

# 입력 확장 스위치 패브릭을 고려한 입력 버퍼링 패킷 스위치

이 현 태  
목원대학교 정보통신공학과

## An Input-Buffered Packet Switch with input expansion switch fabric

Hyeun Tae Lee  
Mokwon University  
E-mail: [hlee@mwus.mokwon.ac.kr](mailto:hlee@mwus.mokwon.ac.kr)

### 요약

본 논문은 입력 버퍼링 구조를 갖는 패킷 스위치에서 입력 확장 스위치 패브릭 구조를 통한 성능 개선에 관한 연구이다. 스위치 패브릭의 처리 능력 개선을 위한 다양한 구조에 대한 성능 및 설계 파라미터를 분석하고, 목적지별로 구분되는 입력 확장 스위치 패브릭 구조를 제안하고 버스트 트래픽 환경에서 제안된 스위치의 성능을 분석한다.

### 1. 서론

패킷 스위치는 스위치의 내부적인 충돌이나 출력 포트에서의 충돌을 피할 수 없고 이를 해결하기 위하여 버퍼링을 필요로 한다. 버퍼링 방식은 버퍼의 위치에 따라 크게 입력 버퍼링, 출력 버퍼링 방식으로 나눌 수 있다[15]. 출력 버퍼링 방식은 스위치 패브릭이 입력과 출력 링크 속도에 비해 최대  $N$ (스위치의 입력과 출력 포트의 수) 배 빠른 속도로 동작하는 구성으로서 최고의 성능을 얻을 수 있다. 그러나 이러한 스위치 패브릭의 처리 속도를 얻기 위해서는 각 출력 단의 버퍼 메모리 속도를 입력 속도의  $N$ 배로 동작시켜야 하고 한 슬롯에 최대  $N$ 개의 패킷을 동시에 출력 단으로 전송할 수 있는 스위칭 패브릭이 필요하므로 구현 비용 측면에서 효과적인 스위치 구성 방법이 아니다. 입력 버퍼링 방식은 입력 포트에서 한 시간 슬롯에 주어진 출력 포트에 하나의 패킷만 전달될 수 있으므로 동일한 출력 포트에 향하는 선택되지 않은 다른 입력 패킷은 입력 버퍼에 저장하는 방식이다. 입력 버퍼링 방식은 구현이 비교적 간단하고 다양한 스위치 패브릭 경로 구성에 적용할 수 있으며 특히 초고속의 스위치 경로 구성이 가능한 장점을 갖고 있으나 입력 버퍼에서 HOL 블러킹에 의하여 성능의 제약을 받는다. 그러나 입력 버퍼링 방식에서 입력 버퍼를 FIFO 방식이

아닌 구성을 사용하여 입력 패킷 스케줄링을 함으로 상당한 성능 개선 효과를 얻을 수 있다[1-3][7][9-10][12-14][16]. 이러한 방식을 통한 성능 개선을 위해서는 스위치에서 동시에 전달할 수 있는 패킷의 수를 최대화할 수 있도록 패킷 스케줄링이 이루어져야 한다. 그러나, 이러한 효과적인 입력 버퍼링 방식을 실현하기 위해서는 메모리 구성이나 제어 기능이 복잡해지므로 추가적인 비용 부담을 요구한다.

스위치 성능을 개선하기 위한 다른 방안으로는 스위치 패브릭의 처리 능력 향상을 통하여 스위치의 성능을 향상시킬 수 있다. 즉, 한 패킷 시간 슬롯에서 입력 포트와 출력 포트에서 처리할 수 있는 패킷의 수를 증가시켜 성능을 개선할 수 있다[4-6][8][11][19]. 그러나 스위치 패브릭의 처리 능력 개선을 위해서는 보다 큰 스위치 패브릭 구성 비용 부담을 요구한다. 지금까지의 기존 연구에서는 출력 확장 스위치 패브릭 구조에 대한 연구가 많이 있었다. 그러나 이러한 출력 확장 스위치 패브릭 구조를 구현하기 위해서는 스위치의 출력 포트 경로 자원을  $r$  배 확장하고 출력 버퍼의 쓰기 속도를  $r$  배 증가시켜야 한다. 출력 버퍼의 쓰기 속도를 증가시키기 위해서는 같은 데이터 폭에서는 쓰기 클럭을  $r$  배 증가시켜야 하며, 같은 쓰기 클럭에서는 데이터 폭을  $r$  배 증가시켜야 한다. 기존의 연구 결과에

