

# PFC 기반의 서비스 복구 QoS 재라우팅 알고리즘에 관한 연구

한 정 수<sup>†</sup> · 정 진 욱<sup>††</sup>

## 요 약

현재 QoS 라우팅의 목적은 사용자 서비스들의 각 QoS 요구사항을 만족하는 최적의 경로를 찾아 서비스하는 것이다. 그러나 네트워크 상황에 따라 장애가 발생했을 경우 이들의 QoS 요구를 만족시킬 수 있는 또 다른 백업 경로를 사용하여 서비스해야 한다. 따라서 본 논문에서는 네트워크 상에 발생된 장애 상황에 대해 서비스되고 있는 각 응용들에 대한 복구 방법으로 서비스 복구 QoS 재라우팅 알고리즘을 제시하고자 한다. 이를 위해 사전에 모든 네트워크 노드를 보호할 수 있는 PFC(Protection Fundamental Cycle) 생성 방법론과 최소의 재라우팅 비용을 사용하는 경로 선택 알고리즘들에 대해 살펴보고 이들에 대한 성능을 분석하기로 한다.

## A Study on Service Recovery QoS Rerouting Algorithm Based on PFC

Jeong-Soo Han<sup>†</sup> · Jin-Wook Chung<sup>††</sup>

## ABSTRACT

The focus of quality-of-service routing has been on the routing of a single path satisfying specified QoS constraints. However upon failure of a node or link on the path, a new path (backup path) satisfying the constraints has to be established. This paper proposed the fault recovery QoS Rerouting algorithms to recovery the service of applications under fault environment of network. For this, we describes the generation method of PFC (Protection Fundamental Cycle) protecting all of network nodes and suggests path selection algorithms using minimization rerouting cost problem and analyzes the performances of these.

**키워드 :** QoS 라우팅(QoS Routing), 복구 경로(Protection Cycle), 재라우팅(Rerouting), Integer Program, 장애복구(Fault Recovery)

### 1. 서 론

사용자가 요구하는 서비스들은 음성, 동영상 등의 멀티미디어 서비스와 같이 차츰 실시간적이고 우선순위가 높은 특징을 갖게 되었으며, 이에 향후 IP 네트워크는 이러한 중요한 데이터를 전송해야 하는 필요성이 생기게 되었다[1, 2]. 하지만 기존의 IP 기반 네트워크에서 사용하는 라우팅 기법으로는 이러한 사용자의 QoS 요구사항을 만족시킬 수 없게 되었다. 이는 네트워크 상황에 대한 올바른 인식 없이 단지 정적인 네트워크 자원만 사용하여 사용자 트래픽을 목적지까지 전달하는 알고리즘을 사용하기 때문이다. 이러한 알고리즘들은 네트워크 상태를 정확히 판단할 방법을 제공하지 않고 있을 뿐만 아니라 네트워크 상태를 더욱 악화시키는 결과를 낳게 되었다. 이에 결국 동적으로 변화하

는 네트워크 상황을 라우팅에 적용하고자 하는 연구들이 생기게 되었으며 이를 QoS 라우팅이라 한다. 이러한 QoS 라우팅에 사용하는 매트릭(matric)은 지연 시간(Delay), 사용 가능한 대역폭(Residual Bandwidth), 이용률(Utilization) 과 같이 동적으로 변화하는 네트워크 자원이다. 결국 QoS 라우팅의 목적은 각 응용에서 발생하는 연결에 대해 보장된 QoS를 제공하기 위해서 적절한 경로(feasible path)를 찾는 것이다[3].

하지만 이러한 연구들은 네트워크 상에 장애가 발생했을 경우에 대해서는 배제하고 있다. 이에 많은 연구들이 정상적인 네트워크 상황뿐만 아니라 장애가 발생했을 경우에 대한 올바른 서비스 제공을 위해 네트워크 생존성(survivability) 문제가 중요시하게 되었다. 이러한 IP 네트워크의 생존성 문제는 네트워크 상에 장애가 발생했을 경우 이를 진단하고 복구하여 사용자 서비스가 중단되는 상황이 발생하지 않도록 하는 것이다[4, 5].

네트워크 생존성 문제는 네트워크 관리자와 사용자 모두

<sup>†</sup> 준 회 원 : 성균관대학교 대학원 정보통신공학부

<sup>††</sup> 종신회원 : 성균관대학교 정보통신공학부 교수

논문접수 : 2002년 7월 26일, 심사완료 : 2002년 9월 3일

에게 중요한 문제가 되고 있으며 점차로 사회적, 산업적으로나 국가적으로 이러한 문제에 지대한 관심을 보이고 있다. 특히, 음성, 데이터, 비디오 컨퍼런스 등과 같은 서비스들이 도시시간이나 회사간에 상대적으로 작은 백본 네트워크 상에 함께 서비스하고 있으며, 점차적으로 광 시스템의 대역폭이 증가함에 따라 그들의 장애 상황에 대한 영향 또한 증가하게 되었다. 따라서 케이블의 단절과 같은 장애 상황에 대해 빠르고 견고한 서비스 복구 기법이 지대한 관심사항으로 대두되고 있다[4].

SONET, WDM과 같은 물리 계층의 전송 시스템은 장애 상황에 대해서 빠른 장애 복구를 제공할 수 있다는 장점에도 불구하고 서비스들의 백업 경로를 위해 전체 대역폭에 대한 투자가 과다하게 되고, 장애 상황에 대해 영향을 받은 플로우들은 단지 라우터의 네트워크 계층에서 동적 라우팅과 같은 형태로 복원되어야 하므로 물리 계층뿐만 아니라 상위 계층에서도 서비스 복구 기법이 도입되어야 한다는 단점을 지니게 된다. 상위 계층(IP 계층)에서 복구 방법을 제공하는 중요한 장점은 대역폭에 대한 분리가 필요하지 않다는 것이다[6]. 즉, 원(working) 대역폭과 여분(spare) 대역폭에 대한 개념이 필요치 않다는 것이다. 이와 함께 IP 계층에서 서비스 복구를 위한 추가적인 대역폭 할당은 서비스 복구 동안 QoS를 제공할 수 있도록 해줄 뿐 아니라 장애가 발생하지 않은 일반적인 상황에서는 사용자 서비스의 성능을 더욱 향상시킬 수 있도록 도와준다. 따라서 물리 계층과 같은 하위 계층에서 제공하는 서비스 복구 방법보다는 상위 계층에서 서비스 레벨로 장애를 복구할 수 있는 새로운 방법이 요구된다.

