

# 공유 버스와 공유 메모리 스위치를 이용한 멀티캐스트 ATM 스위치 구조

정회원 강행익\*, 박영근\*\*

## A Multicast ATM Switch Architecture using Shared Bus and Shared Memory Switch

Haeng-ik Kang\*, Young-keun Park\*\* *Regular Members*

### 요약

멀티미디어 서비스의 증가에 따라 멀티캐스팅(Multicasting)은 ATM 스위치 디자인에 있어 중요성을 더해가고 있다. 기존 다단 연결 구조에서 멀티캐스트에 의한 트래픽 팽창의 문제를 해결하기 위해 본 논문에서는 고속의 버스와 공유 메모리 스위치를 이용한 멀티캐스트 스위치를 제안한다. 고속의 시분할 버스를 연결 매체로 사용하며 공유 메모리 스위치를 단위 모듈로 하는 구조를 채택하여 용이한 포트 확장성을 제공한다. 트래픽 중재 기법을 사용하여 내부 블러킹을 없애며, 시뮬레이션을 통해 데이터 처리율이나 셀 지연 측면에서의 스위치 성능을 확인한다.

### ABSTRACT

Due to the increase of multimedia services, multicasting is considered as important design factor for ATM switch. To resolve the traffic expansion problem that is occurred by multicast in multistage interconnection networks, this paper proposes the multicast switch using a high-speed bus and a shared memory switch. Since the proposed switch uses a high-speed time division bus as a connection medium and chooses a shared memory switch as a basic switch module, it provides good port scalability. The traffic arbitration scheme enables internal non-blocking. By simulation we prove a good performance in the data throughput and the cell delay.

### I. 서론

광파이버를 이용한 기술진전을 기반으로 광대역 종합 정보 통신망은 미래의 초고속 정보화 사회를 실현시킬 수 있는 통신 플랫폼으로 받아들여지고 있다. 광대역 종합정보 통신망 실현의 핵심인 ATM은 다양한 멀티미디어 서비스의 제공, 동적인 대역폭 할당, 통합화된 정보 전달 기능을 수행할 수 있는 광대역 전송 및 스위칭 기술이다<sup>[1]</sup>.

최근 VLSI 기술의 비약적인 발달로 ATM 스위치를 구현하는 데 있어 새로운 디자인 기술이 사용되고 있는 데, 디자인 방법상의 구현 과정은 다르지

만 기본 목표는 대용량의 데이터 처리능력을 가지며 멀티미디어 서비스에 큰 영향을 미치는 데이터 전송 지연(Delay)을 최소화 할 수 있는 스위치를 구현하는 것이다. 네트워크 사용자의 수가 급격히 늘어나면서, 네트워크 노드의 증가에 효율적으로 대처할 수 있도록 대용량의 스위치로 크기 증가가 용이한 포트 확장성(Port Scalability)을 갖는 ATM 스위치 개발이 중요한 요소로 대두되고 있다. 포트 확장성이란 ATM 스위치가 갖고 있는 처리 능력을 넘어서는 트래픽 부하요구에 대해 스위치 기본 구조의 변형 없이 효율적으로 스위치의 크기를 확장시킬 수 있는 능력을 말한다. 대용량 ATM 스위치

\* LG정밀 S&C 그룹(hikang@lgp.co.kr),

\*\* 연세대학교 기계전자공학부(ypark@bubble.yonsei.ac.kr)

논문번호: 98402-0910, 접수일자: 1998년 9월 10일

※ 본 논문은 정보통신부 정보통신분야 특성화 추진사업 제 2차년도 연구과제 연구비에 의해 연구되었습니다.

를 설계하는 기본적인 접근 방법은 소용량 ATM 스위치 모듈을 단위 블록으로 하여 Multistage Interconnection Networks(MINs)를 사용하여 연결하는 것이다. 그러나, 이러한 연결 구조에서의 문제는 각 스위치 모듈간의 링크에서 내부 블러킹에 의한 성능저하가 발생한다는 것이다. 내부 링크의 속도를 높이고, 모듈간의 연결 링크를 늘임으로써 성능의 향상을 취할 수 있을 지라도, 대용량 ATM 스위치로의 확장과 기대치의 처리능력을 얻는 데는 한계가 있게 된다. 따라서, 다단 연결 네트워크 접근과는 다른 구조의 ATM 스위치를 고려하고자 한다.

멀티미디어 서비스의 폭발적 증가와 함께 ATM 스위치 디자인에 있어 고려해야 할 부분으로서 멀티캐스팅(Multicasting)이 큰 비중을 차지해가고 있다. 기존의 서비스가 전화와 같은 음성서비스, 컴퓨터 네트워크의 데이터 서비스와 같은 사용자에게 일대일 통신 즉, 점-대-점 서비스였으나, 현재는 VOD, 비디오 화상회의, 분산 데이터 처리 등과 같이 점-대-다중점 서비스가 계속 증가 추세에 있다<sup>[2][3]</sup>. 이러한 서비스들을 지원하기 위해서는 멀티캐스팅 기능이 필수적인 것이다. 이러한 멀티캐스팅을 가능케 하기 위해서는 각 스위칭 노드가 되는 ATM 스위치에서 단일 VC로 입력되는 멀티캐스트 셀을 그 셀에 의해 요구되는 복사 셀의 수인 Fanout 만큼 복사하여 다수의 VC로 전송시킬 수 있는 멀티캐스트 ATM 스위치를 필요로 하게 된다. 그러나, 멀티캐스트 트래픽은 그 특성상 스위치 내에서 셀의 복사에 의해 과도한 트래픽 팽창을 야기하게 된다. 이 결과는 점-대-점 연결 서비스의 데이터인 유니캐스트(Unicast) 셀과 멀티캐스트 셀의 스위치 내부 블러킹에 의한 셀 손실로 이어짐으로써 스위치 성능의 저하를 가져오게 된다. 특히, 다단 연결 구조에서는 연결 구조가 갖는 특성상 이 멀티캐스트 트래픽은 내부 블러킹을 더욱 가중시키는 결과를 가져오게 된다<sup>[4]</sup>.