따르면 출력 포트 확장을 통한 성능 개선 방식은 유한한 출력 버퍼에서의 패킷 손실을 증가 시키며 이를 줄이기 위해서는 보다 큰 출력 버퍼를 필요로 한다[17]. 본 연구에서는 각 스위치 패브릭 능력 개선 구조에 대한 성능 및 설계 파라미터를 비교 분석하고 성능과 비용간의 타협 설계를 통한 스위치 패브릭 구조를 제안하고자 한다.

## 2. 스위치 패브릭의 성능 개선 구조

### 2.1. 출력 포트 확장 구조

출력 포트 확장 구조는 입력 포트  $i$ 와 출력 포트  $j$ 간에  $r$ 개의 출력 포트를 갖는 경우이다. 따라서 한 패킷 슬롯 시간 동안 한 출력 포트에 최대  $r$  개의 패킷을 전달할 수 있다. 이를 위하여 한 입력 포트와 출력 포트 당  $r$  개의 크로스포인트

$$\{(i_{oe}, j_{oe}, k_{oe}) \mid 1 \leq i_{oe}, j_{oe} \leq N, k_{oe} = 1, 2, \dots, r\}$$

를 갖는다. 따라서 필요한 크로스바의 크로스포인트 수는  $rN^2$  개이다. 그림1은  $r=2$ 인 경우이다.

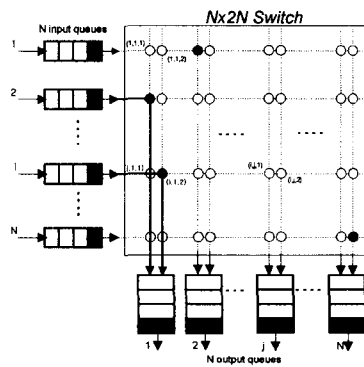


그림 1. 출력 포트 확장 스위치 패브릭 구성

그림2는 출력 확장 비율  $r$ 에 따른 출력 확장 스위치 패브릭의 최대 성능 처리율을 평균 버스트 길이  $l$ 을 변화시키면서 측정한 시뮬레이션 결과이다. 시뮬레이션은 충분히 큰  $N=128$ 인 스위치 크기에서 수행하였다. 트래픽이 랜덤하고 균일한

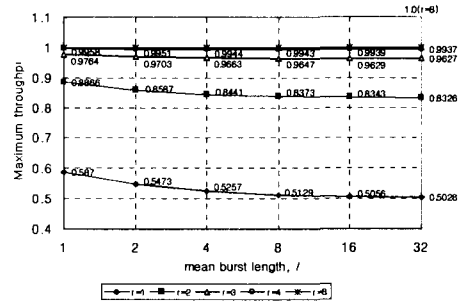


그림 2. 평균 버스트 길이( $l$ )에 대한 출력 확장 스위치의 최대 처리율( $N=128$ )

조건에서 출력 확장 비율( $r$ )이 4에서 99.6%의 최대 처리율을 얻을 수 있다. 트래픽이 버스티한 경우의  $r=1, 2, 4$ 일 경우 버스티 길이가 무한대가 되면 최대 처리율이 각각 50%, 82.8%, 99.3%으로 수렴한다. 출력 확장 스위치 패브릭 구조는  $r=4$ 인 경우 최대 처리율이 99.3%로 일반적인 원도우 방식이나 입력 확장 구조가 버스티 성이 큰 트래픽에 대해 성능이 저하되는 것에 반해  $r$ 이 충분히 큰 경우 버스티 성 트래픽에 대해서도 거의 영향을 받지 않음을 알 수 있다. 이것은 출력 확장에 의해 입력 버퍼 쪽에 패킷을 거의 남겨 두지 않아 버스티 패킷에 의한 HOL 블리킹 현상이 거의 발생하지 않기 때문이다.

