

# 반얀망 ATM 스위치에서의 내부 충돌 제거를 위한 입력 포트 재할당 기법

정회원 이주영\*, 정준모\*, 고광철\*\*, 정재일\*\*

## Input Port re-allocation technique for the elimination of the internal blocking in banyan ATM switches

Joo-young Lee\*, Jun-mo Jung\*, Kwang-cheol Ko\*\*, Jae-il Jung\*\* *Regular Members*

### 요 약

본 논문에서는 반얀망(banyan network)을 적용한 ATM(Asynchronous Transfer Mode) 스위치에서의 스루풋을 크게 감소시키는 셀들간의 내부 충돌을 제거하기 위한 새로운 기법을 제안하고 그 성능을 평가한다. 제안한 기법은 반얀망을 구성하는 SE(Switching Element)와 이들 SE간을 연결하는 내부 링크의 규칙적 연결 특성을 이용하여, 초기 반얀망 스위치에 도착한 셀들간의 내부 충돌 여부를 확인하고, 충돌이 확인된 셀들에 대해 새로운 입력 포트 주소를 재할당 함으로써, 모든 셀들의 충돌 없는 전송이 가능하도록 한다.

제안한 기법을 위하여, 초기 입력된 셀들의 스테이지(stage)별 전송 주소 및 이들 각 셀에 대해 충돌이 가능한 입, 출력 포트 주소를 비교할 수 있는 충돌 확인 테이블을 구성하여, 셀들이 각 스테이지를 거쳐 최종 목적 주소 까지 전송되는 과정에서의 중간 경로 공유 여부를 확인하여 충돌/비충돌 여부를 결정한다. 충돌 셀로 분류된 셀들은 이들 충돌 셀들을 제외된 비충돌 셀들로 구성된 스테이지별 충돌 확인 테이블을 ORing한 입력 주소 할당 테이블을 통해 내부 충돌이 없는 입력 주소를 재할당한다. 제안한 기법은 C 언어로 구현된 NBPG 시뮬레이터를 통하여 그 성능을 평가한다

### ABSTRACT

The banyan network is a popular and basic structure of the multi-stage ATM switches. This paper presents a novel approach to resolve the internal blocking of the banyan network by using a Non-Blocking Permutation Generator (NBPG). The NBPG performs two functions, i.e., the first is to extract the conflict cells from the incoming cells and the second is to re-assign new input port addresses to the conflict cells. As a result, NBPG generates non-blocking I/O permutations. To estimate the performance of the NBPG, we provide the results of several simulations.

### 1. 서 론

반얀망은 구조가 간단하며 동일한 공간 분할형 계열의 망에 비해 I/O 포트 수 증가에 따른 확장이 용이하여 공간 분할형 ATM 스위치에서 주로 사용된다. 같은 공간 분할형 스위치인  $N \times N$  크로스바 망(crossbar network) 스위치의 경우,  $N^2$ 개의 SE로

서 입, 출력 포트들간의 연결 경로를 갖는 것에 비해,  $N \times N$  반얀망은  $(N/2)\log_2 N$ 개의 SE만으로 연결 경로를 가지며, 이러한 연결 경로는 모든 입, 출력 포트간의 독립적이고 유일한 경로가 된다. 그러나 이러한 특징은 반얀망이 갖는 장점인 동시에 단점이 된다. 즉, 소수의 SE만으로 모든 입, 출력간의 셀 전송이 가능한 대신, 상대적으로 적은 수의 SE로 구성되므로 망 구성의 기본 단위인  $2 \times 2$  SE의

\* 서경대 전자공학과

\*\* 한양대 전자전기컴퓨터공학과

논문번호: 020104-0309, 접수일자: 2002년 3월 9일

※ 이 논문은 2001년 한양대학교 교내연구비 지원으로 연구 되었음

두 입력 링크로부터의 셀들은 출력 링크를 모두 공유하므로 이로 인한 셀들간의 충돌이 발생할 수 있다. 내부 충돌로 정의되는 이러한 충돌은 포트 수가 증가함에 따라 함께 증가하여, 결국 망의 스루풋(throughput)을 크게 저하시킨다. 그러므로, 충돌망으로 분류되는 반안망을 라우팅(routing) 망으로 채택하는 모든 반안망 계열 ATM 스위치는 이러한 내부 충돌을 줄이거나 제거하기 위한 방법들을 별도로 마련하고 있다<sup>1)2)3)4)</sup>.

