

# SONET 망에서의 이중 장애 복구 알고리즘의 설계

한경흠\* · 고재상\*\* · 조충호\*

## Design of Self-Healing Algorithm for Double Link Failure in SONET

Kyoungheum HAN · Jaisang KOH · Choongho CHO

### 〈요 약〉

DCS(Digital Cross-connect System)를 이용한 다이내믹 자체 치유 알고리즘들은 주로 대상을 하나의 링크 장애에 대해서 완전 복구될 수 있는 예비 용량을 갖는 망을 대상으로 복구 시간을 줄이는 방향들이 제시되어 왔으나, 이러한 고장난 망의 복구를 위한 자체 치유 복구 알고리즘의 효율성은 이중 장애나 노드 장애에 적용되었을 때의 복구율을 가지고 평가가 된다. 본 논문에서는 단일 링크(single link) 장애뿐만 아니라 이중 링크(double link) 장애에 대해서도 망의 복구 허용 시간 내에서 복구율을 높히는 새로운 망복구 알고리즘을 제안한다. 이를 위해서 망의 복구 허용 시간 내에, 전체적인 예비 용량을 고려하여 선택된 복구 경로의 각 링크의 부하를 균등하게 하는 복구 방법을 적용한다. 또한 제안된 알고리즘의 시뮬레이션 결과를 기존의 복구 방법과 비교 분석 한다.

### 1. 서론

전기 통신 분야 및 전자 부품분야에서 발전된 최신 기술은 광대역 종합 정보 통신망(B-ISDN)의 실현을 가능하게 하였다. 광대역 종합 정보 통신망은 동화상, 정지 화상, 음성 데이터 정보를 종합적으로 전달하는 망으로서 LAN의 상호 연결, 주문형 비디오(video on demand) 그리고 화상 전화 등의 서비스를 제공할 수 있으며, 동기식 전송망(SONET)은 이러한 광대역 서비스를 전송할 수 있는 전송망이다. 높은 전송능력을 갖은 망과 데이터 트래픽(data traffic)이 증가되는 상황에서, 네트워크에서의 장애는 많은 사용자에게 서비스의 단절을 초래하고, 서비스 제공자에게는 경제적

손실을 유발하며, 서비스 단절에 따른 영향력은 기존 망보다 매우 심각하다. 따라서 망의 장애로부터 빠른 복구는 서비스 사용자와 서비스 제공자 모두에게 필요로 하는 요소이며, 망은 노드 또는 링크의 장애에 대처할 수 있는 능력, 즉, Fault-tolerant와 자체 치유 능력(Self-healing)을 가지고 있어야 한다. 망의 자체 치유를 위해서는 복구에 필요한 예비용량과 망의 장애 진단 결과를 신속히 해석하고 수행할 수 있는 망의 지능적인 요소를 필요로 한다[1][2][3][4][12].

망장애를 대처하기 위하여 현재까지 많은 연구들이 이루어지고 있다. 이러한 연구들은 복구 시간, 비용, 적용할 수 있는 망, 그리고 처리할 수 있는 장애에 대하여 서로 다른 특징을 갖게 되며, 물리 계층에서의

\* 고려대학교 전산학과

\*\* 한국전자통신 연구소

복구 구조(physical layer survivability)와 논리 계층에서의 복구 구조(logical layer survivability)로 나눌 수 있다[3]. 물리 계층에서의 복구 방식은 자동 보호 절체(APS : Automatic Protection Switching), RD(Route Diversity)과 자체치유링(SHR : Self-Healing Ring)이 있다. 물리 계층에서의 복구는 각 링크에 해당하는 예비링크가 존재하여 점대점(point-to-point)방식에 의해서 서비스를 보호한다. 논리 계층에서의 복구 방식은 DCS(Digital Cross-Connection System)에 의해서 망을 재구성한다. DCS에 의한 망의 재구성은 경로 테이블(routing table)을 변경하여 망을 재구성하여 장애를 복구한다. 물리 계층에서의 복구 방식은 60ms이내에서의 빠른 복구가 가능하다는 장점이 있지만 노드 장애 또는 복수의 장애에 대해서는 비용 면에서 효과적으로 복구 할 수 없다는 단점이 있다. 반면 논리 계층에서의 복구 방식은 APS나 SHR와 같이 물리적인 예비 용량을 필요로 하지 않고, 경로 테이블을 재구성하므로 경제적으로, 유연(flexible)하게 망을 복구 할 수 있는 장점이 있다.[3]

DCS를 이용한 복구 알고리즘에는 장애가 발생하기 전에 모든 장애 시나리오에 대하여 복구 경로를 미리 결정하는 방법(preplanned route calculation)과 장애가 발생한후 제어 메시지를 인근 노드들에게 전파하여 복구 경로를 찾아내는 동적인 방법(dynamic route calculation)이 있다[4]. 복구 경로를 미리 결정하는 방법은 장애 시나리오에 대하여 복구 경로를 저장하기 위한 데이터 베이스가 필요하고 빠르게 복구할 수 있지만 트래픽 상황의 변화에 따르는 데이터 베이스의 갱신(update)이 필요하며 각 노드에 맞는 경로 테이블을 새로 계산해야 한다는 문제점과 데이터 베이스를 갱신하였을때, 망의 상태와 장애가 발생하였을때의 망의 상태가 불일치한다는 문제점이 존재한다. 동적으로 복구 경로를 찾는 복구 알고리즘은 유연하게 장애를 복구 할수 있지만, 제어 메시지를 통하여 복구 경로를 찾아내므로 복구 시간이 더 길어 질 수 있는 문제점이 있다. 복구 경로를 동적으로 찾는 복구 알고리즘은 크게 선택할 수 있는 복구 경로중에서 예비 용량이 가장 큰 복구 경로를 선택하는 알고리즘과, 복구 경로를 짧게 하여 복구 시간을 줄여주는 알고리즘

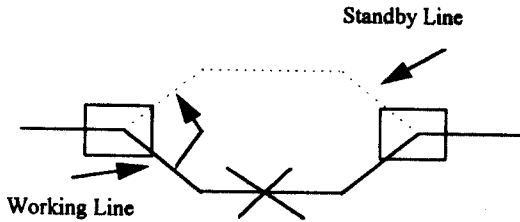
으로 분류할 수 있다. 예비 용량이 큰 복구 경로를 선택하는 알고리즘은 일정한 시간 안에서 복구 경로를 선택하게 되므로 복구 시간이 크며, 짧은 복구 경로를 선택하는 복구 알고리즘은 특정 링크에 트래픽이 집중(congestion)되는 문제가 발생한다.

