

## 동기식 광전송망 복구를 위한 자체 치유 알고리즘

正會員 張 允 瑄\* 正會員 池 尹 圭\* 正會員 金 洪 珠\*

### Self-Healing Algorithm for The Restoration of Synchronous Optical Networks

Youn Seon Jang\*, Youn Kyu Ji\*, Hong Ju Kim\* *Regular Members*

#### 요 약

본 논문은 선로 장애에 대한 분산 제어 자체 치유 알고리즘을 제안한 것으로 복구시간을 향상시키는데 중점을 두었다. 복구를 위해 망에서 교환되는 제어 메시지의 갯수를 작게 하고, 이웃 노드들에게 더이상 복구에 사용될 예비채널이 없음을 미리 알리는 ASC\_NULL(Available Spare Channel\_NULL) 신호를 만들어 복구 경로 설정 과정에서 낭비되는 시간 지연을 줄임으로써 빠른 복구 시간을 얻을 수 있었다. 다른 여러 자체 치유 알고리즘들과의 객관적인 비교는 여러 조건들의 차이로 어려우므로 NETRATS(Network Restoration Algorithm for Telecommunication Systems)와의 비교만을 시뮬레이션 하였다. 시뮬레이션의 결과에서 제안된 알고리즘이 NETRATS보다 향상된 복구 시간을 가진다는 것을 알 수 있었다.

#### ABSTRACT

This paper proposes a distributed control self-healing algorithm for a line failure to improve the restoration time. This proposed self-healing algorithm acquired the fast restoration time by reducing the number of control messages and the queueing delay. And the reduction of queueing delay was attained by the ASC\_NULL(Available Spare Channel\_NULL) message which informs neighbor nodes that a source node has no available spare channel to reduce the waste of time during the rerouting. It is difficult for this proposed self-healing algorithm to be compared with other self-healing algorithms, because each self-healing algorithm has different simulation conditions. So, the performance of this proposed self-healing algorithm was compared with only NETRATS(Network Restoration Algorithm for Telecommunication Systems). The results of the simulation showed that the proposed self-healing algorithm was better than NETRATS in the restoration time.

---

\*韓國科學技術院 電氣·電子工學科  
Dept. of Electrical and Electronics Eng., KAIST  
論文番號 : 9480  
接受日字 : 1994年 3月 14日

## I. 서 론

통신 서비스 분야의 지속적인 발전으로 인한 광대역 서비스에 대한 요구는 전송망의 고속화, 대용량화를 필요로 하게되었다. 그 결과 단일 광섬유에 의한 전송망 장애가 대량의 정보 손실을 유발시켜 심각한 경제적 손실을 초래하게 됨에 따라 이러한 손실을 방지하기 위해 전송망의 설계시부터 서비스 생존성(service survivability)을 가지는 자체적인 치유망(self-healing network)을 고려하는 것이 중요한 과제가 되었다.

최근 동기식 전송망이 대두되면서 회선 분배 장치(DCS : Digital Cross-connect System)를 이용한 메쉬(mesh)형태의 망구성이 용이하게 되었고 각 신호 프레임내의 풍부한 오버헤드 채널들을 이용한 소프트웨어 구동망(software-driven network)구성이 가능하게 됨에 따라, DCS를 이용한 다이나믹 망복구(dynamic network restoration) 방식들이 많이 연구되었다. DCS를 이용한 메쉬망에서의 복구 방식은 운용 시스템내의 예비 채널 용량을 사용하여 자체 치유 알고리즘(self-healing algorithm)에 따라 손실 채널들을 복구하므로 이를 위해 운용 시스템 외의 별도의 전용 보호 장치(dedicated protection facility)를 필요로 하지 않는다는 장점이 있다. 반면, 복구 시간이 자동 보호 절체 방식(APS : Automatic Protection Switching)[1]과 자체 치유 링(SHR : Self-Healing Ring)[2][3]같은 전용 보호 장치를 이용한 복구 방식에 비해 길며, 각 DCS 사이에서 알고리즘에 따라 정보 교환을 제어하고 데이터 베이스를 관리해야 하므로 알고리즘의 구현이 복잡하다는 단점이 있다.

DCS를 이용한 망에서의 자체 고유 알고리즘을 평가하는 중요한 요소로는 복구 시간(restoration time), 복구율(survivability) 그리고 비용(cost)등이 있다. 본 논문은 선로 장애에 대한 분산 제어 자체 치유 알고리즘을 제안한 것으로, 복구 시간을 향상시키는데 중점을 두었다. 복구를 위해 망에섬 교환되는 제어 메시지의 수를 작게 하고, 이웃 노드들에게 더이상 복구에 사용될 예비채널이 없음을 미리 알리는 ASC\_NULL(Available Spare Channel\_NULL) 신호를 만들어 복구 경로 설정 과정에서 낭비되는 시간 지연을 줄임으로써 빠른 복구 시간을 얻을 수 있었다.

본 논문의 내용은, 서론에 이어 2장에서는 DCS를 이용한 자체 치유 복구방식에 대해 전반적으로 간략

하게 살펴보고, 3장에서는 제안한 자체 치유 알고리즘에 대해 설명하고, 4장에서 시뮬레이션 결과를 기존의 NEC에서 제안된 NETRATS(NETwork Restoration Algorithm Telecommunication Systems) 알고리즘[14]의 결과와 비교 분석 하였다. 그리고 5장에서 결론을 맺는것으로 마친다.

## II. DCS를 이용한 자체 치유 복구 방식

DCS는 기존의 수작업에 의한 회선 연결 방법을 소프트웨어에 의한 타임 슬롯교환(TSI) 방법으로 전자식 스위칭으로 대처하고, 전송 회선들을 보다 경제적이고 효율적으로 관리할 수 있도록 기존의 다중화 장치 기능, 광 전송 장치 기능 등 여러 디지털 전송 장치들의 기능을 하나로 통합하여 전 기능이 대규모의 소프트웨어에 의해 제어되도록 구성된 장치이다[4]. 고도화된 통신망의 구조에 적합한 전송망 지능화의 핵심이라고 할 수 있는 DCS는 여러 광전송 선로를 연결할 수 있으므로 메쉬망에서의 적용이 유용하고, 풍부한 데이터 프로세싱 기능과 어떤 정해진 시나리오 없이 다이나믹한 분배기능을 수행할 수 있다.