반안망은 셀들이 도착한 입력 포트를  $x_1, \dots, x_k$ , 그리고 각각의 목적 포트를  $y_1, \dots, y_k$ 라고 할 때, 이 셀들의 목적 포트가  $y_1 < \dots < y_k$  또는  $y_1 > \dots > y_k$ 인 경우, 그리고 셀이 도착한 입력 포트를 활성(active) 입력 포트라고 할 때, 두 활성 입력 포트간의 임의의 입력 포트  $w$  역시 활성 입력 포트인 경우엔 내부 충돌이 발생하지 않는 특성을 갖고 있다. 이러한 특성은 반안망을 이용한 비충돌 스위치를 구성하기 위해 다양하게 이용되고 있으며, 특히, 배치(Batcher)-반안망 스위치는 셀들의 순서를 제어하는 정렬망과 셀들을 전송하는 라우팅 망이 결합한 구조로서, 입력 포트에 도착한 셀들을 목적 포트의 오름차순이나 내림차순으로 재배치하여 내부 충돌을 완벽하게 제거하는 대표적인 스위치이다. 그러나, 배치-반안망 스위치에서 이용하고 있는 목적 포트의 패턴은 목적 포트의 오름차순이나 내림차순만을 이용하는 방식으로서 실제로  $2 \times 2$  SE를 기본 단위로 망을 구성하는 반안망에서 존재하는  $2^{nN/2}$  ( $n=\log_2 N$ )개의 비충돌 목적 포트 패턴 중, 오직 두 가지만을 이용하는 방식이다. 또한, 배치망은 다수개의 비교기로 구성되는 구조로서 실제 구현에 있어, 입, 출력 포트 수 증가에 따른 상당한 복잡도를 필요로 한다.

본 논문은 초기 입력된 셀들 중 충돌을 일으키는 셀들의 입력 포트를 재배치하여 내부 충돌을 제거하는 입력 포트 재할당 기법을 제안한다. 초기에 입력된 셀들은 우선 충돌 셀과 비충돌 셀로 분류되며, 여기서 분류된 충돌 셀들은 본 논문에서 제안하는 테이블 방법을 이용하여 충돌이 발생하지 않는 새로운 입력 포트에 주소 값이 재할당된다. 제안한 기법은 시뮬레이션을 통해, 그 유효성을 입증한다.

## II. 반안망의 구조와 특징

MIN(Multistage Interconnection Network)의 일종인 반안망은 입, 출력 포트 수가  $N \times N$  인 경우,

총  $n(n=\log_2 N)$ 개의 스테이지를 가지며 각 스테이지는  $2 \times 2$  크로스바 망으로 구성된  $N/2$ 개의 SE들로 구성된다. 또한, 스테이지 간 SE는 서로 반안 교환 연결을 갖는다. 입력 포트에 도착한 셀들은 목적 포트의 2진 주소 값에서 각 스테이지에 해당하는 비트 값을 통해 상위 또는 하위 링크로 전송된다. 예를 들어, 한 셀의 목적 포트 2진 주소 값이  $a_0 \dots a_{n-1}$ 과 같다면, 이 셀은 임의의 스테이지  $k$ 에선  $a_k$ 의 비트 정보, 즉, 그 값이 '0'이면 SE의 상위 링크로, '1'이면 하위 링크로 전송된다. 따라서, 총  $n$ 개 스테이지로 목적 포트까지 셀이 전송되기 위해선  $n$ 비트의 비트 정보가 필요하다.

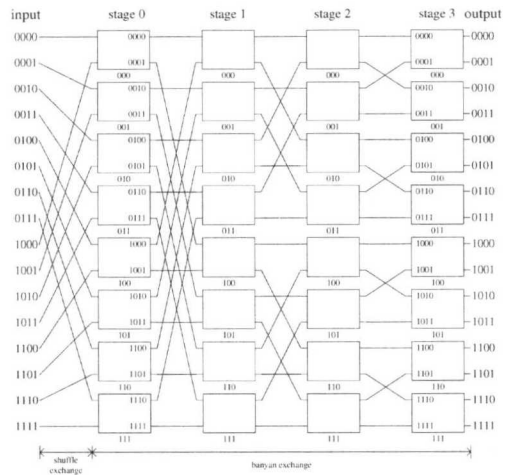


그림 1.  $16 \times 16$  반안망의 예

그림 1에서는  $16 \times 16$  반안망의 예를 들었다. 입력 포트와 반안망의 연결은, 편의상 완전 셔플 교환 연결(perfect shuffle exchange connection)로 구성하였으며, 반안망 내부의 스테이지간 연결은 반안 교환 연결로 연결하였다. 2진수로 표기된 망 좌측의 입력 포트 0000에서 1111까지 진입하는 셀들은 각각 원하는 목적 포트인 망 좌측 출력 포트로서 스테이지 0에서 3까지의 4개의 스테이지를 거쳐 임의의 출력 포트에 전송될 수 있으며, 이때의 각 경로는 해당 입, 출력 포트간의 유일한 경로가 된다. 그러나, 임의의 두 셀은 반안망 내부의 한 SE에 진입하여 동일한 한 SE 출력 링크로 전송을 요구하여 내부 충돌이 발생할 수 있다. 이러한 경우, 한 셀은 드롭(drop) 되고 다른 한 셀만이 전송되게 된다.

이 충돌 과정을 보다 상세하게 알기 위해, 반안망의 물리적 연결 구조를 살펴보면, 한 SE는 반안망의 망 구성 정의에 따라  $2 \times 2$  크로스바 망으로 구

성되므로, 임의의 스테이지, 임의의 SE를 고려하면 이들 구조는 SE를 중심으로 입력 포트 측과 출력 포트 측 각각에 대해 그림 2와 같은 트리(tree) 구조를 가짐을 알 수 있다.