본 논문은 DCS를 이용하여 동적으로 복구 경로를 찾아 주는 알고리즘으로서, 망의 복구 허용 시간 내에서 예비 용량의 어느 특정 링크에 트래픽의 집중되는 현상을 방지하면서 링크 장애를 복구시키며, 이중 링크 장애(Double Link Failure)가 발생하였을 때, 망의 복구율을 높이는 알고리즘을 제안한다. 본 논문의 2절에서는 기존의 서비스 보호망 구성 방식을 설명한다. 3절에서는 복구 과정에서 발생하는 특정 링크의 트래픽 집중 현상을 최소화 하는 방법을 제시하며, 4절에서는 제안된 복구 방법에 대한 시뮬레이션 결과를 제시하고 5절에서는 결론 및 향후 연구를 제시한다.

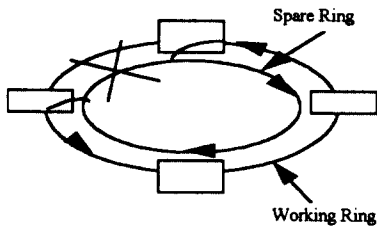
## 2. 기존의 서비스 보호망 구성 방식

서비스 단절 현상을 방지하기 위한 서비스 보호망 구성 방식을 살펴보면 단국 장치를 이용한 자동 보호 절체(APS), 회선 분기 결합 장치를 이용한 자체치유링(SHR) 그리고 DCS 복구 방식이 있다.

APS는 전용 장치를 이용한 보호 방식이며 물리 링크층에서 복구가 이루어진다. 이것은 뜻하지 않은 망의 장애에 대해 신호 복구를 이행하며, 새로운 장치를 설치할 때 지속적인 서비스 제공을 위한 방법으로 쓰인다. APS는 가장 간단하고 빠른 복구 방식이다.[2] 그러나, APS는 운용 선로와 예비 선로가 같은 물리 링크에 존재하므로 링크 절단에 의한 장애를 대처하지 못하다는 단점이 있다. 이런 단점은 <그림 1>에서처럼 예비 선로와 운용선로를 다른 물리 링크에 설치함으로써 링크 절단에 의한 장애를 복구 할 수 있으며, 이런 복구 방법을 RD(Route Diversity)라 한다[6]. SHR은 ADM(Add-Drop Multiplexer)을 이용하며 링형 구조의 망에 적용 가능한 형태이다. 또한, SHR은 APS와 같이 물리층에서 복구가 이루어지며, 노드 사이의 망 노드 장치와 선로들을 서비스 복구 차원에서 서로 공



〈그림 1〉 Route Diversity



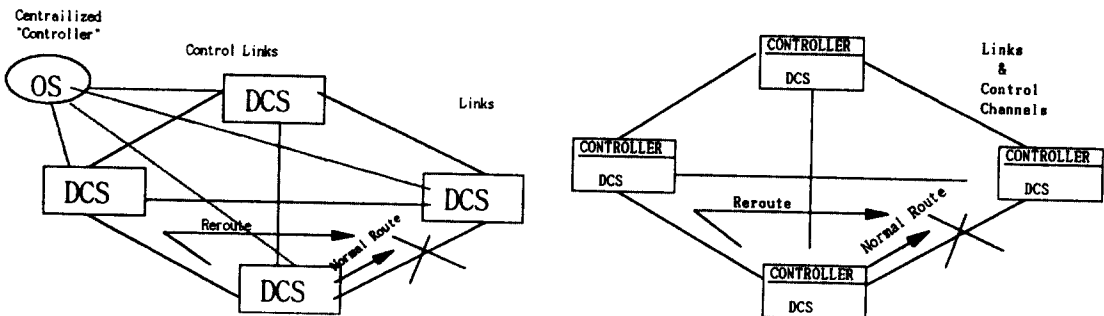
〈그림 2〉 Ring Architecture

유할 수 있고 서비스 자체를 이중화할 수 있다[3]. 〈그림 2〉는 SHR의 복구 과정을 나타내며, 운용링에 장애가 발생하면 예비링으로 데이터는 전송된다. APS와 SHR은 복구를 위한 전용의 예비 용량을 사용하여 장애를 복구하지만 DCS를 이용한 다이내믹 복구는 별개의 전용 보호 장치를 필요로 하지 않고 DCS 경로 테이블의 재구성에 의하여 경제적, 유연한 장애 복구가 가능하다[6].

### 2.1 DCS 복구 방법

망의 생존력(Survivability)을 높이기 위하여 노드나 링크의 장애에 영향을 받는 서비스들을 복구 하기 위해서는 DCS는 액세스 가능한 예비 용량을 가지고 있어야 한다. 이것은 장애 복구를 위하여 DCS가 망 재구성을 용이하게 해준다[4]. DCS의 라우팅(routing)기능은 망에 장애가 발생하였을 때 서비스를 복구하는데 사용된다. DCS는 복구되어야 할 경로의 cross-connects를 변경하여 다른 경로(alternative route)로 복구시키며, 선택가능한 복구 경로는 예비 용량이 존재하여야 한다.

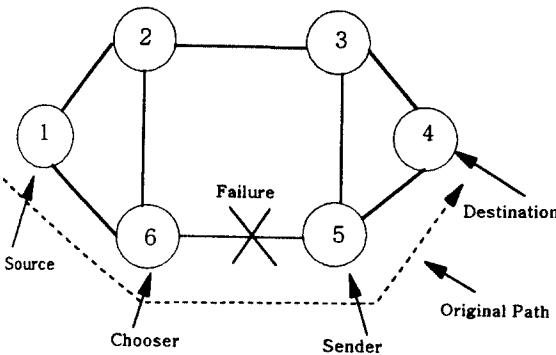
지금까지 DCS 재구성에 의한 복구 방법은 많은 연구가 이루어지고 있다. DCS 재구성에 의한 복구 방법은 〈그림 3〉과 같이 제어 방법에 따라 중앙 제어 방식과 분산 제어 방식으로 구분된다[4]. 중앙 제어 방식은 모든 복구 과정이 하나의 중앙 제어기에 의해 제어된다. 즉, 장애를 검출한 노드가 장애 발생 사실을 중앙 제어기에 알려면, 중앙 제어기는 중앙 제어기가 가지고 있는 네트워크의 모든 정보를 이용하여 장애가 발생된 용량을 다른 경로로 우회시켜 서비스를 복구한다. 분산 제어 방식은 각 DCS가 자신이 가진 정보를 다른 DCS와 교환함으로써 얻어지는 망의 정보를 이용하여 서비스를 복구한다. 분산 제어 방식에서 각 DCS간의 정보 교환은 제어 메시지를 이용하여 정보를 교환하며, 메시지가 갖는 정보는 장애가 발생된 링크, Sender와 현재의 노드까지의 거리를 나타내는



〈그림 3〉 중앙 제어 방식과 분산 제어 방식

Hop Count, 제어 메시지가 통과한 경로와 그 경로의 복구 가능한 용량 등의 정보를 가지고 있다[6].

