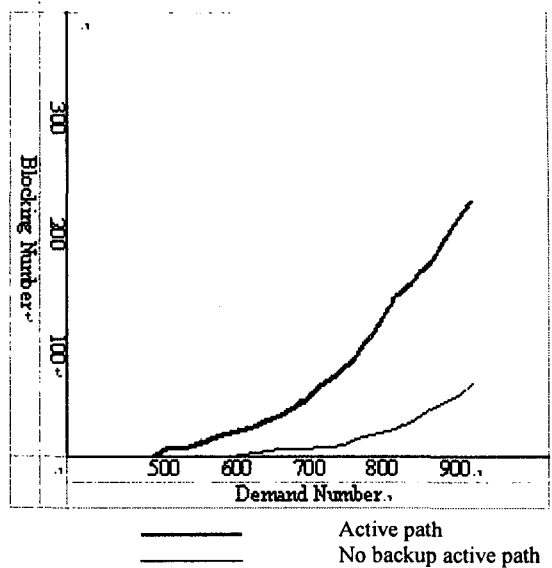


그림 3: 경로 설정 요구에 따른 설정 거부(blocking) (링크의 대역은 균등하게 할당)

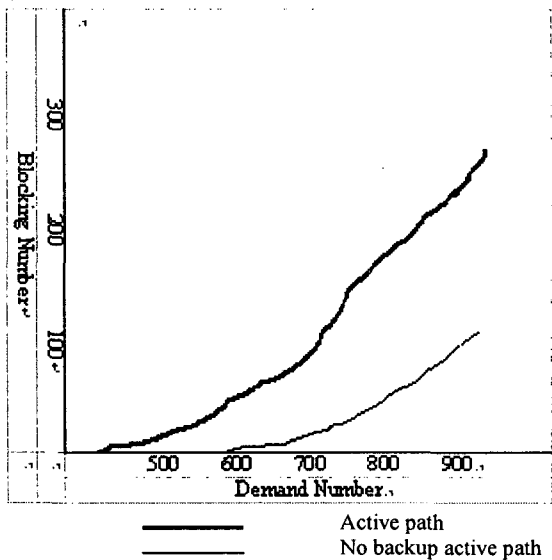


그림 4: 경로 설정 요구에 따른 설정 거부(blocking) (링크의 대역은 랜덤하게 할당)