관련 연구로는 [6]에서는 네트워크 상에 여러 개의 VPC (Virtual Protection Cycle)을 생성함으로써 네트워크의 장애를 보호할 수 있는 방법이 연구되었으며, [7]에서는 회선과 노드 장애에 대해 재라우팅을 수행할 수 있도록 Redundant 트리를 생성하는 알고리즘을 연구하였다. 또한, [8]에서는 재라우팅 기법에 대한 최소 비용 문제를 수학적으로 해석함과 동시에 여러 가지 재라우팅 기법에 대한 성능을 보여주고 있다. 또한, [1]에서는 MPLS 상에서 라우팅 메트릭의 동적 정보에 따른 여러 가지 수학적 모델을 제시하고 있으며, 또한 이들에 대한 성능을 보여주고 있다. 특히, [9, 10]에서는 MPLS 상에서 빠르고 다양한 재라우팅 기법들을 소개하고 있다. 특히 [9]에서는 관리 노드들을 모두 보호하기 위해서 알고리즘으로 토대로 빠른 재라우팅 기법의 성능을 보여주고 있다. 하지만 이 기법은 모든 노드를 보호하기 위해서 많은 경우를 갖고 있기 때문에 이를 획일화하는 방법이 요구된다. [10]은 MPLS 상에서 빠른 재경로(fast re-route) 기법에 대한 연구를 하고 있다.

본 논문의 구성으로는 2장에서 연구의 목적과 범위에 대해 설명하고, 3장에서는 서비스 복구 라우팅에 사용되는 일

반적인 방식에 대해서 설명하기로 한다. 4장에서 본 논문에서 제안하는 서비스 복구 재라우팅 알고리즘을 설명하기로 한다. 이를 위해 PFC에 대한 개념과 생성 방법론 그리고 시간 복잡도에 대해 논하고 이를 각 라우터에서 유지하기 위한 라우팅 요구사항을 설명하기로 한다. 이를 바탕으로 본 논문에서 제안하는 두 가지 재라우팅 기법인 링크 재라우팅(Link Rerouting) 기법과 Directed\_Destination Rerouting 기법에 대해서 설명하고 이들의 성능을 평가하기 위한 성능 평가 항목에 대해서 설명하기로 한다. 5장에서는 제안하는 기법들과 성능 평가 항목을 토대로 시뮬레이션 함으로써 이들의 성능을 분석하기로 한다. 결론은 6장에서 제시하기로 한다.

## 2. 연구 목적 및 범위

본 논문에서 제안하는 IP 계층에서의 서비스 복구 재라우팅 알고리즘은 하부 계층에서 갖는 문제점인 여분의 대역폭을 요구하지 않으며, 동시에 하부 계층에서의 빠른 장애 복구 능력을 상위 계층에 접목함으로써 사용자 서비스들이 빠르고 지능적으로 장애 상황에 대처할 수 있는 알고리즘을 연구하는 것이다. 본 논문의 목적 및 범위로는 첫째, 라우팅 알고리즘이 각 라우터에서 라우팅 테이블을 갱신하기 전까지 기존의 라우팅 테이블 정보를 가지고 가장 적절한 우회로(detour)를 찾는 알고리즘을 제안하는 것이며, 둘째 관리하고자 하는 모든 노드에 대해 자원을 할당하지 않고 사전에 미리 복구 사이클(Protection Cycle)을 형성함으로써 장애 발생 후에도 기존의 IP 계층에서 제공하는 복구 방법보다 더욱 빠르게 복구할 수 있는 능력을 제공하는 것이다. 셋째로는 트래픽 엔지니어링 측면에서 우회로를 선택할 때 재라우팅에 소요되는 비용이 최소화될 수 있도록 최적의 경로를 찾는 알고리즘을 제공하고자 한다. 이는 라우팅에서 트래픽 플로우 모델(Flow Model)과 함께 복구 사이클을 상호 연동함으로써 제공할 수 있다. 넷째 네트워크의 정적 상태 정보와 동적 상태 정보를 재라우팅 메트릭으로 사용함으로써 이들에 대한 성능을 평가하고자 한다. 마지막으로 재라우팅 알고리즘으로 송신지와 목적지간의 재라우팅(End-to-end Rerouting)이 아닌 해당 장애 구간 즉 링크 재라우팅 기법과 장애 발생 위치에서 목적지까지 재라우팅 기능을 수행하는 Directed\_Destination Rerouting 기법의 제안을 통해 이들 간의 성능을 분석하는 것이다.

## 3. 서비스 복구 재라우팅

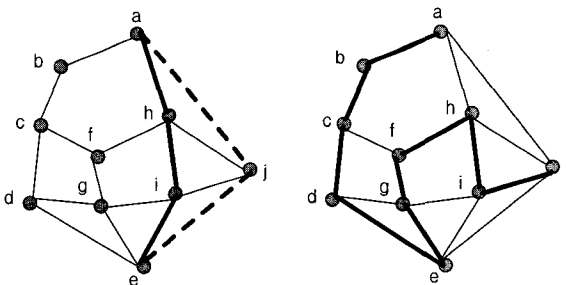
현재 멀티 미디어 서비스들의 QoS를 보장하고 네트워크 장애 상황에 대처하기 위한 여러 가지 복구 모델(recovery model)들이 연구되고 있다. 이러한 복구 모델에 대한 요구

사항으로는 첫째, 엄격한 QoS를 요구하는 서비스들을 만족시키기 위해 연결지향(Connection-oriented) 서비스를 제공해야 하며, 둘째로는 네트워크 자원을 미리 할당하기 위해 자원 예약 방식을 제공해야 한다. 셋째로는 네트워크 상의 장애 구간을 우회하여 서비스할 수 있도록 하는 재라우팅 알고리즘과 우회로를 제공해야 한다. 마지막으로 네트워크 상에 발생한 장애 상황을 감지(detection)하고 통보(notification)할 수 있는 기능을 제공해야 한다는 것이다. 이상과 같은 요구사항 중에서 본 논문에서는 네 번째의 장애 상황에 대한 감지와 통보에 대한 논의는 배제하기로 한다. 이러한 기능은 하부 계층(물리 계층 혹은 데이터 링크 계층)에서 제공하는 것이 빠르고 효과적이기 때문이다. 본 장에서는 서비스 복구 모델 중에서 우회로를 계산하는 시점에 따른 분류로써 복구(Protection) 방식과 재라우팅(Rerouting) 방식으로 나누어 설명하고자 한다[11].

3.1 복구(Protection) 방식

복구 방식은 장애 발생에 대해 우회로를 계산하고자 할 때 장애 발생 전에 즉, 사전에 모든 송신지에서 모든 목적지까지 우회로를 계산해 놓는 방식이다. 일반적으로 하나의 연결(Connection)에 대해서 장애가 발생하기 전의 일반적인 경로를 '원 경로'라 정의하고, 장애가 발생한 후에 연결이 사용하는 우회로를 '백업 경로'라 정의한다. 따라서 복구 방식은 이러한 백업 경로를 사전에 미리 계산하는 방식이다. 이 방식은 사전에 모든 경로를 계산해야 하기 때문에 계산 오버헤드가 발생하는 단점이 있는 반면, 장애가 발생했을 때 신속하게 대응할 수 있다는 점에서 장점을 가지고 있다.

현재는 이러한 복구 방식을 사용하기 위해서 원 경로에 대해 독립적인 백업 경로(link disjoint backup path, node disjoint backup path)를 계산하는 방식과 여러 개의 복구 사이클을 생성하는 방식들로 나누어져 연구되고 있다. (그림 1)는 이러한 복구 방식들 중에 (a)는 복구 방식 중 원 경로(a, j, e)에 대해 독립적인 백업 경로(a, h, i, e)를 보여주고 있으며, (b)는 연결과 상관없이 한 개의 복구 사이클(a, b, c, d, e, g, f, h, i, j)를 생성하여 서비스를 복원하는 방식을 보여주고 있다.



