다중 채널 ATM 스위치에서의 장애 관리

정회원 오 민 석*

Fault Management in Multichannel ATM Switches

Minseok Oh* ***Regular Member***

요 약

다중 채널 스위치 구조의 중요한 이점 중의 하나는 스위치 내부의 장애에 대한 내성 (tolerance)을 스위치 구조에 결합시킬 수 있다는 것이다. 예를 들어 하나의 다중 채널 그룹에 속하는 경로에 장애가 있을 경우, 장애 경로로 통과했어야 하는 트래픽을 나머지 경로가 책임 질 수 있다. 또한 스위치 소자에 발생하는 장애는 ATM 셀 (cell)의 잘못된 라우팅을 야기하거나 순서를 뒤바꾸게 할 수 있다. 본 논문에서는 다중 채널 크로스바 (crossbar) ATM 스위치에서의 장애 위치 알고리즘을 제안하였다. 최적의 알고리즘은 시간적으로 최상의 성능을 보여주지만, 계산상으로는 복잡하여 결과적으로 실제 구현을 어렵게 만든다. 이러한 단점을 극복하기 위해 최적의 알고리즘보다 계산상으로 효율적인 온라인 알고리즘을 제안하였다. 성능은 시뮬레이션을 통해 검증하였으며 그 결과로서 온라인 알고리즘의 성능은 랜덤 (random) 트래픽 및 버스트한 (bursty) 트래픽에 대해 거의 최적에 가까운 성능을 보여 준다. 끝으로 장애 위치 확인 알고리즘에 의해 제공되는 정보를 이용한 장애 복구 알고리즘을 제안하였다.

Key Words : ATM, multichannel; crossbar; fault localization; fault recovery

ABSTRACT

One of the important advantages of multichannel switches is the incorporation of inherent fault tolerance into the switching fabric. For example, if a link which belongs to the multichannel group fails, the remaining links can assume responsibility for some of the traffic on the failed link. On the other hand, if faults occur in the switching elements, it can lead to erroneous routing and sequencing in the multichannel switch. We investigate several fault localization algorithms in multichannel crossbar ATM switches with a view to early fault recovery. The optimal algorithm gives the best performance in terms of time to localization but is computationally complex which makes it difficult to implement. We develop an on-line algorithm which is computationally more efficient than the optimal algorithm. We evaluate its performance through simulation. The simulation results show that performance of the on line algorithm is only slightly sub-optimal for both random and bursty traffic. Finally a fault recovery algorithm is described which utilizes the information provided by the fault localization algorithm.

* LG TeleCom 기술연구소 데이터망개발팀 (msoh@lgtel.co.kr)
논문번호 : 030199-0513, 접수일자 : 2003년 5월 13일

Ⅰ. 서 론

ATM에서 다중 채널 스위치는 처리율, 셀 손실 확률, 지연 등에 있어 보다 우수한 성능을 제공하기 위해 채널 집단화 (grouping)의 개념을 이용한다[1][2][3][4][5][6][7][8]. 셀이 특정한 출력 채널로 고정되어 라우팅되는 것 대신에 그 출력 채널이 속한 채널 그룹의 아무 채널로나 라우팅될 수 있다. 새로운 어플리케이션에 대한 수요가 증가함에 따라 대역폭과 트래픽 특성에 있어 (예를 들어, 세션 길이, 트래픽의 버스트한 정도 등) 다양성이 더욱 요구되고 있다. 이러한 환경에서 고용량의 채널을 통계적으로 (statistically) 공유하는 것의 잇점은 이미 잘 알려져 있다[9].

다중 채널 스위치 구조의 중요한 이점 중의 하나는 스위치 내부의 장애에 대한 내성 (tolerance)을 스위치 구조에 결합시킬 수 있다는 것이다[10]. 예를 들어 하나의 다중 채널 그룹에 속하는 경로에 장애가 있을 경우 장애 경로로 흘렀어야 하는 트래픽을 나머지 경로가 책임을 질 수 있다. 또한 스위치 소자에 발생하는 장애는 ATM 셀 (cell)의 잘못된 라우팅을 야기하거나 순서를 뒤바꾸게 할 수 있다. 본 논문에서는 크로스바에 기반한 다중 채널 구조에서 이러한 문제점을 조사한다.

우리의 관심은 다중 채널 크로스바에 발생한 스위치 요소의 장애를 빨리 찾아내는 것이다. 그러한 장애를 신속히 찾아내는 능력은[11][12][13] 예비 (redundant) 스위치 요소를 이용한 온라인 장애 복구 알고리즘을 가능케 한다.

Ⅱ. MCDC와 MCOC의 개요

크로스바 기반의 스위치 모듈은 면적, 전력 및 클록 면에서 잇점을[7][8][14][15] 제공하며 다중 채널 구조와 잘 조화를 이룬다. 특히 본 논문에서는 다중 채널 디플렉션 크로스바 (multichannel deflection crossbar, MCDC)와 다중 채널 원턴 크로스바 (multichannel one-turn crossbar, MCOC)를 고려하기로 한다[7][14].

1. MCDC

그림 1은 8개의 입력 단자, 8개의 순환 입력 단자와 8개의 출력 채널을 가지고 있는 MCDC를 보여준다. 이는 라우팅 크로스바와 집속기 (concentrator)로 구성되어 있으며, 집속기의 출력 단자를 라우팅 크로스바의 반쪽 상단의 입력 단자로 연결해주는 순환 경로를 갖고 있다. 라우팅 크로스바의 반쪽 하단에 위치한 입력 채널로 셀들이 들

