

광 네트워크에서의 도메인 분리에 의한 효율적인 보호복구

임 순 빈^{*} · 안 현 기^{**} · 이 태 진^{***}

요 약

광 네트워크에서의 효율성과 안정성을 위해 보호 복구가 요구되면서 기존의 여러 보호 복구 방법들이 제안되고 있으며, 많은 연구가 진행되고 있다. 기존 연구 중에서 네트워크 토폴로지를 하나의 사이클로 구성하여 어느 한 링크가 손상되더라도 복구가 가능한 방법들이 제시되고 있다. 본 논문에서는 해밀턴 사이클(Hamiltonian cycle)을 이용하여 네트워크 토폴로지를 하나의 사이클이 아닌 몇 개의 다중 도메인으로 분리하여 장애 발생 시 해당 도메인 내에서 복구경로 설정이 가능한 알고리즘을 제안한다. 제안된 알고리즘을 시뮬레이션을 통해 분석한 결과 복구 경로길이(단일 사이클의 경우에 비해 57% 이상 감소함)를 볼 수 있다. 즉, 다중 도메인으로 분리, 보호복구를 수행 할 경우 복구 용량의 증가는 크지 않으면서 고속 복구가 가능함을 확인할 수 있다.

Efficient Protection/Restoration by Separation of Domains in Optical Networks

Soon-Bin Yim^{*} · Hyun-Ki An^{**} · Tae-Jin Lee^{***}

ABSTRACT

Protection of user service becomes increasingly important since even very short interruption of service due to link or node failure will cause huge data loss and incur tremendous restoration cost in high speed network environment. Thus fast and efficient protection and restoration is one of the most important issues to be addressed. Protection methods have been proposed to provide efficiency and stability in optical networks. In this paper, an original network is separated into several domains using Hamiltonian cycle, and link protection is performed on the cycles of the domains. We have shown that protection path length can be decreased up to 57% with marginal increase of backup capacity. Our proposed method can provide high-speed protection with marginal increase of protection capacity.

키워드 : 광 네트워크(Optical Networks), 보호 및 복구(Protection/Restoration), 해밀턴 사이클(Hamiltonian Cycle), 도메인(Domains)

1. 서 론

인터넷의 수요의 증가에 따라 좀 더 빠르고 효과적인 서비스를 제공하기 위해서 광 네트워크의 필요성이 더 부각되고 있다. 이와 같이 대용량의 데이터를 빠르게 전송할 수 있는 광 네트워크에서, 링크에 장애가 발생하게 되면 그 손실은 막대하게 된다. 본 논문에서는 광 네트워크 내에서의 데이터 전송에 대한 안정성을 보장하기 위해 보호 복구에 관련된 방법을 제시하고자 한다. 광 네트워크에서의 보호 및 복구 방법은 크게 링크 복구 방법과 경로 복구 방법으로 나눌 수 있으며, 또한 이를 기반으로 한 보호 및 복구 알고리즘이 제안되고 있다. 연구 된 보호 및 복구 방법으로는 크게 두 가지로 나눌 수 있는데 하나는 네트워크가 정상적인 상

황일 경우 데이터 전송이 주경로를 통하여 전송되다가 네트워크에 장애가 발생할 경우 부경로를 통해서 전송해주는 방식[1-7]이 있다. 이와 같이 주경로와 부경로를 두어 복구할 경우 주경로 중의 특정 링크가 손상되었다면, 그 경로의 처음으로 다시 돌아가서 복구경로를 설정하는 방법을 사용할 수 있는데 이 방법은 시간적인 낭비뿐만 아니라 효율적인 보호 복구를 할 수 없게 된다. 그러므로 좀더 효율적인 보호복구를 위해서 일정한 부분마다 중복되는 복구 영역(protection domain)을 두어서 빨리 복구 경로를 설정해 주는 방식인 SLSP(Short Leap Shared Protection)방법이 제안되었다[7]. 이 방법은 데이터를 전송하다가 어떤 링크에 장애가 발생하게 되면 신호를 출발지(source)까지 보내지 않고 복구영역에 포함되는 노드까지 신호가 가면 그 복구영역 안에서 경로를 설정해 복구해주는 방법으로 복구 시간을 좀더 줄여주게 된다. 다른 한 가지 방법은 전체 네트워크를 하나의 링 형태를 가지게 함으로써 링크 손상 시 링 내에서 복구해주는 방식[8-11]이다. 이 방법은 네트워크에서 하나의 큰 사이클로 복구 경로를 설정하고 어떤 링크에 장애가 발생되었을 경우 사

* 본 연구는 한국학술진흥재단(KRF 2002-003-D00219)의 지원으로 수행되었음.