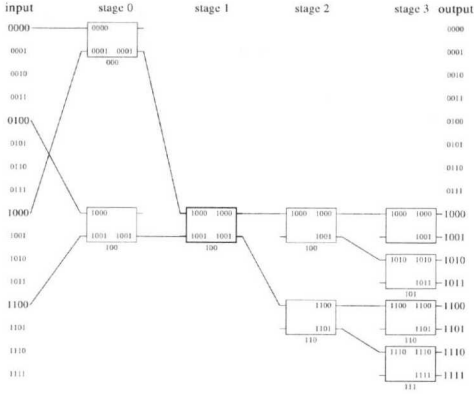


그림 2. 16×16 반얀망에서의 트리 구조의 예

즉, 그림 2와 같이 16×16 반얀망에서 스테이지 1의 SE 100에서 두 셀이 같은 출력 포트 1000에서 1111까지의 전송을 요구한다고 가정할 때, 스테이지 1의 해당 SE 100으로 셀을 보낼 수 있는 입력 포트는 0001, 0100, 1000 그리고 1100이며, 반대로 해당 SE를 통해 전송될 수 있는 출력 포트는 스테이지 2의 SE 100, 110을 통해, 스테이지 3의 SE 100, 101, 110, 111을 거쳐 출력 포트 1000에서 1111까지의 전송이 가능하게 되므로, 해당 SE를 중심으로 좌, 우는 각각 트리 구조의 연결 형태를 갖는다.

이러한 구조적 특성을 이용하여,  $N \times N$  반얀망에서의 임의의 스테이지  $k$ 에서의 충돌을 고려하여 보았다. 우선, 반얀망을 구성하는 링크 연결에 관한 표현은 토폴로지 표기법<sup>[5]</sup>(topology describing rule)에 의해, 다음과 같이 나타낼 수 있다.

$$S[(a_0 \cdots a_{n-1})] = (a_1 \cdots a_{n-1} a_0) \quad (1)$$

$$B_k[(a_0 \cdots a_{n-1})_k] = (a_0 \cdots a_{k-1} a_{n-1} a_{k+1} \cdots a_{n-2} a_k)_{k+1} \quad (2)$$

$S$ 와  $B_k$ 는 각각 서플 교환과 반얀 교환을 나타내고, (2)의 각 괄호 우측 아래 첨자  $k$ 는 해당 스테이지를 나타낸다. 그리고,  $n$ 비트의  $a_0 a_1 \cdots a_{n-1}$ 는 각 스테이지  $N/2$ 개의 SE의 0부터  $N/2-1$ 의 2진 주소 값을 나타낸다. 만일, 반얀망에 전송된 한 셀의 (입력 포트 - 목적 포트) 주소 값의 쌍을 입, 출력 주소쌍이라고 정의하고  $(s_0 \cdots s_{n-1} - d_0 \cdots d_{n-1})$ 라고 할 때,

이 셀은 식 (1)과 (2)에 따라, (3)과 같이 중간 스테이지를 거쳐 스테이지  $k$ 의 SE  $(d_0 d_1 \cdots d_{k-1} s_{k+1} \cdots s_{n-1})_k$ 의 입력 링크  $(d_0 d_1 \cdots d_{k-1} s_{k+1} \cdots s_{n-1} s_k)_k$ 로 전송된다.

$$\text{input port : } S[(s_0 \cdots s_{n-1})] = (s_1 \cdots s_{n-1} s_0)$$

$$\text{stage 0 : } B_0[(s_1 \cdots s_{n-1} d_0)_0] = (d_0 s_2 \cdots s_{n-1} s_1)_1$$

⋮

$$\text{stage } k-1 : B_{k-1}[(d_0 \cdots d_{k-2} s_k \cdots s_{n-1} d_{k-1})_{k-1}]$$

$$= (d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1} s_k)_k \quad (3)$$

또한, 이 셀은 SE  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1})_k$ 의 다른 입력 링크로부터의 셀, 즉, 입력 링크  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1} \bar{s}_k)_k$ 로 입력되는 셀과 SE의 출력 링크를 공유하며, 여기서,  $\bar{s}_k$ 는  $s_k$ 의 보수를 의미한다. 만일,  $S^{-1}$ 과  $B_k^{-1}$  각각을 (1)과 (2)의 역 변환으로 정의하고 입력 링크  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1} \bar{s}_k)_k$ 로 셀을 전송할 수 있는 입력포트 주소 값을 역으로 산출하면  $(x_0 \cdots x_{k-1} \bar{s}_k s_{k+1} \cdots s_{n-1})$ 임을 알 수 있다.  $x$ 는 don't care condition을 의미한다.

SE  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1})_k$ 를 통해 전송될 수 있는 목적 포트 역시, 같은 방법으로 계산할 수 있다.  $(s_0 \cdots s_{n-1} - d_0 \cdots d_{n-1})$ 의 주소쌍을 갖는 셀은 SE  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1})_k$ 에서  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1} d_k)_k$ 의 출력 링크로의 전송을 요구하므로, SE  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1})_k$ 의 출력 링크  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1} 0)_k$ 과  $(d_0 \cdots d_{k-1} s_{k+1} \cdots s_{n-1} 1)_k$ 의 출력 링크를 통해 전송 가능한 최종 출력 포트는  $(d_0 \cdots d_{k-1} x_{k+1} \cdots x_{n-1})_n$ 이 된다.