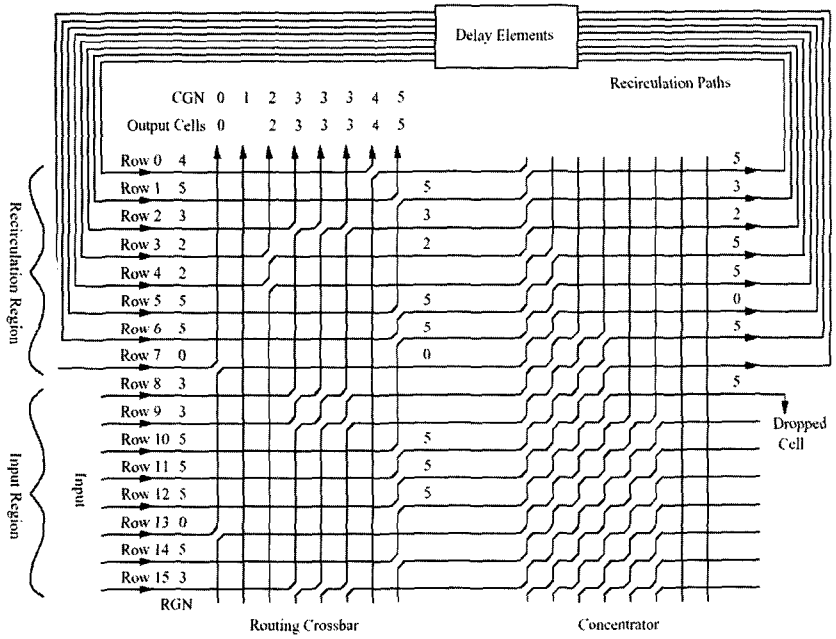


그림 1. 다중 채널 디플렉션 크로스바 구조

어가서 라우팅 크로스바의 상단에 위치한 출력 채널로 나가게 된다. 그림 1에서 표시된 것처럼 입력 셀이 들어가는 열 (row) 부분을 입력 영역 (input region)이라 부르고 순환하는 셀이 들어가는 열 부분을 순환 영역 (recirculation region)이라 부른다. 각 입력 셀은 희망 그룹 번호 (requested group number, RGN)이라 부르는 출력 단자의 주소를 갖고 있다. 이 그룹 번호 정보는 스위치 모듈 내부에서 올바른 출력 단자로 셀을 라우팅하기 위해 이용된다. 출력 채널에는 행 그룹 번호 (column group number, CGN)라 부르는 번호들이 할당되어 있다.

라우팅 크로스바에서 교차점은 2×2 스위치 소자 (switch element, SE)로 되어있다. 2×2 SE의 주요 기능은 두 입력 단자 (교차점의 왼쪽과 아래쪽)와 두 출력 단자 (교차점의 오른쪽과 위쪽) 사이의 연결 방식을 결정해 주는 것이다. 2×2 SE의 상태는 그림 2에서처럼 두 가지 중의 하나가 된다.

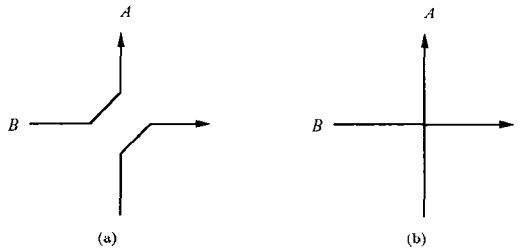


그림 2. 스위치 소자의 상태: (a) match와 (b) bypass

- Match state: 이 상태에서는 $A=B$인 경우에 해당한다. 여기서 A는 세로 선의 그룹 번호 (행의 CGN)이며, B는 가로선에 들어오는 셀의 RGN을 의미한다. 이 상태에서는 왼쪽의 입력 단자는 위쪽의 출력 단자로 연결되고 아래쪽의 입력 단자는 오른쪽의 출력 단자로 연결된다.
- Bypass state: 이 상태는 $A \neq B$의 경우에 해당된다. 이 상태에서는 왼쪽의 입력 단자는 오른쪽의 출력 단자로 연결되고, 아래쪽의 입력 단자는 위쪽의 출력 단자로 연결된다.

매 타임 슬롯마다 2×2 SE의 상태가 계산되고, 계산된 상태로 셋팅이 되면 그 경로를 따라 셀이 이동하게 된다. 셀은 왼쪽 입구로 들어와 오른쪽으로 이동하다가 match 상태인 SE를 만나게 되면 셀은 위쪽으로 굴절하게 된다. 그렇게 이동하는 셀은 다시 match 상태인 SE를 만나게 되면 오른쪽으로 굴절하게 된다. 다시 그렇게 오른쪽으로 이동하는 셀은 또 다른 match 상태의 SE를 만날 때까지 계속 수평으로 이동하고 만약 match 상태의 SE가 없을 경우 집속기의 입력 단자에 도달하여 순환하게 된다.

집속기의 주요 기능은 한 타임 슬롯에서 라우팅 크로스바의 출력 단자로 나갈 수 없는 셀을 순환시켜 다시 라우팅 크로스바의 입력 단자로 보내어 다음 번 타임 슬롯 동안에 나갈 수 있도록 하는 역할을 한다. 집속기는 그림 1에서처럼 라우팅 크로스바와 같은 수의 입력 열 수를 갖고 있다. 집속기의 2×2 SE의 상태는 논리 동작 $A+\overline{B}$에 의해 결정된다. 여기서 A는 현재 SE의 바로 위에 있는 SE의 상태를 의미한다. 위의 SE의 상태가 match이면 $A=1$, bypass이면 $A=0$이 된다. B는 현재 SE의 왼쪽으로 들어오는 셀이 있는가에 의해 결정된다. 셀이 있으면 $B=1$, 없으면 $B=0$이 된다. $A+\overline{B}$의 결과가 1이면 match state, 결과가 0이면 bypass state가 된다. 집속기에 N개의 행이 있다면, 집속기로 들어가는 셀은 최대 N열까지 윗쪽으로 이동하게 되는 것을 알 수 있다.

현재의 티임 슬롯에서 빠져나가지 못한 셀은 집속기로 가게 되어 다음 타임 슬롯에 새로이 도착하는 셀들과 빠져 나가기 위해 경쟁하게 된다. 이와 같이 두 타임 슬롯 간의 분리를 위해 그림 1에서처럼 지연 소자가 필요하다.

2. MCOC

다중 채널을 크로스바로 구현하기 위한 또 다른 방식에는 MCDC에서처럼 여러 번 굴절하지 않고 한 번만 굴절하도록 하게 하는 방식이 있다. 그림 3은 MCOC 구조를 보여준다. 이 또한 라우팅 크로스바와 집속기로 구성되어 있다. 하지만 각각의 SE는 MCDC에서의 동작 방식과는 다르다. MCOC에는 각 SE의 상태를 한 곳에서 제어하는 제어기가 있다. 라우팅 크로스바에서 CGN의 구성과 입력 단자로 들어오는 셀의 RGN에 따라 SE의 상태를 제어하며 집속기에서는 집속기로 들어오는 셀의 RGN에 따라 집속기의 SE의 상태를 제어한다. 제어기는 라우팅 크로스바와 집속기에서 입력 단자와 출력 단자 사이에 단 한번의 굴절이 생기도록 SE를 셋팅한다. (집속기에서의 입력 단자는 라우팅 크로스

바의 출력 단자가 되고 집속기의 출력 단자는 공유 버퍼의 입력이 된다.) 예를 들어, 그룹 번호가 j인 채널 그룹의 (왼쪽에서부터) 첫 번째 행은 RGN이 j인 열 중에서 가장 우에 위치한 열과의 교차점에서 단 한 번 굴절하게 된다. 채널 그룹 번호가 j인 두 번째 행은 RGN이 j인 열 중에서 위에서 두 번째에 위치한 열에서 단 한 번 굴절하게 된다. 한 셀 타임 슬롯에서 들어온 셀 중, 특정 그룹 번호(RGN)를 갖는 셀의 수가 그 RGN과 같은 값을 갖는 CGN으로 할당된 출력 채널의 수보다 많을 경우, 여분의 셀이 들어온 열에는 match state로 셋팅된 SE가 없어 그 여분의 셀은 순환을 위해 집속기로 보내지게 된다.