† 준 회 원 : 성균관대학교 전기전자및컴퓨터공학부

†† 준 회 원 : 성균관대학교 정보통신공학부

††† 정 회 원 : 성균관대학교 정보통신공학부 조교수

논문접수 : 2003년 6월 23일, 심사완료 : 2004년 8월 9일

이클의 복구 경로를 통해 복구해주는 방법이다. [8]에서는 링크 손상이 있을 경우, 각 노드에 세팅되어 있는 보호 복구 스위치를 이용하여 손상 링크를 피하도록 절체 시켜 복구 시켜주는 방법을 제안하였다. 이와 같이 사이클을 형성하여 복구하는 방법으로 오일러 투어(Euler tour)[9]와 해밀턴 사이클(Hamiltonian Cycle)[10],[11]을 이용하는 방법이 있다.

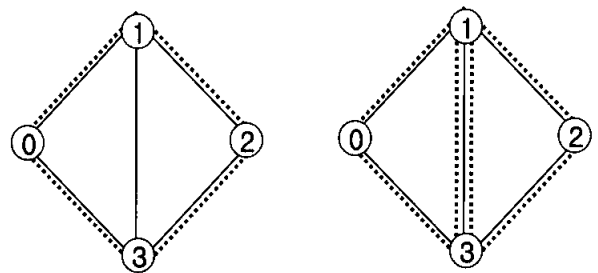
오일러 투어는 연결된 그래프(connected graph)상에서 모든 에지(링크)를 한번씩 지나는 특성을 가지며, 또한 노드의 degree가 홀수인 것이 0개 아니면 2개이어야만 오일러 투어가 성립된다. [9]에서는 이러한 오일러 투어를 이용하여 복구 경로를 설정하는 방법을 보여주고 있다. 즉, 네트워크 토폴로지에 오일러 특성을 적용하여 오일러 투어를 그리기 위해 노드의 degree가 홀수인 것을 찾아 문제노드(problem node)로 정의하고 테이블을 이용하여 가상적인 네트워크 토폴로지를 새로 그려서 오일러 투어를 그리고 복구해주는 방법을 제시하고 있다. 반면에 해밀턴 사이클의 특성은 연결된 그래프 상에서 모든 노드를 한번씩 지나야 한다. 해밀턴 사이클을 형성하기 위해서는 모든 노드의 degree가 $n/2$ (n :노드수)보다 크거나 같아야 하는 충분조건을 갖는다. 그러나 이러한 해밀턴 사이클을 형성하기 위한 필요충분조건이 성립되지 않아(open problem) 연결된 그래프라 해도 해밀턴 사이클이 모두 그려지지 않는다는. [10, 11]에서는 이러한 해밀턴 사이클을 이용하여 배워 망에서 사이클을 구성하고 도메인을 분리했을 경우 도메인끼리 중복되는 링크의 용량을 구하는 여러 가지 방법(HCC, HNC 등)을 제안하고 있다.

본 논문에서는 모든 노드를 한번씩 지나는 특성을 가진 해밀턴 사이클을 이용하여 몇 개의 다중 사이클(multiple domain)을 만들어 보호 복구하는 방법을 제안한다. 이를 위해 본 논문에서는 효율적으로 도메인을 분리하는 알고리즘을 제안한다. 이와 같은 방법을 사용할 경우, 도메인을 분리하지 않고 전체 사이클을 통해서 복구할 경우와 제안된 도메인 분리 알고리즘을 적용 할 경우 복구 경로의 길이(복구시간) 및 복구 용량을 비교한다. 본 논문의 구성은 다음과 같다. 2장에서는 전체 네트워크에서 하나의 사이클을 통한 복구에 대해서 알아보고, 3장에서는 네트워크 내에서 효율적으로 도메인을 분리하기 위한 알고리즘을 제안한다. 4장에서는 네트워크 복구 시에 필요한 최대 용량을 수식적으로 분석하여 도메인 분리하지 않았을 경우와 도메인을 분리했을 경우의 복구 용량 차이를 확인하고, 5장에서는 제안한 알고리즘에 대한 성능분석을 수행한다. 마지막으로 6장에서 결론을 맺는다.

