

# 멀티캐스트 트래픽을 지원하는 입력 버퍼 ATM 스위치에서의 셀 스케줄링 기법

## (Cell Scheduling for Multicast Traffic in Input-Queueing ATM Switches)

조민희<sup>†</sup> 송효정<sup>†</sup> 권보섭<sup>\*\*</sup> 윤현수<sup>\*\*\*</sup> 조정완<sup>\*\*\*</sup>

(Minhee Cho) (Hyojeong Song) (Boseob Kwon) (Hyunsoo Yoon) (Jung Wan Cho)

**요약** 온라인 화상회의, VOD 등의 멀티캐스트(multicast) 특성을 갖는 서비스를 효율적으로 제공하기 위해서는 스위치 수준에서의 멀티캐스트 트래픽 처리를 위한 연구가 필요하다. ATM 스위치 중 입력버퍼형은 하드웨어 구현 복잡도가 낮아 고속의 트래픽 처리와 대용량 스위치 구현에 적합한 반면, 높은 성능을 가지기 위해서는 임의접근(random access) 입력버퍼와 좋은 셀 스케줄링 알고리즘이 필요하다. 본 논문에서는 입력버퍼형 ATM 스위치에서의 멀티캐스트 셀 스케줄링 알고리즘인 무충돌 시간예약(CFTR) 기법을 제안한다. CFTR 기법은 입력버퍼의 셀의 전송시점을 충돌이 없도록 예약함으로써 높은 처리율을 가질 수 있도록 하며, 이를 위해 입력단, 출력단 스케줄러에 예약 테이블을 둔다. CFTR 기법은 각 출력단 스케줄러에서의 예약과정이 간단하고 독립적, 병렬적 수행이 가능하므로 고속 트래픽 처리에 적합하다. CFTR 기법의 성능평가를 위해 시뮬레이션을 통해 기존의 셀 스케줄링 방식과 비교하며, 약간의 하드웨어 추가로 매우 좋은 성능을 보임을 알 수 있다.

**Abstract** In this paper, we propose a novel multicast cell scheduling algorithm, called *CFTR* (*Conflict-Free Time Reservation*), for input-queueing ATM switches. While the existing multicast cell scheduling algorithms have low throughput because of operating in FIFO fashion, CFTR increases the throughput by using non-FIFO input queues and the advance reservation method for contention resolution, for which each input/output keeps a reservation table representing whether time slots in the future are reserved or not. CFTR can operate at high speed because of independent output scheduling and only a few iterations of reservation processes in each time slot. The performance of CFTR is compared with existing unicast and multicast algorithms by simulation. Simulation results show that CFTR performs better than other unicast and multicast algorithms even with the small size of reservation tables and a few iterations of reservation processes.

## 1. 서론

온라인 화상회의, VOD 등의 멀티캐스트(multicast) 특

성을 갖는 서비스를 효율적으로 제공하기 위해서는 스위치 수준에서의 멀티캐스트 트래픽 처리를 위한 연구가 필요하다. 입력버퍼형 ATM 스위치는 출력버퍼형, 공유버퍼형에 비해 하드웨어 구현 복잡도가 낮아 고속의 트래픽 처리와 대용량 스위치 구현에 적합한 반면, HOL(Head of Line) 봉쇄 현상으로 인해

최대 처리율이 58%로 제약된다는 단점이 있다 [1]. 이를 위해 임의접근(random access) 입력버퍼와 좋은 셀 스케줄링 알고리즘을 사용함으로써 스위치의 처리율을 높여주는 연구가 진행되고 있다. 멀티캐스트 트래픽 처리를 위한 셀 스케줄링 알고리즘에 대한 연구는 거의 이루어져 있지 않다. 먼저, 시분할 다중 교환(Time-Division

· 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

† 학생회원 : 한국과학기술원 전자전산학과  
mhcho@calab.kaist.ac.kr  
hjsong@calab.kaist.ac.kr

\*\* 비회원 : 안동대학교 컴퓨터공학교육과 교수  
bxkwon@andong.ac.kr

\*\*\* 종신회원 : 한국과학기술원 전자전산학과 교수  
hyoon@calab.kaist.ac.kr  
jwcho@calab.kaist.ac.kr

논문접수 : 1999년 4월 19일

심사완료 : 2000년 4월 14일

Multiplex Switching) 시스템에서의 멀티캐스트 시간슬롯 할당(Multicast Time Slot Assignment : MTSA) 문제를 ATM 스위치에서의 멀티캐스트 셀 스케줄링에 적용하는 것을 생각해 볼 수 있다.

멀티캐스트 시간슬롯할당 문제는 주어진 셀의 집합을 최단 시간에 출력단으로 모두 전송할 수 있도록 멀티캐스트 셀에 전송 시간슬롯을 할당하는 문제이다. 이를 위한 휴리스틱 알고리즘이 목적지 분할(fanout splitting)의 경우와 목적지비분할(fanout non-splitting)의 경우에 대해 연구되었으나 [2,3,4], 시간복잡도가 높아 ATM 스위치를 위한 멀티캐스트 셀 스케줄링 방식으로는 적합하지 않다. 멀티캐스트 셀 스케줄링 알고리즘에 대한 기존의 연구는 McKeown 등에 의해 이루어져 왔으나 [5,6] 모두 FIFO 방식이어서, 전송되지 못한 선두의 멀티캐스트 셀로 인해 다른 셀의 전송이 방해받는 HOL 봉쇄 현상으로 인해 낮은 처리율을 가진다는 단점이 있다.