DCS를 이용한 자체 치유 복구 방식은 각 DCS 사이에 데이터 통신 채널(DCC : Data Communication Channel)을 통하여 복구 알고리즘에 따라 제어 메시지들을 서로 교환하여 복구 경로를 완성한다. 일반적인 장애 복구 순서는 다음과 같다[5].

1. 장애 검출-적합한 OS(Operating System), 제어기, 또는 망노드등에 의해 장애를 검출한다.
2. 제어 메시지 전파-다른 망노드로 제어 메시지를 전파하고 복구 과정을 시작한다.
3. 루팅(routing selection)-알고리즘에 의해 적당한 새로운 루트를 찾아서 선택한다.
4. 재루트(rerouting)-선택된 새로운 루트로 채널들을 분배한다.
5. 정상 복구-장애가 복구되었음을 확인하고 정상 동작으로 돌아간다.

DCS를 이용한 망복구 기법은 크게 3가지 관점에서 분류할 수 있다. 첫째, 제어 측면에서 중앙 제어(centralized control) 방식과 분산 제어(distributed control) 방식으로 나눌 수 있고 둘째, 재루팅 경로계획 측면에서 선계획(preplanned) 방식과 다이나믹(dynamic) 방식으로 셋째, 신호 복구 계층 측면에서 선로 복구(line restoration) 방식과 경로 복구(path restoration) 방식으로 나눌 수 있다.

### 2.1 제어 방식(중앙/분산)

중앙 제어 방식은 중앙 제어기로부터 모두 제어되는 것을 뜻한다. 그러므로 중앙 제어장치는 각 DCS 노드와 링크에 대한 유용한 운용 정보와 여분의 채널 용량 정보 등에 대한 망 정보를 가지고 있어야 한다. 노드사이에 장애가 발생했을 경우 제어 링크를 통해 장애 메시지를 DCS로부터 중앙 제어 장치까지 수신하게 된다. 이때 중앙 제어 장치는 해당 노드의 장애 메시지에 따라 망루팅 정보를 변경시키고 최적의 경로를 찾아서 최적 경로정보를 각 DCS로 제어 링크를 통해 전달하게 된다. 이때 각 해당 DCS는 스위칭 매트릭스를 제어 정보에 따라 변경하고 장애 신호를 분배 기능을 통해 다른 링크의 여분 채널로 우회시켜 서비스 복구를 완료한다.

이에 반해, 분산 제어 방식은 각 노드내에 자체 제어 장치를 가지고 있다. 분산된 각 DCS 장치는 자체에서 중단되는 링크의 운용/예비 채널에 대한 정보만을 가지며 망 전반의 DCS에 관한 루팅 정보는 가지고 있지 않다. 그러므로 각 노드의 제어기는 링크상에서 장애가 발생하면 광신호 오비헤드내 데이터통신 채널을 통해서 각 노드간 정보를 교환함으로써 서비스 복구를 하게 된다. 루팅 선택을 위해서는 패킷 교환 망에서의 flooding 기법과 유사한 다수의 메시지 송출 기법을 이용한다. 장애 발생의 인접한 노드에서 한쪽 노드는 sender가 되고 다른 노드는 chooser가 되어 sender는 집중된 모든 노드쪽으로 복구 메시지를 송출한다. 일반적인 메시지의 정보 내용은 다음과 같다.

- Sender 주소, sender-chooser 짝번호(pair ID)
- 메시지의 발생지 노드
- 메시지의 목적지 노드
- ASC(available spare ch.)
- HC(hop count)

중간 노드는 수신된 메시지에 따라 해당 경로 루팅 정보를 변경하고, HC를 증가시키는데 등 알고리즘에 따라 그 기능을 수행하고, 새로운 복구 메시지를 만들어 모든 이웃 노드들에 복구 메시지를 진파한다. 이때, HC LIMIT를 넘는 메시지는 제기되며 한번 통과된 노드로는 메시지를 재전송하지 않는다. 복구 메시지가 chooser에 도착하면 chooser에서는 복구 알고리즘에 따라 복구를 위한 경로를 찾고 확인 메시지를 sender를 향해 반대 방향으로 다시 전달한다. 이때 중간 노드들은 확인 메시지에 따라 스위칭 매트릭스를 변경한다. 복구 메시지를 처음 송출한 sender에서 확인 메시지를 수신하면 해당된 장애 채널들을 새로

구성된 경로로 스위칭함으로써 복구 과정이 완료된다[6][7].

중앙제어 방식은 시스템의 구현이 간단하고 서비스 복구율이 높으나 복구 시간이 느리며, 분산제어 방식은 중앙제어 방식에 비해 시스템 구현이 복잡하고 복구율이 낮으나 복구 시간이 빠르다는 장점이 있다. 표 2-1에 이 두 방식을 비교 정리하였다.

표 2-1 중앙 제어와 분산 제어의 비교  
Table 2-1 Comparison between centralized control and distributed control

구분	중앙제어	분산제어
시스템 복잡도	간단	복잡
서비스 복구 확률	높음	낮음
상상 상대 회복	쉬움	어려움
표준화 필요	메시지	알고리즘, 메시지
복구시간	느림	빠름
관리 오비헤드	많음	작음
중앙 OS 의존성	yes	no/yes
여분 용량과 diversity route planning	필요	필요

### 2.2 신호 복구 계층(경로/선로)

경로 복구는 end to end 분리 채널을 복구하는 것으로서 장애가 발생했을 때 영향을 받은 채널에 대해서 각각 새로운 루트를 찾아서 복구한다. 이것은 각 채널의 end point 사이에서 루트가 형성됨을 의미한다[5][8][9]. 선로 복구는 장애가 발생한 장치를 지나가는 모든 채널들을 새로운 루트로 복구시킨다. 표 2-2에 간단히 정리하였다.