2. 전체 네트워크에서의 보호복구

기존의 사이클을 형성하는 방법으로는 앞서 언급한 오일러 투어나 해밀턴 사이클을 이용하여 사이클을 하나만 형성해서 복구하는 방법이 제시되어 있다. 이와 같은 경우 네트워크 토폴로지가 작은 경우에는 큰 문제가 없지만, 네트워크 토폴로지가 커질 경우 전체 네트워크 내에서 손상된 하나의 링크를 복구 해주기 위해서 복구 경로 길이가 길어짐으로 복구 시간이 길어질 수 있는 단점을

가지고 있다. 예를 들어, (그림 3)과 같은 간단한 네트워크 토폴로지가 있다고 가정할 때, 해밀턴 사이클은 (그림 1) (a)에서 보는바와 같이 점선으로 표현할 수 있다. 이와 같은 경우에 만약 0-1 링크가 손상되었을 경우 이를 복구하기 위해서는 해밀턴 사이클을 이용 0-3-2-1로 복구한다. 반면에, (그림 1) (b)에서 보는 바와 같이 두개의 도메인(A:0-1-3, B:1-2-3)으로 나누게 되면 점선과 같이 두개의 도메인에서 사이클을 형성할 수 있다. 이와 같은 경우 0-1사이의 링크가 손상되었을 경우 이를 복구하기 위해서는 도메인 A내에서 0-3-1로 복구할 수 있다. 즉, 위와 같이 도메인을 나누어 복구 경로를 설정 할 경우 복구 경로의 길이가 짧아짐을 볼 수 있다. 이와 같이, 도메인을 분리하여 복구하는 방식을 사용하기 위해 본 논문에서는 효율적인 도메인 분리 알고리즘을 제안하고자 한다. 전체 하나의 사이클을 통해서 복구할 경우와 도메인을 분리해서 여러 사이클을 이용해서 복구 할 경우 복구 경로의 길이(복구시간), 복구 시 필요한 용량을 분석한다.



(a) 전체일 경우

(b) 도메인 분리할 경우

(그림 1) 네트워크 토폴로지 사이클 형성 예.

3. 제안 도메인 분리 알고리즘

본 논문에서는 네트워크를 여러 개의 도메인으로 분리하고, 각 도메인마다 해밀턴 사이클을 두어 링크 장애 시 해당 도메인을 통해 복구하는 알고리즘을 제안한다. 알고리즘에서는 각 링크별로 용량이 주어졌을 때 각 노드에 연결된 모든 링크의 용량의 값을 합하여 가장 작은 용량의 값을 갖는 노드를 선택한다. 처음 도메인을 형성하는 단계는 이렇게 선택된 노드에서 가장 작은 용량의 값을 갖는 링크를 따라 다음 노드로 이동하며, 이 과정을 반복하여 결국 시작 노드로 다시 돌아오게 되면 하나의 사이클이 형성되며, 그 사이클을 하나의 도메인으로 나타낸다. 이 단계에서 고려할 사항은 이미 지났던 노드로 가게 될 경우에는 backtracking을 해서 다른 링크를 선택하는 방법으로 도메인을 찾는다. 첫 번째 도메인을 형성하고 난 후 다음 도메인을 형성하기 위해서 경계 노드의 개념을 도입한다. 경계 노드란 한 도메인에 포함되는 노드들 중에서 다른 어떤 도메인에도 포함되지 않는 노드와 연결되어 있는 노드를 말한다. 도메인을 형성하기 위해서는 경계노드들을 찾아 그 중 가장 작은 용량을 가진 링크와 연결되어 있는 경계노드를 시작노드로 정한다. 그리고 첫번째 도메인을 찾을 때와 같은 과정을 거

치며, 시작된 경계노드의 도메인과 동일한 도메인의 노드에 도달했을 경우, 새로운 도메인 하나가 형성된다. 네트워크에 따라서 여러 개의 도메인이 형성될 수 있으므로, 도메인 형성 과정을 경계 노드가 없을 때까지 반복한다. 제안된 알고리즘의 pseudo code는 (그림 2)와 같다.

4. 복구 용량 분석

도메인을 분리하지 않고 복구를 할 경우 필요한 용량과 제안 알고리즘을 통해 도메인을 분리하고 난 후의 용량 차이를 비교하기 위해 다음과 같이 쓰도록 한다. 도메인을 분리하지 않았을 경우 전체 네트워크 HC(Hamiltonian Cycle)의 용량(C_N)은 다음과 같이 구해진다.

$$C_N = \max_l [C_{H,l}, \frac{1}{2} C_{\bar{H},l}] \times |L_H|, \quad l \in (L_H \cup L_{\bar{H}}) \quad (1)$$

여기서,

L_H : 전체 HC에 포함되는 링크의 집합

$L_{\bar{H}}$: 전체 HC에 포함되지 않는 링크의 집합

$C_{H,l}$: 전체 HC에 포함되는 링크 $l(l \in L_H)$ 의 용량

$C_{\bar{H},l}$: 전체 HC에 포함되지 않는 링크 $l(l \in L_{\bar{H}})$ 의 용량.