출력 포트 확장 구조는 출력 포트에 출력 버퍼를 필요로 한다. 출력 버퍼는 확장된 각 출력 포트마다 별도의 버퍼를 사용하거나 하나의 버퍼를 사용하는 경우  $r$  배의 속도로 버퍼에 저장할 수 있어야 한다(이 경우는 크로스 포인트에 한 패킷 버퍼를 갖고 있는 경우이다).

출력 버퍼가 유한한 경우 출력 버퍼에서 패킷의 손실이 발생할 수 있다. Lee[17]의 연구 결과에 따르면 유한한 출력 버퍼의 경우 출력 확장 비율이 증가할수록 출력 버퍼에서의 패킷 손실이 늘어난다. 따라서 입출력 버퍼 자원, 출력 포트 확장 비율간에 규모 설계가 필요하다.

## 2.2. 입력 포트 확장 구조

입력 포트 확장 구조는 입력 포트  $i$ 와 출력 포트  $j$ 간에  $s$ 개의 입력과  $s$ 개의 크로스포인트  $\{(i_{ie}, j_{ie}, k_{ie}) \mid 1 \leq i, j \leq N, k_{ie} = 1, 2, \dots, s\}$ 를 갖는 경우이다. 필요한 크로스바의 크로스포인트 수는  $sN^2$  개가 필요하다. 그림3은  $s=2$ 일 경우이다.

입력 포트 확장 구조는 넓게 정의하면 각 입력 큐에서 서로 다른 목적 포트 주소를 가진 최대  $s$ 개의 패킷까지 한 패킷 시간 슬롯에서 처리될 수 있는 스위치 패브릭 구조로 정의할 수 있다. 그런데 이 경우  $s$ 보다 많은 패킷이 경합에 참여하고 이 중에 최대  $s$ 개가 선택되는 경우를 포함하게 된다. 본 장에서는 스위치 패브릭 처리 능력에 관심이 있고 다양한 스위치 패브릭 처리 능력 개선 구조간의 상대적인 비교를 하고자 한다. 따라서 본 연구에서는 입력 포트 확장 구조의 정의를  $s$ 개의 패킷이 출력 포트 경합에 참여하고 입력 큐에서 최대  $s$ 개까지의 패킷을 한 시간 슬롯에 처리할 수 있는 스위치 구조로 제한하여 정의한다.

이러한 입력 확장 스위치 패브릭 구조에서 입력 버퍼의 구성은 다음과 같은 두가지 형태의 구성이 가능하다.

첫째는 그림4(a)와 같이 입력 버퍼가 FIRO 방식으로 처리되고 윈도우( $w$ )와 입력 확장 비율( $s$ )이 같은 구성으로 입력 패킷은 도착한 순서대로 저장되고 HOL에서  $s$ 개의 패킷이 출력 포트 경합에 참여할 수 있고 최대  $s$ 개의 패킷이 한 패킷 시간 슬롯에서 처리될 수 있다. 처리된 패킷은 큐에서 제거되고 다음 시간 슬롯에는 처리된 패킷이 제거된 큐에 대하여 같은 방식으로 경합에 참여한다.

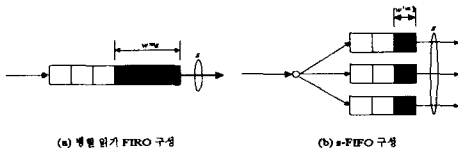


그림4 입력 확장 구조에서의 입력 버퍼 구성 방식

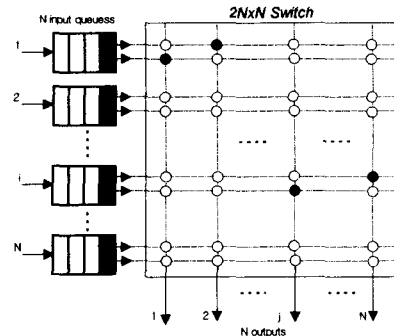


그림3 입력 포트 확장 스위치 패브릭 구성

둘째는 그림4(b)와 같이 입력 버퍼가  $s$ 개의 독립적인 큐로 구성하는 경우이다. 이 경우 각 큐의 HOL 패킷이 매 시간 슬롯마다 경합에 참여한다. 이때 입력되는 패킷을  $s$ 개의 입력 큐에 저장하는 방식으로 다음과 같은 여러 가지 방식을 고려할 수 있다.