본 논문에서는 멀티캐스트 셀 스케줄링 알고리즘인 무충돌 시간예약(Conflict-Free Time Reservation : CFTR) 기법을 제안한다. CFTR 기법은 입력단, 출력단의 예약 상태를 나타내는 입·출력단 예약 테이블을 사용해 출력단 간의 예약시점의 충돌이 없도록 전송시점 예약을 함으로써 높은 처리율을 가진다.

본 논문의 구성은 다음과 같다. 2절에서는 셀 스케줄러를 사용하는 입력버퍼형 ATM 스위치의 구조를 살펴보고, Obara 등에 의해 제안된 셀 스케줄링 알고리즘 [7]과 그 문제점을 알아본다. 3절에서는 이를 개선하여 높은 처리율을 제공하며 멀티캐스트 셀을 처리할 수 있는 셀 스케줄링 알고리즘인 CFTR 기법을 제안하고 그 동작과 하드웨어 구성도를 살펴 본다. 이어서, 4절에서는 CFTR 기법의 성능분석을 위해 여러 트래픽 형태에 대한 컴퓨터 시뮬레이션 결과를 제시하여 기존 알고리즘과 비교, 평가하고, 마지막으로 5절에서는 결론을 내리고 앞으로의 연구 방향을 제시한다.

## 2. 셀 스케줄링 알고리즘

입력버퍼형 ATM 스위치는 입력셀을 저장하기 위해 FIFO 입력버퍼 대신 임의접근 입력버퍼를 사용하고, 동일 출력단에 대한 셀들 간의 경쟁을 해결하기 위해 셀 스케줄러를 사용함으로써 처리율을 향상시킬 수 있다. 이러한 스위치의 전체 구조는 [그림 1]과 같이 입·출력단, 비봉쇄 자가 라우팅 공간분할형 스위치(non-blocking self-routing space switch), 셀 스케줄러로 이루어진다. 각각의 입력단은 입력셀을 저장하기 위한 임의접근 메모리(random access memory : RAM)와 입력

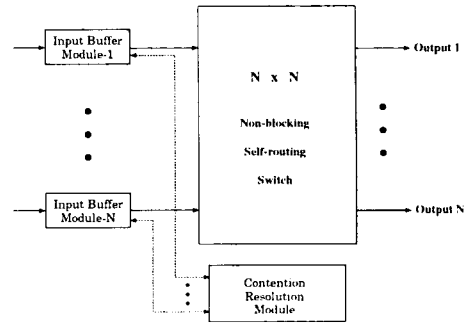


그림 1 입력버퍼형 ATM 스위치의 블록 다이어그램

버퍼 제어기(input-queue controller)로 이루어진다. 셀 스케줄러는 입력버퍼 제어기로부터 정보를 받아서 셀의 전송시점을 스케줄한다. 자가 라우팅 공간분할형 스위치는 셀 스케줄링 모듈에 의해 스케줄된 전송시점에 입력버퍼에 있는 셀을 목적하는 출력단으로 전송한다.

Obara가 제안한 셀 스케줄링 알고리즘 [7]은 각 출력단 스케줄러의 동작이 간단하고 독립적이어서 고속 트래픽의 처리에 적합하다. 또한, 한 셀 타임에 각 입력단마다 하나의 셀이 목적하는 출력단으로부터 전송될 시점을 예약받으므로 멀티캐스트 스케줄링에 쉽게 적용할 수 있다. 그러나, 출력단 스케줄러가 입력단의 예약상태를 모르는 상태에서 스케줄링하기 때문에 동일 입력단의 다른 셀이 이미 예약되어 있는 시점을 스케줄링하는 예약 충돌 현상이 발생할 수 있다. 이로 인해 해당 출력단은 스케줄된 시점에 셀을 전송받을 수 없게 되어 최대 65%의 처리율에 그치는 단점을 가지고 있다.

Obara의 스케줄링 알고리즘 [7]은 각 입력단이 원하는 출력단 스케줄러로 전송요구신호를 보내고, 각 출력단 스케줄러가 요구신호를 보낸 입력단에게 전송시점을 예약해 주는 두 단계로 이루어져 있다. 출력단 스케줄러  $j$ 는 출력단  $j$ 의 예약가능 첫번째 시점을 가리키는  $T_j$ 를 가지고 있다. 출력단 스케줄러  $j$ 가 시점  $t$ 에 어떤 입력단으로부터 요구신호를 받으면,  $T_j$  값을 그 입력단으로 전송하고  $T_j$ 를 1만큼 증가시킨다. 이것은 이 입력단의 해당셀에게 시점  $T_j$ 를 예약해 주었다는 것을 의미한다. 입력단은 예약받은 전송시점 순으로 셀을 버퍼에 보관하게 된다. 그러나, 이 입력단의 다른 셀이 이미 시점  $t' \leq t$ 에 동일한 시점  $T_k (T_k = T_j, k \neq j)$ 를 출력단  $k$ 로부터 예약받았을 수 있다. 입력단은 한 셀 타임에는 하나의 셀만 전송할 수 있으므로 이 경우, 이 셀은 전송시점을 다시 예약받기 위하여 다음 셀 타임  $t+1$ 에 다시 출력단  $j$ 로

요구신호를 보내야 한다. 이 방식은 각 출력단 스케줄러가 하나의 가산기로부터 이루어질 수 있어 구현 복잡도가 낮다는 장점이 있다. 그러나, 위와 같이 출력단간의 예약시점 충돌로 인해 출력단  $j$ 에서 시점  $T_j$ 를 입력단에게 예약해 주었음에도 불구하고 실제로는 시점  $T_j$ 에서 셀을 전송받지 못한다.

