論文96-33B-10-2

Bidirectional MIN에서 효율적인 라우팅을 지원하기 위한 계층적 버퍼링 기법

(Hierarchical Buffering Scheme for Supporting Effective Routing Scheme in Bidirectional MIN)

張 昶 洙 *, 金 聖 天 **

(ChangSoo Jang and SungChun Kim)

요 약

최근 많은 수퍼컴퓨터들은 고성능의 병렬 컴퓨터를 구성하기 위해 스위치를 근간으로 하는 다단계 상호연결망 네트워크(MINs) 구조를 사용하고 있다. 본 논문에서는 웜홀 라우팅 하에서 트래픽이 증가시 성능이 급격히 떨어지는 것을 방지하기 위한 새로운 라우팅 방법인 Hybrid Wormhole and Virtual cut-through (HWCR) 기법을 제안한다. 이 방법은 웜홀 라우팅 시 블럭킹이 발생할 때 플릿의 흐름을 원활케하기 위하여 VCT (Virtual-Cut-Through) 라우팅 방법을 도입하여 블럭된 링크를 해제한 후, 다시 웜홀 라우팅으로 복귀하는 방법이다. 그러기위해 HWCR 방법에서 버퍼는 BMINs의 버퍼의 하드웨어 비용을 줄이고 네트워크의 성능을 향상시키기 위해 버퍼크기를 계층적으로 할당하는 기법을 사용하였다. 버퍼 크기와 통신 지연을 적정화하기 위해 사용된 평균 버퍼량과 평균 패킷 지연을 성능척도로 하였으며, 웜홀과 VCT 그리고 제안된 HWCR기법을 컴퓨터 시뮬레이션을 통해 비교 분석하였다.

Abstract

Many recent supercomputers employ a kind of switch-based Multistage Interconnection network architectures(MINs) for constructing scalable parallel computers. This paper proposed a new routing method, Hybrid Wormhole and Virtual-Cut through Routing(HWCR) for the prevention of rapid performance degradation comming from a conflict in links usage at hot traffic situation. This HWCR is a method to resuming the wormhole routing after the removing of blocked link with the Virtual-Cut through(VCT) for the fast removing temporal stagger, result in seamless flow of packet stream. When the blocked link is removed, wormhole routing is resumed. The HWCR method adopted a hierarchical buffer scheme for improving the network performance and reducing the cost in BMINs. We could get optimum buffer size and communication latency through the computer simulation based on proposed HWCR, and the results were compared to those using wormhole and VCT.

I. 서 론

MIN은 확장가능한 병렬 컴퓨터 시스템(SPC : Scalable Parallel Computer)을 구성하기 위한 스위치를 기반으로 하는 네트워크 구조의 한 종류이다. 이런 MIN에서의 통신 지연은 병렬 컴퓨터의 성능에 영향을 주는 중요한 요소이다. 이러한 통신 지연은 패킷이 소스에서 목적지까지 도달하는 동안 경과된 시간을 말하며, 다음과 같은 세 가지 충돌 형태에 기인한다. 첫 번째는 소스에서 큐잉(queuing), 두 번째는 링크(혹은 버퍼)에서 충돌, 세 번째는 같은 출력으로 다중패킷을 방출할 때 발생하는 출력 충돌 등이다. 만약 네트워크

* 正會員, 國立 麗水 水産大 컴퓨터工學科
(Yosu Nat. Fisheries Univ. Dept. of Computer Eng.)
** 正會員, 西江大學校 電子計算科
(Sogang Univ. Dept. of Computer Science)
接受日字:1996年2月13日, 수정완료일:1996年10月2日

의 트래픽이 복잡하거나, 고르지 않게 분포되면 지연은 높아진다. 이러한 통신 지연은 스위칭 기술에 매우 많이 좌우된다[1]. 이를 개선하기 위해 병렬 컴퓨터 스위칭 방법으로는 회선교환(circuit switching) (예 : BBN GP-1000[2] , TC-2000[2])방법이나 패킷 스위칭 방법 그리고 스토어-앤드-포워드(store-and-forward) 스위칭 방법[3,4,5]들은 병렬 컴퓨터에 광범위하게 연구되고 있다. 특히 웜홀 스위칭 방법은 낮은 통신 지연을 제공하기 때문에 차세대 병렬 컴퓨터[6]에 각광을 받고 있다. 현재 상용 병렬 처리 시스템의 프로세서 모듈 구성은 양방향 링크(bidirectional link)를 갖는 BMIN(Bidirectional MIN)이 선호되고 있다. 또한 BMIN은 기존의 TMIN(Traditional MIN) 보다 경로 상에 더 많은 중복 경로(redundant path)수를 갖음으로서, 링크에서의 충돌확률은 상대적으로 적어진다. 이러한 BMIN에서 라우팅은 대부분 웜홀 라우팅이 사용되는데, 최소 단위정보인 플릿이 전송되고 있을 때 전체 나머지 패킷은 중간 노드에 다 도착할 때까지 기다리지 않고 다음 목적지로 라우팅되는 방식을 취한다. 그러나, BMIN에서 사용되는 웜홀 라우팅은 네트워크 트래픽이 적은 경우에는 효율적이지만, 트래픽이 일정하지 않는 패턴이거나, 과도한 트래픽 집중현상이 일어날 경우에는 상대적으로 많은 채널이 점유되어, 전체적으로 네트워크 처리율이 급격히 저하되는 단점을 갖는다. 본 연구에서는 트래픽이 많은 웜홀 스위칭 하에서 블럭킹이 발생하였을 때, 성능이 급격히 저하되는 것을 방지하기 위한 새로운 라우팅 방법을 제안하였다. 즉 이 방법은 링크가 블럭상태일 때 VCT 라우팅 방법의 장점을 혼합한 라우팅 방법으로 본 논문에서는 HWCR(Hybrid Worm-hole and Virtual-cut through routing)이라 부른다. 제안한 모델의 장점은 VCT 방법 보다는 적은 양의 버퍼를 사용하는데 이는 다단계상호연결망에서 각 레벨 마다 링크의 사용빈도 수가 다르다는 것에 착안하여 각 레벨마다 버퍼 크기를 다르게 할당한 것이다. 이로써 VCT보다는 성능이 우수하고, 웜홀 라우팅의 단점인 과도한 트래픽 하에서의 성능 저하를 줄일 수 있는 장점을 갖는다.