표 2-2 선로 복구와 경로 복구의 비교  
Table 2-2 Comparison between line restoration and path restoration

구분	선로복구	경로복구
복구시간	빠름(수십ms)	복잡
여분용량 사용효율	나쁨	좋음

### 2.3 재루팅 방법(선계획/다이나믹)

선계획 방식에서 각 DCS 제어기들은 각종 장애 시나리오에 대해 망 재구성에 관한 정보들을 모두 가지고 있어야 한다. 이에 비해 다이나믹 방식의 각 DCS

제어기들은 단지 필요한 부분 정보를 가지고 장애가 발생했을때 망의 상태에 따라서 재루팅을 결정한다. 그 특징들은 표 2-3에 잘 비교되어 있다[5].

표 2-3 선계획/다이나믹 재루팅 방법 비교  
Table 2-3 Comparison between preplanned rerouting and dynamic rerouting

구분	선계획	다이나믹
시스템 복잡도	낮음	높음
망 적용	어려움	쉬움
복구 속도	빠름	느림
시스템 신뢰도	낮음	높음
메모리 필요성	높음	낮음

2.4 자체 치유 알고리즘

DCS를 이용하여 제안된 여러 자체 치유 알고리즘으로는 ATRC(Alberta Telecommunications Research Center)에서 제안한 것[10]과, FITNESS (Failure Immunization Technology for Network Service Survivability)[11], FUJITSU에서 제안한 방식[12][13], NETRATS(NETwork Restoration Algorithm for Telecommunication Systems)[14] 등을 들 수 있다. 이들 각 알고리즘의 시뮬레이션에 주어진 조건들의 차이때문에 이 알고리즘에 대해 복

표 2-4 제안된 방식 비교  
Table 2-4 Comparing some self-healing algorithms

제안된 방식	복구방식	선택루트 /주기	메세지 채널	루트선택 기준	메시지 분배
NETRATS [14]	선로/경로 복구	multi-routes	DCC	FIFO	1 메시지 /링크 또는 장애경로**
FITNESS [11]	선로복구	1 route (반복)	DCC	최대 유효 채널*	1 메시지 /링크
DOUBLE SEARCH [15]	선로복구	multi-routes	DCC	FIFO	1 메시지 /1 링크
FUJITSU [12][13]	선로복구	1 route (반복)	DCC	FIFO	1 메시지 /링크

FIFO: First In First Out  
\*: 타임 아웃(time-out) 필요  
\*\*: 선로 복구에는 링크당 1메세지, 경로 복구에는 장애가 일어난 STS-1, STS-3c당 1메세지

구 시간과 복구율 등에 대한 객관적인 평가를 한다는 것은 어렵다. 각 알고리즘들은 표 2-4에 요약되어 있다[5].

III. 제안한 자체 치유 알고리즘

이 자체 치유 알고리즘은 분산 제어 방식과 선로 복구 방식을 채택한다. 본 논문은 여러 알고리즘들 중에서 NETRATS[14] 알고리즘을 기본으로 하여, 향상된 복구 시간을 갖는 알고리즘을 제안하였다.

먼저 특징을 설명하면, 첫째로 ASC NULL 메세지를 들 수 있다. 이 신호는 복구 능력이 없는 노드가 그 이웃 노드들에게 자신은 유효 채널이 없음을 미리 알림으로서, 기존 알고리즘에서 제어 메세지가 복구 능력이 없는 노드로 전송되었다가 다시 NEGATIVE msg.에 의해 이전 노드로 되돌아 오는 것을 방지한다. 그러므로 복구 시간이 단축되는 효과가 있다. 둘째로는 제어 메세지의 감소이다. Selection phase에서 selection 메세지가 search 메세지에 대해서 일대일로 만들어져 전송되므로 이것은 각 복구 채널에 대해 selection 메세지를 만들어 전송하는 것보다 메세지 수가 감소되는 결과를 가져온다. 따라서 각 노드에서 걸리는 대기 지연 시간(queueing delay)을 감소시킨다. 마지막으로는 1개 이상의 선로 장애에 대해 복구를 수행할 수 있다는 것이다. 각 제어 메세지는 fil(failed line identification)라는 장애 선로 번호를 가지고 있으므로 어느 선로에 대한 복구 메세지인지 구분된다.

각 phase를 설명하기전에 이 알고리즘에 사용되는 제어 메세지의 종류와 그 내용을 정리한다. 제어 메세지는 SEA(search) msg., SEL(selection) msg., LINK msg.로 나누어지며, 이들 각각의 메세지들은 그 메세지가 사용되는 phase내에서 정보를 교환하는데 사용된다. 각 메세지의 내용에 대해 살펴보면,

SEA(search) msg. [그림 1(a)]

- ① mtype(message type) : search 메세지임을 표시.
- ② hopc(hop count) : sender로부터 이 메세지가 거쳐온 hop수를 표시.
- ③ asc(available spare channel) : 메세지가 지나온 경로의 유효 예비 채널 수를 표시.
- ④ fil(failed line id.) : 장애가 발생한 선로의 번호를 표시.
- ⑤ tli(transmission line id.) : 메세지가 전송될 선로의 번호를 표시.

⑥ passnd(passed node id.): 메세지가 거쳐온 노드의 번호를 표시.

SEL(selection) msg.[그림 1 (b)]

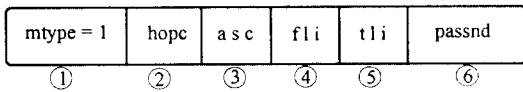
①~⑥: 앞의 search 메세지와 동일.

⑦ neg/pos(negative/positive): selection 메세지에서 제대로 다음 노드로 전송이 되었는지(positive), 안되었는지(negative)를 표시.

