論文97-34C-5-3

NUMA(Non-Uniform Memory Access) 모델 시스템을 위한 Cost-Effective한 다단계 상호연결망

(Cost-Effective Multistage Interconnection Network for NUMA Model System)

崔 昌 勳 *, 金 聖 天 *

(Chang Hoon Choi and Sung Chun Kim)

요 약

지금까지 전통적인 UPP MIN에서 중복 경로를 제공하기 위한 다중 경로 MIN은 스테이지 추가, 중복 데이타 링크, 및 MIN의 복사본등의 추가적인 하드웨어를 첨가하므로써 실현되었다. 그리고 전통적인 MIN에서는 지역 참조성을 활용할 수가 없다 : 즉, 모든 프로세서-메모리 쌍간의 통신은 동일한 시간이 소요되기 때문이다. 이러한 MIN에서의 지역 참조성 활용을 위한 연구도 현재까지는 거의 진행되지 않았다. 본 논문에서는 전통적인 MIN에서의 스위칭 소자의 갯수보다 훨씬 적은 2N-3개의 스위칭 소자를 사용하여 구성할 수 있는 새로운 위상의 MIN인 Hybrid MIN을 제안한다. 비록 Hybrid MIN이 2N-3개로서 기존의 MIN에 비해 적은 수의 스위칭 소자로 구성될지라도 FAC를 만족하고, 중복 경로를 갖게 된다(단, 각 프로세서당 2개의 memory module은 단일 경로임). 더우기 Hybrid MIN에서는 빈번한 데이타 통신(지역 참조성)을 갖는 쌍들간에는 빠른 경로를 제공한다. 이것에 대한 성능은 지역화된 통신의 정도를 변화시키면서 분석되었다.

Abstract

So far, the multiple path MINs to provide redundant paths in the traditional UPP MINs have been realized by adding additional hardware such as extra stages, duplicated data links, or multiple copies of the MIN. And the traditional MINs do not exploit locality: communication with all processor-memory pairs takes the same amount of time. Also so far there has been little progress for exploiting locality of reference in MINs. In this paper, we present a new topology MIN, Hybrid MIN that is constructed with 2N-3 SEs which is far fewer SEs than that of traditional MINs. Although the Hybrid MIN is constructed with 2N-3 SEs, the Hybrid MIN satisfies full access capability(FAC) and has redundant paths(but providing single path for 2 memory modules of each processor). Moreover the Hybrid MIN provides shortcut path between pairs which have frequent data communication(locality of reference). Its performance under varing degrees of localized communication is analyzed.

I. 서 론

다중프로세서 시스템에서 프로세서 및 메모리간의 상호통신(intercommunication)은 시스템의 성능을 저하시킬 수 있는 요인 중에 한가지이다. 따라서 효율적인 상호연결망의 선택은 그 시스템의 성능을 최대로 발휘할 수 있게 하는데 있어 필수적인 요인이 될것이다. 이러한 상호연결망의 위상은 정적 네트워크(static network)와 동적 네트워크(dynamic network)로 나뉘어 진다.

여러 정적 네트워크의 상호연결망중에서 n-cube (hypercube)[3],[11]는 임의의 두 노드간의 직경(diameter)이 n으로서 안정적인 거리를 갖고 있을 뿐만 아

* 正會員, 西江大學校 電子計算學科
(Dept. of Computer Science Sogang University)
接受日字:1996年10月29日, 수정완료일:1997年3月27日

니라 지역 참조성(잦은 데이타 통신)이 높은 노드와는 거리가 1인 이웃(direct neighbor)으로 연결되어, 자주 발생되는 통신에 대하여 짧은 지연을 보장하게 하였다. 그러나 이러한 hypercube는 네트워크의 크기의 증가(노드수의 증가)에 따른 추가적인 통신 포트(communication port)가 필요하게 된다. 따라서 hypercube의 노드에 대한 degree는 n으로서 네트워크의 크기에 비례하여 증가하게 되므로 hypercube의 위상(topology)은 진정한 확장성(scalability)을 갖고 있다고는 말할 수가 없다 [4],[6],[8].

이에 반해 이진 트리(binary tree)구조에서는 degree가 3(루트 노드는 2)으로 고정적으로 일정하므로 확장성 면에서는 좋은 상호연결망이라 할 수 있다. 그리고 지역성이 높은 노드와는 거리가 1인 이웃노드로 연결될 수 있어, 지역 참조 역시 활용할 수 있다. 이러한 트리 네트워크구조는 병렬처리 데이타 베이스 및 여러 응용 분야에서 많이 사용되고 있다[4],[6],[8]. 그러나 이들 이진 트리구조는 원거리 노드(remote node)간의 거리가 매우 길고, 또한 루트 노드에서 병목현상은 이진 트리의 단점이라 할 수 있다. 그러나 이진 트리에 추가적인 링크를 사용함으로써 원격 거리 노드간의 평균거리를 단축시킬 수 있을 뿐만 아니라 루트 노드로 향하는 통신의 양을 감소시킬 수 있는 연구가 J.R. Goodman등[6]에 의해 이루어 졌다. 따라서 본 연구는 각 노드에 링크를 추가한 이진트리의 위상과 아래에서 설명될 동적 네트워크 (dynamic network)의 위상을 결합하여, 이들의 장점들을 모두 얻을 수 있는 새로운 형태의 MIN(Multistage Interconnection Network)을 제안하고자 한다.