(a) disjoint backup path 방식 (b) protection cycle 방식  
(그림 1) 복구 방식들

또한, 경로 매핑(path mapping) 방법에 따라 다양한 복구 방법을 제안하고 있다. 경로 매핑이란 장애가 난 원 경로로부터 복구 경로로 트래픽을 매핑시키는 방법을 말하여 다음과 같은 방법들이 있다.

3.1.1 1-to-1 복구

이 방법에서 원 경로는 단지 그 원 경로를 복구하기 위해 지정된 복구 경로를 사용하게 된다.

3.1.2 n-to-1 복구

이 방법에서는 최대 n개의 원 경로들이 단지 하나의 복구 경로를 사용하여 보호되는 방법이다.

3.1.3 n-to-m 복구

이 방법은 최대 n개의 원 경로들이 m개의 복구 경로를 사용하여 보호되어지는 방법이다.

3.1.4 split path 복구

이 방법에서 여러 개의 복구 경로가 트래픽의 분산에 따라 하나의 원 경로에 있는 트래픽을 복구하기 위해 사용된다. 특히, 이것은 장애가 발생했을 때 원 경로 상에 있는 모든 트래픽을 복구할 수 있는 어떠한 복구 경로를 찾을 수 없을 때 유용하다.

3.2 재라우팅(Rerouting) 방식

재라우팅에 의한 서비스 복구 방법은 장애가 발생한 후에 서비스 트래픽을 복구하기 위해 장애 구간이나 전 경로에 대해 새로운 경로(detour)를 설정하는 기법이다. 이러한 새로운 경로는 장애 정보, 라우팅 정책, 토폴로지 정보를 고려하여 설정될 것이다. 이 방식은 장애 감지와 같은 기법이 수행되어야 하기 때문에 상대적으로 복구 기법보다는 속도가 느린 단점이 있다. 하지만 장애가 발생하고 장애의 위치가 알려지기 전까지는 네트워크 자원을 선점(pre-emption)하지 않기 때문에 더 간단하고 간략한 기법이다. 재라우팅 복구 기법에는 다음과 같은 두 가지 기법을 생각할 수 있으며 (그림 2)에서 나타내고 있다.

3.2.1 Local Repair

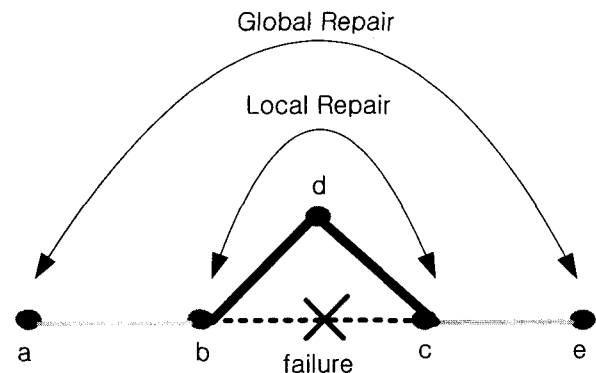
이 기법은 장애 통보에 걸리는 시간을 최소화하기 위해 장애 발생지역의 인접 노드나 링크에 대해서만 재라우팅 기법을 적용하는 것이다. 이 기법에서 장애 발생지역의 바로 전 노드(upstream node)가 복구를 하기 위한 재라우팅 기법이 개시되는 시점이다.

3.2.2 Global Repair

이 기법은 송신지 노드에서 발생한 장애의 경우를 제외한 모든 경로 상의 세그먼트나 경로 상의 노드나 링크 장애에 대해서 보호하는 것이다. 대부분 이 기법에서 사용하

는 복구 경로는 원 경로와 완전히 독립적이어야 하며, 이를 링크/노드 disjoint path라 한다. 이 기법의 장점은 원 경로 상에 있는 모든 장애 노드와 링크들에 대해서 보호한다는 것이다. 그러나 장애 발생 지역에서 복구 행동을 개시하는 노드까지 장애 통보 메시지가 전달되는 시간이 더 오래 걸릴 수 있기 때문에 local repair보다는 서비스 복구에 더 시간이 오래 걸릴 수 있다.

더욱이 이러한 재라우팅 기법을 사용하고자 할 때 우회로를 선택(path selection)하는 기준이 되는 재라우팅 메트릭을 고려해야 한다. 이 경우도 정상적인 라우팅과 마찬가지로 정적인 정보를 사용하는 경우와 동적인 정보를 사용하는 경우로 나눌 수 있다.



(그림 2) 재라우팅 기법들

#### 4. 제안된 PFC 기반의 서비스 복구 재라우팅 알고리즘

본 장에서 제안하는 장애 복구 모델은 복구 기법을 재라우팅 기법에 적용함으로써 보다 빠르고 효율적인 서비스 복구 재라우팅 알고리즘을 제안하고자 한다. 이를 위해서 장애 발생 전에 모든 노드를 복구하는 방식인 PFC(Protection Fundamental Cycle)를 제안하며, 이를 사용하여 서비스를 복구하는 재라우팅 알고리즘을 제안하고자 한다.

##### 4.1 PFC(Protection Fundamental Cycle)

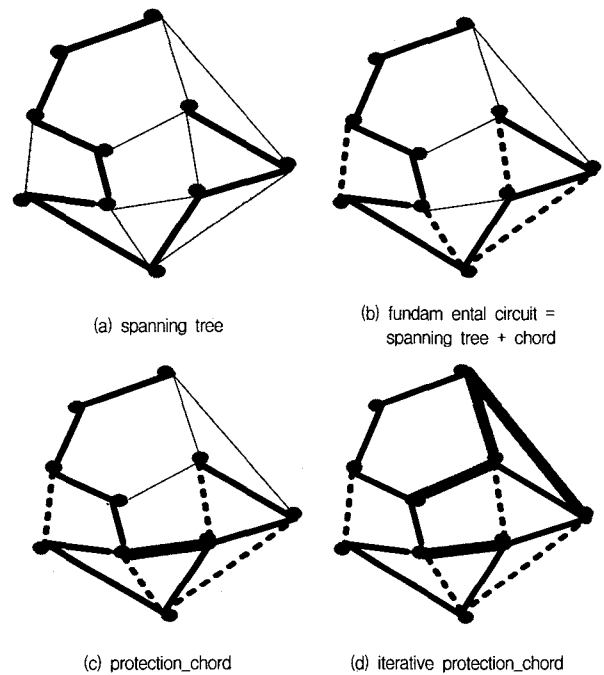
앞 장에서 설명했듯이 복구 기법은 장애가 발생하기 전에 백업 경로를 미리 계산하는 방식이다. 따라서 이 방식은 장애가 발생하였을 때 미리 계산해 놓은 백업 경로를 가지고 빠르게 대처할 수 있다는 장점이 있는 반면, 장애가 발생하기 전에는 사용되지 않고 있다가 장애가 발생되었을 때 사용되기 때문에 네트워크 자원 선점 측면에서는 큰 오버헤드를 가지고 있는 방식이다.

