

입력버퍼 교환기에서의 패킷 동기화 기법

이상호, 신동렬
성균관대학교 산업용네트워크연구소
0331-290-7239 / 0331-290-7231

Synchronization at Input Buffered Switch

Sang-Ho Lee and Dong Ryeol Shin
Sungkyunkwan university
{turtle, drshin}@ece.skku.ac.kr

Abstract

Input queueing is useful for high bandwidth switches and routers because of lower complexity and fewer circuits than output queueing. The input queueing switch, however, suffers HOL-Blocking, which limits the throughput to 58%.

To get around this low throughput, we propose a simple scheduling algorithm called Synchronous Input Port (SIP). This method synchronizes packets and switching without blocking, which is shown to have better performance over the established algorithms.

I. 서론

패킷 교환 장치는 버퍼의 배치에 따라 입력버퍼 방식과 출력버퍼방식으로 나뉘어 지며 이 중 입력버퍼 스위치는 입력버퍼를 FIFO를 사용 할 경우에 발생하는 HOL-블로킹(Head of Line Blocking)은 전체 시스템의 성능을 크게 떨어트리는 요인으로 지적되었으며 이를 해결 할 경우 고속 스위칭을 위한 스위치의 구현은 보다 간단해진다.

HOL-블로킹은 크로스바(Crossbar)와 같은 교환기 구조물(Switch Fabric)로 들어가려는 패킷들 중 어느 하나가 같은 출력포트로 향하는 다른 입력포트의 패킷

의 영향으로 진입하지 못 할 때 블로킹(Blocking)이 발생하는 것으로 이때 해당 포트의 FIFO 에서 대기하는 모든 패킷의 지연이 커지게 됨을 말한다. 이는 특정 패킷에 의하여 다른 패킷들의 대기 시간이 증가하는 것을 의미한다[1].

HOL-블로킹에 대한 문제를 해결하려는 방법은 BSB(Buffer-Space-Buffer)구조의 교환기[2][3]와 이클 링크별 스케줄링까지 고려한 방법[4]이 있으며 가상 출력 버퍼의 방법을 이용하여 가장 긴 큐를 먼저 서비스해주는 LPF(Longest Port First)[5]의 방법이 있다. 이중 BSB-교환기는 입력포트에 대한 평균적인 대기 시간의 감소를 위한 방법을 위하여 수개의 입력포트를 묶어 전체 스위치의 활용도를 높이는 방법과 출력포트를 수개로 묶어 같은 출력포트로 향하는 패킷들에 대하여 발생하는 블로킹이 발생할 확률을 줄이는 방법을 이용한 방법이다.

본 글은 HOL-블로킹을 줄이고 이에 따라 큐에서의 대기 시간을 줄임과 동시에 대기시간의 증가에 대한 한계를 규정 할 수 있는 방법을 입력포트의 버퍼를 각 출력포트별 버퍼로 나누어 블로킹이 발생하지 않도록 동기화 하는 방법을 제시한다. 제안된 동기화 기법을 사용한 입력포트를 SIP(Synchronized Input Port)라 부르고 이를 BSB-교환기와 연동 하여 성능을 분석한다. 2장에서 BSB-교환기의 간략한 소개를 하고 3장에서 제안하는 SIP의 구조를 소개한 후 성능분석을 하기로 한다.

II. BSB-교환기의 구조

BSB-교환기는 입력 측과 출력포트의 상태를 감안하여 스위칭 할 패킷들의 순서를 결정하게 되며 각 포트의 버퍼는 FIFO가 아닌 Random Access 방식의 버퍼를 활용하게 된다. 이 구조는 그림 1에 간략화 하였으며 입력포트들의 묶음과 스케줄러(Scheduler)를 포함하는 입력모듈(Input Module)과 출력포트들의 묶음을 출력모듈(Output Module)로 정의하였다.