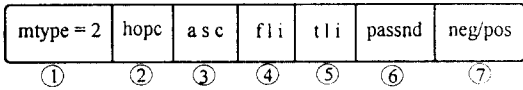
LINK sg.[그림 1(c)]

①~④: 앞의 search 메세지와 동일.

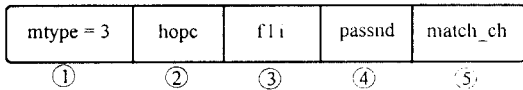
⑤ match\_ch: completing phase에서 복구 채널을 맞추기 위해 sender에서 복구된 손실채널의 번호를 표시.



(a) Search Message



(b) Selection Message



(c) Link Message

그림 1. 각 제어 메세지의 구성

Fig 1. Construction of each control message

이것으로 제어 메세지에 대한 정리를 마치고 다음은 제안한 자체 치유 알고리즘에 대해 자세히 설명한다. 그림 2와 같이 제안된 알고리즘은 7개의 phase로 나누어지며, 각 노드에서 실행되는 phase는 다음과 같다.

정상 상태로 동작하던 망에서 어떤 선로에 장애가 발생하면 그 선로에 연결된 양쪽은 두 노드는 선로 장애 신호에 의해 장애가 발생했음을 알게 된다. 장애 신호를 받은 두 노드중에서 노드 번호가 작은 노드는 sender가 되고, 다른 노드는 chooser가 된다.

**Start phase(sender)**

장애가 감출되면 sender는 먼저 손실된 채널의 수를 파악한 후, SEA msg.를 만든다. 만들어진 메세지는 모든 이웃 노드로 보내진다.

메세지의 내용을 보면, hopc는 1이 되고, asc는 손실된 메세지의 수와 이 메세지가 전송될 선로의 예비 채널(spare channel)의 갯수를 비교하여 이들 중에서 작은 값을 취한다. 만일 asc의 값이 0이라면 이 메세지는 제기되어 전송되지 않는다. 즉 전송 선로에 예비 채널이 없다면 그 선로에는 메세지를 전송하지 않는다. passnd에는 sender의 노드 번호를 입력한다.

그림 3에 장애 선로가 5번일 때의 예를 보였다. 노드 1은 sender가 되고, 노드 5는 chooser가 된다. 손실된 채널 갯수는 3이고, 선로 1의 예비 채널은 1이므로 선로 1로 보낼 메세지의 asc값은 1이된다. 마찬가지로 선로 4로 보낼 메세지의 asc값은 2가 된다.

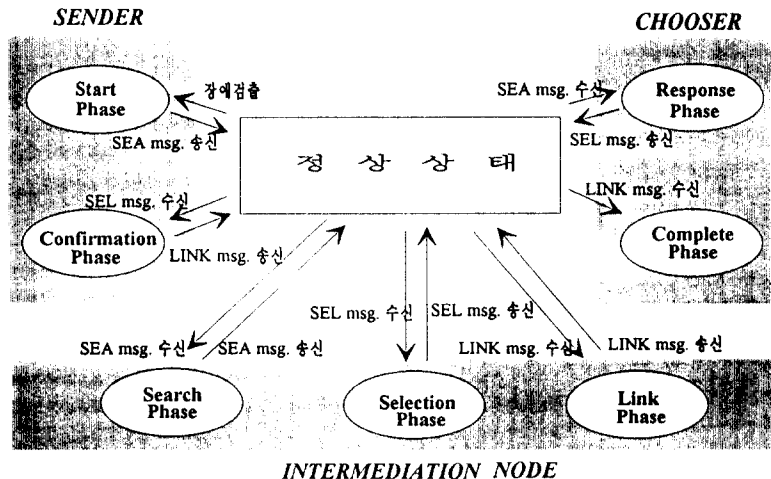


그림 2. 제안된 알고리즘의 구성

Fig 2. Construction of this proposed algorithm

**Search phase(intermediation node)**

제어 메시지를 송수신하는 sender와 chooser를 제외한 노드들은 중간 노드(intermediation node)로 정의된다. 이 노드들은 SEA msg.를 받으면 먼저 그 메시지를 메모리에 저장한다. 그리고 새로운 SEA msg.를 만들어 start phase에서와 같이 모든 이웃 노드로 이 메시지를 전송한다. 이때는 두가지 제약이 있다. 첫째는, 만일 이 새로운 메시지를 수신할 노드가 이미 이 메시지의 passnd에 있는 노드라면 그 노드로는 메시지를 보내지 않는다는 것이다. 둘째로는, 새로운 메시지의 hopc값이 hop count limit를 넘었거나 asc값이 0이면 그 메시지는 제거되어 전송되지 않는다는 것이다.

새로운 SEA msg.의 hopc는 수신된 메시지의 hopc를 1 증가 시킨 값을 가지고, asc는 수신된 메시지의 asc와 전송할 선로의 예비 채널 갯수를 비교하여 작은 값을 취한다.

그림 3의 예에서, 중간 노드 3은 수신된 search 메시지를 메모리에 저장한후 그림 1(a)에 나타난 내용

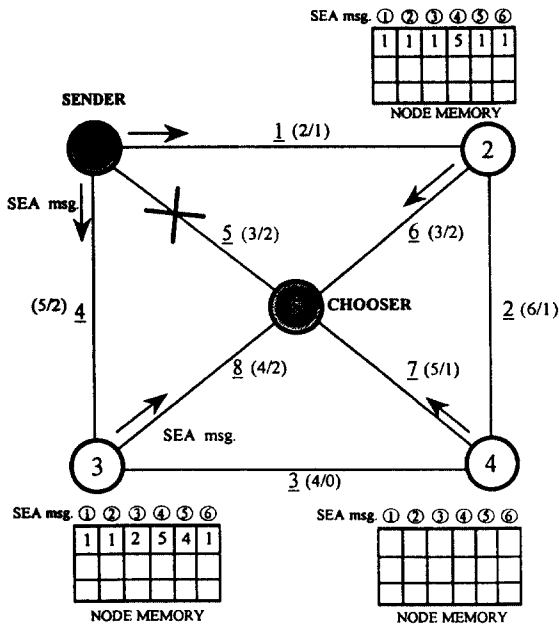


그림 3. Search Phase의 예  
Fig 3. An example of search Phase

순으로 (1, 2, 2, 5, 8, 13)의 내용을 담은 새로운 메시지를 선로 8로 전송한다. 선로 3에 대해서는 (1, 2, 0, 5, 3, 13)의 메시지가 만들어지나 asc값이 0이므로 제거되어 전송되지 않는다. 그림3의 각 노드 메모리에 저장된 숫자는 SEA msg.의 내용을 나타낸다.