Ⅱ. Bidirectional MIN(BMIN)

1. BMIN의 구조

양방향 통신을 허용하기 위해서 스위치의 각 포트는 이중 채널을 갖고, 8-노드 버터플라이 양방향 MIN (Bidirectional MIN)에서 프로세서/메모리 노드들이 네트워크의 왼쪽 편에 있는 single-end 시스템 구조인 8×8 BMIN을 그림 1에 보여준다. 따라서 single-end 구조의 특성상 네트워크의 연결 구성은 최상위 레벨에서 180도 선회(turnaround)하여 접속하는 연결 형태를 갖는다[5,9]. BMIN에서 웜홀 스위칭을 사용한 많은 상업용 SPCs가 있는데, BMIN들의 라우팅 특성은 시스템마다 다양하다.

2. BMIN의 특성

다음은 BMIN의 위상에 연관된 정의이다. 다음 정의에 따라 BMIN을 구성할 수 있으며, 이러한 정의는 라우팅시에도 사용된다.

【정의 1】 BMIN는 $N\times N$ 스위치로 이루어지며, 한 스위칭 소자 SE(Switching Element)의 크기는 $k\times k$로 나타내고, $N=k^n$이므로 전체 스테이지의 수(n)는 $\log_k N$ 이며, 한 스테이지의 SE수는 N/k개 이다.

【정의 2】 하나의 $k\times k$ SE는 양방향 링크를 가지며, 왼쪽에서 오른쪽 방향으로 경로 연결시(순방향 연결) 입력 노드는 위에서부터 차례로 $l_0, l_1, \ldots, l_{k-1}$로, 출력 노드는 $r_0, r_1, \ldots, r_{k-1}$로 인덱스된다.

【정의 3】 소스 주소는 $S=s_{n-1}\ldots s_1 s_0$로 나타내고, 목적지 주소는 $D=d_{n-1}\ldots d_1 d_0$ 로 나타낸다.

【정의 4】 임의의 스테이지는 S_i로 정의하는데, i는 스테이지 번호로 $0\le i\le n-1$이며, 여기서 $n=\log_k N$ 이다. 한 스테이지내의 SE들은 위에서부터 $SE_{i,0}$, $SE_{i,1}$,..., $SE_{i,N/k-1}$ 으로 인덱스된다.

【정의 5】 임의의 채널(또는 링크) 스테이지는 C_i로 정의하는데, i는 채널 번호로 $0\le i\le n-1$ 이다. 여기서 $n=\log_k N$ 이다.

【정의 6】 $t<j<n$ 에서 $s_t\neq d_t$ 이고 $s_j=d_j$가 되는 t 값을 $FirstDifference\ (S, D)=t$ 라고한다.

이 t 값은 소스와 목적지 주소 사이에 왼쪽 첫 번째(Leftmost) 다른 디지트 값을 갖는 비트 위치를 말한다[5,9]. $FirstDifference\ (S, D)=t$, 이 값은 소스에서 목적지 까지 라우팅 시에 선회하는 스테이지를 결정하는 값이 된다.

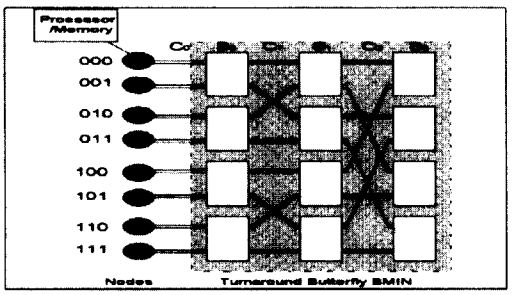

그림 1. 8 x 8 양방향 BMIN
Fig. 1. 8 x 8 BMIN.

