

Non-FIFO 메모리 구조를 사용한 입력버퍼형 스위치에서 개선된 DBP 윈도우 기법

김 훈, 박 성 헌, 박 광 채
조선대학교 전자정보통신공학부
광주광역시 동구 서석동 375번지

An Improved DBP Window Policy in the Input Buffer Switch Using Non-FIFO Memory Structure

Hoon-Kim, Sung-Hun Park, Kwang-Chae Park
School of Electronics, Information and Communications Eng. Chosun Univ.
375 Seoseck-dong, Dong-gu, Kwangju 501-759, Korea
E-mail : candelab@chollian.net

Abstract : In the Input Buffer Switch using the initial stage FIFO memory structure, It has pointed the Throughput limitation to the percent of 58.6 due to HOL(Head of Line) blocking in the DBP(Dedicated Buffer with Pointer) method. During that time, To overcome these problems, The prior papers have proposed the complicated Arbitration algorithms and Non-FIFO memory structures. and These showed the improved Throughput. But, Now, To design high speed ATM Switch which need to the tens of Giga bit/s or the tens of Tera bit/s. It has more difficulty in proceeding the priority of majority and the complicated Cell Scheduling, because of the problem in operating the control speed of the ratio of N to scanning each port and scheduling the Cell

In this paper, To overcome these problems, We could show more the improved performance than the existing DBP Window policy to design high speed ATM Switch

I. 서 론

최근의 정보통신 기술은 기존의 단순한 메시지의 전송에서 발전되어 다차원적으로 부호화된 멀티미디어 데이터를 디지털망에 통합하여 전송하는 광대역 종합정보통신망(B-ISDN)으로 구현되고 있다. 이 B-ISDN은 멀티미디어 데이터를 하나의 디지털망에 통합하는 방식이며, 미래의 고속 패킷 스위칭 시스템을 구현하기 위하여 ATM(Asynchronous Transfer Mode) 스위치를 채택하였다. 이 ATM스위치 시스템은 버퍼의 위치에 따

라서 출력버퍼형 스위치 구조, 공통버퍼형 스위치 구조, 입력버퍼형 스위치 구조 등으로 구분할 수 있다. 여기서 공통버퍼형 스위치 구조는 버퍼를 공동으로 사용하므로 특정 셀 손실률을 달성하기 위해 요구되는 버퍼의 양을 최소화 할 수 있다는 것과 큰 버스트를 흡수하므로 트래픽을 동적으로 수용할 수 있는 융통성이 크다는 장점을 가지고 있으나 이를 초고속 ATM스위치로 구현하기 위해서는 제어부가 1/NV의 속도로 셀을 읽고 쓰고 해야 하므로 포트 속도보다 N배 빠른 속도로 동작해야 하는 문제점으로 인해 전체 Throughput이 메모리의 액세스 시간에 의해 제한적이라는 문제점을 가진다. 또한 대형 ATM스위치를 구현하기 위하여 출력버퍼형 스위치 구조와 공통버퍼형 스위치 구조는 포트의 증가에 따라 버퍼링 공간이 지수적으로 증가하는 문제점으로 인해 하드웨어의 크기가 너무 커진다는 문제점을 가지고 있다. 그러나 입력버퍼형 스위치 구조는 58.6% Throughput을 가지는 FIFO 메모리 사용에도 불구하고 FIFO 메모리의 속도가 입력링크 속도와 같은 V이면 되고, 단순한 구조의 특성으로 인해 대형 스위치화 하기에 다른 방식에 비하여 우월한 특성을 가지고 있다. 그러나 초기의 입력버퍼형 스위치 구조 중에서 FIFO 메모리 구조를 사용하는 입력버퍼형 스위치에서는 HOL(Head of Line) 블로킹으로 인한 58.6% Throughput의 한계를 나타내었고, 그 동안 이를 극복하기 위해 복잡한 Arbitration 알고리즘들과 Non-FIFO 메모리 구조들이 제안되었다. 그러나 이들은 향상된 Throughput을 나타내어 주었지만 다른 면에서 이들 또한 각 포트를 Scanning하고 셀 스케줄링을 하기 위해서 제어부의 속도가 N배의 속도로 동작하여야 하는 문제점을 가지게 되었다. 이는 수십 Giga bit/s 또는 수십