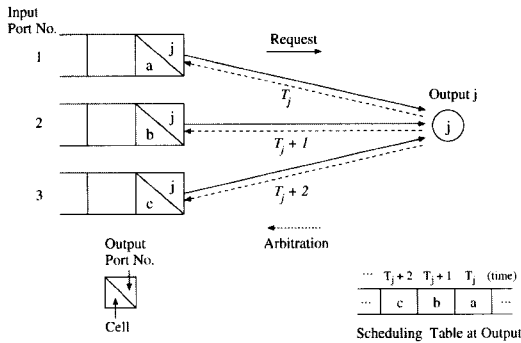


그림 2 Obar의 알고리즘에서 각 출력단에서의 스케줄링

### 3. 무충돌 예약 (Conflict-Free Time Reservation) 기법

본 논문에서 제안하는 CFTR(Conflict-Free Time Reservation) 기법에서는 각 입력단이 입력단의 예약 상태를 나타내는 **입력단 예약 테이블**을 가지며, 각 출력단 스케줄러는 해당 출력단의 예약 상태를 나타내는 **출력단 예약 테이블**을 가지고 있다. 그리고, 입력단에서는 전송 시점이 예약된 셀에 대한 정보를 저장하는 전송 테이블을 관리한다. CFTR 기법은 다음과 같이 요구, 예약, 전송의 세 단계로 이루어진다.

**1. 요구(Request) 단계 :** 각 입력단은 예약셀의 목적 출력단들 중 아직 예약되지 않은 출력단 스케줄러로 입력단 예약 테이블과 함께 요구신호를 보낸다.

**2. 예약(Reservation) 단계 :**

(1) 각 출력단 스케줄러는 요구신호를 보낸 입력단들 중  $k$  ( $1 \leq k \leq N$ )개를 선택한다.

(2) 각 출력단 스케줄러는  $k$ 개의 입력단 각각에 대해 입력단 예약 테이블과 출력단 예약 테이블을 참조하여, 입·출력단 양쪽에 예약되어 있지 않은 시점들 중 가장 빠른 시점을 예약해서 해당 입력단으로 보내고, 출력단 예약 테이블을 변경한다. 이 과정을  $k$ 개의 입력단에 대해 순차적으로 반복 수행한다.

(3) 각 입력단은 출력단 스케줄러로부터 예약받은 시점들에 대해, 전송 테이블에 예약셀이 저장되어 있는 입력버퍼의 주소와 목적 출력단 정보를 저장하고 입력단 예약 테이블을 변경한다.

**3. 전송(Transmission) 단계 :** 각 입력단은 전송 테이블을 참조하여 현 시점에 예약되어 있는 셀을 목적 출력단으로 전송한다.

입력단 예약 테이블 ( $1 \leq i \leq N$ )는 시점  $t$ 에서 본 입력단  $i$ 의 전송 예약상태를 나타내며, 입력단  $i$ 가 시점  $t + k$  ( $0 \leq k < M$ )에 셀을 전송하도록 예약되어 있으면  $I_i[k] = 1$ 이고, 아직 예약되어 있지 않으면  $I_i[k] = 0$ 이다. 마찬가지로, 출력단 예약 테이블 ( $1 \leq j \leq N$ )는 시점  $t$ 에서 본 출력단  $j$ 의 전송 예약상태를 나타낸다.  $ST_i^t$ 는 입력단  $i$ 의 전송 테이블이다.  $ST_i^t[t]$ 에는 시점  $t$ 에 전송할 셀이 저장되어 있는 입력버퍼의 주소와 목적 출력단 정보가 저장되어 있다.

시점  $t$ 에 각 입력단  $i$  ( $1 \leq i \leq N$ )는 예약셀의 목적 출력단 스케줄러들로 입력단 예약 테이블  $I_i^t$ 와 함께 요구신호를 보낸다. 각 출력단 스케줄러  $j$  ( $1 \leq j \leq N$ )는 요구신호를 보낸 입력단들 중  $k$  ( $1 \leq k \leq N$ )개를 선택한다. 그 중 하나가 입력단  $i$ 라면, 입·출력단 예약 테이블  $I_i^t$ 와  $O_j^t$ 를 참조하여 예약시점  $t'(i,j)$ 를 입력단  $i$ 로 보내고,  $O_j^t[t'(i,j)]$ 를 1로 변경한다. 이 때,  $t'(i,j)$ 는 입력단  $i$ 와 출력단 스케줄러  $j$ 에서 아직 예약되어 있지 않은 가장 빠른 시점으로, 식 (1)과 같이 나타낼 수 있다.  $k$ 개의 입력단에 대해 이와 같은 과정을 순차적으로 수행한다. 각 입력단에 대한 예약을 순차적으로 수행하는 이유는 출력단이 특정 입력단에 대해 예약해 준 시점을 다른 입력단에 다시 예약해 주지 않기 위해서이다. 각 입력단  $i$ 는 예약셀의 목적 출력단 스케줄러들로부터 예약시점을 받는다. 만일 출력단 스케줄러  $j$ 로부터 예약시점  $t'(i,j)$ 를 받았을 경우, 전송 테이블  $ST_i^t[t'(i,j)]$ 에 예약셀의 버퍼의 주소와  $j$ 를 저장한다. 그리고  $I_i^t[t'(i,j)]$ 를 1로 변경한다. 그리고, 다음 시점  $t+1$ 을 위해 입·출력단 예약 테이블을 변경시킨다. 즉, 입력단 예약 테이블의 각 엔트리  $I_i^{t+1}[k]$  ( $0 \leq k < M - 1$ )는  $I_i^t[k+1]$ 의 값으로,  $I_i^{t+1}[M-1]$ 은 0으로 초기화되며, 마찬가지로, 출력단 예약 테이블의 엔트리  $O_j^{t+1}[k]$  ( $0 \leq k < M - 1$ )도  $O_j^t[k+1]$ 의 값으로,  $O_j^{t+1}[M-1]$ 은 0으로 초기화된다. 마지막으로, 전송 테이블  $ST_i^t[0]$ 를 참조하여 현 시점  $t$ 에 전송할 셀이 저장되어 있는 입력버퍼의 주소와 출력단 정보를 알아내어 셀을 전송한다.