이상에서의 결과에서, 입력 포트에서  $(s_0 \cdots s_{n-1} - d_0 \cdots d_{n-1})$ 의 주소쌍을 갖는 셀은 스테이지  $k$ 에서  $(x_0 \cdots x_{k-1} \bar{s}_k s_{k+1} \cdots s_{n-1} - d_0 \cdots d_{k-1} x_{k+1} \cdots x_{n-1})$ 의 주소쌍을 갖는 셀들과 충돌함을 알 수 있다.

### III. 내부 충돌 제거를 위한 재할당 기법

#### 1. 충돌 셀의 분류

1장에서  $N \times N$  반얀망엔  $2^{nN/2} (n = \log_2 N)$ 개의 충돌 없는 셀들의 입, 출력 포트 주소쌍 패턴이 존재함을 언급하였고, 2장에서  $(s_0 \cdots s_{n-1} - d_0 \cdots d_{n-1})$ 의 주소쌍을 갖는 셀은 스테이지  $k$ 에서  $(x_0 \cdots x_{k-1} \bar{s}_k s_{k+1} \cdots s_{n-1} - d_0 \cdots d_{k-1} x_{k+1} \cdots x_{n-1})$ 의 주소쌍을 갖는 셀들과 충돌함을 보였다.

반얀망 스위치 입력단에 각각의 목적 포트 정보를 가지고 진입한 셀들을 충돌이 발생하지 않을 입력 포트에 재할당하기 위해선, 우선 초기 입력된 셀들 중에서 서로 충돌을 일으킬 셀들을 분류해야 한다.



2장에서와 같이,  $(s_0 \cdots s_{n-1} - d_0 \cdots d_{n-1})$ 의 주소쌍을 갖는 초기 입력 셀에 대해 충돌을 일으키는 셀들은 (4)와 같이 스테이지  $n-2$ 까지 전송된다.

$$\begin{aligned}
 \text{stage } 0 & : (\bar{s}_0 s_1 \cdots s_{n-1} - d_0 x_1 \cdots x_{n-1}) \\
 & \vdots \\
 \text{stage } k & : (x_0 \cdots x_{k-1} \bar{s}_k s_{k+1} \cdots s_{n-1} - d_0 \cdots d_k x_{k+1} \cdots x_{n-1}) \\
 & \vdots \\
 \text{stage } n-2 & : (x_0 \cdots x_{n-2} \bar{s}_{n-2} s_{n-1} - d_0 \cdots d_{n-2} x_{n-1}) \quad (4)
 \end{aligned}$$

최종 스테이지, 즉 스테이지  $n-1$ 에서의 출력 링크상의 충돌은 출력 충돌로서 동일 목적 포트를 향하는 셀들간의 충돌을 의미하므로 본 논문의 범위에선 고려하지 않는다. 이러한 스테이지별 충돌 가능한 주소쌍을 보다 상세히 알기 위해, 그림 3과 같은 주소쌍을 갖는  $8 \times 8$  반안망의 예를 들었다.

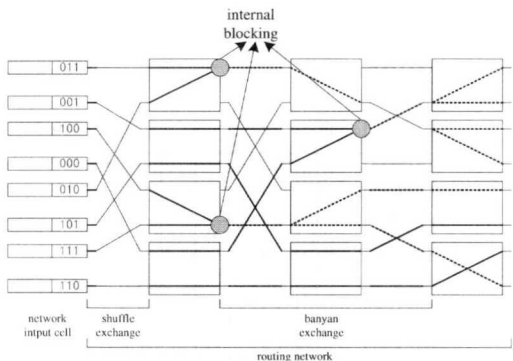


그림 3. 주소쌍이 부가된  $8 \times 8$  반안망의 예

입력 포트의 셀들 각각에 대해 충돌이 가능한 주소쌍을 알아보기 위해, 세로 축 및 가로 축이 각각 입, 출력 포트 2진 주소에 해당하는  $8 \times 8$  크기의 테이블을 마련하고 이를 충돌 확인 테이블로 정의한다. 주어진 셀에 대한 스테이지 0의 충돌이 가능한 주소쌍을 '1'로 표기하고 나머지는 '0'로 표기하며, 주어진 셀의 위치를 동그라미 'o'로 표기하면, 예제 000-011 주소쌍의 셀의 경우, 스테이지 0에서의 충돌이 가능한 입, 출력 주소쌍은 100-000, 100-001, 100-010, 100-011이므로 충돌 확인 테이블은 그림 4와 같이 나타낼 수 있다. 이는 역으로, 주어진 셀의 입, 출력 주소쌍이 000-011의 주소 값을 갖는 경우, 이 셀은 입력 포트 100에 진입한 셀이 목적 주소 값을 000, 001, 010, 011을 가질 경우 서로 충돌함을 의미하며, 100, 101, 110, 111의 목적 주소 값을 갖는 경우엔 서로 충돌하지 않음을

의미한다.

	output port address							
	000	001	010	011	100	101	110	111
000				o				
001								
010								
011								
100	1	1	1	1	0	0	0	0
101								
110								
111								

그림 4. 테이블 구성의 예

이와 같은 방법으로, 스테이지 0에 대한 그림 3의 모든 입력 셀들 각각의 충돌 가능 주소쌍을 나타내면, 그림 5와 같다.