Tera bit/s를 원하는 초고속 스위치를 구현하기 위한 다수의 우선순위와 복잡한 셀 스케줄링을 처리해야 하는 제이계에 어려움을 주었다. 본 논문에서는 이를 극복하고, 초고속 ATM 스위치를 구현하기 위해 Non-FIFO 메모리를 사용한 입력비퍼형 스위치 구조에서 공동비퍼를 사용하여 기존의 DBP 윈도우 기법보다 향상된 성능을 볼 수 있었다.

II. 윈도우 기법

전형적인 입력비퍼 스위치의 구조는 그림 1과 같다.^[1]

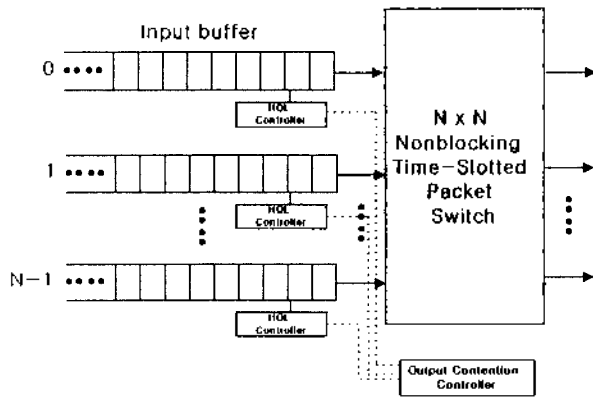


그림 1. 입력비퍼형 스위치 구조

이 방식의 스위치는 FIFO 방식의 입력비퍼와 HOL 제어부를 포함하는 입력비퍼부, CR(Contention resolution block)부, 년 블로킹 스위치 부분의 서로 독립된 세부분으로 구성되어 있다. 고정 길이의 셀은 각 독립된 입력비퍼에 도착하고 저장된다. 이때 입력, 저장된 각 입력셀의 VPI는 HOL 제어부를 경위하여 CR에 저장된다. 여기서 CR은 저장된 VPI를 이용하여 출력포트에서 셀의 충돌을 방지시키는 스케줄링을 한다. 이 스케줄링된 셀은 FIFO로부터 년 블로킹 스위치부로 전달된다. 그리고 년 블로킹 스위치부는 전달되어온 셀들의 목적 출력포트로 라우팅을 한다. 여기서 입력비퍼와 스위치의 속도는 각 포트, 즉 링크속도와 동일하다. 따라서 이 방식은 출력비퍼형 방식과 공동비퍼형 방식들보다 고속으로 동작할 수 있는 이점을 준다. 그러나 위의 방식은 다른 입력으로 동시에 들어오는 입력 셀이 동시에 같은 출력포트로 출력되기를 원할 수 있다. 따라서 이런 형태의 스위치에서는 다른 입력포트들에서 같은 출력포트로 동시에 향하는 셀들의 충돌을 회피하기 위한 Arbitration을 위한 CR 알고리즘이 필요하게 된다. 이 알고리즘은 CR과 HOL 제어부에 속해 실현된다.

DBP 방식의 문제점인 HOL 블로킹 발생시 블로킹이 발생한 입력포트에서만 빈 출력포트로 출력되는 입력셀만은 있으므로 해서 다른 포트에 이 출력포트를 원하는 입력 셀이 있어도 서비스를 할 수 없는 경우가 발생할 수 있다. 이는 DBP 방식에서의 Throughput의 한계를 만드는 문제점이다. 이를 극복하기 위하여 공동비퍼를 사용하여 다른 입력포트에 있는 셀이 서비스를 받을

수 있게 하여 Throughput의 한계를 극복하고자 한다. Reservation Address란 RAM2에서 다음에 서비스를 받을 주소를 미리 할당하여 놓고, 비스트 셀의 순서를 보장하는 것이다.