$$t'(i,j) = \min\{k : I_i^t[k] = 0 \text{ and } O_j^t[k] = 0, 0 \leq k < M\} \quad (1)$$

[그림 3]은 목적 출력단의 개수가 하나 이상인 멀티캐스트 셀에 대한 CFTR의 예약 과정의 예를 든 것이다. 위의 예에서 1번 입력단의 셀 'a'는 1,2번 출력단 스케줄러로부터 동일한 시점  $t+2$ 를 예약받았으므로, 시점  $t+2$ 에 1,2번 출력단으로 복사, 전송된다. 그러나, 2번 입력단의 셀 'b'는 2번 출력단 스케줄러로부터는  $t+3$ 를, 3번 출력단 스케줄러로부터는  $t+2$ 를 예약받는다. 따라서 셀 'b'는 두 셀 타임에 나누어 2,3번 출력단으로 전송된다. 이 예에서, CFTR 기법에서는 예약 충돌이 발생하지 않음을 알 수 있다.

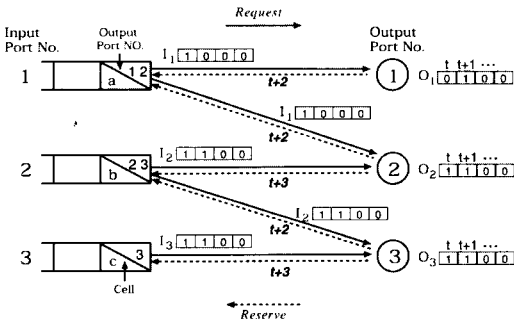


그림 3 CFTR의 예약 과정의 예

3.1 CFTR 기법에 의한 멀티캐스트 셀 스케줄러 설계

[그림 4]는 입력버퍼 모듈의 블럭 다이어그램을 보여 준다. 입력버퍼 모듈로 셀이 들어오면 셀은 입력버퍼(RAM1)에 저장된다. 이 때 RAM3는 셀이 저장될 입력버퍼의 빈 메모리 주소를 제공한다. 셀이 저장된 입력버퍼의 주소와 셀의 목적 출력단 주소는 예약큐(Reservation Queue)에 들어가서 전송시점이 예약되기를 기다린다. 매 타임 슬롯의 처음에 요구신호 생성기(Request Signal Generator)는 셀 스케줄러(Contention Resolution Module)로 예약큐의 선두에 있는 셀의 목적 출력단 정보와 현 입력단의 예약 상태(Reservation State Tag)를 보낸다. 셀 스케줄러에 의해 예약된 전송시점은 전송테이블 제어부(Send Table Controller)로 보내지고, 전송테이블 제어부는 이 정보를 전송 테이블에 저장한다. 전송 테이블은 각 타임슬롯에 전송될 셀의 입력버퍼 주소와 목적출력단, 예약상태 태그(Reservation State Tag)를 저장하고 있다. 전송테이블 제어부는 예약된 전송시점에 해당하는 전송 테이블 주소에 현재 예약큐의 선두에 있는 셀의 입력버퍼 주소와 목적출력단 주소를 저장하고 예약상태를 1로 바꾼다. 그 후, 전송 테이블을 참조하여 현 타임슬롯에 전송할 셀의 입력버퍼 주소와 목적 출력단 주소를 알아

낸 다음, 입력버퍼의 해당 주소에 있는 셀을 목적하는 출력단으로 보낸다. 셀 전송 후 전송 테이블의 해당 엔트리의 예약상태 태그를 0으로 바꾼다. 매 타임 슬롯마다 위에서 설명한 셀의 전송시점 예약과 출력단으로의 실제 전송이 이루어진다. CFTR 기법에서의 입력버퍼 모듈은 Obara의 방식에 비해 전송 테이블의 각 엔트리에 1 비트의 예약상태 태그가 추가되며, 요구신호 생성기가 셀 스케줄러로 예약요구신호를 보낼 때, 이 예약상태 태그들을 함께 보내는 점이 다르다. 이로 인한 전송 테이블의 메모리 요구량 증가는 거의 없으며, 예약상태 정보를 셀 스케줄러로 보내는 것은 고속 ATM 스위치에서도 충분히 가능하다.