하지만 본 절에서 제안하는 PFC는 IP 네트워크 기반의 복구 사이클로써 네트워크 자원을 미리 선점하지 않고 단지 복구 사이클에 대한 정보만 유지함으로써 자원 선점이

라는 단점을 보완하였다. 또한, 관리 네트워크의 모든 노드에 대해서 복구를 제공함과 동시에 각 링크에 대해서 최대 2개의 복구 경로를 제공할 수 있다. 또한, 각 링크에 대해서 가장 기본적인(fundamental) cycle을 생성함으로써 장애가 발생했을 경우 사용자 서비스를 재라우팅하기 위해 해당 목적지까지 가장 최단 거리를 찾을 수 있는 장점을 제공하고 있다.

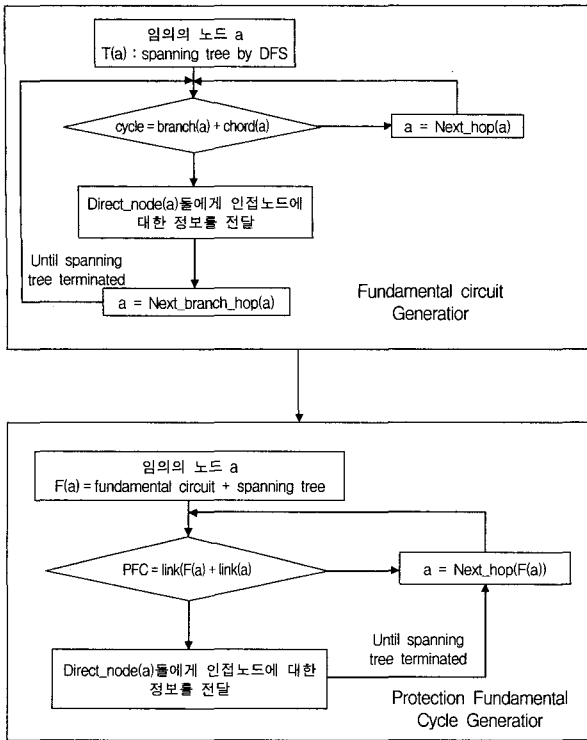
##### 4.1.1 생성 방법 및 알고리즘

본 논문에서 제안하는 PFC 생성 방법은 protection\_chord 개념을 사용하여 생성할 수 있으며, (그림 3)에서 보는 바와 같이 4단계 방법을 사용한다.



(그림 3) PFC 생성을 위한 절차

첫 번째 단계 (그림 3)(a)에서는 해당 네트워크에 대한 스패닝 트리(spanning tree)를 생성하는 것이다. 이것은 임의의 노드에서 단 한번만 수행하게 된다. 각 노드마다 스패닝 트리를 생성하지 않는 것은 스패닝 트리에 따라 PFC가 달라지지 않기 때문이다. 두 번째 단계 (그림 3)(b)는 생성된 스패닝 트리에 chord를 추가함으로써 fundamental circuit을 생성하는 단계이다. 세 번째 단계 (그림 3)(c)와 마지막 단계(그림 3)(d)에서는 PFC를 생성하기 위해 계속적으로 fundamental circuit에 protection\_chord를 추가하는 것이다. 이 때 PFC의 길이를 Cycle length로 정의한다. (그림 4)는 PFC를 생성하기 위한 알고리즘을 보여주고 있다. (그림 4)에서 보듯이 PFC를 생성하기 위해서는 Fundamental Circuit을 생성하는 부분과 여기에 protection\_chord를 추가함으로써 PFC를 생성하는 부분으로 나눌 수 있다.



(그림 4) PFC 생성 알고리즘

자세한 알고리즘을 살펴보면 다음과 같다.

- ① 임의의 노드  $a$ 에 대해 DFS를 사용하여 스패닝 트리  $T(a)$ 를 생성한다.
- ② 주어진  $T(a)$ 상에서  $branch(a)$ 와  $chord(a)$ 를 연결한 것이 cycle을 형성하는지 파악한다. 만약 cycle을 형성하지 않으면 노드  $a$ 의 다음 노드 ( $Next\_hop(a)$ )를 시작으로 해서 ②를 반복 수행하고, 형성하면 3)을 수행한다.
- ③ 형성된 cycle에 대한 정보를 노드  $a$ 의 인접노드 ( $Direct\_node(a)$ )에게 전달한다.
- ④  $branch(a)$ 의 다음 인접 노드 ( $Next\_branch\_hop(a)$ )를 시작으로 ②를 스패닝 트리가 끝날 때까지 반복 수행한다.

위의 알고리즘을 수행한 결과인 fundamental circuit을 사용하여 다음 단계를 수행하여 최종적인 PFC를 생성하게 된다. 자세한 알고리즘은 다음과 같다.

- ① 임의의 노드  $a$ 의 스패닝 트리과 fundamental circuit을 더한  $F(a)$ 를 생성하게 된다.
- ②  $F(a)$ 의 링크 ( $link(F(a))$ )와 노드  $a$ 의 링크 ( $link(a)$ )를 더한 것이 cycle, 즉 PFC이면 ③을 수행하게 되고, 그렇지 않으면  $F(a)$ 의 다음 인접 노드 ( $Next\_hop(F(a))$ )를 대상으로 ②를 반복 수행하게 된다. 이때 PFC가 생성되면 노드  $a$ 의 링크 ( $link(a)$ )는 **protection\_chord**가 된다.

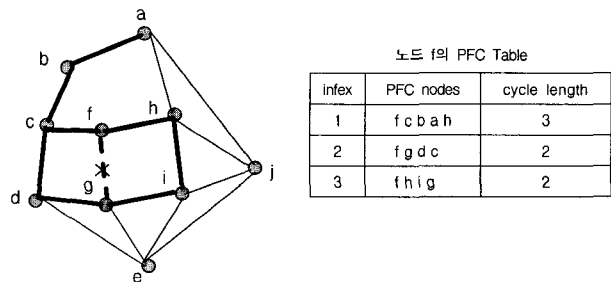
- ③ 생성된 PFC에 대한 정보를 노드  $a$ 의 인접 노드 ( $Direct\_node(a)$ )에게 전달함으로써 PFC의 정보를 공유하게 된다.
- ④  $F(a)$ 의 다음 인접 노드 ( $Next\_hop(F(a))$ )를 대상으로 스패닝 트리가 끝날 때까지 반복 수행한다.

PFC 생성에 대한 시간 복잡도(time complexity)를 살펴보면 스패닝 트리를 생성하기 위해 사용한 DFS 알고리즘은  $n$ 노드에 대해  $O(n^2)$ 이다. 다음으로 fundamental circuit의 생성을 위해 chord를 추가하는 알고리즘은 최대  $n$ 개의 노드를 진행해야 하기 때문에 시간 복잡도는  $O(n)$ 이다. 또한, 최종적으로 fundamental circuit에 protection\_chord를 추가하여 PFC를 생성하는데 걸리는 시간은 최소 한 번 이상의  $O(n)$ 이 걸리므로 최종적으로 PFC를 생성하기 위한 시간 복잡도는  $O(n^2 + n)$ 가 된다. 이것은 스패닝 트리를 생성하는데 걸리는 복잡도와 같이 미미함으로 알 수 있다.