동적 네트워크 위상중에서 MIN은 hypercube와 동일한 $\log_2 N$의 직경을 가지고 있고, 네트워크의 크기의 변화에 무관한 일정한 크기의 스위칭 노드를 사용할 수 있을 뿐만 아니라 간단한 분산 자기제어(distributed self-routing)방식을 사용하여 라우팅을 할 수 있기 때문에 SIMD, MIMD등의 다중 프로세서 시스템(multiprocessor system)에서 많이 사용되고 있는 상호연결망중에 하나이다 [4],[5],[8],[12]. 지금까지 잘 알려진 2x2 스위칭 소자를 사용하는 전통적인 MIN으로는 Omega 네트워크 [4],[8], Baseline 네트워크[8], Generalized n-cube 네트워크[12]등 Banyan(또는 delta)형태의 MIN [5]을 그 예로 들수 있다. 그러나 이러한 전통적인 MIN에서는 한개의 근원지와 목적지간의 쌍(pair)에 대해서 단일 경로(UPP: Unique Path Property)만을 제공하기 때문에 네트워크상에서의 오류및 트래픽에 대한 대체 경로가 없어 시스템 성능이 크게 저하를 초래하게 된다. 더우기 모든 쌍간의 거리는 항상 $\log_2 N$(NxN MIN에서 스테이지수)이기 때문에 MPP(Massively Parallel Processing) 시스템[8]과 같이 많은 프로세서를 사용하는 시스템에서와 같이 네트워크의 크기가 커질 경우 그 지연 시간 또한 그에 비례하여 증가하게 된다. 일반적으로 정렬(sorting)및 Fast Fourier Transforms(FFT)과 같은 많은 알고리즘등은 이웃 노드들간의 지역 참조성이 있기 때문에 빈번한 데이타 교환이 발생하여[6] 이들 쌍에 대한 짧은 경로의 제공이 필요로 하지만 전통적인 MIN에서는 모든 통신 쌍들간에는 스테이지수와 동일한 거리가 항상 유지되기 때문에 지역 참조를 활용할 수 없게 된다. 이러한 지역 참조성의 손실은 시스템 성능을 저하시키는 한가지 요인이 될 수 있다. 일반적으로 단일 프로세서 시스템 환경에 있어서는 대부분의 메모리 참조는 메모리 위치(memory location)상에서 아주 적은 부분(small set)에서만 발생하게 된다. 이러한 연구는 성공적인 캐쉬를 기초로한 시스템(cache based system)의 발전을 꽤할 수 있게 되었다. 이와 유사하게, 다중 프로세서 시스템 환경하에서의 많은 대부분의 응용 프로그램에서는 프로세서간의 통신(interprocessor communication)은 주로 프로세서-메모리들의 작은 그룹(small group)에서 발생하게 된다[1],[4]. 따라서 수많은 프로세서를 갖는 대형 시스템에서 각 프로세서, 메모리 쌍간에 모두 동일한 길이의 연결 경로를 제공하기보다는, 통신이 자주 발생되는 작은 그룹에 더 빠른 경로를 제공함으로써 보다 향상된 시스템 성능을 얻을 수 있을 것이다. 아래의 예)는 참고문헌 [1]을 기초로하여 기존의 전통적인 MIN에서 통신의 빈도수가 높은 그룹에 빠른 경로를 제공함으로써 얻게되는 잇점을 보인 것이다.

Communication Between Processors
(Total communication of each processor normalized to 1)

	P1	P2	P3	P4
P1		0.7	0.2	0.1
P2	0.7		0.1	0.2
P3	0.2	0.1		0.7
P4	0.1	0.2	0.7	

예) 한 시스템에 4개의 프로세서(P1 ~ P4)가 있다고

하자. 그리고 이들 프로세서간의 통신 빈도수를 측정한 결과 아래 표와 같이 산출되었다고 하자.