셀 스케줄러 모듈은 입력버퍼 모듈로부터 목적 출력단의 정보를 받아서 스케줄된 전송 시점을 보내준다. 셀 스케줄러는 독립적이고 병렬적인 수행을 하는 여러 개의 출력단 스케줄러로 이루어져 있어서, 동작이 간단하며 빠르다. 각 출력단 스케줄러는 자신에게 요구신호를 보낸 입력단 중  $k$ 개를 선택하여 전송시점을 예약해 주는 역할을 한다. CFTR 기법에서 출력단 스케줄러가 갖는 특징은 출력단의 예약상태를 관리하며, 입력단 예약상태와 비교하여 전송시점을 예약한다는 것에 있다. 이것은 간단한 하드웨어 구현으로 가능하며 동작도 단순하여 고속의 ATM 스위치에서 사용가능하다.

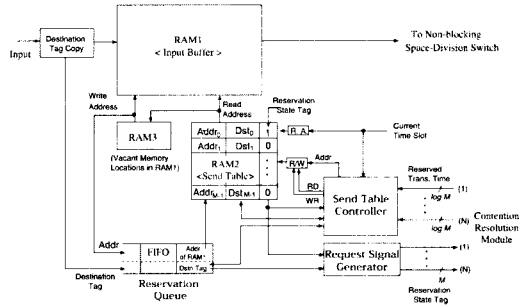


그림 4 입력버퍼 모듈의 블럭 다이어그램

3.2 CFTR 기법의 성질

CFTR 기법에서는 매 시점의 예약 과정이 각 출력단 스케줄러에서 독립적, 병렬적으로 이루어지고 반복회수가 적으므로, 고속 트래픽의 처리가 가능하다. 또한, non-FIFO 입력버퍼를 이용해서 셀을 전송하며 예약 충돌이 발생하지 않도록 전송시점을 스케줄링함으로써 기존의 FIFO 방식의 멀티캐스트 셀 스케줄링 알고리즘에 비해 높은 처리율을 얻을 수 있다. CFTR 기법은 각 출력단 스케줄러의 전송시점 예약과정이 간단하며, 한 시

점에 적은 수의 예약과정 반복회수로도 충분한 예약이 이루어져 높은 처리율을 가질 수 있다. 따라서 고속 트래픽의 처리가 가능하다. 입·출력단 예약 테이블의 크기가 M일 경우, 각 입력단은 현 시점으로부터 최대 M-1 이후의 시점을 예약받을 수 있다. M이 클수록 높은 처리율을 가지게 되나, 너무 크지 않은 M값에서 높은 처리율을 낼 수 있음을 다음 절의 시뮬레이션에서 보인다. 따라서, 약간의 하드웨어의 첨가로 큰 성능의 향상을 가져올 수 있다. CFTR 기법에서는 동일 출력단으로 향하는 같은 입력단의 셀들은 입력단에 도착하는 순으로 출력단으로 전송된다.

#### 4. 성능 평가

본 절에서는 CFTR 기법의 성능 평가를 위해서 기존의 멀티캐스트 셀 스케줄링 알고리즘인 mRRM [6], TATRA와 WBA [5], 그리고 멀티캐스트 처리를 할 수 있도록 간단히 확장한 Obara 방식과 비교해 본다. 이 때 멀티캐스트 셀의 목적 출력단 개수(fanout)는 4로 가정한다. 성능 평가를 위한 척도로는 입력 버퍼에서의 셀의 평균 대기시간(mean waiting time)과 정규화된 처리율(normalized throughput)을 사용한다. 사용된 트래픽 패턴으로는 입력셀의 도착률에 따라서 랜덤(random) 패턴과 버스티(bursty) 패턴 각각에 대해, 각 입력셀이 요구하는 출력단의 분포에 따라 균일(uniform) 패턴과 비균일(nonuniform) 패턴 각각에 대해 시뮬레이션을 수행하였다. 시뮬레이션은 16 x 16 스위치를 가정하였으며, 스위치의 한 셀 타임(time slot)을 단위시간으로 하였다. 시뮬레이터는 C 언어로 작성하였으며, 40000 단위시간 동안을 조사한 것으로 처음의 10000 단위시간은 스위치의 초기상태의 영향을 제거하기 위하여 결과에서 제외시킨다.

[그림 5]는 멀티캐스트 셀이 랜덤 패턴으로 입력단으로 들어올 때 셀의 평균 대기시간을 나타낸 것이다. 또한, [그림 6]은 멀티캐스트 셀이 평균 버스트 길이( $L_b$ )가 5인 버스티 패턴으로 입력될 때 셀의 평균 대기시간을 나타낸 것이다. 두 경우 모두 Obara 방식과 FIFO 방식의 mRRM, TATRA, WBA에 비해 CFTR 기법이 매우 좋은 성능을 보임을 알 수 있다. 이는 CFTR 방식에서는 Obara 방식에서 발생했던 예약 충돌이 일어나지 않으며 non-FIFO 방식의 전송으로 인해 높은 처리율을 가질 수 있기 때문이다. CFTR 기법은 랜덤 패턴에서 입·출력단 예약 테이블의 크기(M)가 16일 때, 95%가 넘는 처리율을 가지며, M=32일 때 98%의 처리율을 갖는다. M=32인 버스티 패턴에서는 85%가 넘는 처리율을 갖는다.