또한 도메인을 제안 알고리즘에 따라 분리하였을 경우 전체 네트워크의 용량은, 경계링크(도메인 간 중복되는 링크)를 공유할 경우와 공유하지 않을 경우 두 가지로 분리될 수 있다. 복구하기 위해 필요한 분리된 각 도메인에서의 최대 용량(C_d)은

$$C_d = \max_d [C_{d,l}, \frac{1}{2} C_{\bar{d},l}], \quad l \in (L_d \cup L_{\bar{d}}) \quad (2)$$

여기서,

L_d : 도메인 d 에 포함되는 링크의 집합

$L_{\bar{d}}$: 도메인 d 에 포함되지 않는 링크의 집합

$C_{d,l}$: 도메인 d 에 포함되는 링크 $l(l \in L_d)$ 의 용량

$C_{\bar{d},l}$: 도메인 d 에 포함되지 않는 링크 $l(l \in L_{\bar{d}})$ 의 용량.

경계링크를 공유하지 않았을 경우와 공유했을 경우에 대해 다시 살펴보면, 경계 링크의 용량을 공유하지 않았을 경우 필요한 복구 용량(C_{T1})은

$$C_{T1} = (\sum_d C_d \cdot |L_d|) + \sum_{l \in L_{\bar{D}}} C_{\bar{D},l} \quad (3)$$

$L_{\bar{D}}$: 어느 도메인에도 포함되지 않는 링크의 집합

$C_{\bar{D},l}$: 서로 다른 도메인 간 연결된 링크 $l(l \in L_{\bar{D}})$ 의 용량(헤밀턴 사이클에 포함되지 않는 링크의 용량).

```

Algorithm HC_domain(G,D)
Begin
/* Definition
G: network graphs, D: set of domains
s_node: starting node to find a domain
n_node: current node
b_node: a domain's node connected with at
least one node which is not included
in any domain
node capacity: total sum of capacity
connected with a node
domain ID: ID of each node
node(d): domain ID of node */
/* initialization */
D=∅, i=1;
s_node=node with the minimum node capacity;
n_node=node connected with the link having minimum link
capacity from s_node;
Find the minimum node capacity;
/* step 1: find domain 1 */
Save s_node;
While(s_node!=n_node){
n_node=node connected with the link having
minimum link capacity from n_node;
Backtracking when n_node is visited before;
Save n_node into domain i;
}
Set domain ID to each node of domain i;
D=D∪domain i;
/* step 2: find domain i */
While(b_node exists){
i=i+1;
s_node=one of b_nodes with the minimum
link capacity;
n_node=node connected with the link having
minimum link capacity from b_node;
While(s_node(d)!=n_node(d)){
n_node=node connected with the link having
minimum link capacity from n_node;
Backtracking when n_node is visited before;
Save n_node into domain i;
}
Finish domain i using the domain n_node(d);
set domain ID to each node of domain i;
D=D∪domain i;
}
End
    
```

(그림 2) 효율적 보호 복구를 위한 제안된 다중 도메인 분리 알고리즘.

경계 링크의 용량을 공유하는 경우 네트워크 용량(C_{T2}) 다음과 같다.

$$C_{T2} = \sum_d (\sum_s C_d \cdot |L_d - L_{sd}|) + \sum_d (\sum_s C_{sd} \cdot |L_{sd}|) + \sum_{l \in L_{\bar{D}}} C_{\bar{D},l} \quad (4)$$

여기서,

L_{sd} : 도메인 s 와 d 가 공유되는 링크의 집합

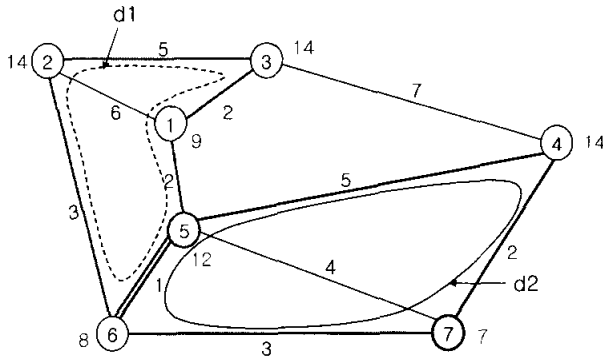
C_{sd} : 도메인 s 와 d 가 공유되는 링크의 최대 용량.