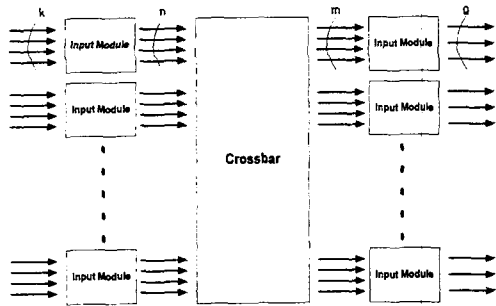


그림 1 BSB 교환기의 구조

입력모듈은 k개의 실제 입력포트들로 묶여지며 이를 n개의 출력으로 이루어져 크로스바에 입력된다. 출력모듈은 크로스바로부터의 m개의 출력을 취하여 이를 g개의 출력으로 출력하며 이때 다시 각 출력포트별로 패킷들을 교환하게 된다. 출력모듈이 m개로 묶여 있음은 같은 시간에 실제 특정 출력포트로 향하는 패킷의 수를 최대 m개까지 허용함을 의미한다. 입력모듈은 각 입력포트에서 도착하는 패킷들을 출력모듈에 관여하여 허용하는 패킷의 수를 고려한 뒤 크로스바로 전송한다. 입력모듈은 특정 입력포트에서 패킷이 도착하지 않더라도 다른 인접한 입력포트의 패킷을 대신 보내는 역할을 하므로 전체 크로스바의 효율을 높게 한다. 출력포트의 상태는 입력모듈로 전달되어 크로스바로 입력될 패킷들을 선택할 때 사용된다[1][2].

BSB-교환기는 각 포트들의 묶음의 수를 크게 할수록 높은 전송률을 가진다. 그러나 실제 구현을 간략하기 위해서는 묶음의 수를 조절할 필요가 있다. 출력모듈들의 경우 패킷들이 실제 향하는 출력포트로 다시 교환을 해야 하므로 출력모듈의 포트 수에 제한이 있게 되며 입력모듈의 경우 스케줄링시간 때문에 제한이 된다. BSB-교환기의 성능을 확인하고 후에 제안된 기법과 비교하기 위하여 모의실험을 통하여 입력버퍼에서의 대기시간을 비교하였다. 모의실험은 16×16의 교환기를 대상으로 하였고 결과는 제안된 구조의 교환기와 비교하기 위하여 4장에 나타내었다.

다음 장에서는 제안된 SIP의 구조에 대하여 살펴본다.

III. 동기방식 입력포트(SIP)의 구조

입력포트에서의 HOL-블로킹은 크로스바를 사용한 교환기의 경우 하나의 출력포트로 향하는 패킷의 수가 같은 시간에 발생하면 발생하게 된다. 이를 방지하는 방법은 각 입력포트에서 패킷들의 전송 순서를 조절할 수 있으나 실제 구현상의 복잡성(Complexity)의 증가가 요구된다. 구현의 간결함을 위해서 SIP는 입력포트에서 각 출력포트로 향하는 패킷들의 순서를 고정적으로 정하여 보내는 간단한 방법이다.

4×4의 교환기의 경우 각 입력포트를 구성하는 SIP는 그림 2와 같은 순서로 패킷을 크로스바에 입력한다.

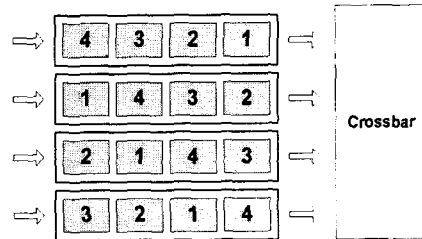


그림 2 패킷들의 동기화

그림 2는 포트별로 4개의 패킷들이 크로스바에 입력되는 순서를 나타냈으며 표시된 번호는 각 패킷들이 향하는 출력포트의 번호이다. 이 경우 크로스바에 입력되는 각 패킷들은 서로 다른 출력포트를 향하므로 B블로킹이 발생하지 않는다. 이와 같은 동작을 위해서 SIP는 각 출력포트별 큐를 가지며 입력되는 패킷을 출력포트별로 나누어 해당 큐에 진입시킨 후 관리하게 된다. 4개의 출력포트를 가지는 경우 각 입력포트에서의 SIP는 4개의 큐를 가지게 된다. SIP의 구조를 그림 3에 나타내었다.

N개의 입력을 가지는 교환기에서 각 각의 입력포트는 SIP를 포함한다. 각 SIP에서 MUX는 각 출력포트별 큐를 하나씩 선택하는데 교환기의 작동이 시작될 때 서로 다른 큐를 선택하도록 초기화를 하고 순차적이고 순환적인 (Round Robin) 방식으로 동작한다. N개의 출력을 가지는 경우 각 SIP는 N개의 큐를 가지므로 하나의 SIP내에서 특정 큐가 서비스를 받기 위해 대기하는 시간은 해당 큐가 선택받은 후 N-1번의 서비스 시간 이후로 결정된다.