[그림 7]은 평균 버스트 길이  $L_b$ 에 따른 정규화된 처

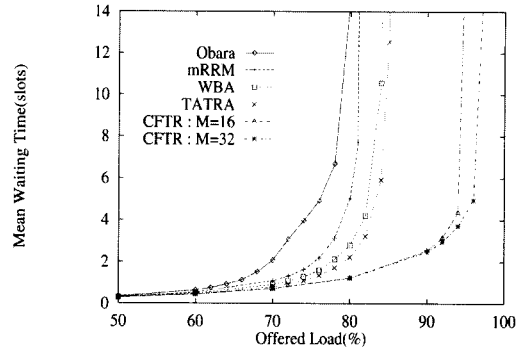


그림 5 멀티캐스트, 랜덤 패턴 하에서 제공부하에 따른 평균대기시간

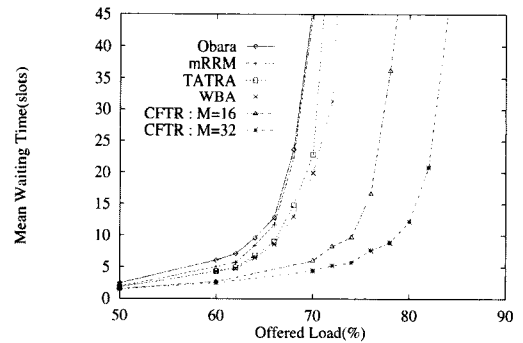


그림 6 멀티캐스트, 버스티 패턴( $L_b = 5$ ) 하에서 제공부하에 따른 평균대기시간

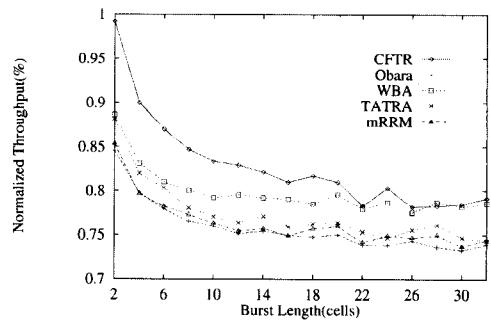


그림 7 멀티캐스트, 버스티 패턴 하에서  $L_b$ 에 따른 처리율 ( $L_d = 90\%$ )

리율을 조사해 본 것이다. 그림에서 알 수 있듯이 평균 버스트 길이가 길어질수록 각 셀 스케줄링 알고리즘의 처리율의 차이가 줄어든다. 이것은 non-FIFO 입력버퍼를 사용하여 셀 스케줄링을 하더라도

버스트의 길이가 길어질수록 전송되지 못하고 대기하고 있는 버스트로 인한 봉쇄 현상이 두드러지기 때문이다.

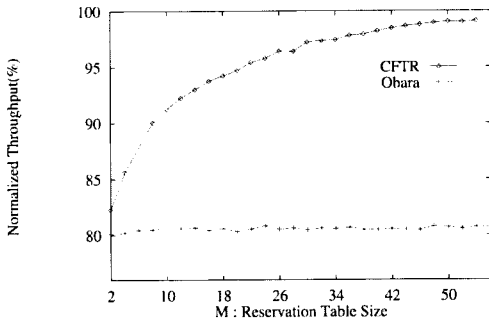


그림 8 멀티캐스트, 랜덤 패턴 하에서 M에 따른 처리율( $L_d = 100\%$ )

[그림 8]은 랜덤 트래픽 패턴에서 CFTR 기법과 Obara의 유니캐스트 알고리즘을 확장한 멀티캐스트 알고리즘에 대해, 제공부하  $L_d=100\%$ 일 때, M에 따른 처리율을 알아본 것이다. 그림에서 볼 수 있듯이 CFTR 기법은 예약 테이블의 크기 M이 증가함에 따라 처리율이 증가한다. 이는 예약 테이블의 크기가 클수록 더 많은 예약이 이루어져서 예약을 받지 못한 셀이 그 뒤에 대기하고 있는 셀의 예약과 전송을 방해하는 현상을 줄여주기 때문이다. CFTR 기법은 거의 100%의 정규화된 처리율(normalized throughput)을 가지는 데 비해, Obara의 방식은 M과 무관하게 80%의 정규화된 처리율에 머무는 것을 볼 수 있다. 이는 Obara의 방식에서 각 출력단 스케줄러 간에 예약 충돌이 발생하기 때문이다. 또한, 약 M=32 이상에서는 처리율의 변화가 거의 없으므로, 약 32 비트의 예약 테이블로 좋은 성능을 얻을 수 있으며 따라서 추가 하드웨어의 비용이 크지 않다.

[그림 9]는 각 출력단 스케줄러가 한 시점동안 예약을 해 주는 입력단 개수 L에 따른 정규화된 처리율을 살펴본 것이다. 제공 부하가 100%인 랜덤 트래픽과 제공 부하가 90%인 버스티 트래픽에 대해 조사하였다. 두 경우 모두 예약 테이블의 크기가 16인 경우 2이상인 L에 대해서 거의 동일한 처리율을 갖는 것을 볼 수 있다. 이것은