만약 이들을 2개의 그룹 {P1,P2}와 {P3,P4}로 나누어 구성시킨다면, 이들에 대한 평균 통신 지연은, 0.7 x log2 + (0.2+0.1) x log4 = 1.3 으로써, 그룹화시키지 않았을 경우의 통신 지연, log4=2 보다 더 적은 통신 지연 시간을 얻을 수 있다. 따라서 일반적으로 많이 그리고 자주 사용되는 응용 프로그램의 통신의 형태(traffic pattern)를 추적도구(tracer)를 이용하여 이들에 대한 통신 분포를 알아낼 수 있다면, 이러한 프로세서-메모리 그룹에 보다 짧은 경로를 제공함으로서 시스템 성능을 보다 향상시킬 수 있을 것으로 기대할 수 있다. 따라서 MIN을 상호연결망으로 사용하고 있는 대형 컴퓨터 시스템의 성능을 향상시키기 위하여 기존 MIN에서의 이러한 연구를 진행시켜야 하는 것은 필수적인 것이다.

또한, 현재까지의 연구에서는 전통적인 단일경로 MIN에서 다중경로를 제공하기 위해 중복링크를 사용하거나 스테이지를 늘리거나 또는 네트워크의 다중 복사본을 사용하여 중복경로를 제공하여 왔다 [11,19],[14]. 그러나 현재 및 미래에 사용될 수십-수천개 이상의 프로세서를 사용하는 MPP 시스템에서는 이렇게 추가되는 하드웨어 비용으로 인한 시스템의 가격의 상승 또한 신중히 고려되어야 할 요소일 것이다.

이러한 문제점을 해결하기 위해 본 논문에서는 새로운 형태의 MIN인 Hybrid MIN을 제안한다. Hybrid MIN은 static 네트워크인 이진 트리 위상의 장점과 dynamic 네트워크인 MIN의 위상 장점을 결합한 상호연결망으로서 기존 전통적인 MIN에서 사용되는 스위칭 소자의 갯수($N/2 \times \log_2 N$)에 비하여, 훨씬 적은 갯수인 단지 2N-3개의 스위칭 소자만을 사용했음에도 불구하고 2개의 중복경로를 제공(단, 각 프로세서당 2개의 memory module은 단일 경로임)할 뿐만 아니라, 통신 빈도수가 높은 지역 참조의 경우에는 보다 빠른 경로를 제공함으로써 자주 발생되는 통신에 대한 지연 시간을 줄일 수 있게 되었다.

Ⅱ. 정적네트워크와 동적 네트워크의 위상 결합

1. 그래프 표현

제안된 Hybrid MIN을 구성할때 MIN의 일반성을 유지시키기 위하여 C.L.Wu와 T.Y.Feng[16]의 그래프 표현 모델을 사용하기로 하겠다. MIN의 그래프 표현에 있어 노드는 스위칭 소자를 의미하며, 에지(edge)는 통신 링크를 의미한다. 각노드는 2-input and 2-output의 degee로써 형성된다.

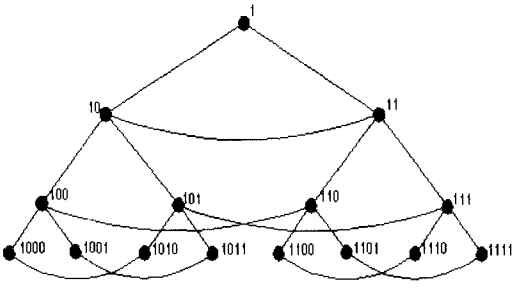


그림 1. Hypertree[6]
Fig. 1. Hypertree[6].

이진 트리등과 같은 정적 네트워크는 약결합 시스템(loosely coupled system)으로서 각 노드는 프로세서와 메모리 모듈이 결합된 PE(Processing Element)를 나타내며, 에지는 동적 네트워크와 동일한 의미를 갖게 된다. 이러한 정적 네트워크 위상을 dynamic 네트워크에 결합시키기 위하여 우선 이진 트리를 고려하기로 하겠다. <그림 1>과 같이 각 노드간에 추가적인 링크를 사용하여 수평 링크(horizontal link)를 제공함으로써 노드간의 평균 거리를 단축시킬 수 있고, 또한 루트로 향하는 통신의 양을 감소시킬 수 있다. 이렇게 Processor-Memory 구조상에서 이진 트리의 위상을 효율적으로 사용할 수 있도록 각 노드에 추가적으로 수평 링크를 제공한 트리를 본 논문에서는 Processor-Memory 트리 라고 하겠다. 이진 트리상에서 수평 링크를 이용한 연구로서 Hypertree[6]가 있었다. 그러한 Hypertree에서의 수평링크 연결은 <그림 1>에서와 같이 트리의 동일 레벨에서 노드들간의 2진 표현으로서, 한 비트만 서로 다른 노드들 중에서의 경로(path)의 길이가 가장 긴 노드들을 서로 연결시켜 이루어진다. 예를 들어 <그림 1>에서 레벨 3에 있는 노드 $8_{10}(1000_2)$이 고려되었을 경우, 이와 동일 레벨에 있는 노드들중에서 이진 표현상의 비트가 1개의 비트만 서로 다른 노드들은 노드 $9_{10}(1001_2)$, $10_{10}(1010_2)$, $12_{10}(1100_2)$ 이다. 그리고 이들간에 경로가 가장 긴 노드는 10_{10}이므로 노드 8_{10}은 노드 10_{10}과 수평 연결이 이루어 진다. 그러나 본 논문에서 사용될 수평 연결 방

법은 다음 절에서 정의 되는 것과 같이 Hypertree와는 전혀 다르게 형성되어질 것이다.