【정의 7】 채널 c_i^r'은 순방향연결에서, 출력 포트(또는 출력 링크)가 $r_j(0\le j\le k-1)$인 $i(0\le i\le \log_k N-1)$번째 채널 스테이지에 있는 채널을 말한다. 그리고, 채널 c_i^l'는 역방향 연결에서, 출력 포트가 l_j인 i번째 링크 스테이지에 있는 채널을 말한다.

3. BMIN의 SE 구조

BMIN에서, 각 SE들의 양방향 입출력 포트들에는 정상적인 웜홀라우팅을 위해 플릿을 저장할 수 있는 플릿 버퍼와 개선된 라우팅 알고리즘을 지원하기 위한 유한 개의 플릿을 저장할 수 있는 패킷 버퍼(모의실험에 사용한 크기: 20flits/packet) 로 구성된다. 이 유한 개의 패킷 버퍼는 진행하려는 링크가 블럭될 경우, 하나의 패킷에 해당되는 여러 개의 플릿들을 모아 저장하기 위한 용도로 사용된다. 보통의 경우에는 웜홀 라우팅을 사용하기 때문에 이 패킷 버퍼는 사용되지 않는다. 그러나 현재 경로를 따라 진행하는 플릿들이 진행 링크가 블럭될 경우에만 이 패킷 버퍼를 사용한다. 그러므로 이 경로 중에 여러 플릿들이 점유하고 있는 헤더 플릿 이후의 블럭된 링크를 자유롭게 하므로서, 링크 자원을 효율적으로 사용할 수 있게 해 준다[12]. 다음 그림 2에서는 스위칭 소자의 구조를 설계하였다. 데이터는 입출력 포트를 통해 양방향으로 전송이 가능하며, 제어 라인을 통해, 진행하려는 링크의 블럭 상태를 감지할 수 있다. 그리고 SE 내에서 데이터의 이동 경로는 두 경우가 존재하는데, 하나의 경우는 일반적인 웜홀 라우팅을 사용할 때에 사용하는 플릿 버퍼를 통한 경로이고, 다른 하나의 경로는 진행할려는 링크에서 충돌이 발생할 때 패킷 버퍼를 통하여 라우팅 하는 경로이다. 그림에서 보여지는 제어라인(control line)은 인접한 스위칭 소자와의 통신을 통해 데이터가 진행할 수 있는지의 여부를 인지하는 라인이다. 일반적인 REQ/ACK 기능을 갖는다고 말할 수 있다. 이에 따라 라우팅하려는 방향에 인접한 스위칭 소자의 링트가 프리이면 데이터를 진행시킬 수 있다. 그러나 제어 라인을 통해 진행 링크의 블럭된 상황이 감지되면, 라우팅하던 모든 플릿들을 각 입출력 포트에 연결된 이 패킷 버퍼에 저장한 후 VCT 라우팅으로 전환하는 것이다. SE에서의 입출력 포트 연결 방식은 순방향연결, 역방향연결, 선회 연결 방식이 존재한다.

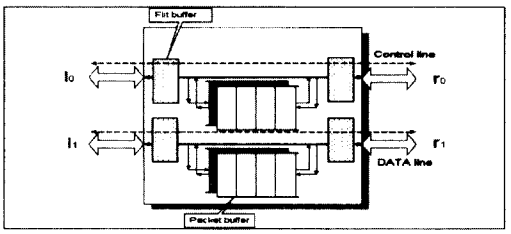

그림 2. SE의 구조
Fig. 2. Structure of SE.

4. 패킷 모델

패킷은 각 싸이클마다 SE의 사이를 이동하며 필요한 데이터를 전송한다. 사용한 패킷의 형태는 다음 그림 3과 같다.

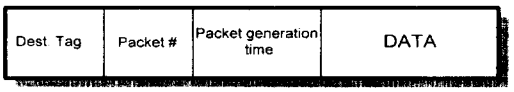

그림 3. 패킷 구조
Fig. 3. Structure of Packet.

목적지 태그는 통신하려는 상대 프로세서 모듈의 주소(디지트 형태: fixed-length)를 나타내며[2,8], 라우팅 형식은 이 목적지 주소를 이용한 셀프 라우팅 방식으로 분산형 라우팅을 한다. 패킷 번호는 메시지에서 해당 패킷의 위치를 나타내며, 데이터는 실제로 그 패킷이전달해야 할 데이터를 말하고, 패킷 생성시간은 입력단에 연결된 프로세서/메모리 모듈(PM)에서 패킷이 생성된 시간을 나타내며, 이것은 이 패킷이 목적지에 도착한 시간과 관련하여 한 패킷의 전송시간을 구하는데 사용되는 요소이다. 그러므로 전송시간은 생성시간과 패킷이 목적지에 도착한 시간의 간격임을 알 수 있다. 이 값은 네트워크 지연을 측정하는데 사용된다. 정

상적인 경우에는 웜홀 라우팅을 사용하므로, 각 패킷은 하나의 헤더 플릿과 여러개의 데이터 플릿으로 구성되어 파이프라인 방식으로 BMIN 내에서의 HWCR 라우팅을 진행한다.