	000	001	010	011	100	101	110	111
000	1	1	1	o	0	0	0	0
001	0	o	0	0	1	1	1	1
010	0	0	0	0	o	1	1	1
011	o	0	0	0	1	1	1	1
100	1	1	o	1	0	0	0	0
101	1	1	1	1	0	o	0	0
110	0	0	0	0	1	1	1	o
111	1	1	1	1	0	0	o	0

그림 5. 스테이지 0의 충돌 확인 테이블

그림 5에서 ①, 즉, 1과 'o'가 동시에 표시된 곳은, 임의의 셀로부터 충돌 가능한 주소쌍으로 지정된 위치에 다른 셀이 입력되어 있는 상태를 의미하는 것으로, 해당 위치에 1을 발생시킨 셀과 이 위치에 진입한 셀들간에 스테이지 0에서 서로 충돌함을 의미한다. 예를 들어 충돌 확인 테이블의 입력 포트 000, 출력 포트 011에서 나타난 ①의 경우, 1을 발생시킨 입, 출력 주소쌍 100-010의 셀은 입력 포트 000의 목적 포트 000, 001, 010, 011의 셀과 충돌을 일으키게 되나, 실제로 입력 포트 000에서 출력 포트 011로의 전송을 요구하는 셀이 진입한 관계로 서로 충돌 관계에 놓였음을 의미한다. 이는 역으로, 충돌 확인 테이블에서의 입력 포트 100, 출력 포트 010에서 나타난 ①과 같이 서로 충돌함을 확인할 수 있다.

따라서, 그림 5의 충돌 확인 테이블에선 총 네개의 셀이 서로 충돌하고 있음을 알 수 있으며, 이는 반안망 상의 서로 다른 두 곳, A와 B에서 네개의 셀이 서로 충돌함을 확인할 수 있다. 이러한 각 충돌은 2×2로 구성된 한 SE 내에 진입한 두 셀이 서로 같은 출력 링크로의 전송을 요구하면서 발생한 충돌이므로, 이 두 셀 중 어느 한 셀만을 제거하면 다른 한 셀은 충돌 없는 전송이 가능해진다. 이런 방식으로 모든 충돌에 대해, 충돌을 일으키는 두 셀 중 한 셀을 제거한 후의 충돌 확인 테이블은 그림 6과 같다. 두 셀 중 어느 셀을 제거하더라도, 망 성능엔 영향을 미치지 않으므로, 각 내부 충돌당 임의의 셀들이 제거된 것으로 가정하였다.

	000	001	010	011	100	101	110	111
000	0	0	0	⓪	0	0	0	0
001	0	⓪	0	0	1	1	1	1
010	0	0	0	0	⓪	0	0	0
011	⓪	0	0	0	1	1	1	1
100	1	1	1	1	0	0	0	0
101	1	1	1	1	0	⓪	0	0
110	0	0	0	0	1	1	1	1
111	1	1	1	1	0	0	⓪	0

그림 6. 그림 5의 충돌 셀 제거 후 충돌 확인 테이블

그림 6에선 그림 5에서 충돌을 일으켰던 주소쌍 000-011, 010-100, 100-010, 110-111의 셀들 중에서 100-010, 110-111의 입, 출력 주소쌍을 갖는 셀들이 제거된 것으로 가정하였다. 그림 5와는 달리, 충돌을 일으킨 셀들을 제거한 상태에서 재구성한 충돌 확인 테이블에선 제거된 셀들의 충돌 가능 주소쌍들이 모두 0으로 표기되며, 셀들이 진입한 것으로 표기되었던 '0'의 표시 또한 제거되었으므로, 그림 6과 같은 충돌 없는 충돌 확인 테이블이 구성되었다. 이러한 방법으로 각 스테이지별 충돌 확인 테이블 구성 및 충돌 셀 제거 작업이 이루어진다. 여기서 제거된 셀들은 추후의 입력 포트 재할당을 위한 대상 셀들이 된다. 그림 7에선 같은 방식을 적용한 스테이지 1에서의 충돌 확인 테이블을 나타낸다.

이상에서와 같이, N×N 반안망의 충돌 셀의 제거를 위해선 총 n-1개의 충돌 확인 테이블이 필요하다. 그리고, 이렇게 마련된 각 스테이지별 충돌 확인 테이블은 각 테이블 구성 과정에서 제거된 셀들의 입력 포트 재할당을 위해 사용된다.

	000	001	010	011	100	101	110	111
000	0	0	0	⓪	1	1	0	0
001	1	⓪	0	0	0	0	1	1
010	0	0	1	1	⓪	0	0	0
011	⓪	1	0	0	1	1	0	0
100	0	0	0	0	1	1	0	0
101	1	1	0	0	0	⓪	1	1
110	0	0	1	1	0	0	0	0
111	0	0	0	0	1	1	⓪	0

그림 7. 스테이지 1의 충돌 확인 테이블

## 2. 제거된 셀들을 위한 입력 포트 재할당

충돌을 발생시키는 셀로 분류되어 제거된 셀들은 자신의 목적 포트 정보는 그대로 유지한 채, 다른 비충돌 셀들과의 충돌이 발생하지 않는 새로운 입력 포트에 그 값을 재할당 받아야 한다. 제안한 방식에선 이러한 셀들의 입력 포트를 재할당 하기 위하여 입력 포트 재할당 테이블을 구성하였다.