여기에서 Sender는 장애가 발생한 링크의 두 노드 중 데이터 흐름의 순방향(Forward)에 있는 노드이며, 역방향(Backward)에 있는 노드는 Chooser가 된다. 또한 장애가 발생한 링크의 경로에서 데이터를 발생한 노드를 시작노드(Source)라 하며, 데이터를 수용하는 노드를 목적지 노드(Destination)라 한다. 예를 들면, 노드 1번에서 4번으로 데이터를 전송하려하고, 선택된 경로 1-6-5-4로 데이터가 전송될 때, 6번 노드와 5번 노드사이에서 링크장애가 발생하였을 때, <그림 4>는 위에서 정의한 노드들을 보여주고 있다.[11]



<그림 4> 정의된 노드

일반적으로, DCS 재구성에 의한 복구 알고리즘들은 다음의 5가지의 기본 단계를 거친다[1].

① Detection and Notification

중앙 제어 방식에서는 DCS가 장애를 검출하고 중앙 제어기에게 장애 발생 사실을 알리며, 분산 제어 방식에서는 DCS가 각 채널을 모니터 한 후 장애를 인식한다.

② Identification

중앙 제어 방식에서는 장애를 인지하고 복구 과정을 시작한다. 분산 제어 방식에서는 장애 사실을 DCS들에게 알린다.

③ Route Selection

알고리즘에 의하여 적당한 복구 경로(route)를 찾는 단계이다. 이 단계는 알고리즘의 성격을 구분 지을 만큼 중요한 단계이며, 일반적으로 다음의 3가지 방법

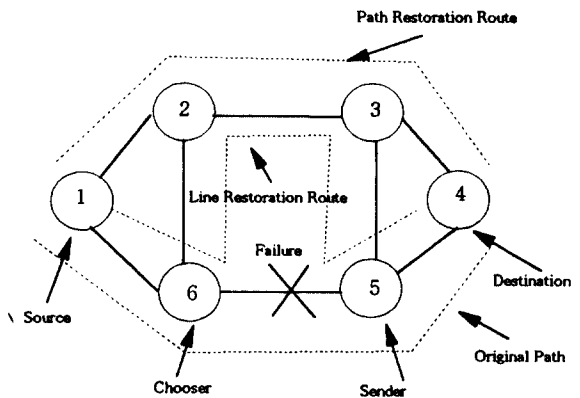
으로 알고리즘이 수행된다.

a. 제어 방법 : 중앙제어와 분산제어 방법[4]

중앙제어 방법은 중앙제어 시스템이 복구 경로를 선택하며, 선택된 복구 경로에 예비용량을 할당한다. 분산제어 방법은 각각의 DCS가 제어 메시지의 교환에 의해서 복구 경로와 예비 용량을 결정한다.

b. 복구 경로의 형태 : 선로(line) 복구와 경로(path) 복구[6]

선로 복구 방법은 장애를 검출한 노드가 복구를 시작하며, Sender부터 Chooser까지의 경로를 다른 경로로 대체하여 복구한다. 경로 복구 방법은 시작 노드와 목적지 노드사이에서 복구 경로를 다른 경로로 대체하여 복구하는 것이다. 예를 들면 노드 1에서 노드 4로 데이터를 전송할 때, 전송 경로가 1-6-5-4이며, 5-6 링크에 장애가 발생하였다면, 선로 경로는 5-6링크를 대체할 경로로 6-2-3-5를 선택할 수 있고 복구 후의 경로는 1-6-2-3-5-4가 될 수 있으며, 경로 복구는 노드 1과 노드 4사이에서 복구가 이루어지므로, 1-2-3-4경로로 복구 할 수 있다. 선로 복구 방식은 빠르고, 쉽게 구현할 수 있고, 경로 복구 방식은 망 예비용량을 효율적으로 사용할 수 있고, 링크 장애뿐만 아니라 노드 장애에 대해서도 복구 할 수 있는 장점이 있다.



<그림 5> 선로 복구와 경로 복구

c. 복구 경로 계산 방법 : 계획된(preplanned)복구 경로와 동적(dynamic)복구 경로[10]

계획된 방법은 여러 가지 장애 시나리오에 대해서 복구 경로를 미리 계산하여 각 노드의 데이터베이스

에 복구 경로를 저장하는 방법이며 동적 방법은 링크 또는 노드의 운용상태와 링크의 예비용량과 같은 현재의 망 상태 정보와 망의 노드사이에서의 복구 제어 메시지를 교환하여 얻어지는 정보를 이용하여 복구를 한다. 계획된 방식은 빠르게 복구할 수 있는 장점이 있지만 트래픽 상황의 변화에 따라서 데이터 베이스의 갱신(update)이 필요하며 각 노드에 맞는 경로 테이블을 새로 계산해야 한다는 문제점과 데이터 베이스를 갱신하였을때, 망의 상태와 장애가 발생하였을때의 망의 상태가 불일치한다는 문제점이 존재한다. 동적방법은 현재의 망 상태를 각 DCS간에 제어 메시지를 교환하여 복구하므로 유동성있게 복구 할 수 있으며 계획된 방식처럼 모든 장애 시나리오에 대한 복구 경로를 가지고 있지 않으므로 노드 비용(cost)은 상대적으로 높다

#### ④ Rerouting

중앙제어 방식에서는 중앙제어기가 DCS에게 특정 cross-connect 명령어를 보내며, 이 명령어에는 분배해야 할 채널의 수에 대한 정보가 포함되어 있다. 분산 제어 방식에서는 Routing Selection 단계에서 얻어진 분배해야 할 채널의 수만큼 예비 채널로 손상된 채널을 분배하게 된다.

#### ⑤ Return-to-Normal

장애가 복구되었음을 확인하고 정상동작으로 돌아간다.

## 2.2 DCS 복구 알고리즘의 성능 평가

DCS를 이용한 복구 알고리즘은 망장애 복구율과 망장애 복구시간 두가지 기준을 가지고 성능을 평가한다.[1]

#### ① 망장애 복구율 (NRR : Network Restoration Ratio)

장애로 인한 발생된 트래픽에 대해서 복구 알고리즘이 복구한 트래픽의 비율로 표시한다. NRR은 복구 알고리즘의 효율성을 표현하며, NRR이 높으면 망의 생존력은 증가하게 된다. 만약 알고리즘에서 사용되는 시범망(test network)의 예비 용량이 단일 링크 장애에 대하여 100% 복구 될 수 있게 설계된다면, 단일 링크 장애(Single Link Failure)는 NRR에 대하여 알고

리즘의 효율성에 의미를 부여하지 못하는 경우가 발생한다. 따라서, 알고리즘의 효율성을 보다 객관적으로 측정하려면 이중 링크 장애나 노드 장애에 대한 복구율을 측정하여야 한다.