본 논문에서는 대용량 멀티캐스트 ATM 스위치를 구현하기 위해 효율적인 포트 확장성을 제공하는 단위 모듈 연결 방식과 멀티캐스팅 기법을 제안한다. 다단 연결 구조가 가지는 단점을 극복하기 위해 공유 매체인, 고속의 시분할 버스(Time Division Bus)를 사용하며, 내부 블러킹을 갖지 않는 공유 버퍼 메모리 스위치(Shared Buffer Memory Switch)<sup>[5][6]</sup>를 단위 모듈 스위치로 하는 모듈형 구조를 사용하여 대용량 스위치로의 확장성을 가지는 ATM

스위치를 제안한다. 버스는 매체의 기본 특성인 브로드캐스팅 성질을 이용하여 쉽게 멀티캐스팅 기능을 구현 할 수 있으며, 효과적인 트래픽 중재(Arbitration) 기법을 이용하여 내부 블러킹이 없으면서 멀티캐스트 트래픽에 대해서도 효율적으로 대처할 수 있는 멀티캐스트 ATM 스위치를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 고속의 시분할 버스를 이용한 멀티캐스트 스위치 구조에 대해 제안하고, 3장에서는 단위 모듈 스위치로써 사용된 공유 버퍼 메모리 스위치에서의 멀티캐스팅에 대해 살펴 본다. 4장에서는 제안한 스위치의 성능에 대한 시뮬레이션 결과와 고찰을 다루며 5장에서 결론을 기술한다.

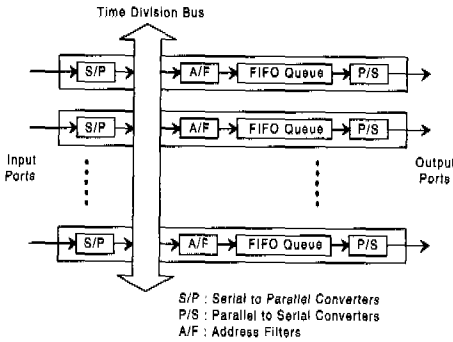
## II. 버스를 이용한 멀티캐스트 ATM 스위치 아키텍처

다단 연결 구조를 이용한 스위치 디자인은 다단 연결 구조가 갖는 스위치 비선형적 스위치 복잡도 특성과 내부경로에서의 셀 블러킹 발생으로 멀티캐스트 특성이 더해졌을 때 스위치 내부에서의 트래픽 폭주에 의한 셀 손실을 증가와 같은 제한요소가 발생하게 된다.

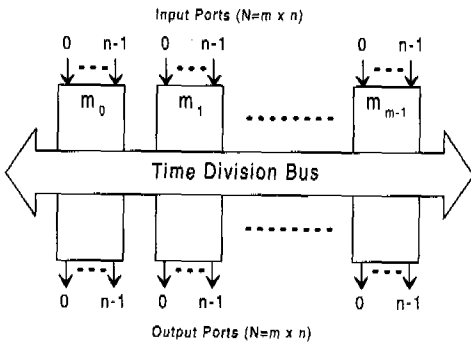
다단 연결 구조를 이용한 연결 방식 외에 버스(Bus)나 링(Ring)과 같은 공유 매체를 이용한 연결 구조를 생각해 볼 수 있다<sup>[7][8]</sup>. 이 구조에서는 모든 포트가 공유 매체를 통해 완전한 스위칭을 이룰 수 있으므로, 다단 구조의 단점을 극복할 수 있는 대안으로 볼 수 있다. 본 논문에서는 공유 버스(Shared Bus)를 이용한 모듈형 구조에 대해 제안한다.

이러한 구조에서 각 모듈의 수를 크게 늘이면 쉽게 대용량 스위치의 구현이 가능하나 버스 구조에는 하드웨어 구현에 있어 큰 제한요소를 갖고 있다. 버스 속도가 제한되어 있기 때문에 모듈의 수를 무한정 늘일 수가 없는 것이다. [9]에서는 대용량 스위치를 구현하기 위해 ATOM 스위치를 단위 모듈로 해서 3단 연결 방식을 이용한 다단 연결 구조를 선택했다. 본 논문에서는 ATOM 스위치의 기본 구조를 골격으로 하고 버스의 속도가 제한되어 있는 문제를 해결하면서 스위치의 크기를 확장하는 방법을 제안해 보았다. 그리고 버스구조는 자체 성질로서 브로드캐스팅 특성을 가지고 있어서 쉽게 멀티캐스팅을 제공할 수 있는 점을 이용하여 버스 구조를 통한 스위치 확장과 멀티캐스팅을 제공하는 스위치의 구조를 본 논문에서 제안하고 있다.

2.1 공유 버스를 이용한 모듈 스위치 연결



(a) ATOM 스위치



(b) 제한된 스위치

그림 1. 공유 버스를 이용한 제한된 ATM 스위치 구조

버스 속도의 제한에 따른 스위치 확장이 문제가 되기는 하지만 버스 구조의 연결은 매우 매력적인 방법임에 틀림없다. 버스 속도의 제한을 극복할 수 있는 방법을 제시한다면 대용량 스위치로의 확장을 기대할 수 있는 것이다. 그림 1(a)의 ATOM 스위치와 달리 각 모듈 스위치의 입/출력포트 수를 하나 이상으로 구성해서 버스를 통해 연결한다면 스위치 크기의 증가는 쉽게 이를 수 있게 된다. ATOM 스위치는 기본적으로 출력 버퍼 방식을 채택하고 있으며, 모듈형 스위치 디자인의 주요 원칙이 되는 확장성과 모듈성을 효율적으로 제공하는 구조를 갖고 있다. 그림 1(b)는 본 논문에서 제안한 멀티캐스트 ATM 스위치의 기본 블록 다이어그램이다. ATOM 스위치의 기본 구조를 변경하여 단위 모듈 스위치의 포트 수를  $n$ 으로 늘리고,  $m$ 개의 단위 모듈을 고속의 시분할 버스를 통해 연결함으로써  $m \times n$  크기의 스위치를 구성시키고 있는 것을 보여주고 있다. 스위치의 크기를 확장하면 물론 요구되는 버스의 속도 또한 증가하여야 한다. 이 문제에 대한

해결은 입력포트에서 들어오는 버스의 트래픽 부하를 각 모듈로 분산시킴으로서 크게 줄일 수 있기 때문에 버스의 속도를 크게 줄일 수 있게 된다. 제안된 스위치구조를 이해하기 위해 2.2절의 버스 중재 기법을 통해 전체적으로 스위치의 기본 동작을 살펴본 후 2.3, 2.4, 2.5절에서 스위치 구조와 동작 및 멀티캐스팅에 대해 살펴본다.

2.2 버스 중재 기법

공유 버스를 이용한 스위치 구조에서 버스의 속도는 전체 스위치 크기를 결정하는 가장 큰 요인이 된다. 따라서, 버스의 속도 제한을 극복할 수 있는 방법이 제시되어야 한다. 본 논문에서는 라운드-로빈(Round-Robin) 방식에 그 기본을 두고 있는 버스 중재(Bus Arbitration) 기법을 제안했다. 이 방법은 각각의 스위치 모듈의 동작 순서를 차례로 시분할된 시간마다 독점적으로 버스를 사용할 수 있도록 정의해줌으로써, 스위치 크기 증가에 따라 요구되는 버스 속도의 증가 문제를 해결할 수 있다. 그림 2는 타이밍 다이어그램을 보여주고 있다.