	000	001	010	011	100	101	110	111
000	0	0	0	⓪	1	1	0	0
001	0	⓪	0	0	1	1	1	1
010	0	0	1	1	⓪	0	0	0
011	1	1	0	0	1	1	1	1
100	1	1	1	1	1	1	0	0
101	1	1	1	1	0	⓪	1	1
110	0	0	1	1	1	1	1	1
111	1	1	0	0	1	1	⓪	0

그림 8. 입력 포트 재할당 테이블

제거된 셀들에게 할당될 충돌이 없는 입력 포트는 모든 스테이지에서 충돌이 발생하지 않아야 하므로, 각 스테이지별로 발생했던 모든 충돌 확인 테이블을 ORing한 테이블에서 그 주소 값을 찾을 수 있으며, 이 테이블은 입력 포트 재할당 테이블로 정의한다. 그림 8은 스테이지 0와 1의 충돌 확인 테이블을 ORing한 입력 포트 재할당 테이블이다. 이전의 예제에서 제거된 셀들의 목적 포트는 각각 000, 010, 111이므로, 이들의 새로운 입력 포트는 입력 포트 재할당 테이블의 출력 포트 000, 010, 111의 축 선상에서 찾아야 한다. 이들 셀들에게 할당될 수 있는 입력 포트엔 이미 다른 비충돌 셀이 배정되어 있으면 안되며, 또한 이미 비충돌 셀들로 결정된 셀들과 충돌하지 말아야 하므로, 입력 포트

재할당 테이블에서 그 교점에 해당하는 값이 0값을 가져야 한다. 그림 8의 경우, 출력 포트 010으로 향하는 충돌 셀은 세로축 A선상에서 B, C, D와 같은 가로축의 빈 입력 포트에 할당될 수 있으나, 그 교점 값이 0인 위치는 오직 B이므로, 입력 포트 011만이 입력 포트에 재할당 될 수 있다.

이와 같은 방법으로, 그림 8에서의 각 할당되어야 할 셀들의 입력 포트를 목적 포트별로 찾아보면, 출력 포트 000의 경우엔 입력 포트 011, 출력 포트 010의 경우엔 입력 포트 100, 출력 포트 111의 경우엔 입력 포트 110이 할당될 수 있다.

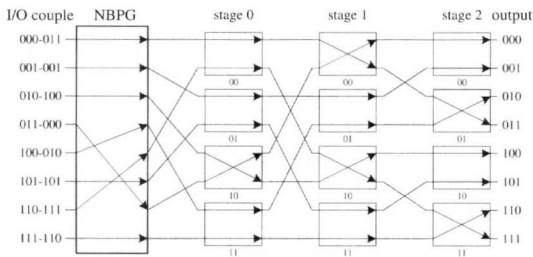


그림 9. 입력 포트 재할당 후의 셀 전송 경로

그림 9는 이와 같이 입력 포트가 재할당된 상태에서의 반얀망에서의 셀 전송 과정을 나타내었다. NBPG(Non-Blocking Permutation Generator)는 본 논문에서 제안하는 기법을 물리적으로 수행하는 가상의 소자로서 나타내었으며, 데이터 BUS를 이용한 구성으로 쉽게 초기 입력 셀들의 입력 포트 주소값을 변경시킬 수 있다.

이상의 예에서는 제거된 셀들 각각의 목적 포트에 할당될 수 있는 입력 포트가 하나씩 존재하나, N×N 반얀망의 경우, 두개 이상 다수개의 할당 가능한 입력 포트가 존재할 수 있다. 이런 경우, 입력 포트의 할당은 최소한의 할당 가능한 입력 포트 수를 갖는 셀들이 우선적으로 입력 포트를 재할당 받아야 하며, 이것은 다수개의 입력 포트가 할당 가능한 목적 포트의 셀이 한개나 소수의 입력 포트가 할당 가능한 셀의 입력 포트를 선점하는 것을 막기 위한 것이다.

IV. 실험 및 검토

제안한 반얀망에서의 내부 충돌 제거를 위한 입력 포트 재할당 기법의 유효성을 입증하기 위하여 C 언어로 구현한 NBPG 시뮬레이터를 통해 최대 스루풋을 비교하여 보았다. 입력 부하는 망이 가질

수 있는 최대의 혼잡 상태를 가정하기 위하여 1.0으로 고정하였으며, 평가를 위해 사용된 최대 스루풋은 다음과 같이 정의하였다.

$$\text{최대스루풋} = \frac{\text{반얀망출력포트에 도착한셀의수}}{\text{총발생한셀의수}} \quad (5)$$

셀의 발생은 임의의 목적 포트 발생 확률이 1/N으로 균일하도록 하였으며, 포트 당 셀의 발생 수는 10<sup>5</sup>개로 하였다.