#### ② Network Restoration Time (NRT)

NRT는 복구 알고리즘이 시작하여, 마지막 채널이 복구되는 시간으로 표현한다. NRT가 적다는 것은 망의 가용도(Availability)가 높다는 것을 의미한다. 2초의 NRT는 서비스의 단절로 미치는 영향을 최소화하기 위해 DCS 구조에서 목표로 잡는 시간이다.

## 2.3 DCS를 이용한 알고리즘들

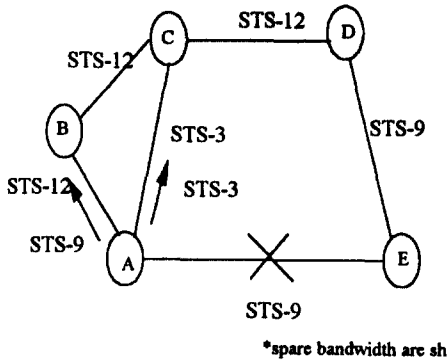
다음 두 알고리즘은 DCS를 이용한 알고리즘들이며, Route Selection에서 사용된 방식은 분산제어 방식이고, 복구 경로는 장애가 발생하였을 때 DCS간에 제어 메시지의 교환에 의하여 동적으로 복구 경로를 찾아내며(dynamic route calculation), 복구 경로는 장애를 검출한 두 노드 사이에서 이루어지는 선로 복구 방식이다.

#### 2.3.1 FITNESS 알고리즘[5]

Chooser에서 복구 메시지를 선택할 때 어느 일정한 시간 동안에 time-out을 두고 그 시간 동안 Chooser가 수신한 제어 메시지를 중에서 복구 가능한 예비 채널을 가장 많이 담고 있는 하나의 메시지를 선택한다. 따라서 Chooser는 어느 일정한 시간 동안에 예비 용량이 가장 큰 복구 루트를 선택할 수 있다. 복구에 사용되는 제어 메시지는 감소하지만 여러 번의 복구 처리 과정이 필요하다. 복구 처리 과정은 다음의 5단계로 구성되어 있다.

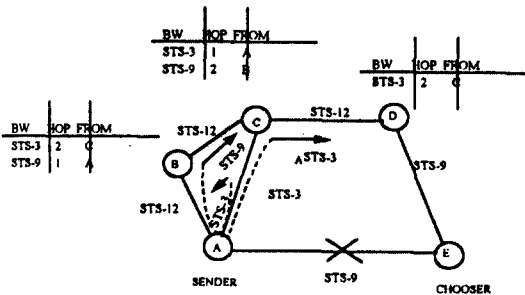
① Send 단계 : Sender는 장애를 검출하고 Help 메시지를 만들어 Sender의 인근 노드들에게 전송한다. Help 메시지는 복구 경로의 Hop수, 복구 경로, 복구 경로가 복원 할 수 있는 용량 등의 정보를 교환한다.

② Spread 단계 : 중간노드들은 수신된 Help 메시지 중에서 가장 작은 Hop Count, 또는 가장 큰 예비 채널을 가진 메시지만을 새로이 저장하며, Help 메시지를 Chooser에게 갈 수 있도록 모든 인근 노드들에게 전



〈그림 6〉 Sender 단계

송하다. 이때, Help 메시지의 Hop Count는 하나 증가하고 Help 메시지가 지나쳐온 경로에 현재의 노드가 포함된다. 그리고 Help 메시지가 복구 할 수 있는 용량과 인근 노드의 용량의 링크를 비교하여 적은 용량을 Help 메시지가 복구 할 수 있는 용량으로 간주하여 그 내용을 갱신한다.

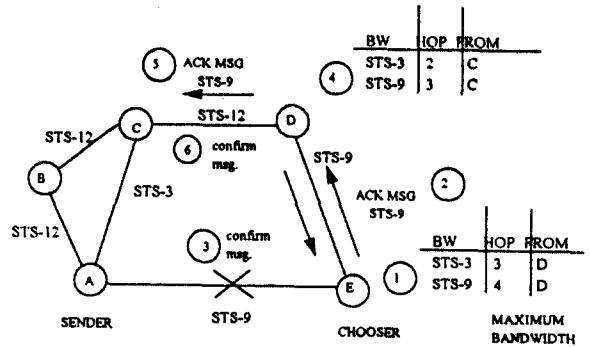


〈그림 7〉 Spread 단계

③ Choose 단계 : Chooser가 Help 메시지를 받으면, 그 Help 메시지를 즉시 처리하지 않고, 일정 시간 동안 여러개의 Help 메시지를 받고, 일정 시간이 경과된 후 가장 많은 예비 용량을 갖는 메시지를 선택한 후에 ACK 메시지를 만들어 Sender를 향해 되돌려 보낸다.

④ Confirmation 단계와 Acknowledge 단계 : ACK 메시지를 수신한 중간노드는 복구에 사용되는 링크의 용량이 충분한가를 검사하여 이전 노드로 Confirmation 메시지를 전송하며, Confirmation 메시지를 받으면

면 Cross-connection은 이루어진다.



〈그림 8〉 Chooser 단계 and Confirmation 단계

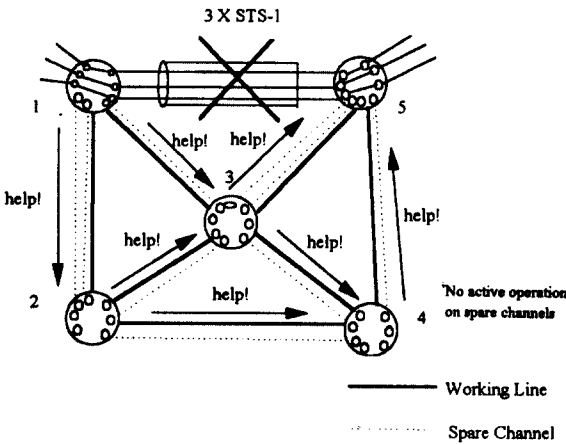
⑤ Cancellation 단계 : 중간 노드가 더 이상의 복구 경로를 찾을 수 없으면, 모든 이전 노드로 Cancel 메시지를 보내며, 탐색된 경로에 대한 복구는 정지된다. 이 단계는 복수의 노드 혹은 링크에 장애가 발생하는 경우에 필요하다. FITNESS 알고리즘에서는 위의 5가지 단계를 거쳐 복구하며, 5가지 단계를 1사이클(Cycle)이라 하며, 1사이클동안에는 Chooser는 오직 하나의 Return 메시지를 전송한다. FITNESS 알고리즘에서는 장애를 1사이클만에 복구 못하는 경우도 있고, 이런 경우에는 여러번의 사이클을 수행한다.