**Response phase(chooser)**

SEA msg.가 chooser에 도착하면, chooser에서는 먼저 도착한 SEA msg.에 대한 SEL msg.를 만든다. 이 SEL msg.의 mtype, hopc, passnd는 새롭게 기록되고 나머지 내용은 수신된 SEA msg.의 값을 그대로 갖는다. neg/pos의 값은 0으로 넣는다.

이렇게 만들어진 메시지는 수신된 SEA msg.를 보낸 노드로 다시 보내진다. 이때 손실된 채널만큼의 채널을 복구하지 못했으면, 그 다음에 도착하는 SEA msg.에 대해서도 앞의 과정을 반복한다. 이것은 손실된 채널을 충분히 복구할 만큼의 충분한 asc가 보내질 때까지 계속 반복된다.

그림 4의 예에서 선로 8에서 SEA msg.가 먼저 도착했다면 chooser에서는 그림 1(b)에 나타난 내용순으로 (2, 1, 2, 5, 8, 0)의 SEL msg.를 만들어 선로 8을 통해 되돌려 보낸다. 이때 asc가 2이므로 손실된 채널을 복구하기엔 불충한 값이다. 그러므로 선로 6에서 온 다음번 SEA msg.에 대해서도 (2, 1, 1, 5, 6, 5, 0)의 메시지를 만들어 보내야한다. 그림4의 각 노드 메모리에 저장된 숫자는 SEA msg.의 내용을 나타낸다.

**Selection phase(intermediation node)**

SEL msg.를 받으면 먼저 메모리에 저장된 SEA msg.를 조사해서 가장 작은 값의 hopc를 가진 메시지를 선택한다. 이 선택된 메시지의 tli와 일치하는 선로로 새로운 SEL msg.를 보낸다.

새로운 메시지는 hopc를 1 증가시키고, asc는 수신된 SEL msg.와 선택된 SEA msg.의 asc값중에서 작은 값을 갖는다. 선택되었던 SEA msg.는 SEL msg.에 사용된 만큼의 채널을 뺀 asc값을 기록하여 다시 메모리에 저장된다. 만일 새로운 asc값이 수신된 asc만큼 복구되지 않으면, 만족될때까지 앞의 과정을 반복한다.

수신된 asc 만큼의 채널을 찾지 못하면 이전 노드로 neg/pos를 set 시킨 NEGATIVE msg.를 보낸다. 이것을 받은 이전 노드는 다른 경로를 찾아서 재전송한다. NEGATIVE msg.의 asc는 전송 경로를 찾지 못한 나머지 채널 갯수이다.

SEL msg.의 전송이 끝나면, 중간 노드는 저장하고 있는 SEA msg.들의 asc를 조사하여 모든 asc값이 0이면 이웃 노드들에게 ASC NULL msg.를 보낸다. 이것은 이웃 노드들에게 이 노드는 더 이상의 복구 경로가 없다는 것을 알림으로써 SEL msg.가 이 노드로 왔다가 다시 이전 노드로 되돌아가는 것을 방지하므로써 복구 시간이 단축된다. ASC NULL msg.는 이 메시지의 til(transmission line identification) 정보를 가지고 있어야하며, 이 메시지를 수신한 이웃 노드들은 til을 참고로 이 번호의 선로로는 메시지를 보내지 않는다.

그림 4의 예에서, 노드 3에 SEL msg.가 도착하면 그림 1(b)에 나타난 내용순으로(2, 2, 5, 4, 53, 0)의 SEL msg.를 선로 4를 통해 전송한다. 그리고 선택되었던 메시지의 asc는 0이 기록되어 다시 저장된다. 그리고 이웃 노드들로 ASC NULL msg.가 전송된다.

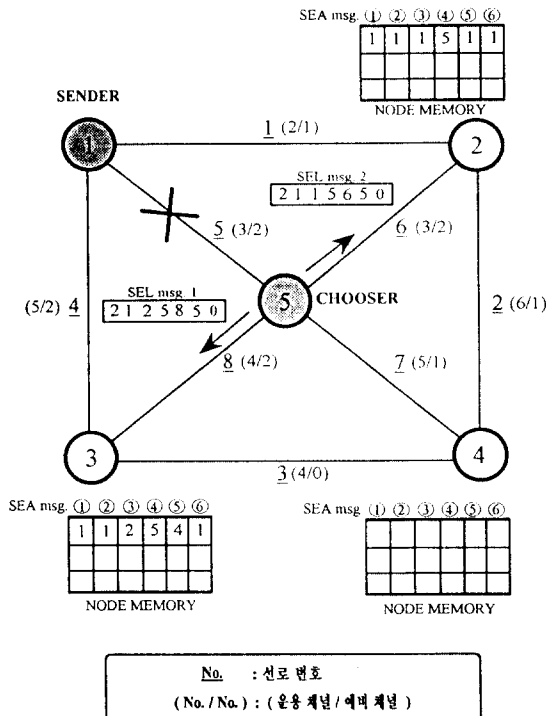


그림 4. Response Phase와 Selection Phase의 예  
Fig 4. An example of response Phase and selection Phase

**Confirmation phase(sender)**

Sender는 SEL msg.를 받으면 그 메시지의 asc값만큼의 손실 채널을 복구 경로로 스위칭 한다. 그런후에 각각의 복구된 채널번호(match ch)를 가진 LINK msg.를 받음이 복구를 위해 선택된 선로를 통해 chooser로 보낸다.