간단한 예를 통해 네트워크의 복구 용량을 구하는 과정을 고려해 본다. (그림 3)은 노드 7개, 링크 11개를 가지는 간단한 네트워크 예이다.

이 경우 전체 네트워크에 대한 해밀턴 사이클이 2-1-3-4-7-5-6과 같이 그려진다. 이와 같은 경우 전체 해밀턴 사이클에 대한 최대 용량을 위의 식에 대해 구해보면, 식 (1)에 의해서 다음과 같이 구해진다.

$$C_N = \max[1, 2, 3, 4, 6, 7, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}] \times 7 = 49$$

또한 제안하는 도메인 분리 알고리즘에 의해 분리된 도메인은 두 개 ($d1 : 6-5-1-3-2$, $d2 : 6-7-4-5$)로 나뉘어진다. 이와 같은 경우 각 도메인에 대한 최대 용량을 위의 식에 대해 구해보면, 식 (2)에 의해서 다음과 같이 구해진다.



(그림 3) 용량 계산을 위한 간단한 네트워크의 예(도메인 d1, d2).

$$C_1 = \max[1, 2, 3, 5, \frac{6}{2}] = 5$$

$$C_2 = \max[1, 2, 3, 5, \frac{4}{2}] = 5$$

복구 시 필요한 네트워크의 용량을 경계링크 공유의 여부에 따라 구해보도록 한다. 경계링크를 공유하지 않았을 경우는 식 (3)에 의해서 다음과 같이 구해지며,

$$C_{T1} = (5 \times 5) + (5 \times 4) + 7 = 52$$

경계링크를 공유했을 경우(공유링크:6-5)는 식 (4)에 의해서 다음과 같이 구해진다.

$$C_{T2} = (5 \times 4) + (5 \times 3) + 5 + 7 = 47$$

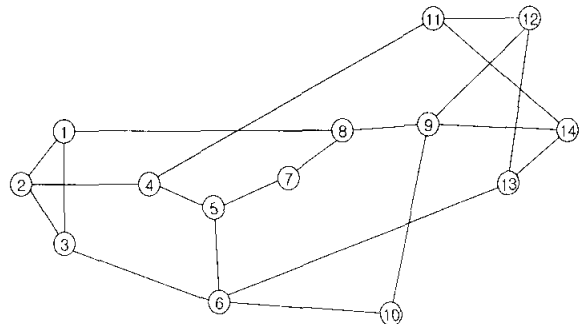
여기서 다른 도메인 간에 연결된 링크 3-4에서 장애 발생 시 도메인 1(6-5-1-3-2)과 도메인 2(6-7-4-5)를 거쳐 복구하기가 어렵다. 즉, 노드 5에서의 트래픽이 3-1-5-4의 트래픽 인지 6-5-4의 트래픽 인지 광 계층에서는 구별이 어렵다. 따라서 별도로 복구 링크가 필요하다(복구용량:7).

위의 예에서 제시된 것처럼 일반적으로 경계링크를 공유하지 않았을 경우 보다 공유했을 때가 더 작은 복구용량을 가지게 된다. 또한 하나의 해밀턴 사이클을 형성해서 복구할 경우(복구용량:49)와 몇 개의 도메인으로 분리해서 복구할 경우(복구용량:52, 47)의 용량의 차이를 보면 위의 예의 경우에는 경계링크의 용량을 공유할 경우가 가장 적은 복구용량을 필요로 하지만 일반적으로는 도메인을 분리해서 복구할 경우가 분리하지 않았을 경우보다 복구용량이 더 많이 필요할 것으로 예상된다.

추가적으로 위의 예를 이용하여 복구 길이에 대하여 살펴보면, (그림 3)에서 2-6링크가 전체 되었을 경우, 전체 해밀턴 사이클을 이용하면 2-1-3-4-7-5-6으로 복구가 되지만, 분리된 도메인 영역 d1을 이용하면 2-3-1-5-6으로, 복구 길이의 차이를 볼 수 있다.