매 시점마다 예약과정이 수행되므로, 한 시점에 두 번의 예약과정으로도 충분한 예약이 이루어지기 때문이다. 따

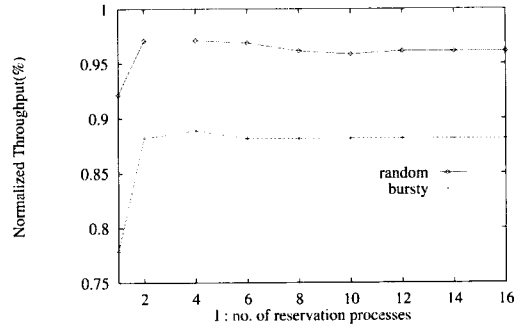


그림 9 L에 따른 처리율

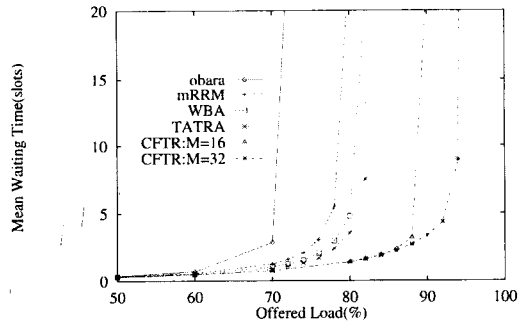


그림 10 제공 부하(%)에 따른 평균대기시간:  $\alpha = 0.1$

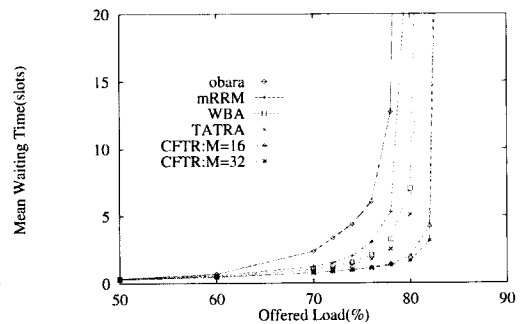


그림 11 제공 부하(%)에 따른 평균대기시간:  $\alpha = 0.1$

라서 한 시점에 예약과정 수행 회수가 적으므로 고속의 트래픽 처리가 가능하다.

[그림 10]과 [그림 11]은 각각 hot ratio  $a$ 가 0.1과 0.3인 비균일(nonuniform) 트래픽 패턴에 대해 제공 부하에 따른 평균대기시간을 측정해 본 것이다. 스위치의 출력단 개수가 16이고 각 셀의 목적 출력단 개수가 4이므로 균일 트래픽의 hot ratio는 0.25라 할 수 있다. hot(cold) node를 1번 출력단이라 하자. 즉  $a = 0.1$ 인 경우는 멀티캐스트 셀의 목적 출력단 중 1번 출력단이 있을 확률이 0.1인 경우, 즉 1번 출력단으로 전송되고자 하는 경향성이 다른 출력단에 비해 작은 경우를 나타낸다. 반대로,  $a = 0.3$ 인 경우는 멀티캐스트 셀의 목적 출력단 중 1번 출력단이 있을 확률이 0.3인 경우, 즉 1번 출력단으로 전송되고자 하는 경향성이 다른 출력단에 비해 큰 경우를 나타낸다. 두 경우 모두 균일(uniform) 트래픽이 경우에 비해 각 멀티캐스트 셀 스케줄링 알고리즘의 처리율이 전체적으로 낮아지는 것을 볼 수 있다. 이는  $a = 0.1$ 인 경우 상대적으로 다른 출력단에 트래픽이 몰리기 때문이며,  $a = 0.3$ 인 경우 1번 출력단으로 트래픽이 몰리기 때문에 1번 출력단으로 전송되지 못한 셀이 입력버퍼에 남아 있음으로써 다른 셀의 전송을 지연시키기 때문이다. 두 경우 모두 CFTR 기법이 균일 트래픽에서처럼 다른 기법에 비해 처리율이 높은 것을 볼 수 있다. 이로써 CFTR 기법이 비균일 트래픽의 처리에도 다른 셀 스케줄링 기법보다 좋은 성능을 보임을 알 수 있다. 본 연구에서는 비균일 트래픽의 처리에 대해 스위치 크기가 32와 64인 경우에 대해서도 시뮬레이션을 수행하였다. 시뮬레이션 결과 각 스케줄링 기법은 스위치 크기가 16인 경우와 유사한 성능을 보였으며, 스위치 크기에 따른 성능의 차이는 거의 없음을 알 수 있었다.

CFTR 기법은 유니캐스트 셀만 입력되는 경우에도 매우 좋은 성능을 보인다. 이 경우 기존의 유니캐스트 셀 스케줄링 알고리즘인 Obara의 방식과 SLIP 방식, 그리고 최적으로 알려져 있는 출력버퍼 방식과 비교한다. 이때 SLIP 방식에서는 입력단에 출력단별로 각각의 버퍼를 두는 것은 실제 구현이 어려우므로 입력버퍼의 선두로부터 16개의 셀들이 요구신호를 보낸다고 가정한다. [그림 12]는 랜덤 트래픽 하에서 제공된 부하(offered load)에 따른 평균대기시간(mean waiting time)을 조사한 것이다. 시뮬레이션 결과로부터 CFTR 기법이 Obara의 알고리즘과 SLIP 알고리즘에 비해 매우 좋은 성능을 보일 뿐 아니라, 출력버퍼형 스위치에 가까운 성능을 보임을 알 수 있다.