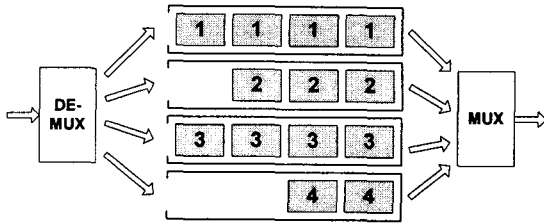


그림 3 SIP의 구조

VI. 성능평가

성능평가는 16×16 교환기를 대상으로 모의실험을 수행하였으며 각 포트에 도착하는 각 패킷들의 출력포트로의 방향은 1/16의 일정한 확률로 발생하게 하였다. 먼저 BSB-교환기를 대상으로 그림 1에서 주어진 k, n, m, g의 값들을 변화시키면서 입력포트에서의 패킷들의 평균대기시간을 비교하였다. 먼저 출력 측의 m과 g를 모두 1로 고정시킨 경우와 입력 측의 k와 n을 1로 고정시킨 것을 모의실험 하였으며 그림 4와 그림 5에 나타내었다.

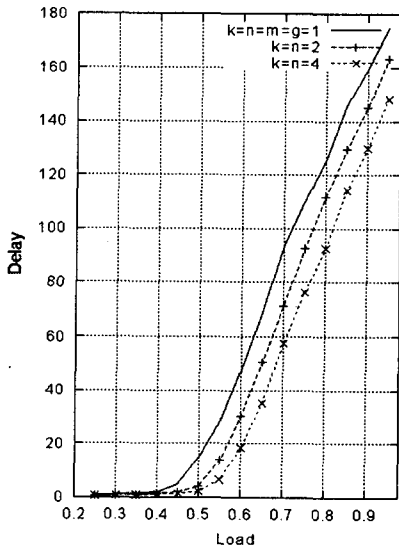


그림 4 k와 m을 변화시킨 결과

그림 4와 그림 5에 나타낸 모의실험결과는 입력포트를 묶어 크로스바의 활용도를 높이는 것보다 출력포트들을 묶어 HOL-블로킹의 발생빈도를 낮추는 것이 성능 향상에 더 효과가 있음을 알게 해주며 HOL-블로킹이 시스템의 성능에 크게 관여함을 보여준다.

이를 기준으로 k, m, g 그리고 n의 값을 함께 증가시켰을 때 보다 좋은 성능을 얻을 수 있으며 이를 그

림 6에 나타내었다.

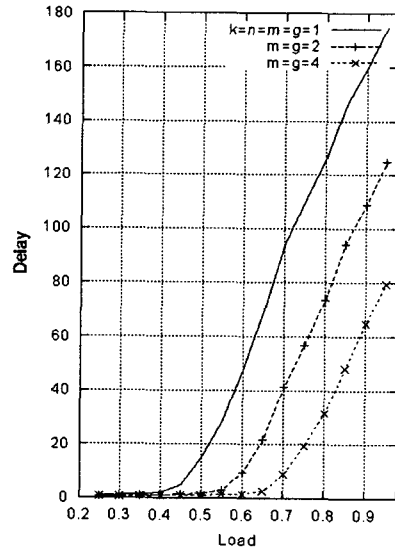


그림 5 출력모듈의 m, g를 변화시킨 결과

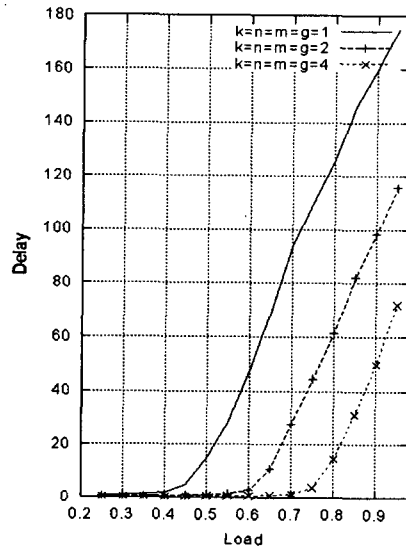


그림 6 k, n, m, g를 함께 변화시킨 결과

본 논문에서 제안된 SIP를 사용한 교환기의 모의실험은 앞서 검증된 BSB-교환기의 입력부분을 SIP로 교체하여 모의 실험을 하였다. 교환기의 모든 각 포트는 SIP 구조를 가지며 출력포트들의 묶음을 바꾸어가며 테스트하였다. 출력 측의 m과 g의 값이 1이상인 경우 출력모듈이 생성되는 것이므로 이 경우 SIP는 출력포트의 수만큼 큐를 가지지 않고 출력모듈의 수만큼 큐를 가지게 된다. 모의실험을 수행한 모델을 그림 7

에 나타내었고 결과를 그림 8에 나타내었다.