5. 성능 분석

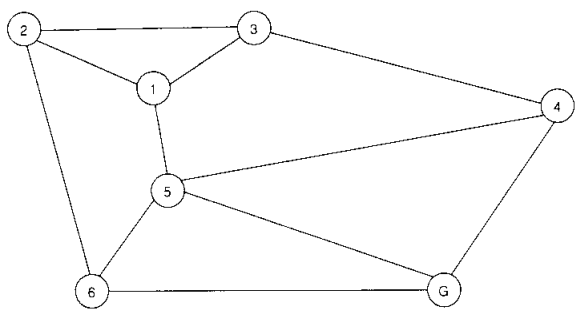
기존의 SLSP 복구방식[7]은 링크 보호 방식과 경로 보호 방식의 혼합 방식으로, 네트워크 전체에서 다중 사이클을 이용 복구하는 본 논문의 링크 보호 방식과 다르므로 본 논문에서는 해밀턴 사이클을 이용한 단일/다중 도메인 보호 방식의 성능을 비교, 분석하였다.



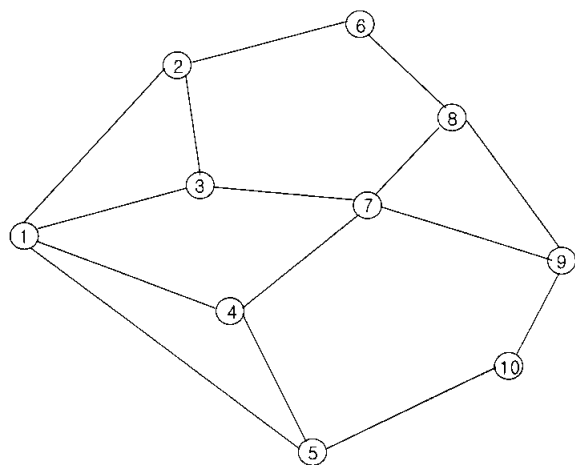
(그림 4) NSFNET.

제안 알고리즘의 성능평가를 위해 시뮬레이션을 수행하였다. 시뮬레이션에 사용된 네트워크는 (그림 4)의 NSFNET(노드 14개, 링크 21개)과 노드수가 각각 7개(그림 5), 10개(그림 6), 18개(그림 7)인 세 개의 네트워크를 이용하였다. 각 링크별 용량은 랜덤하게 1부터 9까지 균일한 분포로 생성하였으며, 복구경로의 길이와 복구 시 필요한 복구용량은 각각의 경우에 1000번씩 수행하여 평균을 도출하였다. 하나의 사이클을 형성(single domain)해서 복구할 경우와 제안 알고리즘을 통해 다중 도메인으로 분리(multiple domains)하여 복구할 경우 복구 길이와 복구 용량을 비교하였다.

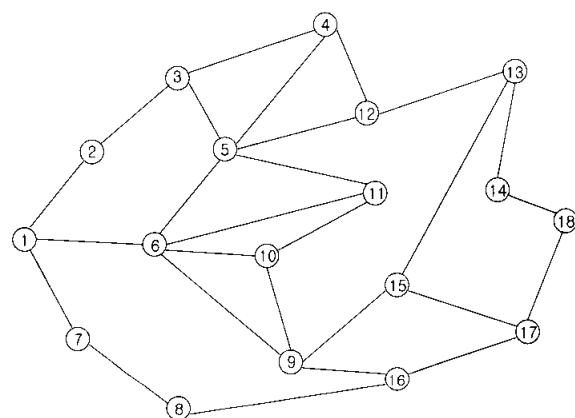
(그림 8)은 각 네트워크마다 평균적으로 몇 개의 도메인으로 분리되는지를 나타내고 있다. 네트워크 토폴로지의 크기가 커질수록 분리되는 도메인수가 평균적으로 증가함을 볼 수 있다. 네트워크의 복구 경로 길이는 네트워크가 손상되었을 때 얼마나 짧은 경로로 얼마나 빨리 복구할 수 있는지와 관련된다. 즉 복구 경로 길이는 복구 시간과 연관됨을 의미한다.



(그림 5) 네트워크 1 (N1).



(그림 6) 네트워크 2 (N2).

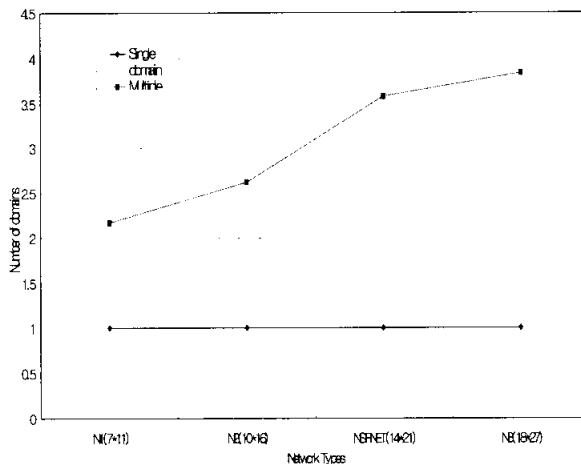


(그림 7) 네트워크 3 (N3).