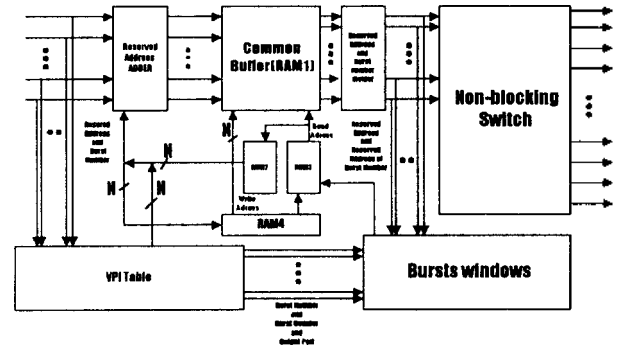


그림 2. 전체 블록도

DBP(Dedicated Buffer with Pointer) 방식의 하드웨어 구성은 그림 3과 같다.^[3]

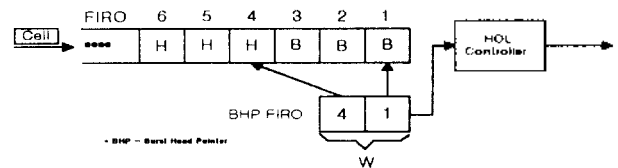


그림 3. DBP 기법을 위한 윈도우 입력비퍼 구현

DBP 방식의 문제점은 비스트 트래픽에서 우수한 성능을 보이던 각 입력포트 당 각각 하나의 윈도우 FIFO(BHP)와 셀을 저장하는 FIFO를 가지고 있다. 이는 HOL 블로킹 발생시 블로킹이 발생한 입력포트만 BHP를 조사하여 빈 출력포트를 원하는 셀이 있을 경우에는 이 셀을 서비스하지만 만약 이 빈 출력포트를 원하는 셀이 없다면 다른 입력포트에 이 빈 출력포트로 원하는 셀이 있다고 하여도 이 출력포트는 빈 출력포트로 서비스되지 않고 남은 포트에 남게 된다. 이는 DBP의 Throughput의 한계를 만드는 하나의 원인이 된다. 그렇다면 이 문제점을 극복, 즉 다른 입력포트에 빈 출력포트를 원하는 셀이 있다면 이를 서비스하여 Throughput를 향상시킨다. 이를 실현하기 위해 입력포트에서 BHP를 확장하는 방법이 있을 수 있다. 그러나 여기서는 공동비퍼를 입력포트 측에 셀을 저장하는 FIFO 대신 사용하여 모든 입력 셀이 이 공동비퍼에 저장되게 하고서 복수 제어를 한다. 이 공동비퍼의 특성 상 어느 주소에 있는 셀이나 원하는 출력포트로 출력시킬 수 있게 된다. 따라서 어느 입력포트에 있는 셀은 서비스를 할 수 있게 된다. 이를 위하여 그림 2에 개별 블록도로 나타내었다. 하지만 이렇게 공동비퍼를 사용한 경우 BHP FIFO는 단지 셀을 저장하는 FIFO에