### 2.3.2 TRANS 알고리즘[6]

DCS를 이용한 분산 제어 방식을 사용하고 링크나 노드의 장애에 대하여 경로와 선로 수준에서 복구할 수 있다. 이 알고리즘은 3개의 단계로 구성되는데 각 노드는 제어 메시지에 따라 단계가 변화된다. 3단계 복구 과정은 다음과 같으며, FITNESS 알고리즘과 구분되는 특징은 Chooser가 복구에 필요한 제어 메시지를 받으면 일정한 시간동안 제어메시지를 수신한 후 처리하는 것이 아니라, 제어메시지를 받는 즉시 처리한다.

① Broadcast 단계: 장애를 검출하고, Sender는 모든 이웃 노드들에게 Help 메시지를 전송한다. 각 중간 노드들은 이 메시지를 메모리에 저장한 다음 이 메시지가 지나오지 않은 모든 이웃 노드들에게 전송한다. Help 메시지는 복구 경로, 복구 경로가 복원할 수 있

는 용량, 복구 경로의 Hop수대 대한 정보를 포함하여야 하며, 인근 노드들에게 Help 메시지를 전송할 때, Help메세지의 정보는 다음과 같이 갱신된다. 첫째 복구 경로는 Help 메시지가 지나온 노드들을 포함하며, Hop수는 Help 메시지가 지나온 노드수를 표현하고, Help 메시지의 복원 할수 있는 용량은 복구 경로의 각 링크가 가지고 있는 최소 예비용량이다.

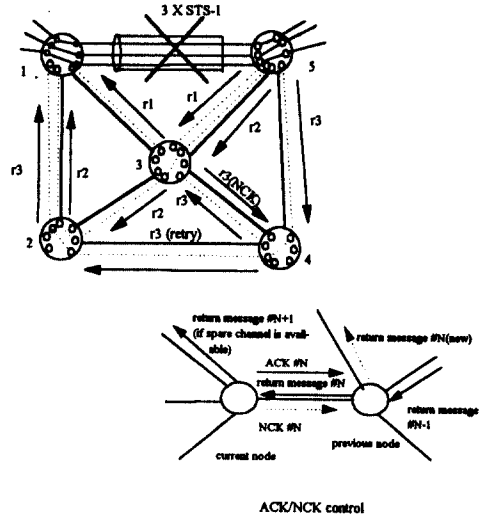


〈그림 9〉 Broadcast 단계

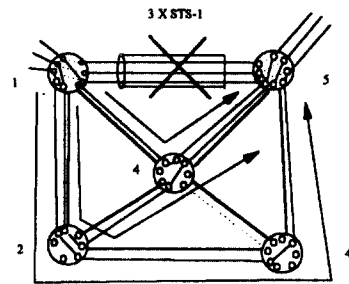
② Selection 단계: Help 메시지가 Chooser에 도착하면, Chooser는 Help 메시지를 받자마자 Help 메시지가 거친 경로가 복구할 수 있는 채널의 수만큼 Return 메시지를 전송한다. Chooser가 보낸 Return 메시지를 중간 노드가 받으면, 중간 노드에서는 Broadcast단계에서 저장한 복구 메시지를 이용하여 Sender까지 최대 짧은 경로가 되는 인근 노드를 선택한 후 선택된 인근 노드로 Return 메시지를 전송한다. 만약 전송할 선로의 예비 용량이 없으면 이전 노드로 NCK 메시지를 보내어 이전 노드에서 재전송이 될 수 있게 한다.

③ Confirm 단계: Sender는 Return 메시지를 받으면 손실된 채널들을 복구 채널로 스위칭 시키고, Link 메시지를 만들어 복구된 경로를 따라 Chooser에게 보낸다.

FITNESS 알고리즘은 Chooser가 복구 경로를 선택할 때 복구 경로의 용량을 고려하여 허용된 복구 시간 내에 복구가 이루어 졌으며, TRANS 알고리즘은



〈그림 10〉 Selection 단계

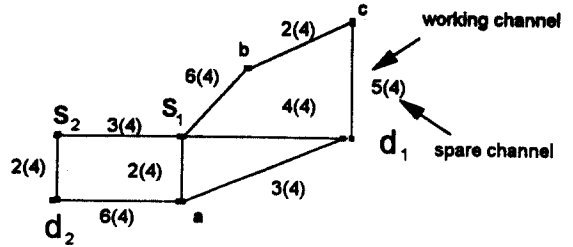


〈그림 11〉 Confirm 단계

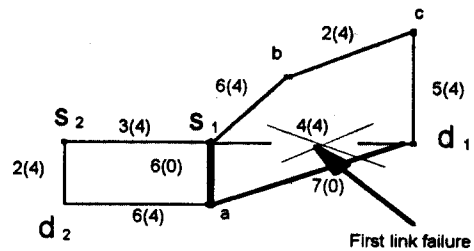
복구 경로의 길이를 고려하여 허용된 복구 시간 내에 복구가 이루어 졌다. FITNESS 알고리즘은 일정한 시간 안에서 복구 경로를 선택하게 되므로 복구 시간이 크며, TRANS 알고리즘은 특정 링크에 트래픽이 집중되는 문제가 발생한다.

본 논문은 DCS를 이용하여 동적으로 복구 경로를 찾아 주는 알고리즘으로서, 망의 복구 허용 시간 내에서 예비 용량의 어느 특정 링크에 트래픽의 집중되는 현상을 방지하면서 링크 장애를 복구시키며, 이중 링크 장애가 발생하였을 때, 서비스 복구율을 높이는 알고리즘을 제안한다. 또한, 복구 시간과 링크의 예비

용량을 고려하여 허용된 복구 시간 내에 예비 용량의 부하를 균등하게(Load balancing)해주는 복구하는 알고리즘을 제안한다. FITNESS 알고리즘과 TRANS 알고리즘에서 사용된 시범 망의 예비 채널은 단일 링크 장애에 대하여 100% 복구가 되도록 설계되었기 때문에 단일 링크 장애 시나리오는 복구 알고리즘의 효율성을 검증하는 복구율에 대한 검증을 못하므로 이중 장애 링크 시나리오를 통하여 제안된 알고리즘의 효율성을 검증하며 망의 장애로 인한 서비스 단절 현상을 최소화하기 위해 DCS를 이용한 복구 알고리즘은 2초의 복구 시간을 가져야 하며[11], 본 논문에서는 제안된 알고리즘이 복구 시간을 만족함을 보인다.