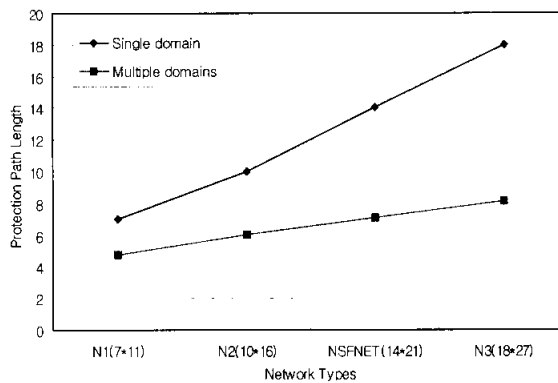
(그림 9)에서 보는 바와 같이 도메인을 분리했을 경우 도메인 내에서 지역적으로 복구하기 때문에 복구 경로 길이는 단일 사이클의 경우에 비해 최대 57% 줄어드는 것을 확인할 수 있다. 또한 네트워크가 커질수록 복구 경로 길이의 감소량이 커짐을 확인할 수 있다. 따라서 네트워크가 더 커질 경우 복구 경로 길이는 더 줄어들 것임을 예상할 수 있다.

복구 시 필요한 용량의 차이는 경계 링크의 용량을 공유할 것(그림 10)인지 공유하지 않을 것(그림 11)인지 각각의 경우에 대해 비교하였다. (그림 10), (그림 11)에서 보는 바와 같이 도메인을 분리하기 때문에 전체적으로 복구용량이 단일 사이클의 경우에 비해 약 54% 증가하지

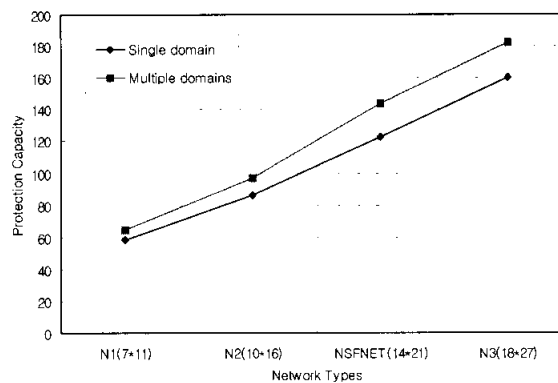
만, 네트워크가 커질수록 복구 용량 증가 분이 작아짐을 알 수 있다. 따라서 복구 용량의 증가를 복구 경로의 감소로 보상할 수 있음을 확인할 수 있다. 또한 경계 링크의 용량을 공유하지 않을 경우에는 각 도메인별로 링크에 하나씩 장애가 발생할 경우도 복구가 가능하다(경계링크 제외). 즉, 다중 장애(multiple failure) 복구가 가능하다. 반면에 경계링크를 공유할 경우에는 도메인이 겹치지 않을 경우에만 다중 장애에 대한 복구가 가능하고, 그렇지 않을 경우에는 다중 장애 복구가 가능하지 않을 수도 있다.



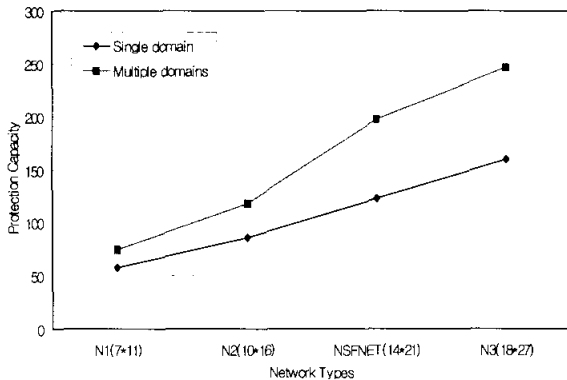
(그림 8) 평균 도메인 수 비교.



(그림 9) 복구 경로 길이 비교.



(그림 10) 경계 링크를 공유할 경우, 복구 용량 비교.



(그림 11) 경계 링크를 공유하지 않을 경우, 복구 용량 비교.