2. 동일 레벨에서의 수평 연결

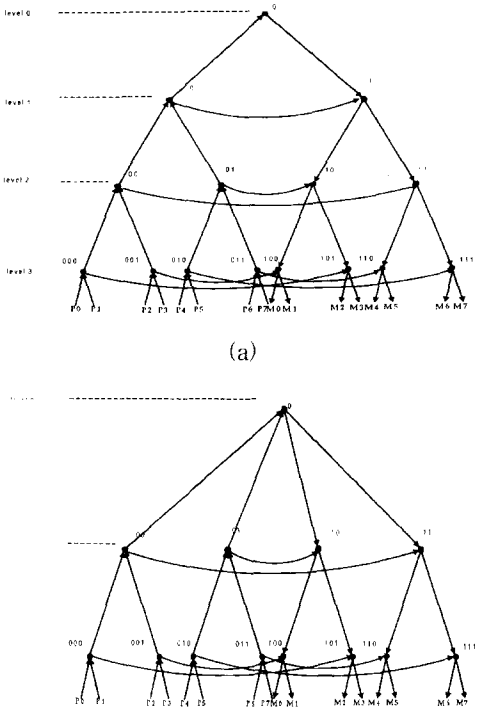
<그림 2(a)>에서와 같이 Processor-Memory 트리의 각 노드에는 이진 번호가 부여되어 있다. 루트 노드는 주소 0을 가지고 있다. 각 노드의 왼쪽 자식 노드는 그 부모 노드의 주소에 "0"을 붙여서 그 주소를 부여하였으며, 오른쪽 자식 노드는 그 부모 노드의 주소에 "1"을 붙여서 그 주소를 결정하였다. 즉, 노드 x의 자식노드들은 $2x$와 $2x+1$로서 각각 그 주소가 부여된다.

또한 동일 레벨간의 노드들을 수평 링크로 연결시킬 경우 그 수평 링크를 통해 연결된 한쪽의 특정 리프 노드(leaf node)로 경로의 수가 치우침이 없이 공평하게 이루어질 수 있도록 설계되어야 한다. 따라서 한 레벨에서 임의의 한 노드가 그 수평 링크의 연결을 통해 연결될 수 있는 리프 노드로의 연결 경로의 수가 공평히 배분될 수 있도록 하는 노드 선택 방법이 중요할 것이다.

[정 의 1]

이진 표현 $d_{n-1}d_{n-2}\ldots d_0$이 주어졌을 때, 임의의 $i((0\le i\le n-1)$에 대하여, 앞에서 n-i개의 이진 비트열 $d_{n-1}d_{n-2}\ldots d_i$를 고려하자. 함수 ρ 는 MSB-LSB-Complement 함수라고 하고, 그 표현은 $\rho(d_{n-1}d_{n-2}\ldots d_{i+1}d_i) = \overline{d_{n-1}}d_{n-2}\ldots d_{i+1}\overline{d_i}$와 같이 정의한다.

이진트리상에서 수평링크의 연결은 MSB-LSB complement function을 이용하여 동일 레벨에서의 노드들간의 수평 연결이 이루어진다. 이러한 Processor-Memory tree에서와 같은 그래프 표현에서의 에지는 MIN의 그래프 모델에서 사용되는 것과 같이 입력 링크와 출력 링크로써 표현될 수 있다. MIN에서의 그래프 모델에서의 각 노드는 2-input degree와 2-output degree를 갖는다. 그러나 <그림 2(a)>와 같은 Processor-Memory 트리에서는 루트 노드를 제외한 모든 노드는 2-input 과 2-output degree를 갖지만, 루트 노드는 1-input 과 1-output degree를 가진다. 따라서 MIN의 그래프 모델에서의 일반성을 그대로 유지할 수 있는 Processor-Memory 트리를 설계하기 위해서 level 1의 노드들을 그 트리에서 제거시킨다. 이렇게 함으로써 <그림 2(b)>와 같이 Processor-Memory 트리의 각 노드가 모두 2-input, 2-output을 갖는 구조로 재설계될 수 있다.



(a)

(b)

그림 2. Processor-Memory 트리의 동일 레벨에서 노드 연결

Fig. 2. Node connection in the same level of Processor-Memory tree.