5. BMIN에서의 선회 라우팅

먼저 BMIN과 같이 선회 지점을 갖는 네트워크 구조에서는 【정의 6】에서 나타낸 *FirstDifference*$(S, D) = t$의 의미가 매우 중요하다. 이 t 값은 소스에서 목적지 까지 라우팅 시에 선회하는 스테이지를 결정하는 값이 된다. 그러나, *FirstDifference*$(S,D)= i$ ($i< n-1$) 인 경우는, 전체 네트워크를 순회할 필요없이 S_i 스테이지에서 선회하면 된다. 즉, 최상위 레벨까지 올라갔다가 다시 내려올 필요가 없다. 다음은 링크 충돌이 없는 일반적인 경우에서의 라우팅 알고리즘이다. 이 알고리즘은 충돌이 없는 경우로 다음 절에서 충돌을 해결하는 알고리즘으로 확장될 것이다. 이 알고리즘은 먼저 *FirstDifference* (S, D)를 구한 다음 순방향경로 배정시에는 임의의 자유로운 링크 (r_i ($0 \le i \le k-1$)) 를 선택하여 랜덤하게 라우팅 하게 된다.

1) 선회 라우팅 알고리즘

Algorithm1 Turnaround routing algorithm in each at stage j

Input : Source address S : $s_{n-1}\ldots s_1 s_0$

Destination address D : $d_{n-1}\ldots d_1 d_0$

Procedure :

1. $t := FirstDifference\ (S, D)$;
2. If $j = t$, 포트 l_{d_j}으로 선회 연결을 택한다.
3. If j<t 이고, 입력 포트 l_m으로부터 메시지가 오면, 임의의 이용 가능한 포트 r_i로의 순방향연결을 택한다($0 \le m, i \le k-1$).
4. If $j< t$ 이고, 입력 포트 r_m으로부터 메시지가 오면, 목적지 태그에 따라 결정된 포트 l_{d_j}로의 역방향연결을 택한다($0 \le m \le k-1$).

2) BMIN의 중복 경로 수

일반적으로 8×8 BMIN에서 *FirstDifference* (S, D)가 2인 경우는 4개의 중복 경로, 또는 대체 경로(alternative path)가 존재하고, *FirstDifference*$(S, D) = 1$인 경우는 2개의 중복 경로가 가능하다. 따라서 $N \times N$ BMIN 네트워크에서 가장 최단 거리의 중복 경로의 수(RD path)는 다음과 같이 결정할 수 있다;

$$RD\ path = k^t \quad (\text{여기서 } t = FirstDifference(S, D),\ 0 \le t \le \log_k N-1)$$

그림. 4는 충돌이 존재하는 경우의 라우팅 예이다. $S_1 = 001 \rightarrow D_1 = 111$, $S_2 = 011 \rightarrow D_2 = 110$로의 2개의 경로가 동시에 존재하는 라우팅이다. 이런 경우 기존의 BMIN에서 웜홀 라우팅을 사용할 경우는 하나의 경로에 우선 순위 경로를 먼저 선택하여 전송 한 후 블럭된 나머지 경로도 재전송을 해야 한다. 그러나 블럭된 시점까지 전송된 패킷(또는 플릿)들을 모두 폐기해야 하거나, 플릿들을 각 스위칭 소자의 플릿버퍼에 저장하고 블럭된 상태로 존재해야하는 문제가 발생한다. 현재 블럭된 지점 이전까지 경로상을 점유하고 있는 플릿들로 인해 또 다른 경로의 진행을 방해하기 때문에 다시 블럭되는 문제가 발생된다. 이런 상황은 비균일한 트래픽 패턴일 경우는 더욱 심각하다. 그림 4에서 $S_2 \rightarrow D_2$ 경로에 우선 순위를 두면, $S_1 \rightarrow D_1$ 경로상에 있는 SE들은 블럭된 링크 $C_1^{t_1}$가 풀릴 때까지 계속 점유된다. 그림 5는 웜홀 라우팅과 부가적으로 패킷버퍼가 요구되는 VCT 라우팅 형태를 보여주고 있다[18]. 그림에서 점선은 블럭킹된 링크를 의미한다.

BMIN에서 두 개의 라우팅 경로 P와 Q가 존재한다고 가정할 때, 이는 순방향 라우팅 시에는 여러 개의 대체 경로로 라우팅이 가능하기 때문에 충돌을 피할 수가 있다. 충돌은 선회 지점까지는 피할 수 있지만, 역방향 라우팅이 시작되는 이후부터 단 하나의 경로만이 존재하기 때문에 충돌 가능성이 항상 존재한다. 이는 패트-트리 구조를 갖기 때문이다[13]. 충돌이 발생하는 경우를 다음과 같이 정의할 수 있다.

【정의 8】 : 두 경로 P와 Q가 스테이지 S_i $(0 \le i \le n-2)$의 한 $SE_j (0 \le j \le N/2-1)$에서 동일한 출력 링크 상에서 충돌이 발생할 경우는 다음의 경우이다[12].