- 랜덤 큐잉(Random Queuing:RQ)

입력되는 패킷을 랜덤하게 입력 버퍼를 선택하여 저장한다. 이 경우 스위치에서 패킷 순서를 보장할 수 없다.

- 목적지별 큐잉(Destination Queuing:DQ)

입력 패킷의 목적 주소에 따라 입력 버퍼를 구분하여 저장한다. 목적지 별로 패킷을 저장하므로 스위치에서 패킷의 순서를 보장할 수 있다.

- 큐 길이에 따른 큐잉(Short Queue Queuing: SQ)

현재 입력 버퍼에 저장된 패킷의 길이가 가장 작은 버퍼를 선택하여 저장한다. 이 경우 스위치에서 패킷 순서를 보장할 수 없다.

그림5는 입력 확장 비율  $s$ 에 따른 입력 확장 스위치 패브릭의 최대 성능 처리율을 입력 트래픽의 평균 버스트 길이를 변화시키면서 측정된 결과이다. 시뮬레이션은  $N=128$ 인 스위치 크기에서 수행하였고 입력 버퍼는 병렬입기 FIRO를 사용하고 패킷 선택 방식은 일반적인 윈도우 방식(H-K 윈도우 방식)으로

Maximum throughput

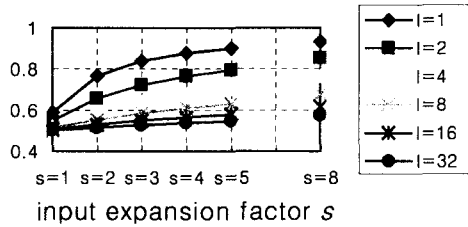


그림5 입력 확장 비율( $s$ )에 대한 입력 확장 스위치 패브릭의 최대 처리율 ( $N=128$ )

선택하며 입력 패킷의 선택은 최대  $s$  개 까지 선택할 수 있는 방식으로 수행하였다.

그림5에서와 같이 입력 확장 구조에서 입력 확장 비율  $s$ 를 증가시키면 최대 처리율이 증가한다. 그러나 트래픽의 버스트성이 클수록 입력 확장에 의한 최대 처리율의 증가율이 감소한다. 이것은 FIRO를 사용하는 입력 확장 스위치 패브릭의 경우 입력 확장 비율  $s$ 가 출력 포트 경합에 참여하는 수와 동일한 경우이므로 평균 버스트 길이  $l$ 이 증가할수록 윈도우 내에 경합에 참여할 수 있는 서로 다른 출력 포트 주소를 가진 패킷의 수가 줄어들어 입력 확장 효과가 줄어들기 때문이다. FIRO 입력 버퍼 구조의 입력 확장 스위치는 평균 버스트 길이가 증가하면 일반적인 FIFO 입력 큐 스위치 구조가 갖는 버스트 트래픽 환경에서의 최대 처리율인 0.5로 수렴한다. 즉, 출력 확장의 경우 확장 비율  $r=4$ 에서 버스트 길이에 관계 없이 최대 성능에 가까운 처리율을 얻을 수 있었으나, 입력 확장 스위치 패브릭의 경우 확장 비율에 따른 처리율의 증가가 출력 확장의 경우보다 완만하고 버스트 성 트래픽의 영향을 많이 받는 것을 보여 준다.

### 2.3. 목적지별 입력 포트 확장 스위치

입력 포트 확장이나 출력 포트 확장 스위치 패브릭 구조의 경우 높은 성능을 얻기 위해서는 구현 측면에서 부담이 되는 큰  $r$ 과  $s$  값을 가져야 한다. 특히 입력 확장 스위치 패브릭의 경우 버스트성이 큰

트래픽에 대해서는 더욱 큰 확장 비율을 가져야 한다. 입출력 확장 구성에서는 스위치의 규모가  $sN \times rN$  규모를 가지므로 역시 구현 측면에서 부담이 크다.