〈그림 12〉 Simple Topology



〈그림 13〉 First Link Failure

### 3. 수정된 TRANS 알고리즘

#### 3.1 문제 분석

DCS를 이용한 TRANS 알고리즘의 복구 방법은 복구 시간에 큰 중점을 두고 있으므로 Route Selection 단계에서 적당한 다른 루트를 선택할 때 짧은 루트를 선택하게 되므로 특정 예비 링크의 부하가 집중되며, 이로 인해 다른 어떤 링크에 장애가 발생하였을 때 복구를 못하는 경우가 발생한다.

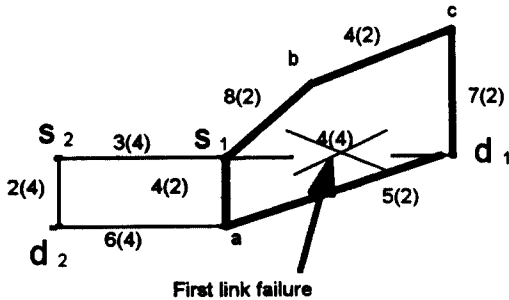
〈그림 12〉는, 현재 사용 중인 링크의 용량과 예비 용량을 나타낸다. 단순하게 각 채널의 대역폭은 1 unit 라고 가정할 때,  $s_1$ 과  $d_1$ 사이에서 링크 장애가 발생한다면 복구해야 할 용량이 4 unit가 되며, TRANS 알고리즘은 복구 경로가 짧은 경로는 선택하게 되므로,  $S_1, b, c, d_1$  경로,  $s_1, s_2, d_2, a, d_1$  경로 그리고  $s_1, a, d_1$  경로 중에서 가장 짧은  $s_1, a, d_1$  경로를 선택하여 복구한다. 복구된 후의 사용 중인 링크의 용량과 예비 용량은 〈그림 13〉과 같다. 만약 정상 복구가 이루어지기 전에 또 다른 링크 장애가 발생하게 되면, 즉,  $s_1$ 과  $d_1$  사이에 장애가 상태가 계속 지속되고 있고  $s_2$ 와  $d_2$  사이에 장애가 발생할 경우에는 복구해야 할 용량이 2 unit라고 하면 〈그림 13〉에서 보는 것처럼  $s_2$ 와  $d_2$  사이에 장애에 대해서는 복구가 이루어지지 않는다.

이와 같은 문제는, 다음 노드를 선택할 때 현재 노드에서 Sender까지의 길이만 고려하는 것이 아니라

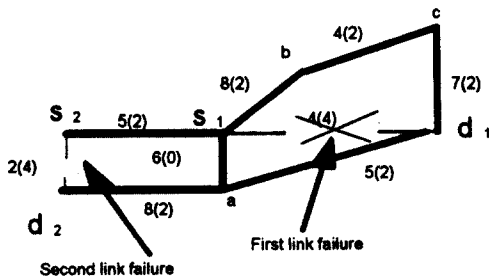
Sender까지의 길이와 함께 복구 경로의 부하를 고려한다. 즉, 제어 메시지를 받은 노드가 다음 노드를 선택할 때, 현재 노드와 인근 노드를 연결하는 링크를 어떤 임계값과 비교하여 임계값보다 예비 용량이 적으면, 링크와 연결된 인근 노드를 HeavyLoad 상태로 분류하고, 임계값보다 예비 용량이 많으면 링크와 연결된 인근 노드를 LightLoad 상태로 분류한다. 현재의 노드가 인근 노드를 선택할 때 우선적으로 LightLoad에 있는 노드들 중에서 하나의 노드를 선택하게 된다. 그리고 복구 경로를 찾는 과정에서 제어 메시지를 받은 현재 노드가 다음 노드를 찾으려고 할 때 LightLoad 상태인 인근 노드들이 존재하면 그 LightLoad 상태의 노드들 중에서 Sender노드와 Chooser노드 사이의 복구 경로가 짧게 될 수 있는 노드를 선택한다. 만약 LightLoad 상태의 어떠한 노드도 존재하지 않는다면, HeavyLoad 상태에 있는 노드들 중에서, Sender노드와 현재 노드 사이의 복구 경로의 길이가 짧게 될 수 있는 인근 노드를 선택하게 된다. 가령 HeavyLoad 상태를 예비 용량이 2이상이고, 예비 용량이 2보다 작을 때를 LightLoad 상태로 간주할 때, 〈그림 12〉에서  $s_1$ 과  $d_1$ 사이에서 장애가 발생하여 4 unit가 복구 되어 할



때, <그림 14>와 같이 2 unit는  $s_1, a, d_1$  루트로 복구된 후,  $s_1, a$  링크는 HeavyLoad상태가 되므로 나머지 2 unit는  $s_1, b, c, d_1$ 로 복구된다. <그림 15>에서는,  $s_1$ 과  $d_1$ 사이에서 장애가 정상 복구로 이루어지기 전에  $s_2$ 와  $d_2$ 사이에서 장애가 발생할 경우에  $s_2$ 와  $d_2$ 사이의 2unit는  $s_2, s_1, a, d_2$ 를 경유하여 복구 할 수 있다.



<그림 14> First Link Failure



<그림 15> Second Link Failure

3.2. 제안 알고리즘

본 알고리즘의 기본 절차는 TRANS 알고리즘을 따른다. 따라서, TRANS 알고리즘의 Broadcast 단계와 Confirm단계는 동일하며, Selection 단계에서 짧은 경로만 고려하는 것이 아니라 예비 용량을 고려하며 망 전체의 링크 부하를 균등하게 하여 또 다른 장애가 발생하였을 때 복구를 가능하게 해준다. 먼저 각 노드와 링크는 각각 고유의 숫자가 부여되었다고 가정을 하며, 다음을 정의하자

$cn$  : current node

$block\_list$  : a set of links which have no spare channel

$con(n)$  : {  $l \mid l$  is the link number which is connected to node  $n$ , not traversed and  $spc(l) > 0$  }

$spc(l)$  : a number of spare channels of link  $l$

Threshold : a number of spare channels separating Light link load and Heavy link load

$light\_links(n)$  : {  $l \mid l \in con(n)$  and  $spc(l) > Threshold$  }

$heavy\_links(n)$  : {  $l \mid l \in con(n)$  and  $0 < spc(l) \leq Threshold$  }

$d(l)$  : distance (a number of hops) between sender node and the node which is connected to link  $l$  and is not  $cn$