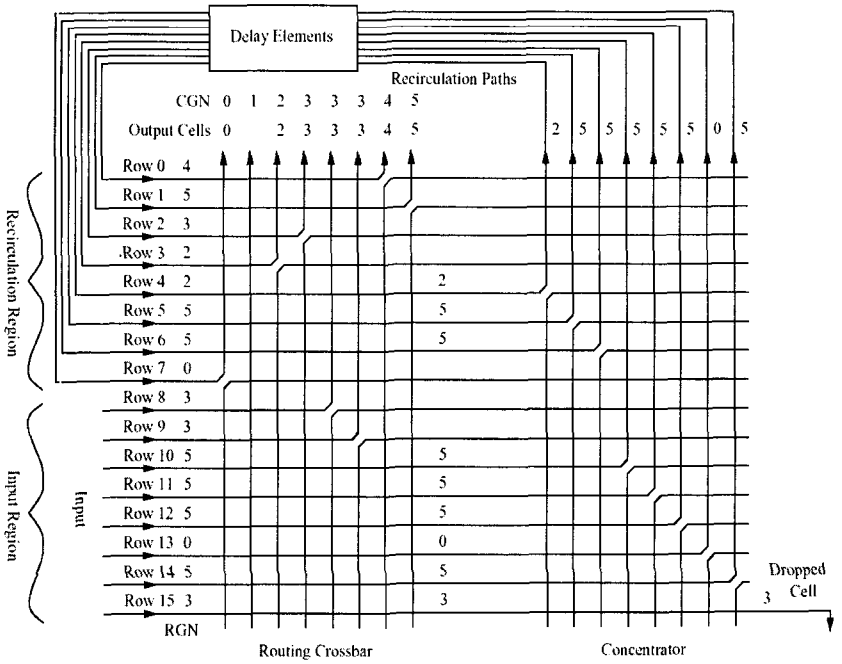


그림 3. 다중 채널 원턴 크로스바 구조

집속기는 또 다른 하나의 라우팅 크로스바로 간주하면 된다. 집속기로 들어오는 셀은 모두 같은 RGN k값을 갖고 있다고 보고 모든 출력 단자는 CGN k값이 할당되어 있다고 보면 된다. 그래서 들어오는 셀이 다음 타임 슬롯 동안 순환 영역에서 비어있는 열이 없이 모두 위쪽으로 셀이 모일 수 있도록 한다.

장애 위치를 찾아내기 위해, 시스템에서는 움직이는 셀이 자기가 들어온 입력 단자의 위치를 기억할 수 있다고 가정한다. 그래서 출력 단자에서 나온 셀이 어느 입력 단자를 통해서 들어 왔는지를 알 수가 있게 된다.

3. 장애 모델

본 논문에서는 2×2 SE에서 일어날 수 있는 장애로서 stuck-at-match (s-a-m)와 stuck-at-bypass (s-a-b)만을 가정하였다. s-a-m 장애는 외부 조건에 상관 없이 영구적으로 match 상태로 고정되어 있는 상태를 의미하며 s-a-b 장애는 영구적으로 bypass 상태로 고정되어 있는 상태를 의미한다. 또한 단지 하나의 장애만을 가정하였다. 실질적으로 여러 개의 장애가 발생하는 경우는 극히 드물다.

4. 표기

설명을 돕기 위해 몇몇 변수를 정의하고자 한다. N은 입력 영역에서 새로 도착하는 셀들을 위한 입력 단자의 수를 의미하고 또한 출력 단자의 수를 의미하기도 한다. R은 바로 직전 타임 슬롯에서 시스템을 빠져 나가지 못한 셀들을 위해 만들어진 순환 경로의 수 혹은 공유 버퍼의 수를 의미한다. M을 라우팅 크로스바에 있는 열의 총 개수라하면, $M=N+R$이 된다. 또한 시스템에서 채널 그룹의 총 개수를 K라 하고, C_k를 k번째 채널 그룹의

그룹 용량 (group capacity)라 한다. 여기서 $k=0,1,\cdots,K-1$. 그룹 용량은 한 채널 그룹에 할당된 출력 채널 (행)의 수로 정의된다. 예를 들어, 그림 1에서 $N=8$, $R=8$, $M=16$, $K=6$, $C_0=1$, $C_1=1$, $C_2=2$, $C_3=3$, $C_4=1$, $C_5=1$가 된다.

Ⅲ. 최적 알고리즘

이 절에서는 장애 SE를 찾아내는데 시간상으로 최적인 알고리즘을 제안한다. 이 알고리즘의 구현은 복잡하다. 궁극적으로 이 알고리즘의 목적은 고장난 SE의 위치를 찾는데 걸리는 시간의 최소값 (lower bound)을 구하는데 있다.

1. 알고리즘

이미 언급된 것처럼 여기서는 단지 두가지의 장애만 (s-a-m과 s-a-b) 고려하기로 한다. 한 개만의 장애가 생길 수 있다는 가정하에서, 이것은 $M\times N$ 라우팅 크로스바에서 $2MN$개의 장애의 경우 수가 있다는 것을 의미한다. 하지만 MCDC와 MCOC의 구조 및 라우팅 방식 때문에 어떤 장애는 정상적인 동작에 영향을 미치지 않는다. 왜냐하면 라우팅 크로스바에서의 어떤 SE는 셀이 지나갈 때는 언제나 match나 bypass 중 늘 같은 셋팅을 유지하게 되며, 바로 그 SE에서 그러한 셋팅으로 stuck-at 장애가 발생하였을 경우는 영향을 미치지 않게 된다. 다른 하나의 경우는 셀이 어느 특정 SE를 전혀 지나지 않는 경우도 있다. 그러한 장애를 감지 불능 장애라 부르고 그러한 장애가 $M\times N$ 라우팅 크로스바 내에 u개가 있다고 하자.

최적 알고리즘은 매 타임 슬롯마다의 라우팅 크로스바의 셋팅과 출력을 이용해서 장애의 위치를 시간적으로 가장 빨리 알아내는 알고리즘이다. 최적 알고리즘은 다음과 같이 동작한다. 우선 장애가 있다고 예상되는 스위치 시스템과 함께 $2MN-u+1$개의 이론적인 (실제 구현이 안되었으며 시뮬레이션을 적용할) 스위치 시스템이 있다고 하자. $2MN-u+1$는 위치 확인이 가능한 장애 각 1개씩 갖고 있는 $2MN-u$개의 스위치 시스템과 1개의 장애 없는 시스템으로 이루어져 있다. 알고리즘의 초기 단계에는 후보 집합에 $2MN-u+1$개의 가정 모두를 포함시킨다. 매 셀 타임 슬롯마다 시스템으로부터 나오는 셀의 입력 단자 정보와 (즉, 몇 번째 열의 단자로 들어왔는지) 다음 번 셀 타임 슬롯에서 순환할 셀이 순환 영역의 어느 입력 단자로 들어 갈 것인가를 기록해 둔다. 첫 번째 셀 타임 슬롯에서 $2MN-u+1$개의 장애 조건에 대해 똑같은 셀과 채널 셋팅을 가지고 시뮬레이션을 한다. 시뮬레이션 상에서 나오는 출력 셀의 정보 (셀의 입력 단자 정보)를 시험하고자 하는 시스템으로부터 나오는 출력 셀의 정보와 비교한다. 그 정보가 일치하면 그 장애에 대한 가정은 후보 집합에 남겨두고, 일치하지 않으면 그 장애 가정을 후보 집합에서 제거한다. 이 과정을 매 셀 타임 슬롯마다 반복한다. 시간이 지남에 따라 후보 집합에 남은 장애 가정은 줄어들 것이며, 결국에 하나만 남게되면 (이것은 확률 1의 값으로 발생하게 된다) 그 가정이 우리가 찾는 장애 조건이 되며 알고리즘은 중단된다.