P 와 Q의 목적지 주소를 $d_{n-1}\ldots d_1 d_0$ 와 $d'_{n-1}\ldots d'_1 d'_0$라고 가정한다. P 와 Q 가 S_i의 SE내에 한 입력 링크로 각각 r_m $(0 \le m \le k-1)$과 $r_n (n \ne m, 0 \le n \le k-1)$를 사용하고, P의 출력 링크 $l_{d_i} (0 \le i \le n-2)$와 Q의 출력 링크 $l_{d'_i}$가 $d_i = d'_i$ 인 관계가 성

립할 경우에 출력 링크에서 충돌이 발생한다.

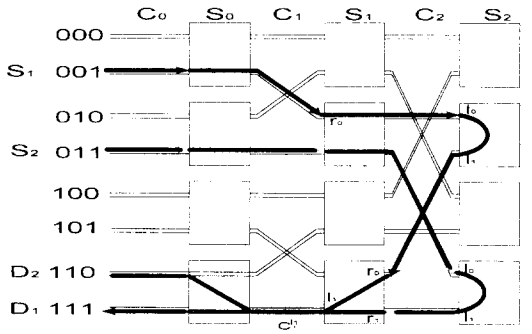

그림 4. 8 x 8 BMIN에서 블록킹
Fig. 4. Blocking in 8 x 8 BMIN.

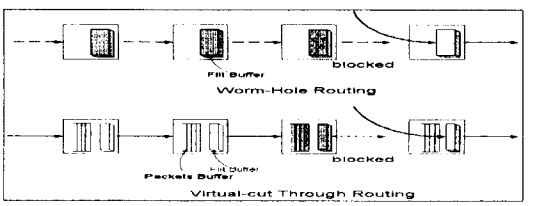

그림 5. 웜홀 라우팅과 VCT 라우팅
Fig. 5. Worm-Hole routing and VCT routing.

Ⅲ. 제안된 HWCR 기법

1. HWCR 알고리즘

다음 알고리즘 2에서 충돌을 해결할 수 있는 HWCR 알고리즘을 설계한다. 이 알고리즘은 현재 BMIN에서 사용되는 선회 웜홀 라우팅의 단점을 보완하였다[11,12]. 웜홀 라우팅으로 진행하는 두 경로가 경로 상의 어떤 링크에서 충돌할 시, VCT 라우팅 방법으로 전환하여 SE 안의 버퍼에 블럭된 플릿들을 모두 모아 저장한다. 그림 6에 나타내었다. 그러므로 블럭된 링크 전에 점유하고 있던 경로 상의 링크를 풀어 주게 되어 다른 경로들이 이 링크를 사용할 수 있게 한다. 이후에 다시 블럭된 링크가 해제되면, 다시 이전의 웜홀 라우팅으로 목적지까지 라우팅을 계속한다.

Algorithm2 Conflict-free HWC turnaround routing algorithm in each at stage j

Input : Source address S : $s_{n-1}\ldots s_1 s_0$

Destination address D : $d_{n-1}\ldots d_1 d_0$

Procedure :

1. $t := FirstDifference\ (S, D)$;
2. IF $j = t$, 포트 l_{d_t}으로 turnaround connection을 택한다.
3. IF $j < t$ 이고, 입력포트 l_m으로부터 메시지가 오면, /* forward path */ 임의의 이용가능한 포트 r_i로의 순방향연결을 택한다($0 \le m, i \le k-1$).
4. IF $j < t$ 이고, 입력포트 r_m으로부터 메시지가 오면, /* backward path */
   ① IF 출력 포트, l_{d_t}에 연결된 링크 $C_i^{l_{d_t}}$($0 \le i \le \log_k N-1$)가 블럭되어 있으면, 현재 SE의 HWC 버퍼에 현재 경로에 있는 모든 플릿을 모은 후 $C_i^{l_{d_t}}$ 링크가 freed일 때 까지 대기한다.
   ② IF 진행할 링크 $C_i^{l_{d_t}}$ 가 블럭되지 않은 링크이면, l_{d_t}로의 역방향연결을 택한다.
   ③ IF 출력 포트(l_{d_t})에 연결된 링크 $C_i^{l_{d_t}}$($0 \le i \le \log_k N-1$)가 블럭 됐었지만, 현재 링크가 free이면, 플릿 단위로 다시 l_{d_t}로 역방향연결을 택한다.

END.

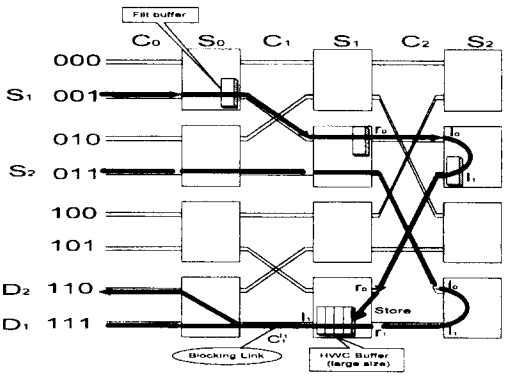

그림 6. 8 x 8 BMIN에서 HWC 라우팅
Fig. 6. HWC routing in 8 x 8 BMIN.