하나의 ATM 셀이 스위칭 되는데 필요한 시간을 타임슬롯  $T_s$ 라 하자. 이 한 타임슬롯을 스위치 모듈의 개수인  $m$ 으로 나누면,  $t_0, t_1, \dots, t_{m-1}$ 의 일련의 타이밍 시퀀스를 만들 수 있게 된다. 각 스위치 모듈은 전송 모드(Send)와 수신 모드(Receive)라는, 두 가지 모드로 동작하게 된다. 그림 2에서 보는 바와 같이 하나의 미니(Mini) 슬롯동안 단지 하나의 스위치 모듈만이 전송 모드 상태가 되며, 나머지  $m-1$ 의 스위치 모듈은 수신 모드 상태가 된다. 각 스위치 모듈은 라운드-로빈 방식에 기초하여 차례로 전송 모드 상태로 들어가게 된다.

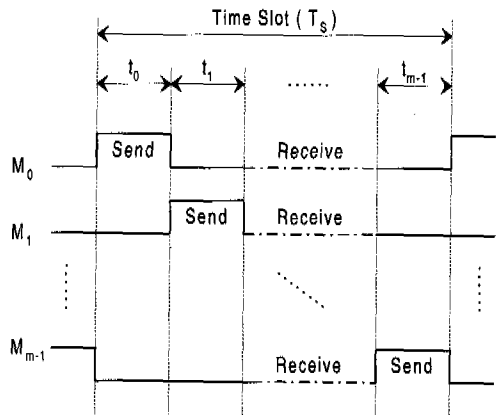


그림 2. 버스 중재를 위한 모듈 동작의 타이밍 다이어그램

따라서, 모듈간의 버스 사용 경쟁이 발생하지 않게 된다. 한 타임 슬롯 동안, 각 모듈은 하나의 미니 슬롯동안 입력포트로부터 혹은  $m-1$  미니 슬롯동안 버스를 통해 다른 스위치 모듈로부터 ATM 셀을 입력받게 된다.

이러한 중재 기법을 위해 버스는 지분할 동작을 필요로 하게 된다. 이 경우 버스의 지분할 동작은 버스에 의해 연결된 스위치 모듈의 스위칭 동작과는 완전히 독립적으로 이루어 지게 된다. 따라서, 제안된 스위치에서는 내부 속도 증가 인자(Speed up factor :  $F$ )는 다음과 같이 된다.

$$F = \max(m, n) \quad (1)$$

반면에, 출력 버퍼 스위치인 경우 제안된 스위치의 크기와 비교했을 때,  $N(=m \times n)$ 의 내부 속도 증가를 필요로 하게 된다. 따라서, 크기  $N$ 의 출력 버퍼 스위치보다 제안된 스위치는 내부 속도를 줄일 수 있게 된다.

### 2.3 스위치 모듈과 버스의 연결

공유 버퍼 메모리 스위치 구조인 스위치 모듈과 모듈간의 셀 전송의 역할을 하는 버스사이의 연결은 버스 인터페이스를 통해 이루어지게 된다. 그림 3은 스위치 모듈(포트 수가 4인 경우)과 버스를 연결하는 구조를 보여 주고 있다. 스위치 모듈과 버스를 연결하기 위한 부분은 버스 인터페이스 블록이 담당하게 된다. 입력포트로부터 들어오는 셀은 현재의 모듈로 향하게 될지, 아니면 다른 모듈로 향하기 위해 버스로 보내지게 될지를 이 버스 인터페이스가 판단하게 된다. 또는, 버스로부터(즉, 다른 모듈로부터 보내온) 입력되는 셀에 대해 자신의 모듈로 향하고 있는 지에 대한 판단을 통해 필터링하는 역할을 하게 된다. 멀티캐스트 셀인 경우의 자세한 구조는 2.4절에서 소개된다.

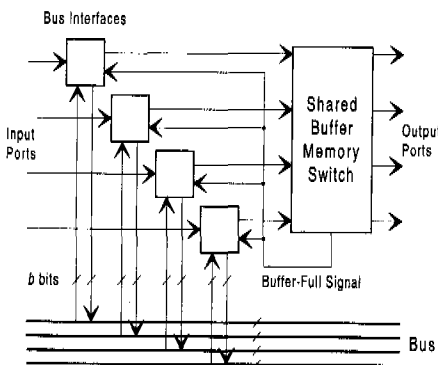


그림 3. 스위치 모듈과 버스의 연결 구조 (n=4일 경우)

균일 랜덤 트래픽에서, 셀이 버스를 통해 다른 모듈로 향하게 될 확률은  $(m-1)\lambda/m$ 이 된다. 여기서,  $\lambda$ 는 셀 도착율이며,  $m$ 은 스위치 모듈의 수가 된다. 확률  $\lambda/m$ 을 가지고, 입력 셀은 자신의 모듈의 출력포트로 향하게 된다.

### 2.4 버스 인터페이스

각 단위 모듈과 버스 연결구조에서는 그림 3에서와 같이 각 포트마다 단위모듈과 버스가 통신하기 위한 연결 장치 즉, 버스 인터페이스를 갖고 있다. 그림 4는 기능 블록 위주로 버스 인터페이스의 내부 구조를 보여주고 있다.

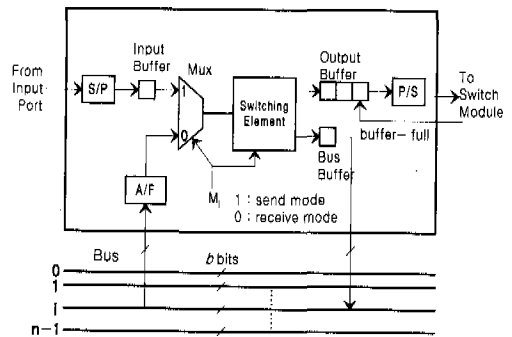


그림 4. 버스 인터페이스

앞서 설명한 중재 방식에서와 같이 각 모듈의 타이밍 흐름에 의한 모듈의 상태 모드에 따라 버스 인터페이스는 셀을 입력포트로부터 읽을지, 또는 버스로부터 받아들일지를 결정하게 된다. 그 결정은 그림 2에 나타나 있는 타이밍 클럭  $M_i$ 에 따르게 된다. 버스 인터페이스의 동작 기능을 전송 모드와 수신 모드에 따라 살펴보기로 한다.