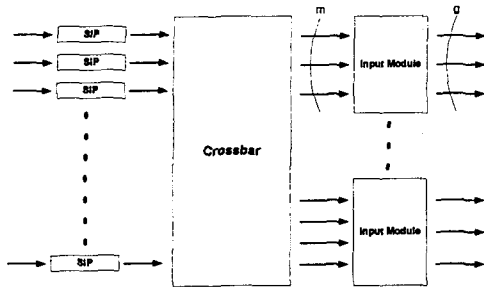


그림 7 SIP를 사용한 교환기 모델

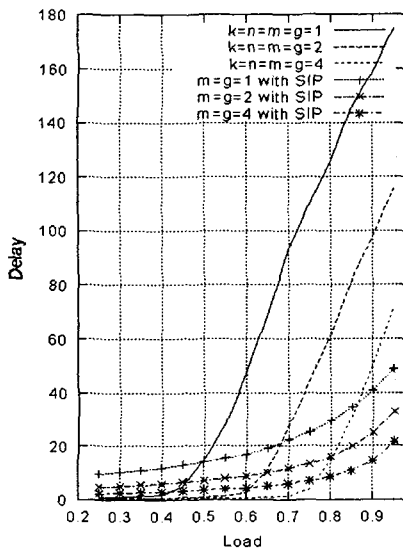


그림 8 BSB-교환기와 SIP를 사용한 교환기의 비교

SIP를 사용한 경우 높은 부하(Load)를 가할 수록 이전의 BSB-구조의 교환기 보다 높은 성능을 나타내며 출력포트들을 묶어 처리할 경우 SIP 내부에서 순환하는 시간 (Round Robin Time)이 적어지므로 더 높은 성능을 가지게 된다. 낮은 부하에서는 일반적으로 대기시간이 크며 이는 SIP내의 각 큐에 대하여 순환적인 서비스에 의하여 만약 특정 큐에만 패킷들이 존재하더라도 한번에 한번 크로스바로 전송되면 해당 큐의 다음 패킷은 다른 큐들이 모두 한번씩 선택된 후에 다시 전송할 기회를 얻게 된다. 낮은 부하에서 보다 높은 성능을 얻기 위해서는 SIP 내에서 순환시간을 줄여야 하며 능동적인 스케줄링 기법이 요구된다.

V. 결론 및 추후 연구

본 논문에서는 같은 출력포트로의 패킷들의 충돌을 제거하기 위하여 입력포트에서 크로스바로 진입하려는 각 패킷들의 순서를 고정적으로 정하였으며 이를 이전에 제시된 BSB 구조와 함께 대기시간에 대한 성능을 평가하였다. SIP를 사용한 교환기는 높은 부하에서 다른 구조에 비해 전송률이 100%에 가까운 높은 성능을 나타내며 최대대기시간의 한계를 정할 수 있음을 확인하였다. 제시된 SIP의 동작은 고정적인 순환(Round Robin)에 의하여 전송할 패킷들을 선택하므로 낮은 부하에서도 이전의 BSB 구조에 비하여 높은 대기시간이 발생한다. 고정적이지 않고 SIP내의 상태를 고려하여 전송하려는 패킷을 선택하기 위해서는 일반적으로 출력포트에서 사용하는 WRR(Weighted Round Robin) 및 WFQ(Weighted Fair Queueing) 기법 등의 공평한 스케줄링 기법이 필요하다[6]. 현재 동적인 스케줄링 방법을 적용하여 낮은 부하에서도 대기시간의 감소와 이에 따른 분석을 하고 있다.

참고문헌

- [1] M. J. Karol, M. G. HLUCHYJ and S. P. Morgan, "Input versus Output Queueing Switch", IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, Sep. 1991, pp. 1347-1355.
- [2] H. Obara, "OPTIMUM ARCHITECTURE FOR INPUT QUEUEING ATM SWITCHES", IEE Electronics Letters, Vol. 27, No. 7, Mar. 1991, pp. 555-557.
- [3] H. Obara, S. Okamoto and Y. Mamazumi, "INOUT AND OUTPUT QUEUEING ATM SWITCH ARCHITECTURE WITH SPATIAL AND TEMPORAL SLOT RESERVATION CONTROL", IEE Electronics Letters, Vol. 28, No. 1, Jan. 1992, pp. 22-24.
- [4] M. J. Karol, Kai Y. Eng and H. Obara, "IMPROVING THE PERFORMANCE OF INPUT-QUEUED ATM PACKET SWITCHES", IEEE INFOCOM '92, 1992, pp. 110-115.
- [5] A. Mekittikul and Nick McKeown, "A Practical Scheduling Algorithm to Achieve 100% Throughput in Input-Queued Switches", IEEE INFOCOM' 98, 1998, pp. 792-797.
- [6] Hui Zhang, "Service Disciplines for Gauranteed Performance Service in Packet-Switching Networks", Proceedings of IEEE, Vol. 83, No. 10, Oct. 1995, pp 1374-1396.