#### 4.1.2 라우터 요구사항(Router Requirements)

본 논문에서 제안하는 기법을 적용하는 각 라우터는 다음과 같은 요구사항들이 필요하다.

- ① 장애가 탐지된 장소까지의 정상적인 라우팅은 Hop-by-Hop 라우팅 알고리즘을 사용한다. 이 방법은 현재 IP 네트워크 상에서 가장 많이 사용하는 라우팅 알고리즘으로써 각 노드에서 모든 목적지까지의 최소 비용을 갖는 경로로 전달하기 위해 다음 노드( $Next\ hop$ )에 대한 정보를 유지한다.
- ② 장애가 발생한 장소를 재라우팅 하기 위해 사용되는 재라우팅 알고리즘은 각 노드에서 네트워크 상황을 재라우팅 비용으로 하여 모든 목적지까지 경로를 결정하는 방법을 사용한다.
- ③ 각 라우터에서는 PFC 정보를 유지하기 위해 라우팅 테이블과는 별도로 PFC Table을 유지한다. 하나의 노드에서 유지하는 최대 테이블의 크기는 인터페이스 개수만큼 유지한다. (그림 5)은 노드  $f$ 에서  $g$ 로 연결된 링크가 장애가 발생했을 때 노드  $f$ 에서 유지하는 PFC 테이블을 보여주고 있다. 이 경우 테이블의 크기는 노드  $f$ 의 인터페이스 개수와 동일함을 알 수 있다.



(그림 5) 노드  $f$ 상의 PFC 테이블 예

4.2 재라우팅 최적화 모델

최적의 라우팅(optimal routing) 목적은 송신지에서 목적지까지 라우팅되는 비용(cost)을 최소화하는 것이다. 이와 같이 재라우팅 역시 재라우팅 되는 구간에서의 비용이 최소가 되는 경로를 찾는 것이다. 본 논문에서 제안하는 PFC 기법을 사용하여 재라우팅의 최소비용 문제를 Integer Program로 표현할 수 있다.

네트워크 노드를 집합  $N(|N|=n)$ 으로 표현할 때,  $L$ 은 각 노드의 해당 링크들의 집합으로 표현하며,  $B_l$ 은 각 링크  $l \in L$ 의 대역폭(bandwidth)으로 정의한다. 다음은 재라우팅 문제를 표현하기 위해 사용되는 정의들이다.

- $d_u$ : 연결  $u$ 가 요구하는 대역폭.
- $K$ : 해당 링크의 장애에 의해 서비스되지 못하는 연결들의 집합
- $R_l$ : 장애가 발생한 후 각 링크  $l$ 의 잔여 대역폭(residual bandwidth)
- $C_k$ : 연결  $k$ 복원을 위해 사용될 수 있는 PFC들의 집합
- $y_{kj}$ :  $C_k$  상의  $j$ 번째 링크가 연결  $k$ 복원을 위해 선택되었는다면 1이고, 아니면 0인 선택 변수,

$$y_{kj} = \begin{cases} 1 & j \in C_k \\ 0 & \text{else} \end{cases}$$

- $W_{kj}$ :  $C_k$  상의  $j$ 번째 링크가 서비스 못했던 연결  $k$ 의 복원을 위한 경로 비용
- $v_l$ : 사용 중인 링크  $l$ 의 비용
- $x_k$ : 연결  $k$ 복원이 실패한다면  $\infty$ 이고, 그렇지 않으면 0인 선택 변수

$$x_k = \begin{cases} \infty & d_k > R_l \quad (l \in C_k) \\ 0 & \text{else} \end{cases}$$

위의 정의를 바탕으로 재라우팅 최소 비용문제는 여러 개의 제약 조건(constraints)을 갖는 목적 함수(objective function)로 표현할 수 있다.

목적함수(objective function)

$$\min_{y_{kj}, x_k} \sum_{k=1}^K \sum_{j=1}^{C_k} W_{kj} y_{kj} + \sum_{k=1}^K x_k \quad (1)$$

제약조건(constraints)

$$\sum_{j=1}^{P_k} y_{kj} + x_k = 1, \quad k = 1, 2, \dots, K \quad (2)$$

$$\sum_{k=1}^K \sum_{j=1}^{P_k} d_k \delta_{kj}^l y_{kj} \leq R_l, \quad l = 1, 2, \dots, L \quad (3)$$

$$y_{kj} = 0 \text{ or } 1, \quad x_k = 0 \text{ or } 1 \quad (4)$$

$$C_k \geq 1 \quad (5)$$

식 (1)의 목적함수는 연결  $k$ 에 대해  $j$ 번째 링크를 재라우

팅 시에 선택하는 비용을 나타내며, 이러한 재라우팅 비용을 최소화하는 것이 이 목적 함수의 목적이다. 좀더 자세히 살펴보면, 식 (1)의 좌측 부분은 연결  $k$ 에 대해 재라우팅이 성공하는 비용이며, 우측 부분은 연결  $k$ 에 대해 재라우팅이 실패하는 비용이다. 제약조건을 살펴보면, 식 (2)는 각 연결의 재라우팅시 서비스가 성공되어야 한다는 것이다. 또 식 (3)은 연결  $k$ 의 재라우팅 시에 사용되는 각 링크의 잔여 대역폭이 연결  $k$ 의 요구 대역폭을 만족해야 한다는 것이다. 식 (4)는 선택 변수들이 가질 수 있는 값들을 나타내며, 마지막으로 식 (5)는 연결  $k$ 에 대한 PFC 집합 수가 1이상이어야 한다는 것을 명시하고 있다. 따라서, 재라우팅의 목적인 식 (1)을 최소화하기 위해서는 다음과 같은 요건을 만족해야 한다.

- ① 각 연결에 대해 재라우팅을 하고자 할 때 사용되는 길의 비용 ( $W_{kj}$ )을 최소화해야 한다는 것이다. 이것은 길을 이루는 각 링크의 비용 ( $v_l$ )의 합과 같다.

$$W_{kj} = \sum_{j=1}^{C_k} y_{kj} v_j \quad (6)$$

- ② 각 연결의 실패 ( $x_k$ )를 최소화해야 한다는 것이다. 즉, 재라우팅을 요구하는 모든 연결에 대해 서비스를 해야 한다는 것이다.
- ③ 재라우팅에 사용되는 링크의 수를 가능하면 최소화해야 한다는 것이다.

위의 요건에서 알 수 있듯이 재라우팅의 목적함수를 최소화하기 위해서는 첫째, 어떠한 길을 재라우팅 시에 선택할 것인가와 둘째, 경로 선택(path selection)시 어떠한 기준으로 선택할 것인가이다. 이러한 질문에 대한 답을 위해 본 논문에서는 경로 선택 문제에 있어서는 사전에 모든 노드를 보호하고 가장 짧은 거리를 선택할 수 있도록 PFC 기법을 제시하고 있다.