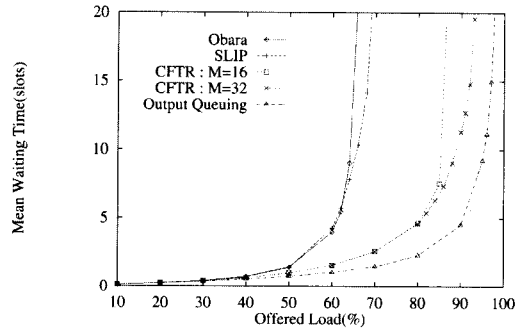


그림 12 유니캐스트, 랜덤 패턴 하에서 제공부하에 따른 평균대기시간

### 5. 결론

본 논문에서는 입력버퍼 ATM 스위치에서의 멀티캐스트 셀 스케줄링 알고리즘인 무충돌 예약(Collision-Free Time Reservation : CFTR) 기법을 제안한다. CFTR 기법에서는 입력단, 출력단의 예약 상태를 나타내는 입·출력단 예약 테이블을 사용해 각 출력단으로의 전송시점을 예약함으로써 입·출력단에서의 예약 충돌을 방지하여 높은 처리율을 제공한다. 각 출력단으로의 전송시점 예약은 독립적, 병렬적 수행이 가능하며, 적은 회수의 예약과정으로도 높은 처리율을 얻을 수 있으므로 고속 스케줄링이 가능하다.

본 논문에서는 CFTR 기법의 성능 평가를 위해 시뮬레이션을 통해 랜덤, 버스티 트래픽 패턴에 대한 평균 대기시간과 처리율을 기존 알고리즘과 비교한다. 또한, 예약 테이블의 크기와 예약과정 반복회수에 따른 처리율을 측정함으로써 셀 스케줄러의 하드웨어 복잡도와 고속수행 가능성을 검토하였다. CFTR 기법은 유니캐스트 입력 트래픽에서 기존의 유니캐스트 셀 스케줄링 알고리즘인 SLIP, Obara의 방식에 비해 높은 처리율을 가지며 멀티캐스트 입력 트래픽에 대해서도 기존의 멀티캐스트 셀 스케줄링 알고리즘인 mRRM, TATRA, WBA에 비해 높은 처리율을 보인다. 이를 위해서는 각 출력단 스케줄러당 작은 크기의 예약 테이블을 두어 적은 회수의 예약과정을 수행하면 되므로, 약간의 부과적인 하드웨어와 동작과정으로 큰 성능의 향상을 얻을 수 있으며 고속수행이 가능하다. 앞으로 이루어질 연구는 CFTR 기법에 의한 셀 스케줄러를 구현함으로써 구현 복잡도와 수행속도를 알아보고, 실제 트래픽에 대한 성능을 측정하는 것이다.

## 참고 문헌

- [1] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," IEEE Trans. Commun., Vol. COM-35, No. 12, pp. 1347-1356, Dec. 1987.
- [2] W. Chen, P. Sheu, and J. Yu, "Time Slot Assignment in TDM Multicast Switching Systems," Proc. of IEEE INFOCOM '91, pp. 1296-1305
- [3] W. Chen, P. Sheu, and J. Yu, "Time Slot Assignment in TDM Multicast Switching Systems," IEEE Trans. on Commun., Vol. 42, No. 1, pp. 149-165, Jan. 1994,
- [4] C. Kim and T. T. Lee, "Call Scheduling Algorithms in a Multicast Switch," IEEE Trans. Commun., Vol. 40, No. 3, pp. 625-635, March, 1992
- [5] B. Prabhakar, N. McKeown and R. Ahuja, "Multicast Scheduling for Input-Queued Switches," Submitted to IEEE JSAC, 1996
- [6] N. McKeown and B. Prabhakar, "Scheduling Multicast Cells in an Input-Queued Switch," Proc. of IEEE INFOCOM '96, pp. 271-278, 1996
- [7] H. Obara, "Optimum Architecture for Input Queueing ATM Switches," Electronics Letters, Vol. 27, No. 7, pp. 555-557, March, 1991

## 권 보 섭



1983년 경북대학교 전자공학과 학사.  
1990년 충남대학교 전자공학과 석사.  
1997년 한국과학기술원 전산학과 박사.  
1983년 ~ 1985년 LG전자 연구원. 1985년 ~ 1998년 한국전자통신연구원 선임 연구원. 1998년 ~ 현재 안동대학교 컴퓨터교육과 교수. 관심분야는 상호연결 네트워크, 병렬 컴퓨터 구조, 컴퓨터 성능평가임

## 윤 현 수

정보과학회논문지: 정보통신  
제 27 권 제 2 호 참조

## 조 정 완



1964년 서울대학교 공과대학 전자공학과 졸업. 1968년 와이오밍 주립대학 전기공학과 석사학위 취득. 1973년 노스웨스턴 대학 박사학위 취득. 1968년 ~ 1973년 미국 IBM 연구원. 1982년 ~ 1984년 삼성전자 고문. 1985년 ~ 1988년 금성소프트웨어 대표, 고문. 1973년 ~ 현재 과학기술원 전산학과 교수. 관심분야는 computer architecture, parallel processing, knowledge systems, man-machine interfaces임

## 조 민 회



1995년 한국과학기술원 전산학과 학사.  
1997년 한국과학기술원 전산학과 석사.  
1997년 ~ 현재 한국과학기술원 전자전산학과 전산학전공 박사과정 재학 중. 관심분야는 ATM 스위치 구조, 네트워크, 이동통신망, QoS 등

## 송 효 정



1992년 연세대학교 전산학과 학사.  
1994년 한국과학기술원 전산학과 석사.  
1998년 한국과학기술원 전산학과 박사.  
1998년 ~ 현재 University of California San Diego. 관심분야는 병렬 처리, 워홀라우팅임