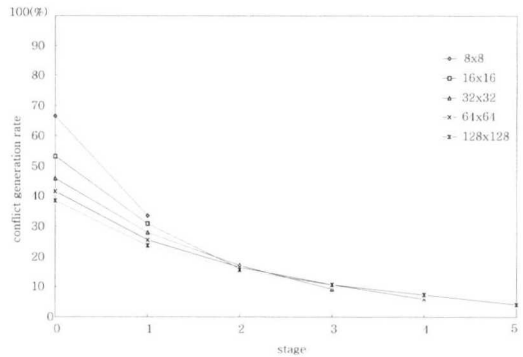


그림 10. 반얀망 스위치의 입, 출력 포트 증가에 따른 스테이지별 충돌 발생률

발생한 초기 입력 셀의 특성을 알기 위해, 반얀망 스위치의 입, 출력 포트 수를 8, 16, 32, 64, 128로 점차 늘려가며 스테이지별 충돌 발생률을 살펴보았다. 스테이지별 충돌 발생률은 총 충돌 셀에 대한 해당 스테이지의 충돌 셀의 수를 백분율로 나타낸 것으로 정의하였다. 실험 결과, 반얀망 스위치에서 충돌은 대부분 전 단계의 스테이지에서 주로 발생함을 알 수 있었다. 이것은 (s<sub>0</sub>...s<sub>n-1</sub> - d<sub>0</sub>...d<sub>n-1</sub>)의 주소쌍을 갖는 임의의 셀과 충돌하는 셀들이 (4)와 같은 주소쌍의 셀들과 충돌할 확률이 하므로 낮은 스테이지에서 높은 충돌 발생률을 갖기 때문이다.

제안한 입력 포트 재할당 기법을 적용하기 위하여, 포트 수 증가에 따른 충돌 셀 발생률을 알아보았다. 실험 결과, 그림 11에서와 같이 총 셀 발생률을 1.0으로 하였을 때, 포트 수가 8×8에서 128×128로 증가함에 따라 충돌 셀 발생률 약 0.4에서 약 0.8로 크게 증가함을 알 수 있었다.

따라서, 그림 10과 11에 따라, 반얀망은 입, 출력 포트의 수가 증가할수록 셀 손실률이 크게 증가함을 확인할 수 있었다. 그림 10의 특성을 갖는 셀들을 그림 11과 같은 충돌 셀을 발생하는 반얀망에 발생시켜, 제안한 기법을 적용한 반얀망과 일반 반얀망과의 최대 스루풋을 비교해보았다.



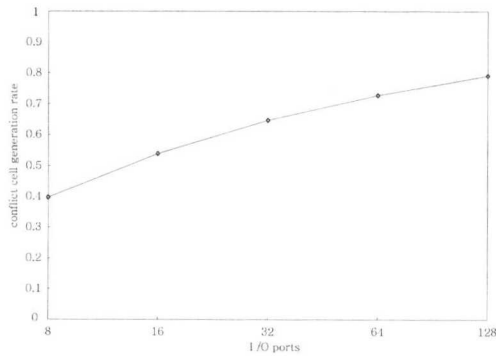


그림 11. 입, 출력 포트 수 증가에 따른 충돌 셀 발생률

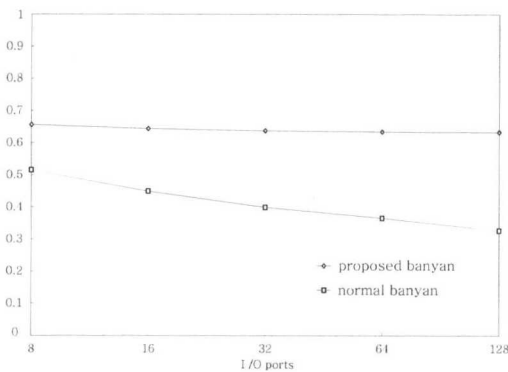


그림 12. 제안한 기법을 적용한 반얀망 대 일반 반얀망의 최대 스루풋

그림 12에서와 같이, 제안한 기법을 적용한 경우, 적용하지 않은 경우에 비해 높은 최대 스루풋을 가지며 입, 출력 포트 수가 증가함에도 일정한 최대 스루풋은 0.6과 0.7 사이의 일정한 레벨을 유지함을 알 수 있었다. 이러한 결과는 제안한 기법을 적용할 경우, 입, 출력 포트가 크게 증가하더라도 셀 손실이 일정 수준으로 제어가 되고 있음을 보여주며, 그림 11에선 포트수가 증가함에 따라 충돌 셀이 크게 증가함에도 불구하고 증가된 충돌 셀 모두가 충분히 제어되고 있음을 보여준다.

제안한 기법을 적용한 경우에서의 셀 손실에 대한 자세한 분석을 위하여 최대 스루풋과 초기 입력된 셀들의 동일 목적 포트에 진행함에 따른 출력 충돌로 인한 셀 손실률을 비교하여 보았다.

실험 결과, 제안한 방식의 최대 스루풋과 출력 충돌 셀 발생률을 비교한 결과 그림 13과 같은 값을 얻을 수 있었으며, 이들 두 값을 합한 결과 1.0에 아주 근사 하는 값을 얻을 수 있었다. 이러한 결과는 그림 12 및 그림 13에서의 최대 스루풋의 값은 출력 충돌로 인한 셀 손실에 따른 결과이며, 출력

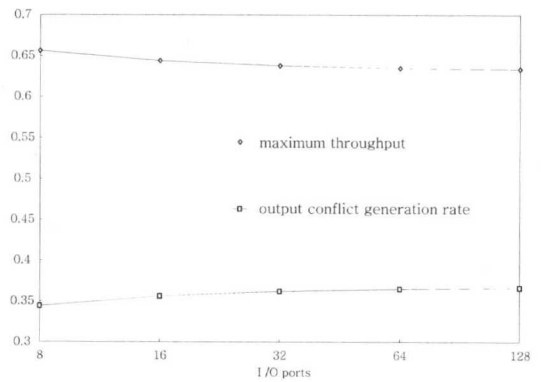


그림 13. 최대 스루풋 대 출력 충돌 발생률

충돌을 제외하고는 내부 충돌의 대부분이 제거된 결과를 확인할 수 있었다.