2. HWCR에서 버퍼링 방법

BMIN의 특성은 최상위레벨에서 선회하는 구조이므로 팻-트리(Fat-tree)[5,10]와 함수적으로 동형으로 볼 수 있다. 따라서 single-end 구조를 갖는 BMIN의 트래픽은 상위 레벨에 가까운 쪽에 많은 패킷 충돌이 발생한다. 스위치 크기가 k×k 일 경우, $t = First\ Difference\ (S, D)$값에 따라 블록킹 횟수는 k^t가 된다. 이것은 t-스테이지에서 경로의 수를 계산함으로써 쉽게 구할 수 있다. 순방향 경로에 있는 스테이지들로 부

터 선회하는 스테이지까지는 여러 개의 중복경로가 존재하기 때문에 최악의 상태에서도 충돌을 피할 수 있으므로 고려 할 필요가 없지만, 역방향 경로에 있는 *SEs*의 버퍼 크기는 스테이지가 감소하는 레벨 크기 순서로 버퍼 크기를 할당한다. 예를 들어, $N=8$ 인 경우, 선회 스테이지가 2 라면, 중복경로를 사용하여 라우팅이 가능하기 때문에 순방향 경로에서 스테이지 2까지는 충돌을 피할 수 있다. 그러므로 스테이지 1, 스테이지 0순으로 버퍼 크기를 할당하면 된다. 그런 후 성능분석을 통해 각 레벨에 따른 최적의 버퍼 크기를 결정한다. 일반적으로 버퍼링은 상위 레벨부터 아래로 점차로 감소하는 크기의 계층적 버퍼(hierarchical buffer)를 할당한다. 즉, 상위레벨에 큰 버퍼를 할당하는데 이것은 상위레벨에서의 채널 충돌 가능성이 증가하기 때문이다.

3. 버퍼 모델

BMIN의 각 *SE*들은 양방향 입출력 라인을 갖으므로, 입출력 쪽에 모두 플릿 버퍼가 필요하고, 각 링크당 한쌍의 패킷 버퍼가 필요하다. 이 패킷 버퍼는 패킷 충돌을 효율적으로 관리하기 위해 사용하며, 보통때에는 플릿 버퍼만이 사용된다. 그러나 진행 링크에서 블럭된 상황이 검출될 때는 자동적으로 패킷 버퍼 경로로 패스가 스위치 된다. 본 논문에서는 BMIN이 팻-트리 구조를 가지고 있으므로, 계층적 버퍼할당 전략을 사용였다. 이것은 팻-트리의 특성상 상위 레벨에서 패킷 충돌이 많이 일어난다는 관찰에 근거한다[5,10]. 그림 7은 N=16일 경우에 계층적 패킷 버퍼 할당전략에 따른 패킷 버퍼 모델을 도식화한 것이다. 물론 각 SE당 양방향의 입출력포트에 한 쌍씩의 플릿 버퍼가 존재한다. 그림에서 버퍼의 크기는 $2^{i-1}(0\le i\le \log_k N-1)$의 비율로 할당한다.

Ⅳ. 성능분석

BMIN의 성능을 분석하기 위해서는 기존의 웜홀 라우팅을 사용하는 BMIN과 제안된 HWCR 알고리즘을 사용하는 BMIN의 성능을 네트워크 통신 지연과 사용된 버퍼 측면에서 분석해 본다. 각 레벨에 따라 계층적으로 증가된 버퍼 크기를 갖지만, 통신지연을 증가시키지 않고 일정한 수준의 네트워크 처리율을 유지할 수 있는 최적의 버퍼 크기를 결정하기 위해 다음 시뮬레이션을 통해 분석하였다.

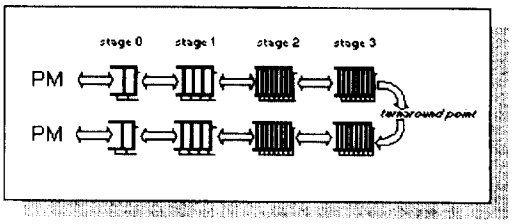

그림 7. 버퍼 모델
Fig. 7. Buffer Model.

1. 시뮬레이션 결과 분석

BMIN 시뮬레이션을 위한 가정들은 다음과 같다;

① BMIN의 스위칭 소자는 동기적(synchronously)으로 동작 한다.
② 스테이지 사이클은 10 클럭 주기로 한다.
③ N개의 프로세서 모듈로 부터 BMIN에 도착하는 패킷은 포아송 분포를 가지며, 그리고 모든 목적지는 동일한 확률, 1/N으로 참조된다[5].
④ 각 스위칭 소자는 유한개의 버퍼를 갖는다.
⑤ 두 개의 플릿이 동시에 같은 출력 포트로 전송을 요구하는 경우, 임의의 하나를 선택하고, 나머지는 버퍼에 저장한 후 다음 사이클에 전송한다.
⑥ 하나의 패킷은 20개의 플릿으로 나누어 지며, 각 플릿의 크기는 8 바이트로 가정한다.