장애 조건은 $2MN-u+1$개가 있다. 각각의 장애를 발견할 확률이 1이라면 우리의 장애 위치 확인 알고리즘은 시간적으로 최적이된다. 이것을 증명해 보자. 다음과 같은 장애를 가정하자. H_i, $i=1,2,\cdots,2MN-u+1$. 그리고 H_i와 연관된 발견 가능한 장애를 f_i라 하자. O_i를 시스템으로부터 나오는 셀의 입력 단자 정보와 다음 번 셀 타임 슬롯에서 순환할 셀이 순환 영역의 어느 입력 단자로 들어갈 것인가에 대한 정보라 하자. 그러면, 알고리즘이 하는 것은 현재 셀 타임 슬롯에서

$$P(f_i \mid O_1, O_2, \cdots, O_T) > 0 \qquad (1)$$

이면, H_i를 후보 집합에 남기는 것이고

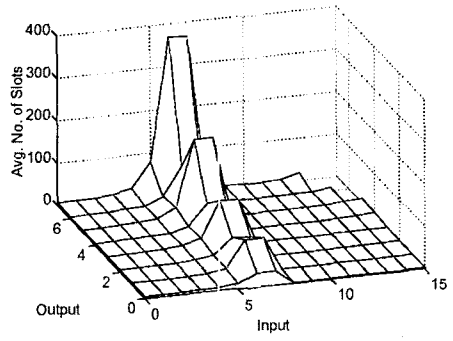
$$P(f_i \mid O_1, O_2, \cdots, O_T) = 0 \qquad (2)$$

이면, H_i를 후보 집합에서 제거하는 것이다. 만약 (2)가 만족되기 전에 후보 집합에서 H_i를 제거하면, 즉 f_i와 상관있는 가상 시뮬레이션의 출력과 시험 시스템의 출력이 일치하는데도 불구하고 제거한다고 하자. 그러면 가정 H_i가 사실일 수도 있기 때문에 올바르게 장애를 감지할 확률은 1보다 작아지게 된다. 이는 주어진 조건을 위배하는 것이 된다. 그러므로 알고리즘 장애를 감지하는데 있어 시간적으로 최적임이 증명된다.

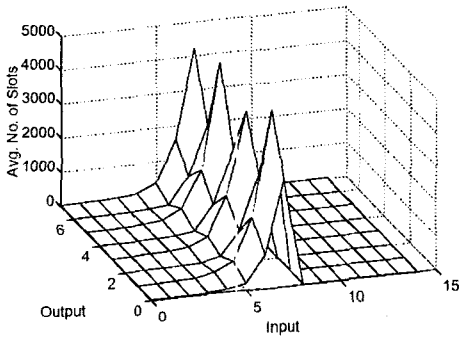
2. 최적 알고리즘의 시뮬레이션 결과

그림 4는 최적 알고리즘을 이용해서 각 SE에

서 발생한 s-a-m 및 s-a-b 장애에 대한 장애 위치의 확인 때까지 걸린 평균 타임 슬롯의 수를 보여준다. 여기서 $M=16$, $N=8$을 사용하였으며 출력 단자에서의 CGN 할당으로 (0,0,1,1,2,2,3,3)을 사용하였다. 입력 셀들의 RGN 분배는 시스템에서 할당된 채널 그룹의 그룹 용량에 비례하고 시간적으로 균일하게 분포 (uniformly distributed)되었다고 가정하였다. 즉, 우리가 든 예에서 입력 셀이 RGN 0,1,2,3을 가질 각각의 확률은 1/4, 1/4, 1/4, 1/4이 된다. 또한 한 입력 단자의 트래픽은 다른 입력 단자의 트래픽에 독립적 (independent)이라고 가정하였다. 실험에서 0.7의 호 밀도 (traffic intensity)를 사용하였다. 즉, 평균적으로 10개의 셀



(a) s-a-m



(b) s-a-b

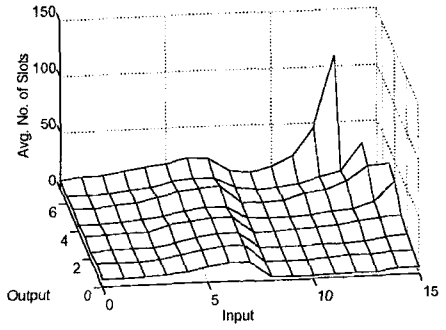
그림 4 최적 알고리즘을 MCDC에 적용하였을 경우의 슬롯 수

타임 슬롯 중 7번이 들어오는 것을 의미한다.

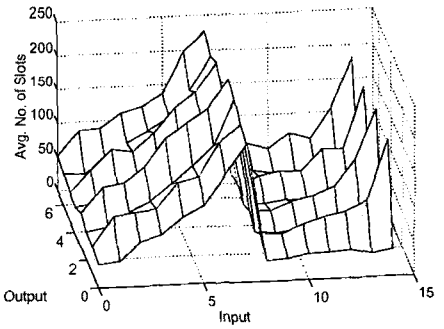
우선 s-a-m 장애 위치 확인에 걸리는 시간보다 s-a-b에 걸리는 시간이 평균적으로 크다는 것을 알 수 있다. 이것은 s-a-m (s-a-b)의 위치 확인은 해당 위치의 SE가 bypass (match)로 셋팅되어 있고 (이러한 반대의 셋팅은 셀로 하여금 다른 방향으로 가게끔하여 장애가 없는 경우와는 다른 셀의 출력을 야기하게 하기 위함이다), 한 셀 타임 슬롯에서 SE가 match로 셋팅될 확률이 bypass로 셋팅될 확률보다 작기 때문이다.

s-a-m과 s-a-b의 경우 모두 장애 위치 확인을 위해 소요되는 시간이 다른 위치보다 순환 영역의 아래 부분 (5, 6, 7열)에서 큰 것을 알 수 있다. 이것은 셀들이 집속기로 들어가면 MCDC의 경우 최대 N열까지 위쪽으로 이동하여, 순환 경로를 지나온 셀들이 순환 영역의 아래 부분으로 들어가게 될 확률이 작기 때문이다.