TRANS 알고리즘의 2단계인 Selection 단계와 제안된 알고리즘의 Selection은 다음과 같다.

```

<TRANS 알고리즘>
cn = chooser
block_list = {}
do {
  If (  $\exists x$ , such that  $x \in con(cn)$  AND  $x \notin block\_list$  )
    y = min {  $y \mid y = d(x)$  }
    cn = the node connected to the y link
  Else
    insert cn to block_list
    cn = 이전 노드
    이전 노드로 return메시지를 전송한다.
} while (cn != sender)

<제안 알고리즘>
cn = chooser
block_list = {}
do {
  If (  $\exists x$ , such that  $x \in light\_links(cn)$  AND  $x \notin block\_list$  )
    y = min {  $y \mid y = d(x)$  }
    cn = y
  Else If (  $\exists x$ , such that  $x \in heavy\_links(cn)$  AND  $x \notin block\_list$  )
    y = min {  $y \mid y = d(x)$  }
    cn = y
  Else
    insert cn to block_list
    cn = 이전 노드
    이전 노드로 return메시지를 전송한다.
} while (cn != sender)
    
```

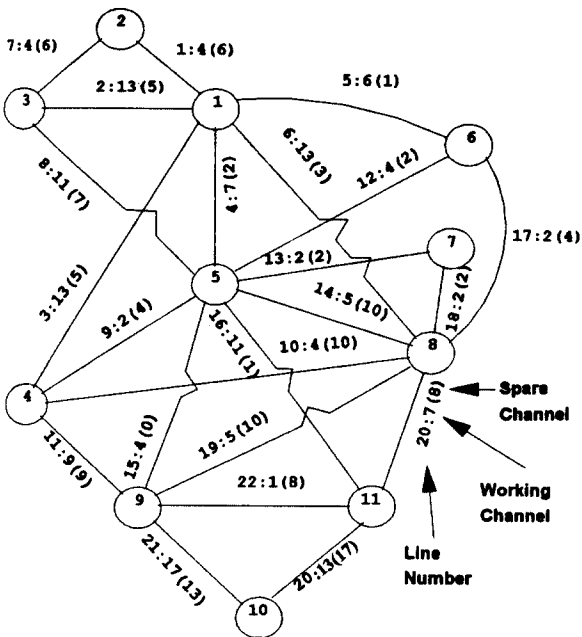
4. 시뮬레이션

4.1 시뮬레이션 환경과 조건

시뮬레이션은 TRANS 복구 알고리즘에서 제안된 알고리즘을 사용하였을 때와 사용하지 않을 경우를 비

교하였다. 제안된 알고리즘의 성능을 비교하기 위해서 TRANS 알고리즘에서 사용된 것과 똑같은 조건을 시뮬레이션에 적용한다.

- 메세지의 프로세싱 타임을 1-9ms의 임의의 값을 갖는다.
- 메세지의 전송률은 64Kbps로 한다.
- 3 단계에서 각각 사용되는 Search, Selection, Link 메세지는 20byte로 구성되었다고 가정한다. 그러므로 시뮬레이션에서 각 메세지 전송 시간은 2.5ms이다.
- cross-connection되는 시간은 100ms이다.
- Sender와 Chooser사이의 복구 경로의 길이는 7을 넘지 않는다.
- 시뮬레이션에 사용된 망 모델은 <그림 16>과 같다. 이 망 모델은 미국의 실제 LATA(Local Access Transport Area)중 한곳을 근거로 만들어졌다. 이 망 모델은[5][6]에서 사용되었으며, [6]에서 사용된 링크의 용량과 예비용량을 사용하였다.
- 제안된 알고리즘에서 HeavyLoad 상태와 Light-



<그림 16> Network Model

Load 상태의 임계값은 예비 용량의 2를 기준으로 하였다. 예비 용량이 3 이상이면 LightLoad 상태이고 2 이하이면 HeavyLoad 상태로 간주하였다.

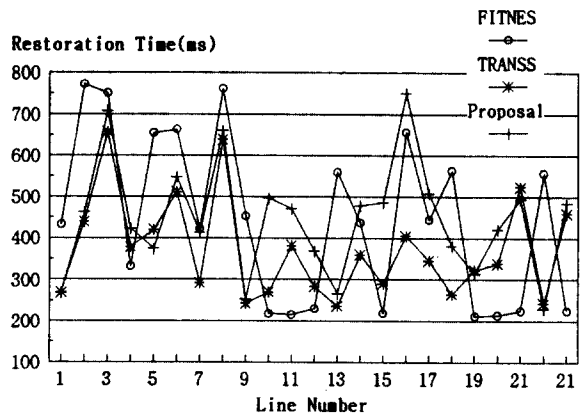
#### 4.2 시뮬레이션 결과 및 분석

시뮬레이션은 링크에 장애가 발생하였을 때의 복구 시간과, 복구율, 복구 된 후에 예비 링크의 과부하 (Congestion)를 FITNESS, TRANS와 제안 알고리즘과 비교하였다. 또한 링크가 장애가 계속 지속된 상태에서 또 다른 링크 장애가 발생하였을 때의 복구율을 비교하였다.

첫 번째 링크에 장애가 발생하였을 때 복구율은 FITNESS, TRANS알고리즘과 제안 알고리즘은 모두 100% 복구가 되었다.

##### 4.2.1 복구 시간

<그림 17>은 23개중 하나의 링크에 장애가 발생하였을 때 복구 시간을 나타내고 있다. FITNESS알고리즘은 전반적으로 TRANS알고리즘보다 복구 시간이 많이 걸린다. 이것은 복구 제어 메시지를 선택할 때 일정 시간동안 가장 많은 예비 용량을 선택하고, 선택된 복구 경로의 길이를 고려하지 않았기 때문이다. TRANS의 최대 복구시간과 제안된 알고리즘의 최대 복구시간의 차이는 약 100ms이며, 두 알고리즘의 복구 시간은 비슷한 분포를 나타내고 있다. 두 알고리