#### 1) 전송 모드

각 타임 슬롯동안 모듈  $i$ 에서 입력포트로부터 들어오는 셀들은 우선 S/P 변환기를 거쳐 입력 버퍼에 저장된다. 그 저장된 셀은 다음 타임 슬롯 동안에 처리되기 때문에 단지 하나의 버퍼만을 필요로 하게 된다. 전송모드에서 단위 모듈 스위치  $i$ 에 속해 있는 모든 버스 인터페이스는 클럭  $M_i$ 에 의해 입력포트로부터 셀을 받을 수 있는 상태로 전환하게 된다. 전송 모드일 때, 즉, 클럭  $M_i$ 이 "1"일 때, 입력 버퍼에 저장된 셀은 멀티플렉서에 의해 스위칭 소자(Switching Element)로 보내지게 된다. 스위칭 소자는 들어오는 셀의 헤더 정보를 분석하여 셀을 자신의 모듈 스위치로 보낼지 또는 버스로

보낼지를 결정하게 된다. 멀티캐스트 셀인 경우, 셀 포맷의 MMP에 의해 복사 과정을 거쳐 자신의 모듈과 버스 양쪽 모두에게 보낼 수 있다. 버스 버퍼로 보내진 셀은 버스에 의해 수신 모드상태에 놓여진  $m-1$  개의 모듈로 브로드캐스팅되며, 출력 버퍼로 보내진 셀은 한 타임 슬롯당 하나의 비율로 P/S 변환기를 거쳐, 단위 모듈 스위치인 공유 버퍼 메모리 스위치로 보내지게 된다. 공유 버퍼 스위치의 버퍼가 빈 공간이 없을 경우 공유 버퍼 스위치로부터 Buffer-full 신호가 출력 버퍼로 피드백되어 그 셀은 출력 버퍼에서 잠시 대기하는 상태가 된다. 따라서, 스위치 내에서 셀 손실이 발생할 수 있는 유일한 곳은 바로 이 출력 버퍼가 된다.

2) 수신 모드

단위 모듈 스위치  $i$  에서 수신모드는 타이밍 클럭  $M_i$  이 "0"일 경우에 해당한다. 수신 모드 상태의 모든 모듈 스위치는 항상 버스의 트래픽을 모니터링하다가 버스상의 셀이 있을 경우 그 셀을 받아들인다. 어드레스 필터는 그 셀의 라우팅 정보를 이용하여 셀이 자신의 모듈로 향하는 셀인지를 판별하여 만일 자신의 모듈 스위치의 출력 포트에 향하는 셀일 경우 그 셀을 멀티플렉서로 보내게 되며, 그렇지 않을 경우, 셀을 버리게 된다. 필터링된 셀은 멀티플렉서를 통해 출력 버퍼로 보내지게 된다. 이 과정은 타이밍 클럭  $M_i$  이 계속해서 수신 모드 상태인 "0"일 경우 지속된다.

2.5 스위칭 소자

그림 4의 버스 인터페이스내에서 셀의 라우팅 경로를 설정해주는 스위칭 소자에 대해 알아본다. 그림 5는 버스 인터페이스에서 스위칭 소자의 기능적 모델을 보여주고 있다.

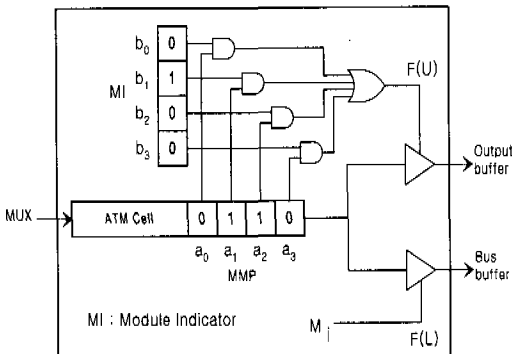


그림 5. 스위칭 소자

주어진 그림은 스위치 모듈의 수가 4이며, 그 스위칭 소자가 속한 모듈이 #2임을 MI를 통해 알 수가 있다. 또한, 그림 6의 셀 포맷에서 설명된 비트 패턴인 MMP를 통해서, 입력된 셀은 모듈 스위치 #1, #2로 향하는 멀티캐스트 셀임을 알 수 있다. 따라서, 이 스위칭 소자는 입력된 멀티캐스트 셀을 현재의 모듈 스위치 #2뿐만 아니라, 스위치 모듈 #1로도 보내기 위해 셀을 복사해야 한다.

스위칭 소자에서 셀 스위칭은 셀의 헤더 정보인 MMP와 각 단위 모듈을 표시해주는 비트 패턴 MI와 타이밍 클럭  $M_i$ 에 의해 이루어진다. 스위칭 소자에서 일어나는 셀 스위칭 패턴은 표 1에서 보는 바와 같이 4가지로 나타낼 수 있다. 표에서 나타난 바와 같이 비트 함수  $F(U, L)$ 에 의해 스위칭 패턴이 정해지는데 함수  $F(U, L)$ 는 다음과 같이 나타낼 수 있다.

$$F(U) = a_0 \cdot b_0 + a_1 \cdot b_1 + \dots + a_{m-1} \cdot b_{m-1} \tag{2}$$

$$F(L) = M_i \tag{3}$$

여기서,  $a_i$ 는 MMP의 비트를,  $b_i$ 는 MI의 비트를,  $M_i$ 는  $i$ 번째 모듈의 타이밍 클럭을 가리킨다. 그리고 식에서  $\cdot$ 는 AND 연산,  $+$ 는 OR 연산을 나타낸다.

표에서 볼 때, 함수  $F(U, L)$ 가 '00'일 경우는 어떠한 스위칭도 일어나지 않고 있음을 주의할 필요가 있다. 나머지 세 경우는 함수 값에 의해 단일 경로로 스위칭 되거나 또는 양쪽 경로로 멀티캐스팅 되어 진다. 함수  $F(U, L)$ 가 '01'일 경우는 입력된 셀이 하위 경로 즉, 버스로 향하고 있음을 나타내며, '10'인 경우는 그 셀이 자신의 모듈 스위치로만 향하고 있음을 나타낸다. '11'인 경우는 그 셀이 자신의 모듈과 다른 모듈로도 향하고 있음을 나타내는 것으로, 스위칭 소자는 그 셀을 복사하여 상위 경로와 하위 경로 양방향으로 셀을 전달하게 된다.

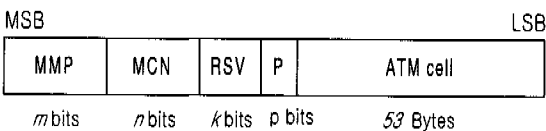
표 1. 스위칭 소자에서의 비트 함수에 의한 스위칭 동작

$F(U, L)$	스위칭 동작
0 0	없음
0 1	하위 포트
1 0	상위 포트
1 1	양쪽 포트

2.6 셀 포맷

그림 6은 제안된 스위치에서 사용하게 되는 멀티캐스트 ATM 스위치의 셀 포맷이다. 멀티캐스트 모듈 패턴(MMP)은 일종의 비트 패턴이다. 입력포트에서 들어오는 셀은 헤더분석에 의해 스위치 내부에서의 라우팅 경로가 정해지게 되는데, 이 모듈 패턴에 의해 입력 셀은  $m$ 개의 모듈중 어느 모듈로 향하는 셀인지를 판별하게 된다. 예를 들어, 입력 셀의 모듈 패턴의  $i$ 번째 비트가 "1"로 되어 있으면, 그 셀은  $i$ 번째 모듈로 보내지게 된다. 멀티캐스트 셀인 경우는 "1"로 지정된 비트가 하나 이상일 수 있으며, 유니캐스트 셀인 경우는 단지 하나의 "1" 비트를 갖게된다. 사용되는 비트의 수는 모듈의 수와 같은  $m$ 비트가 되며, 각 비트가 모듈 하나와 연관지어지게 된다.