## V. 결론

본 논문은 반얀망에서의 각 스테이지간 SE의 링크 연결 특성을 기초로 하는 충돌 확인 테이블을 통해 초기 입력 셀에서 충돌 셀과 비충돌 셀을 분리하였고, 충돌 셀로 분류된 셀들은 각 스테이지에서 발생된 충돌 확인 테이블을 ORing한 입력 포트 재할당 테이블을 통하여 새로운 입력 포트 주소를 부여하였다. 시뮬레이션을 통한 실험 결과, 제안한 방식을 적용할 경우, 반얀망에서 발생하는 내부 충돌을 완전히 제거할 수 있었으며, 입, 출력 포트 수가 증가하더라도 그 성능이 유지되었다.

제안한 기법은 기존의 배처-반얀망으로 대변되는 비충돌 반얀망을 전혀 다른 방식으로도 구현할 수 있음을 보여주는 것으로, 기존의 배처망이 이용한 오름차순이나 내림차순의 출력 주소 패턴만이 아닌, 반얀망이 갖는 고유의 비충돌 패턴을 모두를 이용하였으며, 배처-반얀망이 갖는 복잡한 구조 대신, 보다 간단한 구조로 비충돌 반얀망을 구현할 수 있는 가능성을 제시하였다. 실제 반얀망을 이용한 ATM 스위치 구현에 있어선, 완전한 퍼뮤테이션 (permutation) 입, 출력 주소쌍을 갖는 셀들이 입력될 경우, 제안한 방식을 적용하면 내부 충돌이 완전히 제거된 손실 없는 전송이 가능하나, 출력 충돌 셀들이 공존하는 실제의 패킷 입력을 고려할 경우 셀 손실은 불가피하다. 그러나, NBPG 전단에 FIFO 구조나 개량된 구조의 입력버퍼를 적용할 경우 출력 충돌로 인한 셀 손실을 상당한 부분 제거할 수 있으므로, 그 성능이 크게 개선될 것으로 기대된다. 앞으로의 연

구 과제는 NBPG의 실제 구현 시에 문제가 될 수 있는 실시간 ATM 패킷 스위칭을 위하여 최적의 충돌 가능 셀 판별 로직 회로를 구현하는데 있다.

참 고 문 헌

- [1] Chen, Thomas M. and Stephen S. Liu, "ATM Switching Systems", *Artech House Incorporated*, Chap. 5-10, pp. 81-233, 1995
- [2] K. E. Batcher, Sorting networks and their application, "Sorting network and their application", Proc. AFIPS Spring Joint comp. Conf., pp.307-314, 1968.
- [3] James N. Giacopelli et al., Sunshine: A High-Performance Self-Routing Broadband Packet Switch Architecture, *IEEE J. Select. Areas. Commun.*, Vol. 9, No. 8, Oct., 1991.
- [4] Huang A. and Knauer S., Starlit: A Wideband Digital Switch, *GLOBECOM '84*, pp. 121-125, Dec., 1984.
- [5] Wu C. L. and Feng T. Y., On a Class of Multistage Interconnection Networks, *IEEE Trans. Comput.*, Vol. 29, pp.694-702, 1980.

이 주 영(Joo-young Lee) 정회원



1990년 : 한양대 전자공학과  
공학사  
1992년 : 한양대 전자공학과  
공학석사  
1991년 : 한양대 전자공학과  
공학박사  
2001년~2002년 : 한양대 BK21  
사업단 연구원

2002년~현재 : 서경대 전자공학과 전임강사

정 준 모(Jun-mo Jung) 정회원

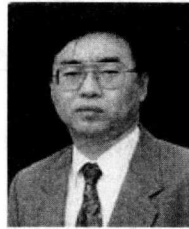


1985년 : 한양대 전자공학과  
공학사  
1987년 : 한양대 전자공학과  
공학석사  
1992년 : 한양대 전자공학과 박사  
1991년~1995년 : 부천대학  
전자계산기과 교수

1995년~1999년 : 서경대 컴퓨터과학과 교수

1999년~현재 : 서경대 전자공학과 교수

고 광 철(Kwang-cheol Ko) 정회원



1982년 : 한양대 전자공학과  
공학사  
1986년 : 日本 東京工業大學  
공학석사  
1989년 : 日本 東京工業大學  
공학박사  
1990년~현재 : 한양대 전자전기  
컴퓨터공학부 교수

정 재 일(Jae-il Jung) 정회원



1981년 : 한양대 전자공학과  
공학사  
1984년 : 한국과학기술원 전기  
및 전자공학과 공학석사  
1993년 : 프랑스 ENST 네트워크  
공학과 공학박사  
1984년~1997년 : 한국통신 통신  
망 연구소 선임 연구원

1993년 : 프랑스 국립 전기통신 연구소 연구원

1997년~현재 : 한양대 전자전기컴퓨터공학부 교수