그림 4(a)의 s-a-m 경우에 장애 위치 확인을 위해 필요한 슬롯의 수는 입력 단자에서 멀리 떨어진 경우가 가까운 경우보다 크다는 것을 알 수 있다. 반면 그림 4(b)의 s-a-b 경우 입력 단자로부터의 거리에 상관 없이 균일한 것을 알 수 있다. 이것은 다음과 같은 이유에서이다. 장애를 감지하기 위해서는 하나 이상의 셀에서 나온 정보가 일치한지 않아야 한다. 일치하지 않는 셀 정보를 갖기 위해서는 s-a-m (s-a-b) 장애는 그 위치의 SE는 bypass (match)로 셋팅되어야 하고 그 SE로 셀이 지나 가야만 한다. 입력 단자로부터 멀리 떨어져 있는 SE에 s-a-m이 발생했을 경우 셀이 그 SE에 도달하기



(a) s-a-m



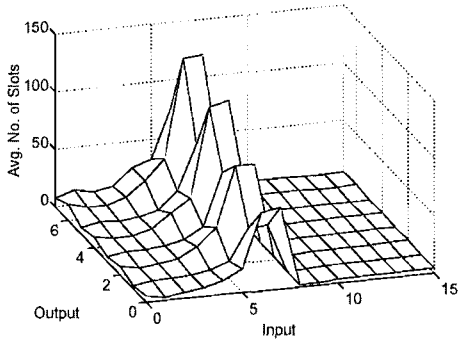
(b) s-a-b

그림 5. 버스트한 트래픽에 (B=5) 대해 최적 알고리즘을 MCDC에 적용하였을 경우의 슬롯 수

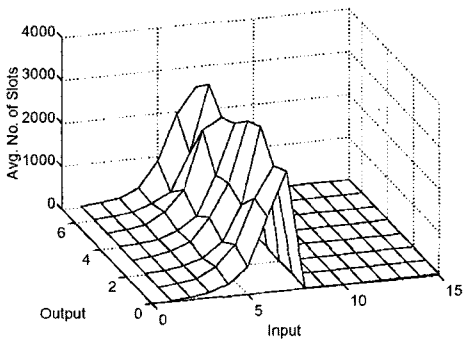
전에 시스템의 출력 단자로 나오기 때문에 감지될 확률이 작게 된다. 그러나 s-a-b의 경우, 장애가 일어난 위치에서 꺽어지기만 하면 장애 감지가 되고 꺽어지려는 셀의 분포가 균일하기 때문에 장애 위치 확인을 위해 걸리는 시간은 균일한 분포를 갖게 된다.

그림 5는 입력 트래픽이 버스트할 때에 장애 위치 확인에 걸리는 시간을 보여준다. 셀이 도착하는 프로세스가 ON/OFF 프로세스라고 가정하였다. 셀의 도착은 ON 상태에서 이루어진다. ON과 OFF 상태가 주기의 길이는 서로 독립적이며 각각 $1/\alpha$와 $1/\beta$를 평균값으로 갖는 지수 분포를 갖는다. 여기서 버스트 지수 (burstiness)를 ON/OFF 프로세스에서 ON 상태에 대한 지수 분포의 평균값으로, 즉 $1/\alpha$로 정의하고 호 밀도를 β/α로 정의한다. 시뮬레이션을 위해 5의 버스트 지수를 사용했으며 0.7의 호 밀도를 사용했다.

버스트한 트래픽의 경우 (그림 5) 버스트하지 않은 트래픽이 들어왔을 때보다 (그림 4) 장애 위치 확인에 걸리는 시간이 확연히 줄어드는 것을 알 수



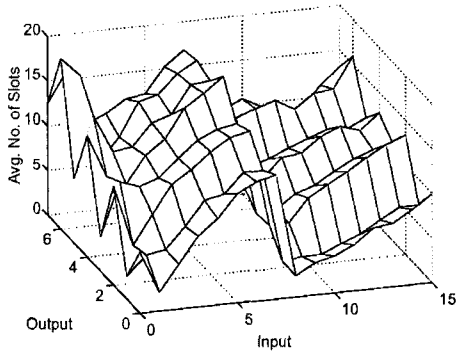
(a) s-a-m



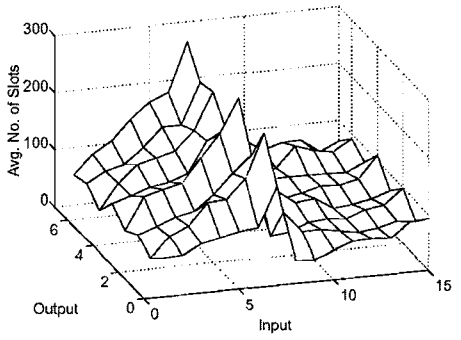
(b) s-a-b

그림 6. 최적 알고리즘을 MCOC에 적용하였을 경우의 슬롯 수

있다. 이것은 입력 셀의 버스트한 특성이 순환하는 셀을 보다 많이 만들어서 순환 영역의 아래 부분에 도달하는 셀들의 수가 많아지기 때문이다.



(a) s-a-m



(b) s-a-b

그림 7. 버스트한 트래픽에 (B=5) 대해 최적 알고리즘을 MCOC에 적용하였을 경우의 슬롯 수

그림 6은 MCOC에 최적 알고리즘을 적용해서 각 SE에 s-a-m과 s-a-b의 장애가 생겼을 경우, 장애 위치 확인에 걸리는 시간을 보여준다. MCDC의 경우와 같은 조건을 이용하였다. 이 경우 순환 영역에서 걸리는 시간의 경사가 MCDC보다 MCOC의 경우가 입력 (input) 축을 따라서 완만한 것을 알 수 있다. 이것은 MCDC의 집속기가 셀을 최고 N열 만큼 밖에 위로 이동시키지 않아 (N 이상을 위로 이동시키면 빈 열 없이 채울 수 있음에도 불구하고) 순환 영역의 아래 부분에 셀이 나타나게끔 한다. 반면 MCOC에서는 집속기가 셀을 빈 열이 없도록 위로 이동시키기 때문에 순환하는 셀은 늘 순환 영역의 위쪽을 채우게 되기 때문에 순환 영역에서 조금만 밑으로 내려가도 SE가 셀을 만날 수 있는 기회가 줄어들기 때문이다. 기회가 줄어들면 그 만큼 장애 위치 확인에 시간이 더 걸리게 된다.

그림 7은 버스트한 트래픽의 경우에 최적 알고리즘을 사용하였을 때에 MCOC의 각 SE에 s-a-m과

s-a-b의 장애의 위치를 확인하는데 걸리는 시간을 보여준다.

Ⅳ. 장애 복구 알고리즘

지금까지 장애 위치 확인 알고리즘에 대해 설명하였으며, 본 절에서는 위치를 알아낸 장애에 대해 복구하는 알고리즘을 제안한다.