멀티캐스트 채널 패턴(MCN)은 각각의 모듈 스위치에서 필요한 만큼의 셀을 복사해서 다시 그 모듈 스위치의 출력포트로 멀티캐스트하기 위한 정보를 담고 있는 부분이다. 이 부분의 정보를 이용해서 다음 3장에서 논의될 스위치 모듈(공유 버퍼 메모리 스위치)은 다시 그 모듈내에서 지원되는 멀티캐스트 기법에 따라 멀티캐스팅을 수행하게 된다.



MMP : Multicast Module Pattern    MCP : Multicast Channel Pattern  
 RSV : Reserved for future use    P : Priority  
 MSB : Most Significant Bit        LSB : Least Significant Bit

그림 6. 라우팅을 위한 스위치 내부에서의 ATM 셀 포맷

우선 순위(P) 부분은 각 단위 모듈 스위치에서의 셀의 선택, 전송에 있어 셀 충돌시 어느 셀을 선택할 지에 대한 기준을 제공한다. 우선 순위는 비트수에 의해 다단계로 분리될 수 있다. 비트 수가  $p$ 비트일 경우 우선 순위의 단계는  $2^p$ 가지로 나뉘어진다.

RSV 비트는 나중 사용을 위해 부가 비트로 첨부되며, 그 뒤로 실제 데이터를 담고 있는 ATM 셀이 이어진다. RSV 비트는 전체 셀 크기를 2의 지수승으로 만들도록 설정되어진다. 셀 포맷에서 원래의 데이터인 53Bytes의 ATM 셀을 제외하고 스위치 내에서 라우팅을 위한 앞부분의 추가 헤더는 스위

치를 빠져나갈 때 다시 제거되어 다음의 스위치 노드로 전송되게 된다. 이 추가 헤더는 단지 스위치 내부에서 셀 스위칭을 위해 스위치 컨트롤러에 의해 첨가되었던 부분인 것이다.

III. 공유 버퍼 메모리 스위치

이제까지 버스를 통한 멀티캐스트 ATM 스위치의 구조를 살펴보았다. 지금 부터는 단위 모듈 스위치로 선택한 공유 버퍼 메모리 스위치를 멀티캐스팅 기능 지원 측면에서 살펴 보기로한다. 단위 모듈 스위치로서 공유 버퍼 메모리 스위치를 택한 이유 중 하나는 바로 버퍼 공유 효과(Buffer Sharing Effect)에 의해 스위치 구현에 있어 필요한 버퍼 메모리의 양을 줄일 수 있다는 것이다. 공유 버퍼 메모리 스위치 구조에서는 출력 버퍼 스위치와 같이 버퍼를 공유하지 않을 경우 주어진 셀 손실율을 만족시키기 위해 필요한 버퍼양보다 작은 버퍼만을 필요로 한다. 버퍼 공유 효과에 의해 공유 버퍼 메모리 스위치는 랜덤 트래픽에서의 셀 손실율 뿐만 아니라 버스트(Bursty) 트래픽에서 셀 손실율을 크게 감소시킬 수 있는 장점을 갖고 있다<sup>10)</sup>. 공유 버퍼 메모리 스위치에서 가장 중요한 사실은 버퍼 공유에 의해 출력포트 경쟁이 발생하지 않으며 따라서 최적의 데이터 수율, 셀 손실률, 그리고 셀 지연 성능을 가지므로 대용량 ATM 스위치의 단위 모듈로서 사용할 수 있는 큰 매력을 갖는다고 할 수 있다. 지금 부터는 공유 버퍼 메모리 스위치에서 멀티캐스팅 기능을 중심으로 살펴 보기로 한다.

3.1 멀티캐스팅을 고려한 어드레스 큐잉

제안된 멀티캐스트 ATM 스위치가 전체적으로 멀티캐스팅을 할 수 있기 위해서는 지역적으로 즉, 단위 모듈 스위치에서도 멀티캐스팅을 지원할 수 있어야 한다. ATM 셀을 버퍼 메모리에 저장하고 출력하는 과정은 셀이 저장된 메모리의 어드레스를 제어하는 스위치의 컨트롤러에 의해 이루어지게 된다. 그리고 멀티캐스팅 기능 또한 이 컨트롤러 회로에 의해 지원할 수 있다. 그림7은 멀티캐스팅을 지원하는 FIFO 어드레스 큐를 갖고 있는 컨트롤러의 구조를 보여주고 있다<sup>11)</sup>.

구조는 크게 네 가지 기능 요소로 분리되어 있다. 셀의 쓰기 동작을 담당하는 쓰기 어드레스 제어(WAC : Write Address Control), 읽기 동작을 담당하는 읽기 어드레스 제어(RAC : Read Address

Control), 비어있는 버퍼 메모리 어드레스를 저장하는 Idle Address FIFO Buffer(IAFB), 그리고 셀이 저장된 메모리 어드레스를 기록해두는 어드레스 큐(AQ : Address Queue)로 구성된다.