[정 의 2]

Inheritance Number Set, **INS**$(d_{n-1}d_{n-2}\ldots d_i)$ = $\{d_{n-1}d_{n-2}\ldots d_id_{i-1}\ldots d_k\ldots d_0 \mid$ for all d_k=0 or 1, $0\le k\le i-1\}$을 정의한다. 따라서 inheritance number set, **INS**$(d_{n-1}d_{n-2}\ldots d_i)$는 n-i개의 MSB들, 즉 $d_{n-1}d_{n-2}\ldots d_i$로 시작되는 모든 이진 표현을 나타낼 수 있다.

임의의 한 디지트(digit) d_{n-1} = 0에 대해 **INS**(0) ∪ **INS** ($\overline{0}$) 는 총 2^n개의 숫자를 나타낸다. 즉, $\{d_{n-1}d_{n-2}\ldots d_i\ldots d_0 \mid d_i$=0 or 1, $0\le i\le n-1\}$. 따라서 이러한 0은 모든 수를 나타낼수 있는 근원이므로 atom number라고 하겠다. 또한 이진 트리상에서 atom number를 갖는 노드(루트 노드)를 atom number node라고 하겠다.

Processor-Memory 트리의 레벨 i $(0\le i\le n-1)$에서 만약 임의 한 노드, $d_{n-1}d_{n-2}\ldots d_i$가 고려되어 진다

면, **INS**($d_{n-1}d_{n-2}\ldots d_i$)로 부터 연결될 수 있는 모든 자식 노드로 연결될 수 있음을 표현한다. 또한 그 고려된 노드와 수평으로 연결될 노드는 ρ($d_{n-1}d_{n-2}\ldots d_i$)로써 결정된다. 다시 말해서, 고려된 한노드와 수평 연결은 그노드와 MSB와 LSB가 모두 다른 이진 번호를 갖는 노드와 연결시키는 것이다. 또한 그 노드와 연결될 수 있는 모든 자식 노드는 **INS**(ρ($d_{n-1}d_{n-2}\ldots d_i$)) 로써 모두 표현될 수 있다.

[정 리 1]

임의의 한 고려된 노드로 부터 MSB-LSB Complement function을 이용한 수평 링크에 의해 연결된 모든 자식 노드는 atom number node를 제외한 그의 모든 부모 노드로 부터 수평 링크로 연결되는 모든 노드와 다르게 구별된다. 이것은 Processor-Memory 트리에서 수평링크를 사용한 연결이 어떤 특정 리프 노드로 연결 경로의 갯수가 어느 한쪽 리프 노드에 치우침이 없이 공평히 이루어질 수 있다는 것을 의미한다.

증명)

Processor-Memory tree의 레벨 i에서 임의의 한노드 $d_{n-1}d_{n-2}\ldots d_i$를 고려할때, 그 고려된 노드와 수평 링크를 통해 연결될 수 있는 노드는 **INS**(ρ($d_{n-1}d_{n-2}\ldots d_i$))로 나타낼 수 있다. 또한 그 고려된 노드 $d_{n-1}d_{n-2}\ldots d_i$의 모든 부모 노드(atom number node는 제외)로 부터 수평 링크를 통하여 연결시킬 수 있는 모든 자식 노드는 $\bigcup_{j=i-1}^{n-2}$ **INS**(ρ($d_{n-1}d_{n-2}\ldots d_j$))로 표현될 수 있다. 따라서 이 정리를 증명하기 위해 **INS**(ρ($d_{n-1}d_{n-2}\ldots d_i$)) $\cap$ $\bigcup_{j=i+1}^{n-2}$ **INS**(ρ($d_{n-1}d_{n-2}\ldots d_j$)) = φ이라는 것을 증명하여 보이면 된다.

i) i=0에 대하여

$$\begin{aligned}\textbf{INS}(\rho(d_{n-1}\ldots d_0)) &= \textbf{INS}(\overline{d_{n-1}}\ldots\overline{d_0})\\ &\subseteq \textbf{INS}(\overline{d_{n-1}}\ldots d_1)\\ &=\{\overline{d_{n-1}}\ldots d_1d_0 \mid d_0=0 \text{ or } d_0=1\}\\ &=\{\overline{d_{n-1}}\ldots d_1\mathbf{0},\ \overline{d_{n-1}}\ldots d_1\mathbf{1}\}\end{aligned}$$

$\bigcup_{j=1}^{n-2}$ **INS**(ρ($d_{n-1}\ldots d_j$)) = $\bigcup_{j=1}^{n-2}$ **INS**($\overline{d_{n-1}}\ldots\overline{d_j}$),

따라서 **INS**($\overline{d_{n-1}}\ldots d_1$) $\cap$ $\bigcup_{j=1}^{n-2}$ **INS**($\overline{d_{n-1}}\ldots\overline{d_j}$) = φ 이므로, **INS**(ρ($d_{n-1}\ldots d_0$)) $\cap$ $\bigcup_{j=1}^{n-2}$**INS**(ρ($d_{n-1}\ldots d_j$)) = φ는 명백하다.