전체 스위치 시스템을 교체하는 것 대신에, 장애 SE에 대한 대안으로 열과 행을 공유하는 방식을 이용하기로 한다. 그림 8은 정상적인 라우팅 크로스바에 두 개의 행 및 두 개의 열을 추가한 모습을 보여준다. 한 개의 열, 한 개의 행, 한 개의 입력 시프터 (input shifter) 행, 한 개의 출력 시프터 (output shifter) 열이다. 입력 시프터 행과 출력 시프터 열은 경로를 보정하기 위해서 필요하다. 앞서 설명된 장애 위치 확인 알고리즘을 통해 i열, j행에 발생한 2×2 SE 장애가 검출되었다고 하자. 그러면 그림에서 보여주는대로 추가한 열과 행을 이용하여 열과 행을 재할당함으로써 발생한 장애가 올바른 스위치 동작에 영향을 주지 않도록 할 수 있다. 다시 말해 셀로 하여금 장애가 발생한 경로를 비켜 가게끔하는 것이다.

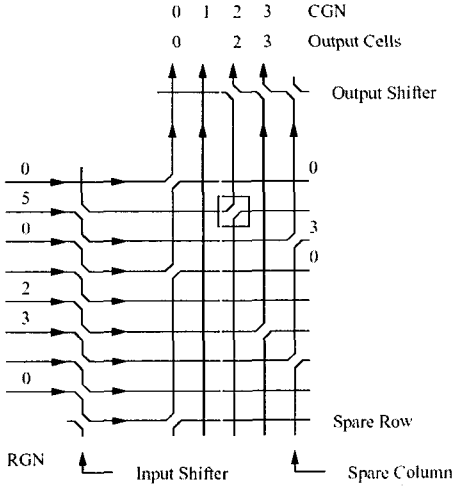


그림 8. 라우팅 크로스바에서의 장애 복구

라우팅 크로스바에서 발생한 한 개의 장애를 복구하기 위해서 두 개의 열과 행이 필요하다. 그림 8에서 정사각형 상자로 표시된 SE에 장애가 발생하였다고 가정하자. 그러면 입력 시프터는 셀들을 장애가 발생한 열의 아래 쪽에 있는 모든 열에 대해 한 열씩 아래로 이동시키게 된다. 또한 장애가 발생한 행을 사용하지 않고, 장애 행의 오른쪽으로 한 행씩 (즉, CGN을) 이동하게 한다. 출력 시프터는 출력 채널과 정렬시키기 위해 다시 한 행씩 왼쪽으로 이동시키도록 만든다. 이러한 방식은 k개의 장애를 마스킹하는데 이용될 수 있다. 예를 들어, 2개의 장애를 마스킹시키기 위해 4개의 추가 열 (2개의 입력 시프터와 2개의 추가 열)과 4개의 추가 행(2개의 출력 시프터와 2개의 추가 행)이 필요하다.

Ⅴ. 온라인 장애 위치 확인 알고리즘

이절에서는 온라인 장애 위치 확인 방식을 제안한다. 이 방식은 계산상으로 최적 알고리즘보다 효율적이며, 성능은 거의 최적 알고리즘에 가깝다. 먼저 우리는 제안한 방식의 기본 개념을 설명한다. 그리고 장애 위치 확인 알고리즘을 설명한다. 끝으로 제안된 알고리즘에 의해 한 개의 SE까지는 장애의 위치가 확인되지 않고 대략적인 위치를 묶음으로까지만 위치 확인이 가능한 불완전 위치 확인 장애에 대해 논한다.

1. 알고리즘

여기서도 새로 도착하거나 순환하는 셀들의 입력 단자 정보를 이용한다. 일단 입력 셀들의 RGN과 출력 단자에서의 CGN이 알려지면 우리는 앞에서 설명된 MCDC나 MCOC의 구성 규칙에 따라 장애가 없을 때에 셀이 흐르는 경로를 파악할 수 있다. 그러한 장애 없는 온전한 시스템으로부터의 온전한 출력 셀 정보와 SE 장애가 있을 것으로 추정되는 시스템으로부터의 출력 셀 정보를 비교하게 된다.

여기서 s-a-m 지수 (indicator) 행렬 H_m과 s-a-b 지수 행렬 H_b를 정의한다. 지수 행렬의 크기는 라우팅 크로스바의 크기 ($M \times N$)와 동일하다. H_m과 H_b의 각 원소는 각각 s-a-m 및 s-a-b 장애에 대한 의심값, 즉 심증의 정도를 나타낸다. 즉 H_m과 H_b의 원소에서 높은 숫자는 각 s-a-m와 s-a-b에 대한 장애가 있을 확률이 높음을 알려준다.

의심값은 초기에 0으로 셋팅된다. 위치 확인 절차가 진행됨에 따라 의심값은 매 셀 타임 슬롯에 온전한 시스템의 출력과 시험 시스템의 출력이 일치하지 않았을 때마다 일치하지 않는 셀의 온전한 경로를 따라 1씩 증가시킨다. 여기서 온전한 경로는 장애가 없었을 경우에 가야 할 경로를 의미한다. 시간이 흐름에도 불구하고 H_m과 H_b에서 원소의 값이 계속 0으로 유지되는 것은 그 위치에 각각 s-a-m과 s-a-b

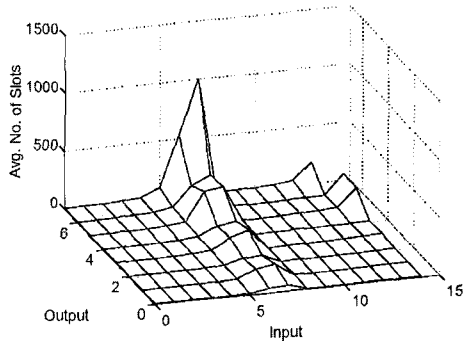
장애가 있을 확률이 작다는 것을 의미한다.

우리는 이미 입력 단자로 들어가는 셀의 RGN과 출력 채널의 CGN을 알기 때문에 시스템으로 들어가는 셀들의 경로를 계산할 수 있으며 출력 단자로부터 무슨 셀이 나와야 한다는 것을 예측할 수 있다. 이미 설명한대로 MCDC나 MCOC의 내부를 흐르는 셀들은 각각의 입력 단자 정보를 유지하고 있다. 우리는 시험 대상인 MCDC (혹은 MCOC)로부터 나오는 셀의 입력 단자 정보와 온전한 MCDC (혹은 MCOC)로부터 나오는 셀의 입력 단자 정보를 비교한다. 만약 두 정보가 일치하지 않는다면, 셀의 경로에 한 개 이상의 장애가 있다는 것을 의미하고 H_m과 H_b에 대응하는 원소의 값을 다음의 규칙에 따라 증가시킨다. 만약 온전한 경로 상의 SE가 bypass 상태로 셋탱되었다면 H_m의 대응하는 원소의 값을 1 만큼 증가시키고, SE가 match 상태로 셋팅되었다면 H_b의 대응하는 원소의 값을 1 만큼 증가시킨다. 만약 두 정보가 일치한다면, 우리는 셀의 경로 상에 장애가 없다고 판단하여 경로 상의 의심값을 0으로 리셋한다. 온전한 경로상의 원소가 bypass 상태로 셋팅되었다면 H_m의 대응하는 원소의 값을 0으로 만들고, match 상태로 셋팅되었다면, H_b의 대응하는 원소의 값을 0으로 만든다. 일단 0으로 리셋된 원소는 장애가 없는 것으로 판단하여 다시는 의심값을 증가시키지 않는다. 이러한 절차를 H_m과 H_b에서 단 하나의 최대값이 나올 때까지 계속 반복한다.