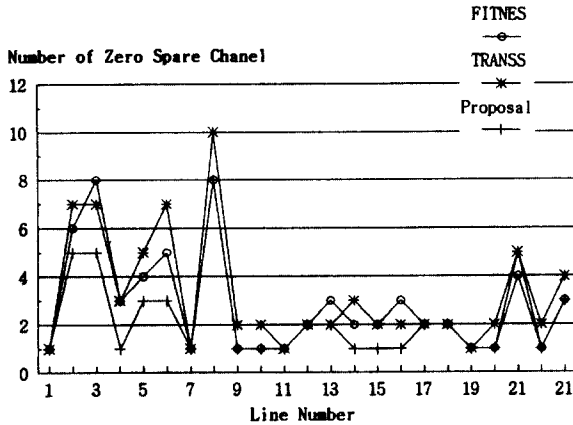


<그림 17> 하나의 링크장애에 대한 복구시간

증의 복구는 모두 1초보다 낮은 시간 시간에 복구된다. 따라서 이 수치는 DCS 자체 치유 망의 서비스 복구 요구 시간인 2초를 만족시킨다

4.2.2 예비 링크의 congestion

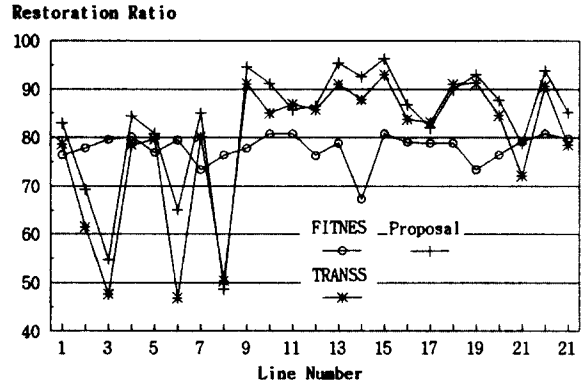
장애 링크를 복구한 후, 예비 링크의 트래픽 집중 현상은 각 링크를 복구한 후 예비 링크의 용량이 0인 링크의 수로 간접 비교하였다. <그림 18>에서처럼 복구된 후 예비 링크의 용량이 0인 링크의 수는 제안된 알고리즘에서 적은 것을 알 수 있다.



<그림 18> 단일 링크 장애 후 예비 용량이 0인 링크의 수

4.2.3 이중 링크 장애에 대한 복구율

<그림 19>는 첫 번째 장애가 발생하고 그 장애를 복구하였지만 장애 링크가 정상 복구되지 않았을 경우에 또 다른 링크 장애가 발생하였을 경우의 복구율을 나타내고 있다. <그림 19>에서의 복구율은 두 번째 장애가 모든 링크에 일어날 수 있다고 가정하여 첫 번째 링크가 정상 복구가 이루어지지 않은 상태에서 각 링크에 두 번째 장애가 발생하였을 때의 평균 복구율을 나타낸 것이다. <그림 19>에서 보는 것처럼 제안된 알고리즘은 모든 링크에서 복구율은 향상되었으며, FITNESS알고리즘보다 복구율이 증가하였다 또한 LATA망에서의 이중 링크 장애가 발생하였을 때의 전체 복구율은 FITNESS알고리즘의 79%에서 제안된 알고리즘은 83%으로 향상되었으며, 제안된 알고리즘의



<그림 19> 이중 링크 장애 후의 복구율

효율성이 보다 나음을 알 수 있다.

5. 결론 및 연구 방향

본 논문에서는 링크에 장애가 발생하였을 때 시간과 용량을 고려하여 허용된 복구 시간 내에 예비 용량의 부하가 균등하게 되면서 장애를 복구하는 알고리즘을 제안하였다. 제안된 알고리즘에서는 단일 링크 장애에서는 100% 복구율을 이루었으며 이중 링크에서는 예비 용량의 과부하를 줄여서 복구율은 기존 TRANS방식보다 높아졌고, 제안된 복구 알고리즘은 효율성은 증가하였다.

앞으로의 연구 과제로는 LightLoad와 HeavyLoad의 임계값이 일정한 상수 값이 아닌 망 용량과 예비 용량의 비율로 선택하는 문제를 다룰 수 있고, Selection 단계에서 발생하는 Return 메세지 수를 줄임으로서 대기 지연시간을 감소시키는 연구가 필요하며, 예비 용량을 초기에 지정하는 할당 메카니즘은 이중 장애 링크를 고려해야 할 것이다. 또한 각 링크에서 예비용량의 이용률을 성과측정계수를 제시하는 연구가 필요 하겠다.

[참고 문헌]

[1] Bellcore Special Report , Digital Cross-connect systems in Transport Network survivability, SR-NWT-002514, Issue 1, 1993

- [2] T. Wu, "A Novel Architecture for Optical Dual Homing Service Fiber Networks", Proc. IEEE GLOBECOM'90, pp.309.3.1-7, 1990
- [3] A. Zolfaghari and F. J. Kaudel, "Framework for Network Survivability Performance", IEEE Journal on Selected Areas in Comm.(JSAC), Vol 12. No.1, pp 46-51, 1994.
- [4] J. Sonsosky "Service Applications for SONET DCS Distributed Restoration", IEEE JSAC, Vol 12. No. 1, pp59-68, 1994
- [5] C. Yang "FITNESS Failures Immunization Technology for Network Service Survivability", Proc. IEEE GLOBECOM'88, pp.47.3.1-6, 1988
- [6] S. Hasegawa, "Control Algorithms of SONET Integrated Self-Healing Network", JSAC, Vol 12. No.1, pp.110-118, 1994
- [7] Y. Kajiyama, N. Tokura and K. Kikuchi, "An ATM VP-Based Self-Healing Ring", JSAC, Vol 12. No.1, pp.171-178, 1994
- [8] Y. Kajiyam, "An ATM VP-Based Self-Healing Ring", JSAC, Vol 12. No.1, pp.171-178, 1994
- [9] J. Sosnosky, "Service Applications for SONET DCS Distributed Restoration", JSAC, Vol 12. No.1, pp. 59-68, 1994
- [10] R. Kawamura, K. Sato and Ikuo Tokizawa, "Self-Healing ATM Networks Based on Virtual Path Concept", JSAC, Vol 12. No.1, pp.120-127, 1994
- [11] D. E. McDysan and D. L. Spohn , ATM Theory and Application, McGraw-Hill, New York, p.524, 1994
- [12] 지윤규, 장유선, 김홍주 "동기식 전송망의 service survivability", 전자공학회지, 20권 4호, pp.85-94, 1994.4



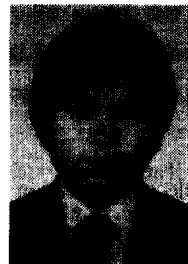
한경흠(韓慶欽)

1995 고려대학교 전산학과 학사  
 1995~현재 고려대학교 대학원 전산학과 석사  
 관심 분야 : ATM 망 복구, 지능형 에이전트



조충호(趙忠鎭)

1981 고려대학교 공과대학 산업공학과(학사)  
 1983 고려대학교 대학원 산업공학과(석사)  
 1986 프랑스 INSA de Lyon 전산학과(석사)  
 1989 프랑스 INSA de Lyon 전산학과(박사)  
 1990.3-1994.2 순천향대학교 전산통계과 조교수  
 1994.3-현재 고려대학교 전산학과 부교수  
 관심 분야 : ATM 망 복구, 지능망, 망성능분석, 지능형 에이전트, 퍼지 전문가 시스템



고재상(高在相)

1980 고려대학교 산업공학과 졸업  
 1985 전남대학교 경영학 석사  
 1995 전남대학교 산업공학과 박사과정 수료  
 1982-현재 한국전자통신 연구소 신뢰성 공학 연구실장  
 관심 분야 : ATM 교환기, 시스템 신뢰성 기술, 통신망 신뢰성 기술