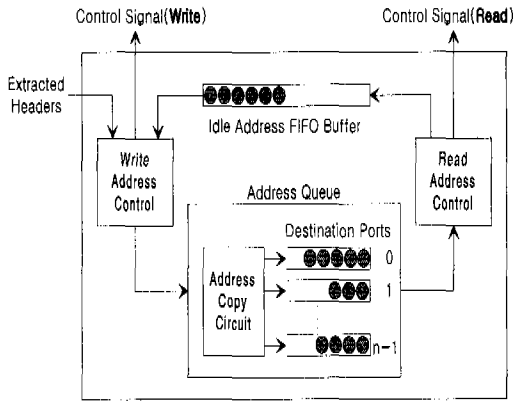


그림 7. FIFO 어드레스 큐잉 방식의 스위치 컨트롤러의 블록 다이어그램

WAC는  $N$ 개의 입력포트로부터 차례로 들어오는 셀들의 헤더 변환에 의해 그 셀의 목적지 출력포트를 알게 되며, IAFB로부터 읽어들이는 메모리 어드레스들을 각각의 셀에게 차례로 저장 어드레스로 할당한다. 그리고, 그 입력된 셀들에게 할당된 어드레스를 셀의 목적지 출력포트에 따라 AQ에 각각 저장하게 된다. 입력된 셀의 목적지 출력포트가  $i$ 일 경우,  $i$ 로 향하는 셀들의 메모리 어드레스만을 저장하는 AQ  $i$ 에 넣게 된다. 만일 입력된 셀이 목적지 출력포트를 여러 개 갖는 멀티캐스트 셀인 경우는 그 멀티캐스트 셀이 저장되는 메모리 어드레스를 그 셀의 Fanout 수만큼 복사하여 멀티캐스트 되어질 출력포트의 AQ로 각각 저장되어진다. RAC는 셀의 읽기 동작을 수행하게 된다. RAC는 AQ의 각 큐로부터 차례로 큐의 맨 앞의 어드레스를 읽어들이고 그 어드레스에 저장된 셀을 읽어 출력포트로 내보내게 된다. 그리고, 읽혀진 메모리 어드레스는 다음 사용을 위해 IAFB에 저장되게 된다. 만일 멀티캐스트 셀이 저장된 메모리 어드레스는 여러 번 액세스해야 하기 때문에 한 번 읽고 나서 IAFB에 넣게 되면 안 된다. 따라서, 멀티캐스트 셀 카운터(MCC : Multicast Cell Counter)와 같은 장치를 사용하여 멀티캐스트 Fanout 수만큼 그 메모리 어드레스를 읽고 나서 IAFB에 넣도록 하는 것이 필요하다.

### 3.2 단일 메모리 어드레스 큐잉 방법

그림 7의 분리형 FIFO 어드레스 큐잉 방법은 단순한 구조로 멀티캐스팅을 구현할 수 있지만 어드레스 큐의 속도와 크기에 있어서 몇 가지 단점을 지니고 있다. 앞서 얘기된 바와 같이 FIFO 어드레스 큐잉에서는 멀티캐스트 셀의 메모리 어드레스를 Fanout 만큼 복사하여 AQ에 저장하게 되는데, 최악의 경우,  $N$ 입력에서 들어오는 모든 셀이  $N$ 출력포트로 브로드캐스팅되면 결과적으로  $N^2$ 개의 메모리 어드레스가 생기게 된다. 이 결과는 AQ의 속도에 큰 영향을 미치게 된다. 그리고, 입력되는  $N$ 개의 셀이 모두 하나의 출력포트로 향할 경우 셀 손실을 없애기 위해서는 각각의 AQ는 전체 버퍼 메모리 크기 보다 더 커야 한다.

[12]에서는 이러한 분리형 FIFO 어드레스 큐잉의 단점을 극복하기 위해 단일 메모리 어드레스 큐잉(Single Memory Address Queuing)방법을 제안했다. 이 방법은 각각의 출력포트마다 별개의 AQ를 두는 것이 아니라 공유 버퍼 메모리 방식의 기본 개념인 공유 효과를 적용하기 위해 분리되어 있는 각각의 메모리 어드레스 큐 또한 하나의 단일 큐에 의해 공유하는 것이다. 그림 8에 단일 어드레스 큐의 구조를 나타내었다.

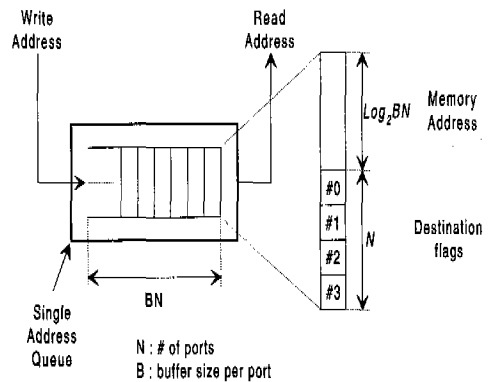


그림 8. FIFO 어드레스 큐

이 방식에서 스위치 컨트롤러의 구조는 그림 7을 따르지만 어드레스 큐의 구조만이 다를 뿐이다. 셀이 입력포트로 들어오면, 그 셀은 일정 버퍼 메모리에 저장되면서 그 저장된 버퍼 메모리의 어드레스는 FIFO 어드레스 큐에 저장된다. 큐의 구조는 셀이 버퍼 메모리에 저장된 위치를 가리키기 위한 어

드레스부분과 멀티캐스트 셀인 경우 Fanout 포트를 나타내기 위한 플래그(Flags)로 이루어져 있다. 플래그는 셀이 어느 출력포트로 나가게 될지를 표시하게 되는데, 만일 비트가 '1'로 셋팅되어 있으면 그 셀은 셋팅된 출력포트로 향하고 있음을 표시해준다. 따라서, 멀티캐스팅은 이 플래그 비트에 의해, 셋팅된 비트 수만큼 메모리 읽기 동작을 수행함으로써 이루어진다. 물론 지정된 어드레스의 셀을 한번 읽은 후 어드레스를 나중에 입력되는 셀의 저장장소로 할당하기 위해 IAFB에 보내면 안 된다. Fanout 만큼 더 읽기 동작을 위해 완전히 멀티캐스팅이 끝날 때까지 그 셀은 버퍼 메모리에 저장되어 있도록 하는 로직을 필요로 한다. 완전히 멀티캐스팅이 끝나게 되면 그 어드레스는 다음 셀의 저장을 위해 IAFB에 보내어진다.

#### IV. 시뮬레이션 결과 및 분석

##### 4.1 시뮬레이션 환경

본 논문에서는 제안한 멀티캐스트 ATM 스위치의 성능을 C언어로 작성한 Event-Driven 시뮬레이션 프로그램을 이용하여 검증하며, 스위치에 적용되는 트래픽은 순수(Pure) 멀티캐스트 트래픽, 그리고 유니캐스트 트래픽과 멀티캐스트 트래픽이 혼합된 경우에 대해 시뮬레이션한다.

순수 멀티캐스트 트래픽은 입력 되는 셀이 요구하는 복사셀의 수에 따라 크게 다른 트래픽 환경을 만들게 된다. 셀당 요구되는 복사셀의 수를 흔히 Fanout 이라 하게 된다. 순수 멀티캐스트 트래픽은 단지 멀티캐스트 셀들의 입력으로만 구성된다. 따라서, 입력에서의 제공 부하(Offered Load)또는 셀 도착율이 최대( $\lambda = 1.0$ )일 경우, 멀티캐스팅에 의해 스위치 시스템에 전달되는 실제 입력 부하(Effective Offered Load)는 1.0 을 초과하게 된다. 이런 상황에서는 스위치가 안정상태를 유지 할 수 없게 된다. 예를 들어, 입력 셀당 평균 Fanout 수가 2 이고, 입력 부하가 0.9 일 경우 실제 입력 부하는  $0.9 \times 2 = 1.8$  이 되게 된다. 따라서, 안정상태에서 스위치의 성능 분석을 위해서는 실제 입력 부하를 1.0 보다 작게 유지해 주어야 한다.