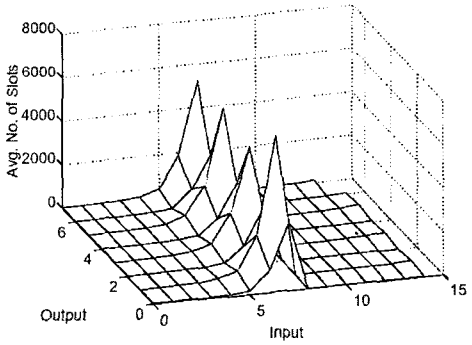
명제: 만약 H_m과 H_b에서 의심값의 최대치가 하나만 있을 경우, 그 원소에 대응하는 SE가 장애 원소가 된다.

증명: 온전한 경로 상의 장애에 의해 셀이 잘못 라우팅되었을 경우 온전한 경로에 대응하는 의심값이 증가하게 된다. 장애 위치 확인이 가능한 (즉, 위에서 설명한 알고리즘에 의해 장애 위치 확인이 가능한 위치; 위치 확인의 조건은 아래에 설명된다) 위치 A에 s-a-m 장애가 발생하였다고 가정하자. 그리고 H_m과 H_b 중 H_m의 A 위치가 아닌 다른 위치인 B에서 유일한 최대값이 생겼다고 하자. H_m의 위치 A에서의 의심값을 S_A라고 하고 B의 의심값을 S_B라고 하자. 가정에 의해 $S_A < S_B$이 된다. 불일치하는 출력 셀 정보는 모두 A에 위치하는 s-a-m 장애에 의한 것이기 때문에 출력 셀 정보가 불일치할 때 마다 알고리즘은 늘 H_m의 A에 위치하는 의심값을 증가시키게 된다. 그러므로 $S_A \geq S_B$, 즉, 모순이 된다. 이것으로 s-a-m 경우에 대한 명제의 증명이 된다. s-a-b 장애의 경우에도 증명은 마찬가지가 된다. **Q.E.D.**

이 알고리즘은 H_m이나 H_b에서 유일한 최대값이 하나가 나올 때까지 반복된다. 시간이 충분히 지나게 되면 대부분의 의심값이 0으로 된다. 이는 셀의 경로에 장애가 없는 것으로 판명이 되면 대부분의 SE가 s-a-m이나 s-a-b 장애가 없는 것으로 표시되기 때문이다.



(a) s-a-m



(b) s-a-b

그림 9. 온라인 알고리즘을 MCDC에 적용하였을 경우의 습못 수

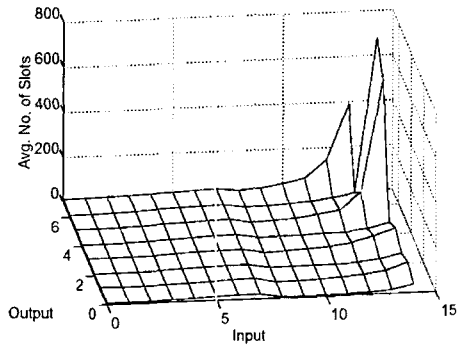
2. 위치 확인이 불완전한 장애

위에서 제안된 알고리즘이 한 개의 장애까지 정확히 위치를 찾아내지 못하는 경우가 있다. 이러한 경우는 H_m이나 H_b에서 유일한 최대값이 없을 경우에 발생한다. 어떤 경우에는 셀이 장애가 있는 SE에 전혀 도달하지 않는 경우도 있다. 혹은 장애 SE가 s-a-m이나 s-a-b 둘 중 하나의 상태로 굳어져 있고 어떤 셀들이 들어오더라도 그 SE의 원하는 셋팅이 이미 굳어진 상태인 경우도 있다. 그러한 경우에는 H_m이나 H_b의 의심값이 증가하지 않는다. 이러한 감지 불능의 장애는 CGN 구성인 변하지 않

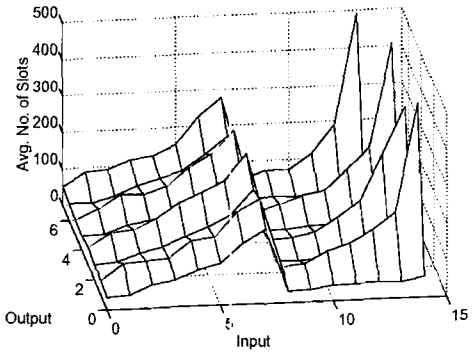
는 한 정상적인 동작에 영향을 미치지 않는다.

3. 온라인 위치 확인 알고리즘의 시뮬레이션 결과

최적 알고리즘의 경우와 같은 조건을 적용하였다. 즉, $M=16$, $N=8$이며, (0,0,1,1,2,2,3,3)의 CGN 구성을 이용하였다. 입력 셀의 RGN이 CGN의 그룹 용량에 비례하여 균일하게 분포되었다고 가정한다. 호 밀도는 0.7이며 한 입력



(a) s-a-m



(b) s-a-b

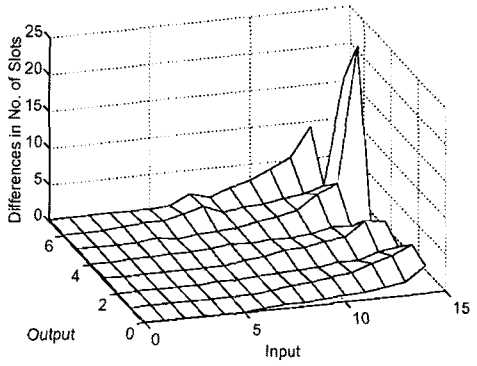
그림 10. 버스트한 트래픽에 (B=5) 대해 온라인 알고리즘을 MCDC에 적용하였을 경우의 슬롯 수

단자의 트래픽은 다른 입력 단자의 트래픽에 독립적이라고 가정한다.