**Link phase(intermediation node)**

LINK msg.를 수신한 중간 노드는 이 메시지의 복구 경로에 따라 예비 채널로 스위칭을 완료한다.

**Completion phase(chooser)**

최종적으로 LINK msg.가 chooser에 도착하면 chooser가 수신된 채널을 복구채널로 스위칭하므로써 복구 과정을 마친다.

**IV. 시뮬레이션 결과 비교 및 고찰**

시뮬레이션 프로그램은 event driven[11] 개념을 기본으로 한다. 제한된 알고리즘의 성능을 비교하기 위해서 NETRATS에 사용된 것과 동일한 조건을 이 논문의 시뮬레이션에 적용한다.

망의 각 노드는 연결된 모든 선로들에 대해 수신 메시지를 위한 queue와 송신 메시지를 위한 queue를 가지고 있다. 그리고 각 메시지는 time\_stamp를 가지고 있다. 이것은 알고리즘 수행, queueing delay, 메시지 전송등의 프로세싱 타임을 고려해서 총 실행 시간을 기록하는 것이다. Time stamp의 값이 작은 것부터 먼저 수행된다.

**시뮬레이션의 조건**

- 1) 메시지의 프로세싱 타임은 1-9 msec의 랜덤한 값으로 uniform distribution을 갖는다. 이것은 OS나 CPU에 의존한다.
- 2) 메시지의 전송율은 64kbps로 한다.
- 3) Search, selection, link의 각 제어 메시지는 20 bytes로 구성되었다고 가정한다. 그러므로 시뮬레이션에서 각 메시지 전송 시간은 2.5msec이다.
- 4) Hop count limit는 6으로 한다.
- 5) 시뮬레이션에 사용된 망 모델은 그림 5와 같다. 이 망모델은 미국의 실제 LATA(Local Access Transport Area)중 한곳을 근거로 만들어졌으며, [11], [14]에서 사용된 것이다.

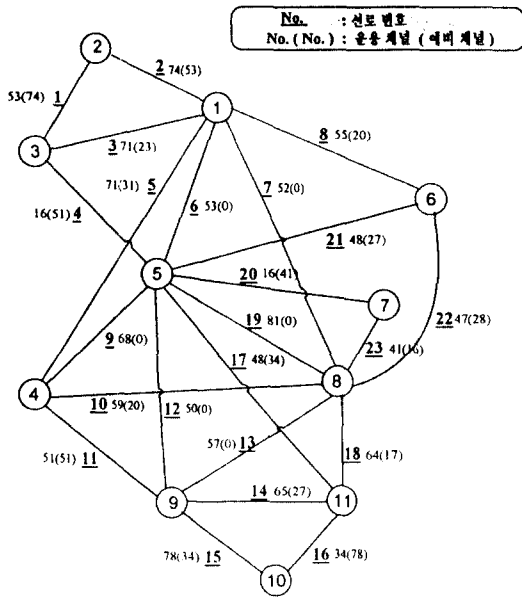


그림 5. 시뮬레이션 망 모델  
Fig 5. Network model for simulation

먼저 각 선로 장애에 대해서, 제안된 알고리즘에서의 제어 메시지 수가 NETRATS의 제어 메시지 수보다 얼마나 감소되었는지에 대해서 그림 6에 보였다. 평균적으로 37%의 감소를 보였다.

이러한 제어 메시지수의 감소와 ASC\_NULL 메시지에 의해 향상된 복구 시간을 살펴보면, single-failure 경우에 대해서는, 그림 7은 선로 7에서 장애가 발생했을 경우의 복구 시간과 복구율을 보이고 있다. 두 알고리즘 모두 100%의 복구율을 가지며 복구 시간에 있어서는 제안된 알고리즘이 150msec 정도 빠름을 볼 수 있다. 그림 8은 선로 12에 대한 복구 시간과 복구율을 보였으며 복구 시간이 크게 단축되었음을 알 수 있다. 그림 9는 single-failure에 대해서 각각의 복구 시간을 비교 하였다. 평균적으로 제안된 알고리즘이 NETRATS 보다 280msec 정도 빠름을 알 수 있다.

Double-Failure 경우에 대해서, 그림 10은 선로 7과 12에 장애가 발생했을 경우이다. 선로 7과 12가 각각 single-failure에 대해서는 100% 복구가 되었으나, double-failure에서는 선로 12는 대략 90%의 복구율만을 보였다. Double-failure의 경우에서도 150msec 정도 제안된 알고리즘의 복구 시간이 빠르다. 이 결

과들은 30개의 결과치를 평균한 값이다.

복구 시간이 개선된 이유를 분석하면, 첫째 ASC\_NULL메세지가 복구 시간의 지연을 방지하기 때문이다. 기존의 NETRATS 알고리즘[14]은 복구에 사용된 예비 채널을 보유하고 있는 노드와 보유하고 있지 않은 노드의 구별없이 selection 메시지를 전송하므로, 예비 채널을 보유하고 있지 않은 노드로 전송된 메시지는 결국 다시 그 이전 노드로 되돌아 온다. 이렇게 낭비되는 시간을 줄이기 위해서 본 논문에서 제안한 알고리즘은 ASC\_NULL 메시지로써 미리 예비 채널을 보유하고 있지 않은 노드를 인접 노드들에게 알림으로서 예비 채널을 보유한 노드로만 selection 메시지를 전송하여 복구 시간의 지연을 방지한다. 둘째는 기존의 제안된 알고리즘은 selection phase에서 각 복구된 채널에 대해 selection 메시지를 전송하므로 망 내에 전송되는 메시지의 수가 많다. 그러나, 본 논문의 알고리즘은 여러 채널 정보(asc)를 가진 search 메시지에 대해 selection 메시지를 만들어 전송하므로 망 내의 메시지수를 줄일 수 있다. 이것은 각 노드에서의 queueing delay를 감소시키는 결과를 가져온다. 그러므로 좀 더 향상된 복구 시간을 얻을 수 있다.