ii) i=k에 대하여,

INS(ρ($d_{n-1}\ldots d_k$)) $\cap$ $\bigcup_{j=k+1}^{n-2}$ **INS**(ρ($d_{n-1}\ldots d_j$)) = φ이라고 가정하자.

iii) for i=k+1에 대하여,

$$\begin{aligned}\textbf{INS}(\rho(d_{n-1}\ldots d_{k+2}d_{k+1})) &= \textbf{INS}(\overline{d_{n-1}}\ldots d_{k+2}\overline{d_{k+1}})\\ &\subseteq \textbf{INS}(\overline{d_{n-1}}\ldots d_{k+2})\\ &\subseteq \textbf{INS}(\overline{d_{n-1}}d_{n-2})\\ &=\{\overline{d_{n-1}}d_{n-2}d_{n-3}\ldots d_j\ldots d_0\\ &\quad \mid \textbf{for all } d_j, 0\le j\le n-3\}\end{aligned}$$

$\bigcup_{j=k+2}^{n-2}$ **INS**(ρ($d_{n-1}\ldots d_j$)) = $\bigcup_{j=k+2}^{n-2}$ **INS**($\overline{d_{n-1}}\ldots\overline{d_j}$)

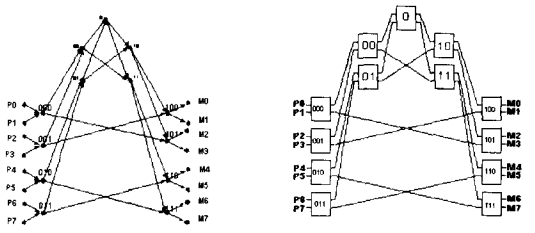
따라서 **INS**($\overline{d_{n-1}}\ldots d_{k+2}$) $\cap$ $\bigcup_{j=k+2}^{n-2}$ **INS**($\overline{d_{n-1}}\ldots\overline{d_j}$) = φ 이다. 그러므로 레벨 i에서 고려된 노드 $d_{n-1}d_{n-2}\ldots d_i$에 대하여 **INS**(ρ($d_{n-1}d_{n-2}\ldots d_i$)) $\cap$ $\bigcup_{j=i+1}^{n-2}$ **INS**(ρ($d_{n-1}d_{n-2}\ldots d_j$)) = φ이 성립한다. □

예를 들어 <그림 2 (a)>에서 레벨 3에 있는 노드 2(010) 와 노드 7(111)은 서로 수평 링크를 통하여 연결되어 있다. 즉 ρ(010)=111. 이 두 노드의 주소에서는 오직 MSB와 LSB의 bit들만 서로 다르다. 또한 이러한 수평 연결은 ***[정 리 1]***에 의해 공평한 수평 링크를 보장할 수 있다.

Ⅲ. Hybrid MIN의 구조

1. 그래프 모델에서 Hybrid MIN의 유도 형성

<그림 2(b)>에서와 같은 tree의 모든 노드들은 각각 2-input, 2-output degree를 가지고 있기 때문에 2x2 스위칭 소자를 갖는 MIN을 형성할 수 있게 된다. <그림 2 (b)>에서 tree의 아래 leaf 노드들중에서 프로세서 노드들은 왼쪽 그리고 메모리 모듈들은 오른쪽으로 양쪽을 잡아 늘린다면, <그림 3 (a)>와 같은 형태가 될것이다. 이와 같이 늘린 형태의 그래프에서 각 노드를 2x2 스위칭 소자로 바꾸어 표현하면, <그림 3 (b)>와 같은 MIN을 형성시킬 수 있다. 이렇게 형성된 MIN을 8x8 Hybrid MIN이라고 하겠다.



(a) (b)

그림 3. 8x8 Hybrid MIN의 유도
Fig. 3. Derivation of Hybrid MIN.

2. Hybrid MIN의 위상

한개의 NxN Hybrid MIN은 N개의 입력과 N개의 출력 터미날을 가지고 있다.(여기서 N은 2의 멱승, 즉 $n=\log_2 N$이다. 또한 Hybrid MIN은 총 2n-1개의 스테이지를 가지고 있다. 이러한 Hybrid MIN의 구성에 있어서 스테이지 $i(0\le i\le n-2)$에서는(이하 FRONT_STAGE라고 칭함) $2^{n-i}/2$개의 스위칭 소자들로 구성된다. 스테이지 $j(n-1\le j\le 2n-2)$에서는(이하 REAR_STAGE라고 칭함) $2^{j-n+2}/2$개의 스위칭 소자로 이루어졌다. 그러므로 한개의 NxN Hybrid MIN은 총 2N-3개의 스위칭 소자로 구성된다.