본 논문에서는 스위치 패브릭이 출력 포트 주소에 따라 논리적으로 별도의 평면으로 구성되는 입력 확장 스위치 패브릭 구조를 제안한다.

그림6의 예는 입력 버퍼를 짝수의 목적 포트로 향하는 패킷을 저장하는 Even 입력 큐(EQ)와 홀수의 목적 포트 패킷을 저장하는 Odd 입력 큐(OQ)로 구성된 경우를 예시한다. Even 입력 큐는 Even 스위치 패브릭의 입력 포트에 연결되고 Odd 입력 큐는 Odd 스위치 패브릭의 입력 포트에 연결된다. 그리고, 각 스위치 패브릭의 출력 포트는  $r$ 의 비율로 확장된 출력 포트를 갖는다. 출력 확장이 없는 경우는  $r=1$ 인 경우이다. 제안된 스위치 패브릭은  $sN \times rN$  규모를 가진다. 본 연구에서는 출력 확장 규모를  $r=1$ 인 Even/Odd 입력 확장 구조 경우와  $r=2$ 인 Even/Odd 입출력 확장 구조를 고려한다. 그림6에서 볼 때, 입력 포트  $i$ 와 출력 포트  $j$  간에는 각 2개의 입력 포트와  $r$ 개의 출력 포트를 갖고 있다. 여기서 입력 포트  $i$ 와 출력 포트  $j$  간에 2개의 크로스포인트  $\{(i_{deo}, j_{deo}, k_{deo}) \mid 1 \leq i_{deo}, j_{deo} \leq N, k_{deo} = 1, 2\}$  가 있으며 출력 포트가 짝수 번호 포트인  $\{(i_{deo}, j_{deo}, k_{deo}) \mid i_{deo} = 0, 2, 4, \dots, N\}$  크로스포인트는 Even

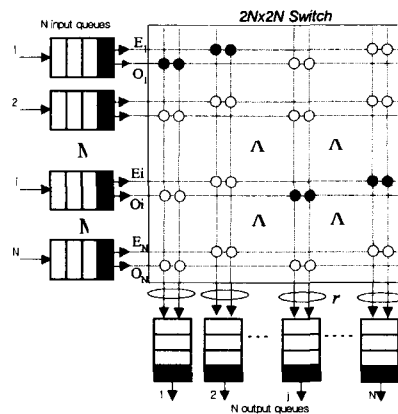


그림6 Even-Odd 입출력 확장 스위치 구성

스위치 패브릭의 입력 포트  $E_i$  상에 존재하고 홀수 번호 포트인  $\{(i_{deo}, j_{deo}, k_{deo}) | i_{deo} = 1, 3, 5, \dots, N-1\}$ 는 Odd 스위치 패브릭의 입력 포트  $O_i$  상에 존재한다. 이 경우 총 필요한 크로스포인트 수는  $2(N \times r(N/2)) = rN^2$ 이다. 즉, 목적지별 입력 확장 스위치 패브릭의 경우 필요한 총 크로스포인트의 수는  $rN^2$ 이고 순수한 입력 확장만 하는 구조( $r=1$ )는 크로스포인트 수가  $N^2$ 이다. 이는 출력 확장 스위치 패브릭의 경우  $rN^2$ 인 경우와 비교할 때 보다 경제적인 패브릭 구성이 가능함을 보여준다.