버스트 트래픽의 헤더 셀 순시안을 보존하므로 인해서 버스트한 트래픽의 다음 헤더 셀이 공동버퍼 어느 주소에 저장되어 있는지 알수 없게 된다. 따라서 BHP FIFO에 버스트 트래픽 셀의 헤더 순서가 아닌 공동 버퍼의 주소를 사용하여야 한다. 또한 헤더 셀의 주소만을 기억하게 되므로 버스트 트래픽 셀의 주소를 가진다고 하더라도 이는 다만 버스트 트래픽 셀의 주소 정보 뿐이므로 버스트한 셀의 나머지는 공동버퍼의 어느 주소에 저장되어 있는지 알수 없는 상황이 발생하게 된다. 따라서 BHP에 저장되어 있는 주소는 항상 서비스된 셀이 주소를 가지고 있어야 한다. 이를 실현하기 위한 방법으로 입력 셀이 공동버퍼에 저장될 때 쓰기 주소로 미리 예약되어 있는 주소를 주며, 이는 BHP에도 항상 다음에 서비스될 주소의 데이터로 주어지게 된다. 이는 하나의 셀이 들어오면 미리 예약된 주소에 지장을 주며 다음에 들어올 입력 셀을 위하여 하나의 주소만을 예약하며 이는 저장되는 셀과 같이 주어진다. 여기서 미리 예약된 주소는 그림 1의 RAM에 저장되어 있다가 쓰기 주소로 주어진다. 이렇게 하면 이 예약된 주소는 버스트한 셀을 서로 링크하게 되어 버스트 트래픽의 연계를 나타낼 수 있는 논리적인 FIFO 버퍼를 구성할 수 있게 된다. 따라서 BHP에서 버스트 트래픽의 한 셀이 서비스가 되면 그림 2에서 Reserved 주소를 분리하는 모듈에서 예약된 주소를 분리하여 해당하는 BHP에 적어서 다음에 서비스될 셀의 위치를 알린다.

또한 BHP 모듈에서는 분리되어 들어오는 예약된 주소만으로는 어느 버스트인지를 알수 없으므로 이를 식별하기 위해 부가적인 데이터가 필요하게 된다. 이는 각 버스트 트래픽이 들어오면 이 버스트에 번호를 주어서 이를 식별할 수 있게 한다. 또한 각 버스트의 폭주 해결을 위해서는 원하는 출력포트 번호가 있어야 하므로 이 또한 폭주 해결을 위하여 부가하며, 하나의 버스트 트래픽이 서비스가 되면 이 버스트 이전에 서비스되어 남아 있는 셀의 카운터 값에서 1만큼 감한다.

그림 3에서는 부가된 각 데이터의 역할을 설명하였으나, 이는 버스트 윈도우에 각 입력포트 별로 그림 5와 같이 구성한다.

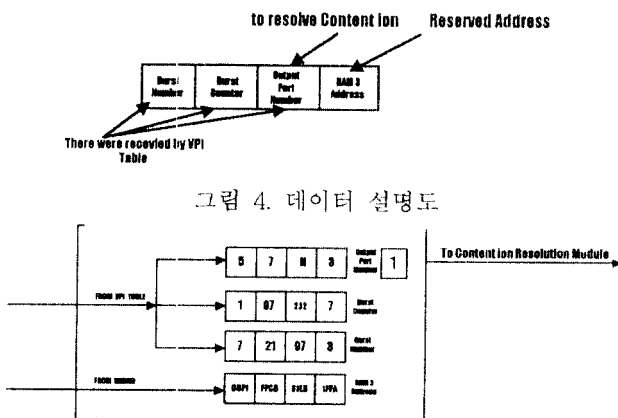


그림 4. 데이터 설명도

그림 5. 버스트 윈도우 내의 한 포트 분포도

본 논문에서 제안한 방식은 DBP의 Throughput 한계를 극복하고, First-In-First 서비스를 실현할 수 있어 공정성을 부과할 수 있다. 공동버퍼를 사용하게 되므로써 공동버퍼의 문제점인 입력포트의 속도에 N (입력포트 수)배하는 속도로 동작해야 하는 문제점을 가지게 된다. 따라서 이는 고속/대형 ATM스위치를 구성하는데 문제점이 된다. 그렇다면 여기서 적당한 Throughput을 원하며, 고속/대형 ATM스위치를 원하게 된다면 입력포트를 몇 개의 그룹으로 나누어서 속도를 감당한 실계를 하였다.