이미 앞에서 보여진 <그림 3 (b)>에서의 스위칭 소자에 대해 부여된 2진 표현은 이진 트리의 위상의 속성에서 Hybrid MIN을 유도하기 위해 부여된 번호이다. 따라서 Hybrid MIN의 독자적인 고유의 위상 설정을 위해 Hybrid MIN의 각 스위칭 소자의 번호를 아래와 같은 방법으로 재 부여되어야 할 것이다.

NxN Hybrid MIN의 위상을 표현하기 위하여, T.Y. Feng과 S.W. Seo[13]에서 사용된 것과 유사한 방법을 사용하여 topology describing rule을 정의한다. FRONT_STAGE에서 각 스위치 노드는 스테이지 $i(0\le i\le n-2)$에 있는 물리적인 위치에 따라 이진 표현, $b_\ell b_{\ell-1}\ldots b_{i+1}$ (여기서 $\ell=n-1$)을 사용하여 나타낸다. 각 스위칭 소자에 연결된 각 링크는 이진 표현 £$(b_\ell b_{\ell-1}\ldots b_{i+1}0)$와 £$(b_\ell b_{\ell-1}\ldots b_{i+1}1)$로써 레이블을 붙였다. 한 스위칭 소자에 연결된 출력 터미날의 링크 표현에 있어서 처음 $\ell-i$개의 bit들은 그 스위칭 소자의 번호의 이진 표현과 동일하고 마지막 디지트는 0과 1로서 UP 링크(link)와 DOWN 링크를 각각 나타낸다.

REAR_STAGE에서 각 스위칭 소자는 스테이지 j에 있는 물리적인 위치에 따라 이진 표현($b_\ell b_{\ell-1}\ldots b_{2n-j-1}$)으로서 나타낼 수 있다 (여기서 $n-1\le j\le 2n-2$, $\ell=n-1$). REAR_STAGE에서 각 링크의 표현방법은 FRONT_STAGE에서의 표현과 동일하다.

[정 의 3]

Hybrid MIN의 topology describing rule은 아래와 같이 나타낼 수 있다. f_i 또는 r_j에서 상위 첨자 0(1)은 그 스위칭 소자의 UP(DOWN) 링크를 나타낸다.

FRONT_STAGE Topology Describing Function

$f_i^0\ (b_\ell b_{\ell-1}\ldots b_{i+1})_i=(b_\ell b_{\ell-1}\ldots b_{i+2})_{i+1}$
for link £$(b_\ell b_{\ell-1}\ldots b_{i+1}0)$, at stage $i\ (0\le i\le \ell-2)$

$f_i^0\ (b_\ell)_{\ell-1}=(0)_\ell$
for link £$(b_\ell 0)$, at stage $\ell-1$ (for atom node)

$f_i^1\ (b_\ell b_{\ell-1}\ldots b_{i+1})_i=(b_\ell b_{\ell-1}\ldots \overline{b_{i+1}})_{2n-2-i}$
for link £$(b_\ell b_{\ell-1}\ldots b_{i+1}1)$, at stage i $(0\le i\le \ell-1)$

REAR_ STAGE Topology Describing Function

$r_j^0\ (b_\ell b_{\ell-1}\ldots b_{2n-j-1})_j=(b_\ell b_{\ell-1}\ldots b_{2n-j-1}0)_{j+1}$
for link £$(b_\ell b_{\ell-1}\ldots b_{2n-j-1}0)$

$r_j^1\ (b_\ell b_{\ell-1}\ldots b_{2n-j-1})_j=(b_\ell b_{\ell-1}\ldots b_{2n-j-1}1)_{j+1}$
for link £$(b_\ell b_{\ell-1}\ldots b_{2n-j-1}1)$, at stage $j\ (\ell\le j\le 2\ell-1)$.

이것을 간단히 표현하면, $r_j^{0(1)}\ (b_\ell b_{\ell-1}\ldots b_{2n-j-1})_j=(b_\ell b_{\ell-1}\ldots b_{2n-j-1}X)_{j+1}$ 이된다. (여기서 bit X는 don't care bit, 즉 0 또는 1 이다)

<그림 4>는 16x16 Hybrid MIN에서 스위칭 소자와 링크에 부여된 번호 및 연결 형태를 예로써 보인 것이다. 예를 들어, 스테이지 0에 있는 스위칭 소자 (011)는 스테이지 1에 있는 스위칭 소자(01)과 스테이지 6에 있는 스위칭 소자 (010)와 각각 UP 링크 £(0110)와 DOWN 링크 £(0111)를 통하여 연결되어 있다.

3. Hybrid MIN에서 Non-Uniform Memory Access(NUMA) 모델의 구현

Hybrid MIN은 NUMA 모델(model)로서 통신의 빈도수에 따라 메모리를 엑세스(access)하는데 소요되는 시간을 달리하여 구성할 수 있다.