4.3 PFC 기반의 재라우팅 알고리즘

본 논문에서 제안하는 재라우팅 알고리즘들은 링크의 비용을 홉 수(Hop Count)와 잔여 대역폭(Residual Bandwidth)으로 하는 기법들을 제안하고자 한다. 제안하는 알고리즘은 크게 두 가지로 나눌 수 있다. 첫째 장애 구간만 재라우팅을 수행하는 링크 재라우팅 알고리즘과 장애가 발생한 위치에서 목적지까지 직접 재라우팅 기능을 수행하는 Directed\_Destination 재라우팅 알고리즘이다. 제안하는 링크 재라우팅 알고리즘은 재라우팅 시 사용하는 링크의 비용을 홉 수로 하는 SHC(Shortest Hop Count) 기법과 잔여 대역폭으로 하는 MRB(Maximum Residual Bandwidth) 기법으로 나눈다. 또한, 제안하는 Directed\_Destination 재라우팅

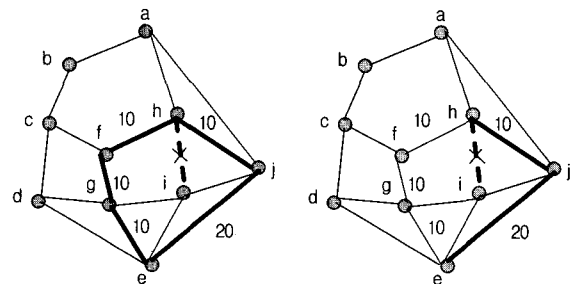
알고리즘의 링크 비용은 홉 수와 잔여 대역폭을 동시에 감안하는 다중 매트릭스(multi-matrices)를 지원하는 방법인 Shortest\_Widest 기법과 Widest\_Shortest 기법을 제안하고자 한다.

4.3.1 SHC(Shortest Hop Count) 링크 재라우팅 기법  
가장 고전적이고 간단한 재라우팅 기법이다. 장애 구간에 적어도 1개 이상 존재하는 PFC에 대해 각 cycle length를 조사하여 가장 작은 값을 갖는 길을 선택하여 재라우팅을 수행하게 된다. 이 때 각 링크의 비용인  $v_l=1$ 로 정의한다. 이 기법은 현재 IP 네트워크 상에서 많이 사용하는 라우팅 프로토콜과 똑같다. 즉, 이 기법은 홉 수로 경로를 선택하기 때문에 가장 짧은 경로를 선택함으로써 재라우팅 시간을 줄일 수 있는 장점이 있는 반면, 네트워크 상황을 제대로 반영할 수 없는 단점을 가지고 있다.

4.3.2 MRB(Maximum Residual Bandwidth) 링크 재라우팅 기법  
이 기법은 네트워크 상황을 제대로 반영할 수 없다는 SHC의 단점을 보완한 기법으로 각 링크의 잔여 대역폭을 기반으로 재라우팅 비용을 계산하는 방법이다. 즉, 장애가 발생한 구간의 재라우팅 경로를 찾기 위해 가장 짧은 홉으로 찾는 것이 아니라 여러 개의 PFC 중에서 잔여 대역폭이 가장 많은 경로로 전송하는 방법이다. 이 기법에서 각 링크의 비용은  $v_l = R_l$ 이다. 이 기법의 장점은 네트워크의 상황을 반영하여 재라우팅 경로를 선택하기 때문에 각 연결에 대한 실패율을 감소시킬 수 있는 반면에 홉 수면에서는 더 긴 경로를 선택할 수 있기 때문에 재라우팅 시간에 있어서는 단점을 가지고 있는 기법이다. 특히 이 기법은 네트워크 자원을 감안하여 재라우팅 경로를 선택하기 때문에 로드 밸런싱(load balancing)의 기능을 제공하고 있다.

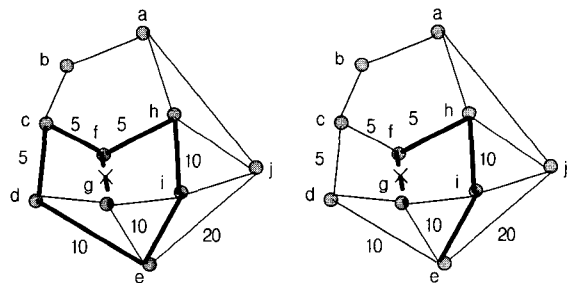
4.3.3 SW(Shortest\_Widest) Directed\_Destination 재라우팅 기법  
이 기법은 다중 매트릭스를 지원하는 재라우팅 기법이다. 먼저 장애 구간에 의해 서비스되지 못한 연결  $k$ 가 재라우팅할 수 있는 여러 개의 가능한 경로가 존재한다고 가정하자. 이 여러 개의 가능한 경로 중에 잔여 대역폭이 가장 넓은 경로를 선택하게 된다. 만약 최대 잔여 대역폭 같은 경로가 여러 개가 존재하면 그 중에서 가장 작은 홉 수를 갖는 경로를 선택하게 되는 다중 매트릭스 기법이다. 이 기법은 중요한 속성은 병목현상이 발생한 링크를 피할 수 있다는 점이다. 따라서 네트워크 자원에 대해 로드 밸런싱(load balancing) 기능을 제공하고 있다는 점이 특징일 수 있으며, 더욱이 홉 수를 기준으로 지연(delay)에 민감한 경로를 선택할 수 있다는 점이다. 따라서 이 기법은 서비스 복구 시간보다는 네트워크 자원에 맞게 서비스를 분산시키는데 주

목적이 있다.  
또한, 이 기법은 장애 구간만 재라우팅 하는 것이 아니라 장애 발생 지점부터 목적지까지 직접 재라우팅을 시도하는 기법이다. 따라서 본 논문에서 제안하는 PFC 기법은 장애 발생 지점에서 가능한 경로를 검색하기 위해 다른 노드들을 선택하는 기준으로 사용되고 있고, 이를 이용하여 목적지로 직접 경로를 선택하기 때문에 PFC를 사용하는 기법보다 지연시간 측면에서 더 짧은 경로들에 대해서 네트워크 상황을 고려하는 기법이다.  
(그림 6)에서 보는 바와 같이 연결  $k$ 가 송신지  $h$ 에서 목적지  $e$ 로 서비스 되고자 할 때 해당 링크  $(h, i)$ 가 장애 구간으로 판별되었다고 가정하자. 이에 PFC 기반의 Directed\_Destination 경로는  $(h, f, g, e)$ 와  $(h, j, e)$ 로 두 개의 경로가 생겨 날 수 있다. 이는 PFC 기반으로 인터페이스가 선택되기 때문에  $h$ 노드에 연결된  $f$ 와  $j$  노드가 선택되기 때문이다. 결과적으로 이 기법을 사용하면 두 개 경로의 잔여 대역폭이 모두 30으로 같기 때문에 지연시간이 가장 짧은(홉 수가 가장 적은)  $(h, j, e)$  경로가 선택되어질 것이다.



(a) SW 기법의 path 선택 전 (b) SW 기법의 path 선택 후  
(그림 6) SW 기법의 경로 선택 과정

4.3.4 WS(Widest\_Shortest) Directed\_Destination 재라우팅 기법



(a) WS 기법의 path 선택 전 (b) WS 기법의 path 선택 후  
(그림 7) WS 기법의 경로 선택 과정

이 기법 역시 다중 매트릭스를 지원하는 재라우팅 기법으로써 연결에 대해서 송신지와 목적지 사이에 가능한 모든 경로들 중에 가장 지연시간인 짧은(홉 수가 가장 적은) 경로를 선택하는 기법이다. 만약 지연시간이 짧은 경로가 여

러 개가 생길 경우 그 중 가장 넓은 잔여 대역폭을 가지고 있는 경로를 선택하게 된다. 따라서 이 기법은 각 연결에 대해 네트워크 상의 분산보다는 각 연결의 서비스 복원 시간에 좀더 장점을 가지는 기법이다.