III. 시뮬레이션 모델 및 결과

두가지 윈도우 방식의 실험을 비교하기 위해서 컴퓨터 시뮬레이션을 이용하고, 일정한 기하학적 버스트 트래픽 모델 상태에서 작동하는 $N \times N$ 번 블럭킹 입력버퍼형 스위치의 Throughput(즉, 100% 입력 부하)을 알기 위해 버스트사이의 시간 간격이 0이라고 가정하였다¹⁴⁾.

공정한 비교를 위해 각각의 입력(FIFO와 FIRO버퍼를 포함해서)들에 대한 버퍼의 크기는 256 셀들로 모든 선계에서 같다. DBP 방법을 위해서 각 입력은 256 셀 크기의 FIRO 대기행렬을 가지고 있으며(BHF FIRO 대기행렬은 포함되지 않는다). DSB 방법을 위해서 각 입력은 각각 256/W 크기의 W FIFO 대기 행렬들을 가지고 있다. 여기에 나타나 있는 시뮬레이션들은 $N=32$ 에 대한 것이다. 각 데이터의 점은 10^6 이상의 셀들이 모든 입력포트들에 걸쳐서 수집된 것이다.

그림 6, 7, 8, 9는 각각의 $W = 2, 4, 8, 16$ 에 대한 평균 버스트 길이에 대한 최대 처리량을 나타낸 것이다. 더 큰 평균 버스트 길이에 대해서는 즉, $L \geq 5$ 이고, $W = 2, 4, 8, 16$ 일 때는 DBP 설계가 가장 좋은 실행 결과를 볼 수가 있다. 그림 10, 11은 $L=5$ 일 때와 $L=15$ 에서 W에 대한 Throughput을 나타낸 것이다. 이 실험에서 윈도우는 서로 다른 버스트들의 시작점을 나타내는데 이용된다. 그러므로 주어진 디업 슬롯에서 폭주하는 패킷들은 다른 출력포트들로 결정되어지고, 그 결과로 높은 Throughput을 갖게 된다. 각각의 버퍼속으로 각기 분리된 버스트 위치에 대해 W 버퍼로 사용하는 DSB 설계는 특히 윈도우 크기 증가에 대하여 DBP 설계만큼 효율적이지 못하였다.