입력에서의 셀 도착이 서로 독립적이고, 균일한 분포를 갖는다고 가정하자. 주어진 타임 슬롯동안, 입력 부하가  $\lambda$  이며, 입력 셀당 요구되는 복사 셀의 수를 C 라 하면, C는 다음의 식으로 주어지는 Truncated 기하 분포를 갖는 랜덤 변수가 된다.

$$\Pr(C = k) = \frac{(1-q)q^{k-1}}{1-q^N} \quad 1 \leq k \leq N \quad (4)$$

평균 복사 셀의 수는 다음과 같다.

$$E(C) = \sum_{i=1}^N iP\Pr(C=i) = \frac{1}{(1-q)} \cdot \frac{Nq^N}{1-q^N} \quad (5)$$

따라서, 실제 입력 부하  $\lambda_{eff}$ 는 다음과 같이 된다.

$$\lambda_{eff} = \lambda E(C) \quad (6)$$

입력 멀티캐스트 셀에 의한 복사 수는 자신을 포함하여  $2 \leq E(C) \leq N$ 이 된다.

복사셀의 수가 Truncated 기하 분포를 따르지 않고 일정 상수 C를 가지는 경우의 실제 입력 부하는 다음과 같이 주어지게 된다

$$\lambda_{eff} = \lambda \cdot C \quad 2 \leq C \leq N \quad (7)$$

순수 멀티캐스트 트래픽인 경우와 달리 혼합(Mixed) 트래픽인 경우는 멀티캐스트 셀과 유니캐스트 셀이 입력에서 혼합되어 도착하는 경우에 해당된다. 입력되는 모든 소스 셀의 10%가 멀티캐스트 셀이라 가정하고, 각 멀티캐스트 소스 셀당 평균 복사 수를 4라 하면, 전체 트래픽 중에서 40%(=10×4)가 멀티캐스트 트래픽이 된다. 실제 스위치에 입력되는 입력부하는 멀티캐스트 셀에 의해 다음과 같이 주어지게 된다.

$$\lambda_{eff} = \lambda [(1-f) + E(C) \cdot f] \quad (8)$$

여기서, f는 입력 되는 모든 소스 셀 가운데 멀티캐스트 셀이 차지하는 비율을 나타낸다.

##### 4.2 시뮬레이션 결과 및 분석

그림 9는 순수 멀티캐스트 트래픽에서 최대 수율(Throughput)을 스위치 크기와 평균 복사셀 수에 대해 보여주고 있다. 전체 스위치 크기는 수율에 크게 영향을 미치지 않으나, 평균 복사 수( $E(C) : Fanout$ )는 전체 스위치 수율에 영향을 준다는 것을 알 수 있다. 평균 복사셀 수가 2 이고, 스위치 크기가 32 인 경우의 최대 수율은 0.884 이다. 스위치 크기와 수율과는 크게 상관 관계를 갖지 않고 있음을 보여 주고 있다. 이는 제안된 스위치 크기의 확장성의 우수함을 보여주고 있는 결과이다. 그림 10은 혼합 멀티캐스트 트래픽인 경우에 해당하는 결과로, 평균 복사셀 수와 멀티캐스트 트래픽의 비



율을 변화시키면서 스위치의 최대 수율을 구해 본 것이다. 멀티캐스트 비율은 일정하게 유지하면서 복사셀 수를 증가시킬 경우 최대 수율은 감소하게 되며, 멀티캐스트 비율이 클수록 감소 폭은 커짐을 알 수 있다.

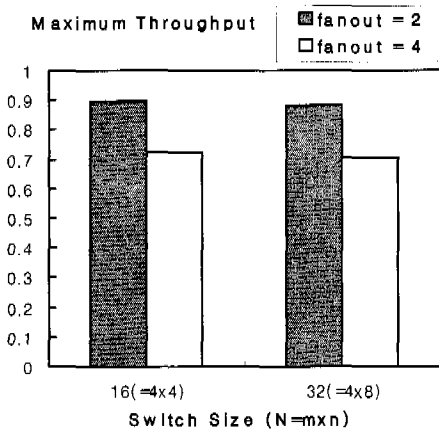


그림 9. 순수 멀티캐스트 트래픽일 경우의 최대 수율

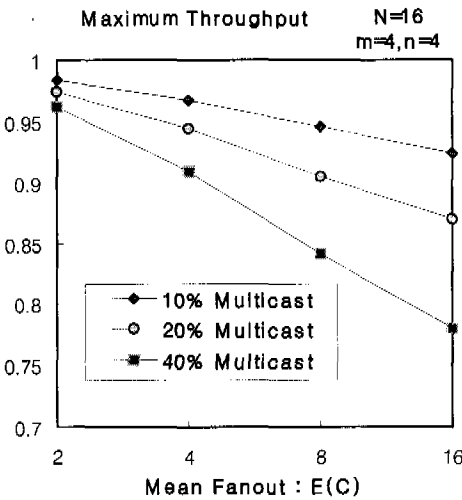


그림 10. 혼합 멀티캐스트 트래픽일 경우의 최대 수율

그림 11은 순수 멀티캐스트 트래픽인 경우의, 평균 셀 지연에 대한 시뮬레이션 결과를 입력포트에서의 셀 도착율에 대해서 보여주고 있다. Fanout이 2인 경우, 스위치가 안정 상태를 유지하기 위해서는 셀 도착율이 0.5 보다 작아야 한다. 그림에서 보면, 셀 도착율이 0.5에 근접함에 따라 실제 제공 부하는 1.0에 근접하므로, 셀 지연은 무한대로 향하게 된다. 즉, 스위치에 입력되는 멀티캐스트 셀의

도착율이 0.5 라면, 버스 상에 셀 복사기 발생하여 실제로 모듈내의 공유 버퍼 메모리로 입력되는 셀의 도착율은 Fanout만큼이 곱해지게 되므로, 실제로 스위치 내부에서의 셀 지연은 거의가 공유 버퍼에서 발생하게 된다. One-Shot Scheduling, Call Splitting, Revision Scheduling과 같은 기존의 멀티캐스팅 방식에서는<sup>[13][14]</sup>, 평균 복사셀 수  $E(C)$ 가 4일 경우, 셀 도착율이 0.2에 이르기 전에도 스위치는 안정상태를 벗어나게 되어 셀 지연이 크게 증가하게 된다. 그러나, 그림에서 알 수 있듯이 제안된 스위치의 셀 지연은 실제 제공 부하가 1.0 ( $\lambda=0.25$ )이 되는 경우에 무한대로 증가하게 되며, 이 결과는 출력 큐잉을 사용하는 멀티캐스팅 스위치에서의 셀 지연<sup>[14]</sup>과 거의 동일하다는 것을 알 수 있다.