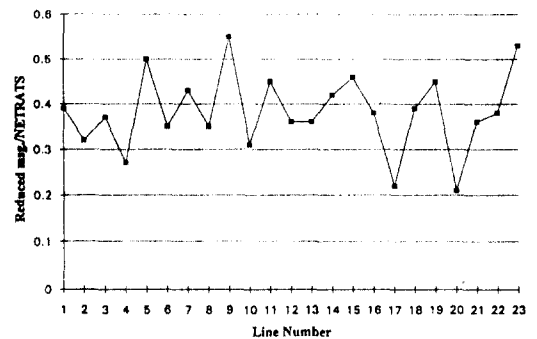


그림 6. 각 sing-failure에 대한(감소된 제어 메시지 수/NETRATS 제어 메시지 수)  
Fig 6. (Reduced control messages./NETRATS control messages) vs. each single-failure



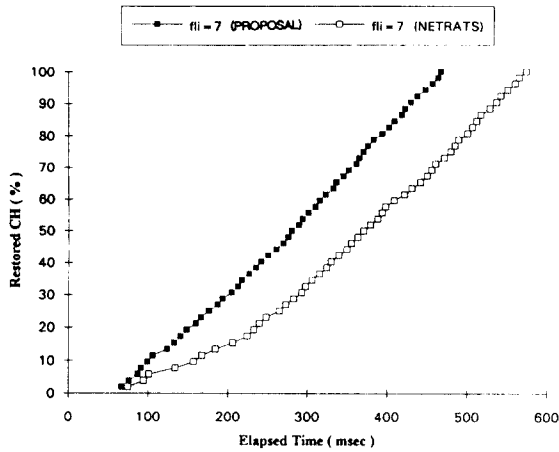


그림 7. FI = 7인 경우의 복구율과 복구 시간  
 Fig 7. Restoration rate vs. Elapsed time in the of FI = 7

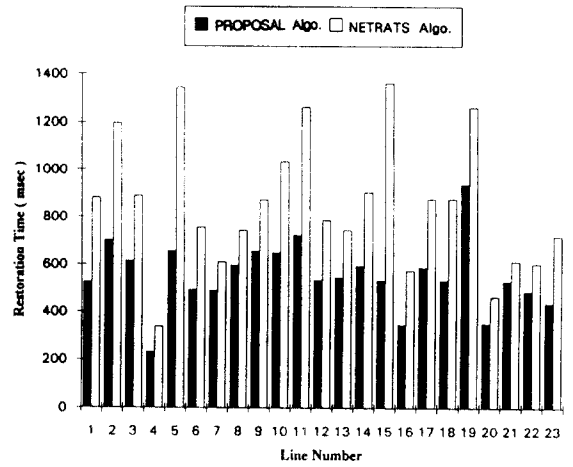


그림 9. 각 Single failure에 대한 복구 시간  
 Fig 9. Elapsed time on each single failure

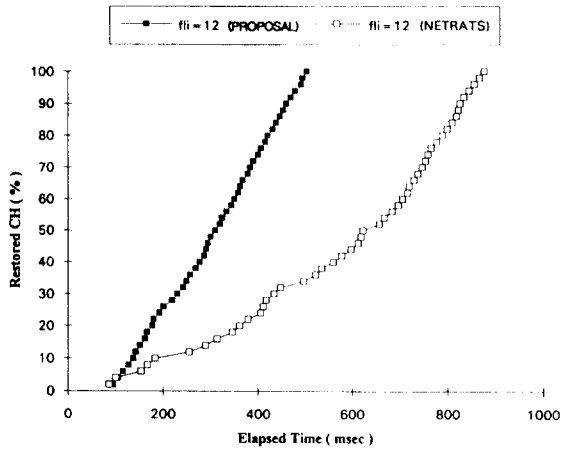


그림 8. FI = 12인 경우의 복구율과 복구 시간  
 Fig 8. Restoration rate vs. Elapsed time in the case of FI = 12

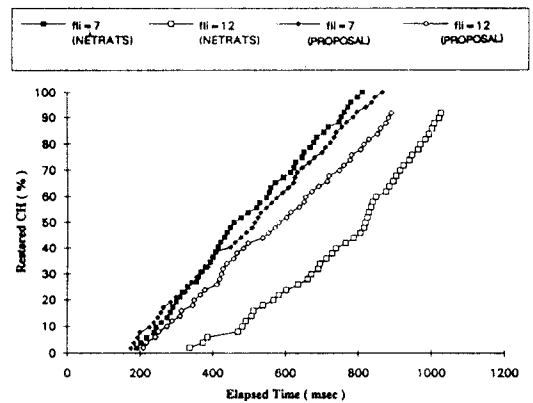


그림 10. Double failure(FI = 7, 12)인 경우의 복구율과 복구 시간  
 Fig 10. Restoration rate vs. Elapsed time in the case of double failure(FI = 7, 12)

## V. 결 론

본 논문은 동기식 전송망에서, 선로의 절단등과 같은 선로 장애로 인하여 손실된 서비스를 복구하는 서비스 보호망에 대한 연구로서 서비스 복구를 위해 DCS 복구 방식의 자체 치유 알고리즘을 제안하였다.

제안한 자체 치유 알고리즘은 분산 제어 방식이며, 선로 절체 방식이다. Single-failure 뿐만 아니라 multi-failure에 대해서도 서비스 복구를 수행한다. 이 자체 치유 알고리즘은 좀 더 빠른 복구 시간을 얻는데 중점을 두었으며, ASC\_NULL 신호를 제안하였고 제어 메시지의 수를 감소시키므로써 빠른 복구 시간을 얻을 수 있었다. 다른 여러 자체 치유 알고리즘들과의 객관적인 비교는 시뮬레이션에서 주어진 여러 조건들의 차이로 어려우므로 NETRATS와의 비교만을 시뮬레이션 하였다. 시뮬레이션의 결과에서 제안된 알고리즘이 NETRATS보다 향상된 복구 시간(restoration time)을 가진다는 것을 알 수 있다.