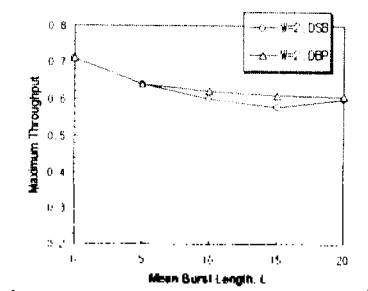


그림 6. $N = 32, W = 2$ 일 때 최대 Throughput 대 평균 버스트 길이

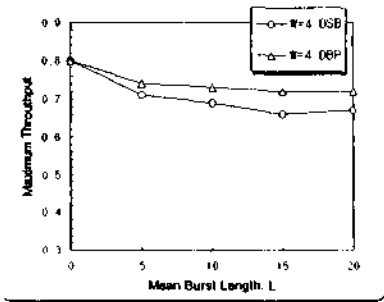


그림 7. $N = 32, W = 4$ 일 때 최대 Throughput 대 평균 버스트 길이

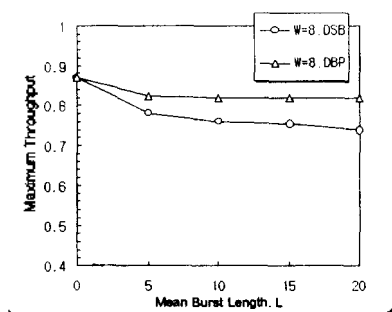


그림 8. $N = 32, W = 8$ 일 때 최대 Throughput 대 평균 버스트 길이

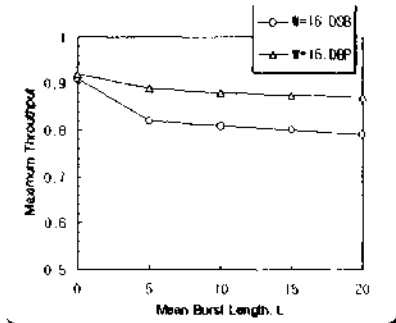


그림 9. $N = 32, W = 16$ 일 때 최대 Throughput 대 평균 버스트 길이

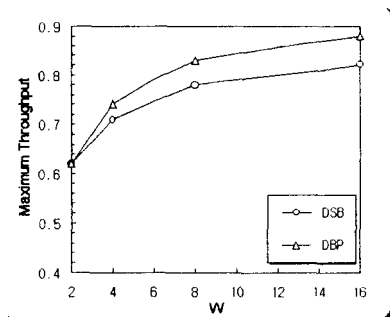


그림 10. $N = 32, L = 5$ 일 때 최대 Throughput 대 W

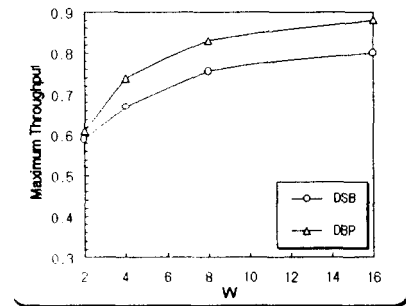


그림 10. $N = 32, L = 15$ 일 때 최대 Throughput 대 W

IV. 결 론

입력 버퍼형 스위치의 처리능력(Throughput)은 HOL 블록킹에 제한을 받는다. IOL 블록킹은 입력 버퍼가 사용된 엄격한 FIFO 큐잉의 결과이며, 윈도우 방식은 사용함으로써 감소시킬 수 있다. 이 윈도우 방식은 엄격한 FIFO 큐잉을 완화시킨다. 윈도우 방식을 채택한 스위치의 처리능력은 윈도우 크기, 윈도우 방식, 트래픽 패턴에 의해 결정된다.

본 논문에서는 DBP 윈도우 기법을 위해 DSB 윈도우 방식을 서로 다른 트래픽 패턴하에서 비교해 보았다. 이 비교는 랜덤 트래픽 패턴과 버스트 트래픽 패턴하에서 Throughput에 의해 수행되고, 수행 필요조건에 대해서도 비교하였다. 예전에 제안된 공유 버퍼 윈도우 방식이 비록 N배의 속도향상을 가진 큐가 필요하지만 랜덤 트래픽 조건하에서 최대 성능을 가진다는 것을 보여준다^[5]. 그리고 N은 네트워크 크기를 나타낸다. 버스트 트래픽 조건하에서 DBP 윈도우 방식은 최대 성능을 가진다. 이 방식은 포인터에 의한 수행이 필요하므로 비용 절감효과는 기대하기 어렵다.

참 고 문 헌

- [1] Uyless Black, "ATM : Foundation for Broadband Networks", Prentice-Hall, 1995.
- [2] James P. G. Sterbenz, "ATM/B-ISDN Tutorial notes", GTE Telecommunication Research Lab., August, 1995
- [3] J. J. Li, "Improving the Input-queueing Switch Under Bursty Traffic," Electronics Letters, 25th May, 1995, Vol. 31, No. 11, pp. 854~855.
- [4] S. C. Liew and K. W. Lu, "Comparison of Buffering Strategies for Asymmetric Packet Switch Modules," IEEE Journal on Selected Areas in Communications, Vol. 9, No. 3, April, 1991, pp.428~438.
- [5] G. Thomas and J. Man, "Improved Windowing Rate for Input Buffered Packet Switches," Electronics Letters, 18th February, 1993, Vol. 29, No. 4, pp. 393~395.