#### 2.4. FIRO와 S-FIFO

제안된 Even/Odd 입력 확장 스위치 구조가 갖는 목적지별 큐잉의 성능 효과를 분석하기 위하여 FIRO 입력 버퍼 구성과 s-FIFO 입력 버퍼 구성을 갖는 각 입력 확장 스위치의 성능을 시뮬레이션으로 측정하였다. FIRO 입력 버퍼를 갖는 입력 확장 스위치(이하,  $FIRO(w=s)$  스위치로 구분하여 명명한다.)는  $s$ 개의 윈도우를 갖는 FIRO 입력 버퍼( $FIRO(w=s)$ )와 입력 확장 비율이  $s$ 인 입력 확장 스위치 패브릭( $sN \times N$ )으로 구성된다. s-FIFO 입력 버퍼를 갖는 입력 확장 스위치(이하, s-FIFO-s 스위치로 명명한다.) FIFO의 앞 첨자는 FIFO의 갯수를 나타내고 뒤 첨자는 스위치 패브릭의 입력 확장 비율을 나타낸다.)는 입력 확장 비율  $s$ 와 같은 수의 목적지별 FIFO(s-FIFO)와 목적지별 입력 확장 스위치 패브릭( $s(N \times N/s)$ )으로 구성된다. 시뮬레이션은 각 구성 방식에 대하여 입력 확장 비율과 평균 버스트 길이에 따르는 최대 처리율을  $N=32$ 의 스위치 크기에서 측정하였다.

그림7은  $FIRO(w=s)$  스위치와 s-FIFO-s 스위치의 평균 버스트 길이에 대한 최대 처리율을 나타낸다. 측정 결과, 트래픽이 랜덤( $l=1$ )한 경우 두 스위치 구성의 최대 처리율이 거의 같게 측정된다. 이것은 랜덤 트래픽에서는 각 스위치 구조가 갖는 서로의 장단점이 상쇄되기 때문이다. 즉, 같은 목적 주소 패킷이 연속적으로 나타날 경우는 s-FIFO-s 스위치가

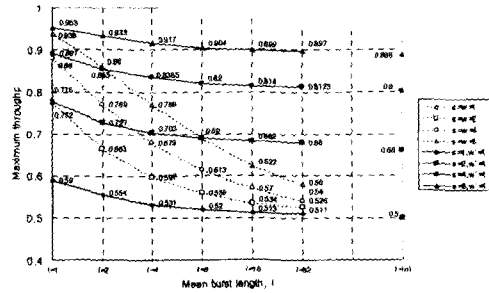


그림7 평균 버스트 길이에 대한 FIRO( $w=s$ ) 스위치와 s-FIFO-s 스위치의 최대 처리율( $N=32$ )

유리하지만 서로 다른 목적 주소의 경우  $FIRO(w=s)$  스위치는 모두 처리할 수 있으나  $FIRO(w=s)$  스위치 경우에는 같은 목적 주소 그룹의 패킷인 경우 블리킹이 되는 경우가 발생한다. 평균 버스트 길이가 커질 수록  $FIRO(w=s)$  스위치와 s-FIFO-s 스위치의 성능이 저하한다.  $FIRO(w=s)$  스위치의 경우는 평균 버스트 길이가 커질 수록 최대 처리율이 0.5에 수렴한다. 이 수렴 값은 일반적인 FIFO 입력 버퍼 스위치에서  $l = \infty$ 에서의 최대 처리율이다[18]. 이와 같이 s-FIFO-s 스위치의 경우 버스트의 길이가 커질 수록 성능이 저하되고 입력 확장 효과가 줄어드는 것은 동시에 경합에 참여할 수 있는 패킷의 수가 윈도우 내의 버스트가 증가할 수록 줄어들기 때문이다. 그런데 s-FIFO-s 스위치 경우는 트래픽의 버스트성이 커지면 실험 결과와 같이 최대 처리율이 입력 확장 비율( $s=2, 4, 8$ )에 따라 각각 0.66, 0.75, 0.88값에 수렴한다. 이것은 목적 주소별로 버퍼를 구별하므로 버퍼의 크기가 충분히 클 경우 버스트가 존재하여도 서로 다른 목적 포트 그룹 주소를 갖는 패킷은 경합에 참여할 수 있기 때문에 얻을 수 있는 최대 처리율이다.

### 3. 결론

본 논문은 입력 버퍼링 패킷 스위치에서 스위치의 성능 개선을 위한 스위치 패브릭의 구조에 대한 연구를 수행하였다.