(그림 7)에서 보는 바와 같이 연결  $k$ 가 송신지  $f$ 에서 목적지  $e$ 로 서비스 되고자 할 때 링크  $(f, g)$ 가 장애 구간으로 판별되었다고 가정하자. 이에 PFC 기반의 Directed\_Destination 경로는  $(f, c, d, e)$ 와  $(f, h, i, e)$ 로 두 개의 경로가 생겨 날 수 있다. PFC 기반으로 인터페이스가 선택되기 때문에  $f$ 노드에 연결된  $c$ 와  $h$  노드가 선택되어진다. 이 두 개의 경로들의 홉 수는 모두 같이 3이기 때문에 가장 넓은 잔여 대역폭( $35 = 5 + 10 + 20$ )을 갖는  $(f, h, i, e)$  경로를 선택하여 재라우팅을 하게 된다.

### 5. 실험 및 분석

#### 5.1 시뮬레이션

본 논문의 시뮬레이션에서 사용하는 네트워크 범위는 노드 수를 20노드로, 회선은 전이중 방식으로 155Mbps(OC-3)로 구성하였으며, 각 회선의 큐잉 요소(Queuing component)는 무시하였다. 또한, 회선의 지연은 대칭적이나 회선의 비용은 비대칭적으로 설정하였다. 즉,  $D(u, v) = D(v, u)$ 이지만,  $C(u, v) \neq C(v, u)$ 이다. 이는 회선의 길이가 같으면 지연도 같게 되지만, 회선의 비용은 전이중 방식으로 다르게 표현되기 때문이다. 본 논문에서 실험 네트워크를 생성하기 위해 NS-2(Network Simulator-2)와 함께 제공하는 GT-ITM 토폴로지 생성기를 사용하여 순수 랜덤 그래프(pure random graph)를 생성하였다. 생성된 그래프의 평균 노드 정도(degree)는 3이상으로 설정했으며, 시뮬레이션 횟수는 연결 요청 횟수로 5000회를 선정하였으며, 시뮬레이션의 신뢰성을 높이기 위해서 5000회 실험을 3회 반복하여 평균값으로 실험치를 도출하였다. 다음은 본 논문의 시뮬레이션을 위해 사용한 가정들이다.

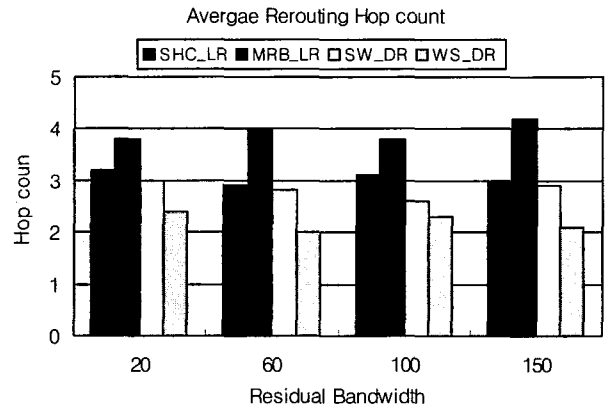
- ① 시뮬레이션 시 네트워크 상의 장애 노드(fault node)는 임의의 노드를 선정하여 지속적으로 실험한다.
- ② 각 연결 요청 도착 간격은  $\lambda=5$ 인 exponential 분포를 따른다.
- ③ 연결 설정 후 데이터 전송 지속시간(duration)은 임의의 시간을 선택한다.
- ④ 다중 송신지와 다중 목적지를 기본으로 한다. 즉, 연결 설정 요청은 20노드에 서 랜덤으로 발생할 수 있으며 수신지는 송신지를 제외한 모든 노드가 될 수 있다.
- ⑤ 모든 회선의 지연시간은 일정하다. 따라서 송신지에서 목적지까지의 지연시간은 중간 경로 수로 결정된다.
- ⑥ 모든 방식에서 사용되는 라우팅 테이블 갱신 주기는 모두 같다.

비교하는 성능 항목으로는 재라우팅 시 사용되는 경로의 길이인 ARH(Average Rerouting Hop count)와 복구를 원하는 서비스에 대해 재라우팅 시 ARB(Average Rerouting Blocking rate)로 나누어 분석하기로 한다.

#### 5.2 결과 및 분석

먼저 본 논문에서 비교하는 성능 항목은 위에서 설명했듯이 ARH와 ARB이다. 이것은 각 연결들이 장애 구간을 만나 재라우팅 기법을 통해 서비스될 때 발생하는 성능 항목이다. 즉, 각 연결들이 재라우팅 기법을 수행할 때 대해서만 평균 홉 수와 평균 실패율을 계산하게 된다.

본 논문에서 제안하는 4가지 기법들에 대해서 ARH와 ARB에 대한 성능을 살펴보면 (그림 8)과 (그림 9)와 같다. 먼저 (그림 8)은 4가지 기법들에 대한 ARH의 성능을 보여주고 있다. 여기서 알 수 있듯이 재라우팅 시 사용되는 총 홉 수는 WS\_DR 기법이 가장 좋은 성능을 보여주고 있다. 즉, 각 연결에 대해 재라우팅을 수행하게 되면 WS\_DR 기법이 가장 짧은 홉을 통해 서비스할 수 있다는 것이다. 이것은 장애 시작 지점에서 목적지까지 직접 재라우팅을 수행함과 동시에 가장 짧은 홉 수에 대한 경로를 선택하고 나서 그 중에서 가장 넓은 대역폭을 갖는 경로를 선택하기 때문이다.

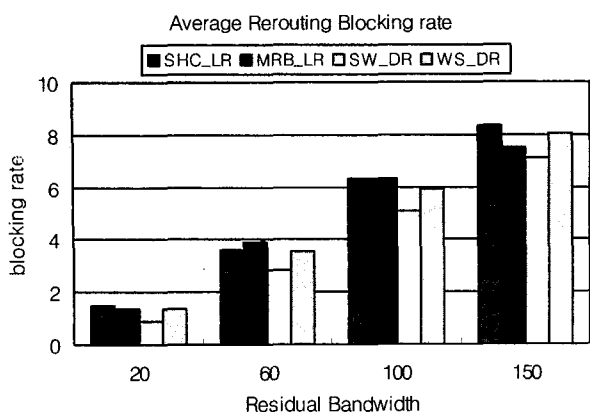


(그림 8) 각 기법들에 대한 ARH 성능 평가

그 다음 성능은 SW\_DR과 SHC\_LR 기법이 거의 비슷한 보여주고 있으며, MRB\_LR 기법이 가장 낮은 성능을 보여주고 있다. 이는 장애 구간에 대해서만 가장 넓은 대역폭을 갖는 경로를 찾기 때문이다. 즉, 홉 수와 상관없이 잔여 대역폭을 매트릭스로 사용하기 때문에 이러한 현상이 발생하게 된다.