시뮬레이션은 SLAM II를 사용하여 이산사건 모델로 모델링되어 수행되었다. 시뮬레이션에 있어서 스위치 내부에 존재하는 버퍼의 크기는 네트워크의 성능을 유지하는 중요한 요소이다. 다음은 BMIN 성능 분석을 위해 본 논문에서 사용한 네트워크 성능 척도이다.

2. 평균 패킷 지연시간

평균 패킷 지연시간(average packet latency)은 전체 메시지의 패킷이 근원지 프로세서 모듈에서 생성된 다음 부터 마지막 플릿이 목적지에 도착할 때까지의 평균 시간을 나타낸다. 이 값은 패킷 헤더(플릿 헤더)가 가지고 있는 패킷 생성 시간과 이것이 목적지에 도착한 시간차로 부터 측정된다. 우리는 이 값을 기존의 웜홀 라우팅과 VCT, 그리고 제안된 HWCR과 비교하였다. 그림 8의 그래프가 $N=64$인 경우의 평균 패킷 지연시간을 보여주고 있다. 본 논문에서 제안된 HWCR 기법은 VCT 라우팅 보다 적은 양의 패킷 버

퍼를 사용하였지만, 지연 시간은 거의 VCT 라우팅에 가까운 성능을 나타내고 있다. 이 값은 기존의 BMIN에서 사용하는 웜홀 라우팅 보다 현저히 적은 값이다. 이 결과에서, HWCR 방법은 웜홀 라우팅 방법 보다 통신 지연이 적어지므로, 다양한 통신 트래픽 형태를 지원할 수 있으며, 또한 웜홀의 단점인 복잡한 트래픽 하에서도 본 기법은 좋은 성능을 기대할 수가 있다.

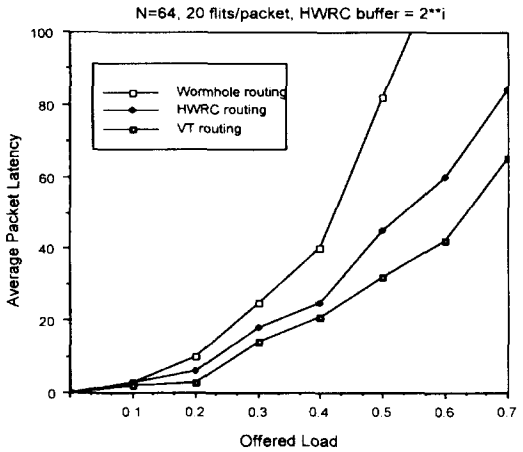

그림 8. 평균 패킷 지연시간($N=64$, packet : 20 flits)
Fig. 8. Average Packet Latency (N=64, packet : 20 flits).

3. 평균 버퍼 크기

버퍼 요구 정도를 측정하기 위해서 각 SE에서 사용된 평균 패킷 버퍼의 크기를 척도로 사용하였다. 그림 9의 그래프는 스테이지 당 평균 버퍼 크기를 보여주고 있다.

이 그래프에 의하면, 스테이지 0와 스테이지 5 에서는 패킷 버퍼가 필요 없었으며, 2, 3 스테이지에 최대 VCT 에 설치한 패킷버퍼를 설치하면 된다. 그러므로 제안된 HWCR 기법이 VCT 보다 훨씬 적은 양의 버퍼를 필요로 함을 알 수 있다. 그리고 입력 부하가 적을 시에는 실제 설계된 버퍼 양 보다 상당히 적은 양만이 사용됨(평균적으로 각 SE의 버퍼에 저장된 패킷이 적을 때)을 보여준다.

V. 결 론

본 논문에서는 트랙픽이 많은 웜홀 스위칭하에서 링크의 충돌이 발생하였을 때, 성능이 급격히 저하되는 것을 방지하기 위한 HWCR 라우팅 기법 및 버퍼 할당기법을 제안하였다.

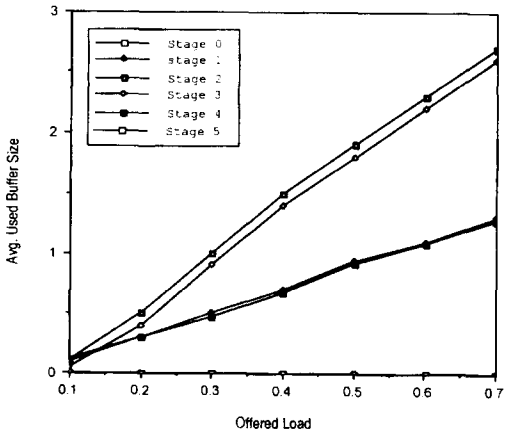

그림 9. 평균 버퍼 크기($N=64$, packet : 20 flits)
Fig. 9. Average Buffer Size (N=64, packet : 20 flits).