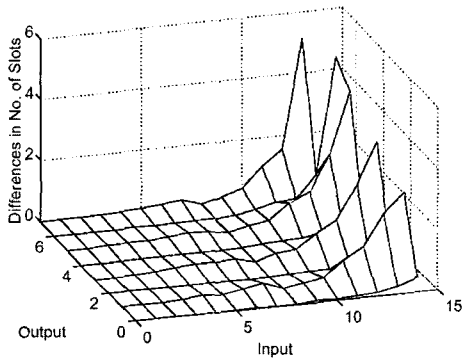
그림 9는 MCDC에서 대응하는 위치에서 s-a-m와 s-a-b 장애에 대해 100번의 시뮬레이션에 대한 평균치를 보여준다. MCOC의 경우에도 결과 그래프는 유사한 형태를 보여준다.

또한 버스트한 트리픽에 대해서도 시뮬레이션을 하였다. 그림 10은 MCDC에서 버스트한 입력에 대해 위치 확인에 걸리는 셀 타임 슬롯의 수를 보여준다.

그림 11은 온라인 알고리즘을 이용해서 장애 위치 확인에 걸리는 평균 시간과 최적 알고리즘으로 걸리는 시간의 차이를 보여준다. 온라인 알고리즘을 통한 셀 타임 슬롯의 수를 n_H라



(a) s-a-m



(b) s-a-b

그림 11. MCDC에서 최적 알고리즘과 온라인 알고리즘을 이용한 슬롯 수의 차이

하고 최적 알고리즘을 통한 셀 타임 슬롯의 수를 n_O라 하면, 그래프에서 보여주는 수는 $(n_H-n_O)/n_O$가 된다. 그래프는 대부분의 위치에서 온라인 알고리즘을 통한 셀 타임 슬롯의 수가 최적 알고리즘을 통한 셀 타임 슬롯에 가깝다는 것을 보여준다.

VI. 결 론

우리가 분석한 문제는 다중 채널 크로스바 스위치에서 발생할 수 있는 2×2 소자의 장애의 위치를 빨리 찾아 내고자하는 것이었다.

우리는 stuck-at-match (s-a-m) 및 stuck-at-bypass (s-a-b) 장애에 대해 다중 채널 크로스바 스위치의 표준인 Multichannel Deflection Crossbar (MCDC) 및 Multichannel One-turn Crossbar (MCOC)에서

최적 알고리즘을 포함한 2개의 장애 위치 확인 알고리즘을 제안했다.

최적 알고리즘은 시간상으로 가장 좋은 성능을 갖고 있으나 계산상의 복잡성 때문에 구현이 어렵다. 반면 온라인 알고리즘은 최적 알고리즘보다는 계산상으로 보다 효율적이다. 시뮬레이션 결과는 s-a-b 장애가 위치를 확인하는데 걸리는 시간이 s-a-m보다 평균적으로 더 걸리는 것을 보여주었다. 이것은 s-a-m (s-a-b) 장애의 위치 확인을 위해서는 장애 SE의 셋팅이 bypass (match)로 되어야 하며 한 슬롯에서 match로 셋팅될 확률이 bypass로 셋팅될 확률보다 적어 s-a-b 장애의 위치 확인에 보다 많은 시간이 필요로 하게 된다. 시뮬레이션 결과는 버스트한 입력 트래픽이 장애 위치 확인에 걸리는 시간을 줄여 준다. 이것은 입력 셀의 버스트한 성질이 셀로 하여금 버스트하지 않은 트래픽으로는 도달하기 힘든 순환 영역에 도달하게끔 한다.

열과 행을 추가하여 장애 소자를 우회하도록하는 장애 복구 알고리즘을 제안하였다.

온라인 알고리즘은 계산상으로 최적 알고리즘보다 효율적이나 성능은 최적에 가깝다. 만약 온라인 알고리즘의 계산 상 복잡성이 비현실적이라 한다면 출력 셀의 정보를 저장하여 나중에 이 정보를 이용하여 장애 소자의 위치를 확인할 수 있다.

참 고 문 헌

[1] A. Pattavina, "Multichannel bandwidth allocation in a broadband packet switch," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1489-1499, Dec. 1988.

[2] R. L. Cruz, "The statistical data fork: A class of broad-band multichannel switches," *IEEE Transactions on Computers*, vol. 40, no. 10, pp. 1625-1634, Oct. 1992.

[3] H. S. Kim, "Multichannel ATM switch with preserved packet sequence," in *IEEE International Conference on Communications*, 1992, vol. 3, pp. 1634-1638.

[5] P. S. Min, H. Saidi, and M. V. Hegde, "Nonblocking architecture for broadband multi-channel switching," *IEEE/ACM Transactions on Networking*, vol. 3, no. 2, pp. 181-198, 1995.

[6] T.-H. Cheng, "Design and analysis of a multichannel transmission scheme," *Computer Networks and ISDN Systems*, vol. 29, no. 2, pp. 209-220, Jan. 1997.

[7] P. Y. Yan, K. S. Kim, P. S. Min, and M. V. Hegde, "Multi-channel deflection crossbar (MCDC): A VLSI optimized architecture for multi-channel ATM switching," in *Proceedings of INFOCOM '97, Kobe, Japan*, Apr. 1997, pp. 12-19.

[8] K.-B. Kim, P. Y. Yan, K.-S. Kim, O. Schmid, and P. S. Min, "A growable ATM switch with embedded multi-channel multicasting property," in *IEEE GLOBECOM*, Nov. 1997, pp. 222-226.

[9] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 2nd edition, 1992.

[10] T. Anderson, *Fault Tolerance: Principle and Practice*, Prentice Hall, 1981.

[11] A. T. Bouloutas, S. Calo, and A. Finkel, "Alarm correlation and fault identification in communication network," *IEEE Transactions on Communications*, vol. 42, pp. 523-533, 1994.

[12] I. Katzela and M. Schwartz, "Schemes for fault identification in communication networks," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 753-764, 1995.

[13] A. A. Lazar, W. Wang, and R. H. Deng, "Models and algorithms for network fault detection and identification: A review," in *Communications on the Move. ICCS/ISITA '92, 1992*, vol. 3, pp. 999-1003.

[14] P. Y. Yan, *Crossbar Architecture for Broadband Switching*, D. Sc., Washington University, St. Louis, MO, 1997.

[15] M.-S. Oh and P. S. Min, "Reliability analysis for one-turn and deflection crossbar architectures and distributed fault recovery scheme," in *IEEE GLOBECOM*, Nov. 1997, pp. 227-231.

[16] M.-S. Oh, *Detection, Localization, and Recovery of Faults in Communication Networks*, D. Sc., Washington University, St. Louis, MO, 1998.

오 민 석(Minseok Oh) 정회원

1987년 2월 : 서울대학교 전기 공학과 졸업
1993년 5월 : Columbia University 전기 공학과 석사
1998년 5월 : Washington University 전기 공학과 박사
1998년 7월 : minMax Technology
1999년 3월 : AT&T
2000년 3월 ~ 현재 : LG TeleCom

<주관심분야> 망관리, 이동통신