순수한 출력 확장, 입력 확장을 통한 스위치 성능 개선은 구현 측면에서 부담이 크므로 비용과 성능간의 타협 설계를 통하여 스위치 패브릭이 출력 포트의 주소에 따라 별도의 평면으로 구성되는 입력 확장 스위치 패브릭 구조를 제안하였다. 제안된 스위치 패브릭 구조는 입력 확장 비율과 관계 없는 스위치 패브릭의 크로스포인트 복잡도를 가지며, 목적지별로 입력 버퍼를 구성하는 스위치 구조이므로 버스트성 트래픽에 대한 성능 개선 효과를 얻을 수 있었다.

#### 참고 문헌

- [1] T. E. Anderson, S.S. Owicki, J. B. Saxe and C. Cp. Thacker, "High Speed Switch Scheduling for Local Area Networks," ACM Transaction on Computer Systems, Vol.11, No.4, pp.319-152, Nov. 1993.
- [2] M.K.M. Ali and M. Youssefi, "The Performance of an Input Access Scheme in a High-speed Packet Switch," IEEE INFOCOM'91, pp.454-461, 1991.
- [3] Christos Kolias and Leonard Kleinrock, "Throughput Analysis of Multiple Input-Queueing in ATM Switches," Broadband Communications 96, pp.382-393, Canada, 1996.
- [4] J. S. Choi, "A Nonblocking ATM Switch with a Single Plane or Multiple Planes Combined with a Window Scheme," ICC'96, pp.1680-1684, 1996.
- [6] Chugo Fujihashi, Hideki Hikita, "Speed-Up of Input Buffer Asynchronous Transfer Mode Switch by Introducing of Parallel Read-Out Structure," GLOBECOM'96, pp.819-824, 1996.
- [7] M. G. Hluchyj and M. J. Karol, "Queueing in High-performance Packet Switching," IEEE HSAC, Vol 6., pp.1587-1597, Dec. 1988.
- [8] Y. C. Jung, C. K. Un, "Performance analysis of packet switches with input and output buffers," Computer Networks and ISDN Systems 26, pp.1559-1580, 1994.
- [9] R. O. Lemaire, D. N. Serpanos, "Two-dimensional Round-robin Schedulers for Packet Switches with Multiple Input queues," IEEE/ACM Trans. Networking, Vol.2, No.5, pp.471-482, 1994.
- [10] N. McKeown, "Achieving 100% Throughput in an Input-Queued Switch," INFOCOM'96, pp.296-302, 1996.
- [11] Y. Oie, M. Murata, K. Kubota and H. Miyahara, "Effect of Speedup in Nonblocking Packet Switch," ICC'89, pp.410-414, June 1989.
- [12] E. D. Re and R. Fantacci, "Performance Evaluation of Input and Output Queueing Techniques in ATM Switching Systems," IEEE Trans. On Communications, Vol.41, No.10, pp.1565-1575, Oct. 1993.
- [13] D. Stiliadis and A. Varma, "Providing Bandwidth Guarantees in an Input-Buffered Crossbar Switch," INFOCOM'95, pp.960-968, 1995.
- [14] V. Yau and K. Pawlikowski, "A Conflict-free Traffic Assignment Algorithm," INFOCOM'96, pp.1277-1284, 1996.
- [15] Y. Awdeh and H. T. Mouftah, "Survey of ATM Switch Architecture," Computer Networks and ISDN Systems, Vol.27, pp.1567-1613, 1995.
- [16] G. Thomas, "On High Speed Packet Switches with Windowed Input Buffers," IEEE GLOBECOM'93, pp.1406-1410, 1993.
- [17] M. J. Lee, "Cell Loss Analysis and Design Trade-Offs of Nonblocking ATM Switches with Nonuniform Traffic," IEEE Transac. On Networking, Vol.3, No.2, 1995.
- [18] S. C. Liew, K. W. Lu, "Comparison of Buffering Strategies for Asymmetric Packet Switch Modules," IEEEJSAC, Vol.9, No.3, 1991.
- [19] K. Genda and Naoaki Yamadaka, "TORUS-switch: Scalable Tb/s ATM Switch Architecture based on the Internal Speed-up ATM Switch," GLOBECOM'95, pp.1738-1745, 1995.