(그림 9)은 각 기법에 대한 ARB에 대한 성능 평가를 보여주고 있다. 여기서 볼 수 있듯이 재라우팅 시 각 기법에 대한 ARB에 대한 성능 평가 결과는 SW\_DR 기법이 가장 좋은 성능을 보여주고 있다. 이는 장애 시작 지점에서 목적지까지 갈 수 있는 여러 가지 경로 중에서 가장 넓은 대





(그림 9) 각 기법들에 대한 ARB 성능 평가

역폭을 갖는 경로들을 먼저 선정하고 다음으로 가장 짧은 경로를 선정하기 때문이다. 즉, 장애 시작 지점에서 목적지까지 갈 수 있는 경로가 다른 기법들보다 많이 존재할 수 있기 때문에 각 연결에 대해서 더 많이 서비스할 수 있는 때문이다. 그 다음의 성능으로는 MRB\_LR과 WS\_DR 기법이 비슷한 성능을 보여주고 있으며 가장 낮은 성능을 나타내는 기법으로는 SHC\_LR 기법이다. 이는 장애 구간에 대해서만 가장 짧은 경로를 찾기 때문에 네트워크 상황을 제대로 반영하지 못한다는 점과 장애 구간에 대해 재라우팅할 수 있는 구간이 다른 기법보다 적게 존재하기 때문이다.

## 6. 결 론

본 논문은 네트워크 상에 발생하는 장애 상황에 대해서 각 연결들이 수행하는 재라우팅 기능을 보다 빠르고 효율적으로 제공하기 위한 기법을 연구하였다. 이를 위해 IP 네트워크 모든 노드를 자원 예약 없이 사전에 복구할 수 있는 PFC를 생성하고 이를 기반으로 라우팅 테이블의 갱신 없이도 재라우팅 경로에 대한 정보를 유지함으로써 빠르게 우회로를 찾을 수 있게 되었다. 또한, 재라우팅 최소 비용 문제를 통해 PFC 상의 경로 선택을 사용 메트릭에 따라 최적으로 선택함으로써 최소 비용을 갖는 재라우팅 기법들을 연구하였다. 이러한 재라우팅 기법들은 SHC\_LR, MRB\_LR, WS\_DR, SW\_DR 등으로 이들에 대한 성능은 ARH와 ARB에 따라 각기 다른 성능을 보여주고 있음을 알 수 있다. 즉, ARH를 줄이기 위해서는 장애 구간의 가장 짧은 홉 수를 계산하여 재라우팅해야 하나 이 때 사용하는 재라우팅 비용은 네트워크에서 정적인 값을 갖는 홉 수이기 때문에 네트워크 상황을 전혀 반영할 수 없게 된다. 즉, ARH를 줄이는 기법을 선호하게 되면 반대로 재라우팅을 통해 서비스되는 연결들이 제대로 서비스되지 못하는 현상이 발생하게 된다.

반면, 네트워크 상황을 반영할 수 있는 재라우팅 비용을 사용할 경우에는 ARB가 올라갈 수 있다. 물론 이 기법은

홉 수와 상관없이 가능한 모든 경로에 대한 동적인 값을 기본으로 경로를 결정하기 때문에 ARH는 다소 올라가는 현상이 발생할 수 있다. 하지만 네트워크 상황에 맞게 각 연결을 분산할 수 있다는 점에서 장점을 가지고 있다.

결론적으로 말해 ARH와 ARB는 서로 상관 관계에 있기 때문에 어느 하나의 항목을 위한 기법에 치중해서는 안 되고, 서로 절충하여 최적의 성능을 나타내는 기법을 선택하여 재라우팅 시에 사용해야 함으로 알 수 있었다.

## 참 고 문 헌

- [1] 한정수, 정진욱 "암시적 3단계 정보를 갖는 분산 QoS 라우팅 성능 연구", 정보처리학회논문지 C, 제9-C권 제1호, pp.141-145, 2002.
- [2] Anindo Banerjee "Fault Recovery for Guaranteed Performance Communications Connections," IEEE/ACM Transactions on Networking, Vol.7, No.5, pp.653-668, October, 1999.
- [3] Murali Kodialam and T. V. Lakshman, "Restorable Dynamic Quality of Service Routing," IEEE Communications Magazine, pp.72-81, June, 2002.
- [4] D. Medhi and R. Khurana "Optimization and Performance of Network Restoration Schemes for Wide-Area Teletraffic Networks," Journal of Network and Systems Management, Vol.3, No.3, Sept., 1995.
- [5] Christopher Metz, "IP Protection and Restoration," Internet Computing, pp.97-102, March-April, 2000.
- [6] Demetrios Stamatelakis, Wayne D. Grover "IP Layer Restoration and Network Planning Based on Virtual Protection Cycles," IEEE Journal on Selected Area in Communications Vol.18, No.10, pp.1938-1949, Oct., 2000.
- [7] Muriel Medard, Steven G. Finn, Richard A. Barry, Robert G. Gallager, "Redundant Trees for Preplanned Recovery in Arbitrary Vertex-Redundant or Edge-Redundant Graphs," IEEE/ACM Transactions on Networking, Vol.7, No.5, pp.641-645, October, 1999.
- [8] W-P. Wang, D. Tipper, B. Jaeger, and D. Medhi, "Fault Recovery routing in Wide Area packet Networks," in Proceedings of 15th International Teletraffic Congress, Washington, DC, June, 1997.
- [9] Bartos, R, Raman, M, "A heuristic approach to service restoration in MPLS networks," Communication, 2001, ICC 2001, IEEE International conference on, Vol.1, pp.117-121, 2001.
- [10] Jozsa, B. G, Magyar, G., "Reroute sequence planning for label switched paths in multiprotocol label switching networks," computers and Communications, 2001, Proceedings, 6th IEEE Symposium on, pp.319-325, 2001.
- [11] Ken Owens, Vishal Sharma, Srinivas Makam, Changcheng Huang, "A Path Protection/Restoration Mechanism for MPLS Networks," IETF Draft Jan., 2002.



### 한 정 수

e-mail : jshan@songgang.skku.ac.kr

1997년 성균관대학교 공과대학 정보공학과  
졸업(학사)

1999년 성균관대학교 전기전자 및 컴퓨터  
공학부 석사 졸업

현재 성균관대학교 전기전자 및 컴퓨터공학  
부 박사수로

관심분야 : 네트워크 관리, 트래픽 분석, 인터넷 QoS, QoS 라우팅,  
Rerouting 알고리즘



### 정 진 욱

e-mail : jwchung@songgang.skku.ac.kr

1974년 성균관대학교 전기공학과 학사

1979년 성균관대학교 대학원 전자공학과 석사

1991년 서울대학교 대학원 계산통계학과  
박사

1982년~1985년 한국과학기술 연구소 실장

1981년~1982년 Racal Milgo Co. 객원연구원

1985년~현재 성균관대학교 전기전자 및 컴퓨터공학부 교수

2001년~현재 한국정보처리학회 회장

관심분야 : 컴퓨터 네트워크, 네트워크 관리, 네트워크 보안