6. 결 론

본 논문에서는 광 네트워크에서의 복구 경로 길이를 단축하기 위해 영역을 분리하여 고속 복구가 가능하도록 하는 알고리즘을 제안하였다. 전체 네트워크를 여러 도메인으로 분리해서 복구하게 되면 하나의 사이클을 구성하여 복구하는 것보다 복구 경로 길이가 감소됨을 확인할 수 있었다. 이는 네트워크에서 장애가 발생했을 경우 복구하는 시간을 단축할 수 있음을 의미한다. 또한 복구 시 필요한 복구 용량은 증가 하지만, 복구용량의 증가율에 비해 복구 경로 길이의 감소율이 더 크게 됨을 확인할 수 있었다. 도메인 분리 알고리즘을 이용하여 복구할 경우에는 단일 링크 손상 뿐 아니라 다중 링크 손상까지 어느 정도 복구가 가능하게 된다.

참 고 문 헌

[1] R. Ramaswami and K. N. Sivarajan, "Optical Networks - A Practical Perspective," Morgan Kaufmann Publishers, 2002.
 [2] C. Siva Ram Murthy and M. Gurusamy, "WDM Optical Networks-Concepts, Design, and Algorithms," Prentice Hall, 2002.
 [3] S. Ramamurthy and B. Munkherjee, "Survivable WDM Mesh Networks, Part I - Protection," Proc. of IEEE INFOCOM, vol.2, pp.744-751, Mar., 1999.
 [4] S. Ramamurthy and B. Munkherjee, "Survivable WDM Mesh Networks, Part II - Restoration," Proc. of IEEE ICC, Vol.3, pp.2023-2030, 1999.
 [5] W. D. Grover and J. Doucette, "Increasing the Efficiency of Span-restorable Mesh Networks on Low-connectivity Graphs," Proc. of International Workshop on the DRCN 2001. pp.1-9, 2001.
 [6] W. D. Grover and J. Doucette, "Design of a Meta-Mesh of Chain Subnetworks: Enhancing the Attractiveness of Mesh-Restorable WDM Networking on Low Connectivity Graphs," IEEE Journal on Selected Areas in Communications, Vol.20, no.1, pp.47-61, Jan., 2002.
 [7] P. H. Ho and H. T. Mouftah, "A Framework of a

Survivable Optical Internet using Short Leap Shared Protection(SLSP)," Proc. of IEEE Workshop on High Performance Switching and Routing, pp.21-25, 2001.
 [8] G. Ellinas, A. G. Hailemariam and T. E. Stern, "Protection Cycles in Mesh WDM Networks," IEEE Journal On Selected Areas in Communications, Vol.18, No.10, pp.1924-1937, Oct., 2000.
 [9] A. Sen, B. Hao, B. H. Shen and G. Lin, "Survivable Routing in WDM Networks - Logical Ring in Arbitrary Physical Topology," Proc. of IEEE ICC, Vol.5, pp.2771-2775, 2002.
 [10] H. Huang and J. A. Copeland, "Multi-Domain Mesh Optical Network Protection Using Hamiltonian Cycles," Proc. of IEEE Workshop on High Performance Switching and Routing, Proc. of Merging Optical and IP Technologies Workshop, pp.83-87, 2002.
 [11] H. Huang and J. A. Copeland, "A series of Hamiltonian Cycle-Based Solutions to Provide Simple and Scalable Mesh Optical Network Resilience," Proc. of IEEE Communications Magazine, pp.46-51, Nov., 2002.



임 순 빈
 e-mail : sbyim@skku.edu
 1998년 한서대학교 전자공학과(공학사)
 2004년 성균관대학교 전기전자및컴퓨터공학부(공학석사)
 관심분야 : 광네트워크 라우팅 및 보호/복구, 무선LAN/PAN, ad-hoc 네트워크



안 현 기
 e-mail : puppybit@ece.skku.ac.kr
 2004년 성균관대학교 정보통신공학부(공학사)
 2004년~현재 성균관대학교 정보통신공학부(석사과정)
 관심분야 : 광네트워크 라우팅, 무선 LAN /PAN, ad-hoc 네트워크



이 태 진
 e-mail : tjlee@ece.skku.ac.kr
 1989년 연세대학교 전자공학과(공학사)
 1991년 연세대학교 전자공학과(공학석사)
 1995년 University of Michigan, Ann Arbor (M.S.E.)
 1999년 University of Texas, Austin (Ph.D.)

1999년~2001년 삼성전자 중앙연구소 책임연구원
 2001년~현재 성균관대학교 정보통신공학부 조교수
 관심분야 : 무선 PAN/LAN, ad hoc 네트워크, 광 네트워크, 통신 네트워크 성능 분석 및 설계