본 논문은 DCS 복구 방식에 대한 기본적인 알고리즘 연구로서 앞으로 구축될 동기식 전송망의 경제성과 신뢰성의 향상을 위한 기반이 될 것이다.

## 감사의 글

본 연구는 한국전자통신연구소의 위탁 연구 과제로서 수행중이며, 본 연구에 많은 조언과 도움을 주신 지윤규 교수님과 김홍주 선배님께 고마움을 전합니다.

## 참 고 문 헌

1. Tsong-Ho Wu, "Survivable Network Architectures for Broadband fiber Optic Networks: Model and Performance Comparison," *J. of lightwave Tech.* Vol.6, NO.11, pp. 1698-1709, Nov. 1988.
2. Yasuyo OKANOUE, Hideki SAKAUCHI, and Satoshi HASEGAWA, "DESIGN AND CONTROL INTEGRATED SELF-HEALING NETWORKS," in *Proc. IEEE GLOBECOM '91*, pp. 221.1-6.
3. Steven H. Hersey, Mark J. Soulliere, "ARCHI-

- TECTURES AND APPLICATIONS OF SONET IN A SELF-HEALING NETWORK," in *Proc. IEEE ICC '91*, pp. 443.1-7.
4. 정옥희, 윤원우, 이경국, "광대역 디지털회선분배장치(B-DCS)와 고속 digital 전용망 구성," 전자공학회지, pp. 68-75, 1993.4.
5. Tsong-Ho Wu, *Fiber Network Service Survivability*, Artech House, 1992.
6. Arun Nellary, Koji Mizushima, "INTELLIGENT TRANSPORT NETWORK SURVIVABILITY : STUDY OF DISTRIBUTED AND CENTRALIZED CONTROL TECHNIQUES USING DCS & ADMS," in *Proc. IEEE GLOBECOM '90* pp. 705B.3.1-5.
7. Tsong-Ho Wu, Haim Kobrinski Kipak Ghosal and T.V. Lakshman, "A Service Restoration Time Study for Distributed Control SONET Digital Cross-Connect System Self-Healing Networks," in *Proc. IEEE ICC '93*, pp. 893-899.
8. Jiro Yamada and Akiya Inoue, "INTELLIGENT PATH ASSIGNMENT CONTROL FOR NETWORK SURVIVABILITY AND FAILURESS," in *Proc. IEEE ICC '91*, pp. 22.3.1-5.
9. Satoshi Hasegawa, Osamu TABATA, Yasuyo OKANOUE and Hideki Sakauchi, "INTEGRATED SELF-HEALING NETWORK FOR STS-1/STS-3c PATH LEVEL RESTORATION," in *Proc. NOMS '92*, pp.9.1.1-12.
10. W. D. Grover, "THE SELF HEALING NETWORK: A FAST DISTRIBUTED RESTORATION TECHNIQUE FOR NETWORKS USING DIGITAL CROSSCONNECT MACHINES," in *Proc. IEEE GLOBECOM '87*, pp. 28.2.1-5.
11. C. Han Yang and Satoshi Hasegawa, "FITNESS : FAILURE IMMUNIZATION TECHNOLOGY FOR NETWORK SERVICE SURVIVABILITY," in *Prco. IEEE GLOBECOM '88*, pp. 47.3.1-6.
12. Hiroaki Komine, Takafumi Chujo, Takao Ogura, Keiji Miyazaki, and Tetsuo Soejima, "A DISTRIBUTED RESTORATION ALGORITHM FOR MULTIPLE-LINK NODE FAILURES OF TRANSPORT NETWORKS," in *Proc. IEEE GLOBECOM '90*, pp. 403.4.1-5.

13. Takafumi Chujo, Hiroaki Komine, Keiji Miyazaki, Takao Ogura, and Tetsuo Soejima, "THE DESIGN AND SIMULATION OF AN INTELLIGENT TRANSPORT NETWORK WITH DISTRIBUTED CONTROL," in *Proc. NOMS 90*, pp. 11.4.1-12.

14. Hideki Sakauchi, Yasuyo Nishimura and Satoshi Hasegawa, "A SELF-HEALING NE

TWORK WITH AN ECONOMICAL SPARE-CHANNEL ASSIGNMENT," in *Proc. IEEE GLOBECOM 90*, pp. 403.1.1-6.

15. Hiroyuki FUJII, Toshiaki HARA, Noriaki YOSHIKAI, "Characteristics of Double Search Self healing Algorithm for SDH Networks," in *IEICE CS91 48, 1991*, pp. 71-76.



張 允 璿(Jang Youn Seon) 정회원  
 1970년 3월 30일 생  
 1992년 2월 : 경북대학교 전자공학과(학사)  
 1994년 2월 : 한국과학기술원 전기 및 전자공학과(석사)  
 1994년 ~ 현재 : 한국과학기술원 박사과정 재학중

※주관심분야: Fiber network survivability, WDM Network work

池 尹 圭(Youn Kyu Ji) 정회원  
 1978년 : 서울대학교 공과대학 전자공학과(학사)  
 1980년 : 서울대학교 공과대학 전자공학과(석사)  
 1980년 ~ 1984년 : The Univ. of Texas at Austin(박사)  
 1984년 ~ 1989년 : (미국) AT & T Bell Labs.(MTS)

1989년 ~ 현재 : KAIST 전기 및 전자공학과, 현재 부교수  
 ※주관심분야: 광통신, 광대역통신, 광교환 등

金 洪 珠(Hong Ju Kim) 정회원  
 1960년 3월 27일 생  
 1983년 2월 : 경북대학교 전자공학과(학사)  
 1985년 2월 : 경북대학교 전자공학과(석사)  
 1994년 6월 ~ 현재 : 한국과학기술원 전기 및 전자공학과(박사과정)

1985년 1월 ~ 1994년 6월 : 한국전자통신연구소 전송방식 연구실, 선임연구원

※주관심분야: 광대역통신망, 디지털전송