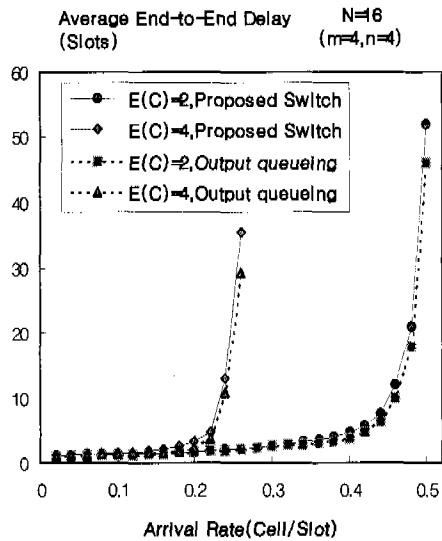


그림 11. 순수 멀티캐스트 트래픽에서의 셀 지연

그림 12는 혼합 멀티캐스트 트래픽인 경우의 평균 셀 지연을 보여주고 있다. 여기서, 평균 복사셀 수는 2이다. 지연을 구하는 데 있어서는 출력포트로 셀이 나가는 수율을 기준으로 하여 멀티캐스트 트래픽의 비율의 변화에 따른 결과를 구하였다. 0%의 멀티캐스트 트래픽인 경우는 유니캐스트 트래픽 중에서도 균일 랜덤 트래픽인 경우에 해당하는 셀 지연과 일치한다. 그림에서 알 수 있듯이 멀티캐스트 비율이 증가함에 따라 셀 지연은 늘어남을 알 수 있다. 그러나, 전체 트래픽의 10% 정도가 멀티캐스트일 지라도 수율이 0.8 이하에서의 셀 지연은

거의 균일 랜덤 트래픽과 같은 결과를 보이고 있는데, 이는 출력 큐잉 방식의 셀 지연과도 거의 같은 값이 된다. 수율이 0.9 이고 멀티캐스트 비율이 20% 인 경우의 셀 지연은 대략 58 타임 슬롯을 필요로 한다.

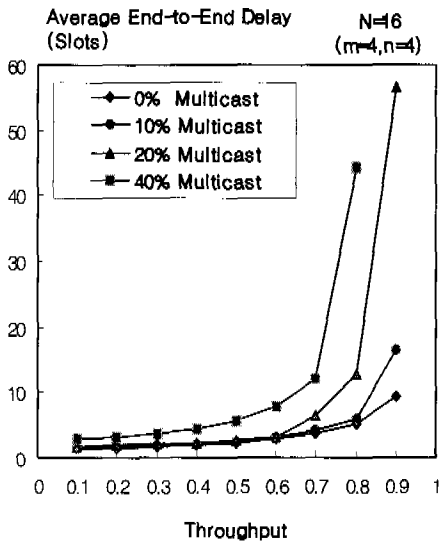


그림 12. 혼합 멀티캐스트 트래픽에서의 셀 지연( E(C)=2 )

그림 13은 평균 복사셀 수  $E(C)$  를 4로 했을때의 평균 셀 지연을 보여주고 있다. 그림 14와 거의 같은 분포를 가지고 있으나 전체적으로 평균 복사셀 수가 증가할 경우 같은 비율의 멀티캐스트 트래픽이 입력 될 지라도 지연은 더 증가하고 있음을 알 수가 있다. 이는 평균 복사셀의 증가는 공유 버

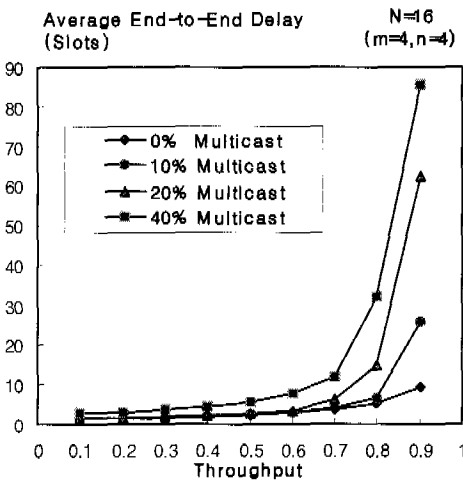


그림 13. 혼합 멀티캐스트 트래픽에서의 셀 지연( E(C)=4 )

퍼 메모리 스위치에서 한 셀당 다수의 읽기 동작을 요구하기 때문에 공유 버퍼에서의 셀 지연의 증가에 따른 결과이다. 그러나, 그림 12에서와 마찬가지로 전체 트래픽의 10% 정도가 멀티캐스트일 지라도 수율이 0.8 이하에서의 셀 지연은 거의 균일 랜덤 트래픽과 같은 결과를 보이고 있다.

### V. 결론

본 논문에서는 대용량으로의 확장성과 모듈러 구조를 제공하며 용이하게 멀티캐스팅이 지원 가능한 멀티캐스트 ATM 스위치를 제안하였다. 브로드캐스팅을 속성으로 하는 버스를 이용하여 쉽게 멀티캐스팅을 구현할 수 있었으며, 다양한 트래픽 조건에서 버퍼 메모리량을 줄일 수 있는 공유 버퍼 메모리 스위치를 단위 모듈로 사용함으로써 스위치 전체의 버퍼 메모리 양도 줄일 수 있는 스위치 구조임을 알 수 있었다. 제안된 ATM 스위치는 멀티미디어 서비스에서 중요한 셀 지연 측면에서 가장 나은 성능을 보이는 출력 큐잉 방식의 멀티캐스트 스위치와 거의 동일한 결과치를 보였다.

### 참고 문헌

- [1] Recommendation I.121, CCITT Document AP IX-143-E, July, 1988
- [2] Balaji Probhakar, Nick McKeown, and Ritesh Ahuja, "Multicast Scheduling for Input-Queued Switches," IEEE Journal on Selected Areas in Communications, vol. 15, no. 5, June 1997.
- [3] Ka Lun Eddie Law, Alberto Leon-Garcia, "A Large Scalable ATM Multicast Switch", IEEE Journal on Selected Areas in Communications, vol. 15, no. 5, June 1997.
- [4] H. Jonathan Chao, et al., "Design and Implementation of Abacus Switch: A scalable Multicast ATM switch," IEEE Journal on Selected Areas in Communications, vol. 15, no. 5, pp. 830-843, Jun. 1997.
- [5] J. Garcia-Haro and A. Jajszczyk, "ATM shared-memory switching architecture," IEEE Network, pp. 18-26, July/Aug. 1994.
- [6] H. Kuwahara, et al., "A shared buffer memory switch for an ATM exchange," in Proc. Int. Conference on Communications (ICC89), pp.