HWCR은 이러한 블럭킹이 일어난 상황에서 VCT 라우팅 기법의 장점을 혼합하여 라우팅하는 방법으로서, 링크의 블럭시 VCT 라우팅 방법으로 전환하여 플릿들의 흐름을 원활케하고, 다시 블럭된 링크가 해제되면, 원래의 웜홀 라우팅 방법으로 되돌아가는 방법이다. 이로인해 HWCR의 링크 자원을 효율적으로 사용하게 되어 네트워크 성능이 향상됨을 보였다. 또한 실험 결과 버퍼크기가 네트워크의 첫 몇 개 스테이지 부터 증가하다 이 후 몇 스테이지에서 선회 지점까지 계속 커질 필요가 없다는 버퍼 할당 전략의 구현 가능성을 제시하였다. 또한, 제안된 HWCR, VCT 라우팅, 웜홀 라우팅과의 시뮬레이션 결과에서 HWCR이 버퍼 크기와 통신지연에 의한 처리율이 우수함을 보였다. 차기 연구과제로는 복잡한 트래픽 패턴에 대한 성능분석과 하드웨어 비용과 성능간에 심도있는 분석, 그리고, 다양한 네트워크에서의 성능비교가 이루어 질 수 있을 것이다.

참 고 문 헌

[1] L. M. Ni and P. K. McKinley, "A survey of wormhole routing techniques in direct network," IEEE Computer, vol. 26, pp. 62-76, Feb. 1993.

[2] Kai Hwang, "Advanced Computer Archi-

tecture: Parallelism, Scalability, Programmability" pp. 5-18, 1993.

[3] C. Kruskal and M. Snir, "The Performance of multistage interconnection networks for multiprocessors," IEEE Transactions on Computers, vol. C-32, pp. 1091-1098, Dec. 1983.

[4] J. Ding and L. N. Bhuyan, "Finite buffer analysis of multistage interconnection networks," IEEE Transactions on Computers, vol. 43, pp. 243-247, Feb. 1994.

[5] Lionel M. Ni, Yadong Gui and Sherry Moore, "Performance Evaluation of Switch-Based Wormhole Networks," ICPP, Aug. 1995.

[6] W. J. Dally, "Virtual channel flow control," in Proc. of the 17th ISCA, pp. 60-68, May 1990.

[7] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessing systems," in Proc. of the First ISCA, pp. 21-28, 1973.

[8] Hyunmin Park and Dharma P. Agrawal, "WICI:An Efficient Switching Scheme for Large Scalable Networks," IEEE 6th PDP, 1994.

[9] Hong Xu, Ya-Dong Gui and Lionel M. Ni, "Optimal Software Multicaste in Wormhole-Routed Multistage Networks," IEEE 6th PDP, 1994.

[10] C.E.Leiserson, "Fat-Trees:Universal Networks for hardware-efficient supercomputing," IEEE Trans. on Computers, vol. C-34, pp. 892-901, Oct. 1985.

[11] ChangSoo Jang, SungChun Kim, "A Design and Analysis of A New Hybrid WC Routing Scheme in the Bidirectional Multistage Interconnection Network", the Fifteenth IASTED International Conference, Innsbruck, Austria, Feb. 19-22, 1996.

[12] C.S. Jang, S.C. Kim, "HWCR : A Cost-Effective Routing Scheme in Bidirectional Multistage Interconnection Network", The IASTED International Conference on MODELLING, SIMULATION AND OPTIMIZATION, Gold Coast, Australia, May 6-9, 1996.

저 자 소 개

張 昶 洙(正會員)

1976년 3월 ~ 1980년 2월 조선대학교 공과대학 전자공학과 졸업. 1980년 9월 ~ 1982년 8월 건국대학교 대학원 전자공학과 졸업. 1990년 9월 ~ 1993년 서강대학교 대학원 전자계산학과 수료. 1984년 10월 ~ 현재 국립 여수 수산대학교 컴퓨터공학과 부교수. 1991년 3월 ~ 1992년 2월 여수 수산대학 전자계산소 소장. 1995년 12월 ~ 현재 여수반도 지역 정보센터 사무국장 및 이사. 1996년 2월 ~ 현재 여수지역 정보센터 사무국장 및 이사. 관심분야는 병렬처리 시스템, 마이크로프로세서, 고속컴퓨터네트웨크

金 聖 天(正會員)

1975년 서울대학교 공과대학 공업교육학(전기전공)학사. 1976년 ~ 1977년 동아 컴퓨터(주) Sys. Eng. 1977년 ~ 1978년 스페리 유니백 Sales Rep. 1979년 Wayne State Univ. 컴퓨터공학 석사. 1982년 Wayne State Univ. 컴퓨터공학 박사. 1982년 ~ 1984년 캘리포니아 주립대 조교수. 1984년 ~ 1985년 금성반도체(주) 책임 연구원. 1986년 ~ 1989년 서강대학교 공과대학 전자계산학과 전자계산소 부소장. 1986년 ~ 1991년 서강대학교 공돠대학 전자계산학과 학과장. 1985년 ~ 현재 서강대학교 공과대학 전자계산학과 조교수, 부교수, 교수. 1989년 ~ 현재 한국정보과학회 병렬처리시스템 연구회 부위원장, 위원장 대한전자공학회 및 한국통신학과 논문지 편집위원. 관심분야는 병렬처리 시스템(Parallel computer architecture, Interconnection network) 컴퓨터 네트워크(이동통신, ATM 네트워크)