[정 의 4]

NxN Hybrid MIN에서 n-레벨 클러스터(level cluster)를 정의한다. Hybrid MIN에서는 트래픽 로드(traffic load)에 따라 n개의 레벨 클러스터 L_i(1 ≤i≤n)가 존재한다. 임의의 한 프로세서 P($P_\ell P_{\ell-1}\ldots P_0$)에서 연결시킬 수 있는 각 레벨 클러스터 L_i는 아래와 같다.

2^1 Memory Modules, $P_\ell P_{\ell-1}\ldots\overline{P_i}X_{i-1}\ldots X_1X_0$ for L_i, $1\le i\le \ell$ ①

2^n Memory Modules, $X_{n-1}\ldots X_0$ for L_i, $i = n$ ② (여기서 X들은 don't care bits이다). <표 1>은 ***[정 의 4]***를 정리한 것이다.

표 1. 레벨 클러스터에 따른 Memory Module
Table 1. Memory Modules according to the level cluster.

memory / window	Memory Modules
level cluster L_1	2^1 Memory Modules, $P_\ell P_{\ell-1}\ldots\overline{P_1}0$, $P_\ell P_{\ell-1}\ldots\overline{P_1}1$
level cluster L_2	2^2 Memory Modules, $P_\ell P_{\ell-1}\ldots\overline{P_2}00$, $P_\ell P_{\ell-1}\ldots\overline{P_2}01$, $P_\ell P_{\ell-1}\ldots\overline{P_2}10$, $P_\ell P_{\ell-1}\ldots\overline{P_2}11$
...	.....
level cluster L_i	2^1 Memory Modules, $P_\ell P_{\ell-1}\ldots\overline{P_i}X_{i-1}\ldots X_1X_0$
...	....
level cluster L_n	2^n Memory Modules,, $X_\ell X_{\ell-1}\ldots X_0$

여기서 임의의 한 프로세서 $P_\ell P_{\ell-1}\ldots P_0$가 모든 메모리와 연결할 수 있다는 FAC를 만족시키기 위해서 이들이 n개의 모든 레벨 클러스터에 있는 메모리 모듈과 연결이 가능한다는 것을 보여야 한다.

4. Hybrid MIN의 특성

[정 리 2]

각 레벨 L_i (1≤i≤n)에 있는 모든 메모리 모듈은 프로세서 P($P_\ell P_{\ell-1}\ldots P_0$)와 연결될 수 있다.

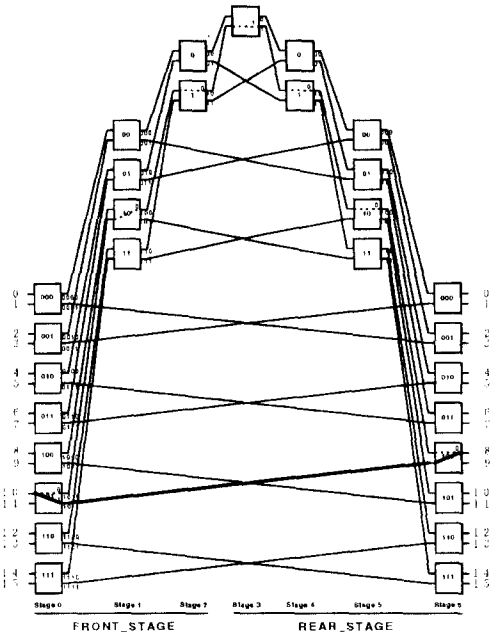


그림 4. 16x16 Hybrid MIN
Fig. 4. 16x16 Hybrid MIN.

증명)

모든 레벨 L_i (1≤i≤n)에 대해, 이 정리는 topology describing function을 적용시켜 증명될 수 있다. 만약 topology describing function으로 부터 유도된 2진 표현이 ***[정 의 4]***에 의한 Table I의 L_i에서의 2진 표현과 같다면, 이 정리의 증명은 명백해 진다. 연산자 ⊙를 composite mapping [13]이라고 하면 FRONT_STAGE에서 topology describing function f를 사용하여 UP 또는 DOWN 링크를 통하여 도달시킬 수 있는 스테이지 k에 있는 스위칭 소자를 $[\,f_{k-1}^{0(1)} \odot [\,\ldots \odot [\,f_1^{0(1)} \odot [\,f_0^{0(1)}\,]\,]\ldots] = \prod_{i=0}^{k-1} f_i^{0(1)}$와 같이 표현시킬 수 있다. (REAR_STAGE topology describing function r에서도 동일한 방법이 적용됨).

$$[\,\prod_{v=2n-i-1}^{2n-3} r_v^{0(1)} \odot [\,f_{i-1}^{1} \odot [\,\prod_{u=0}^{i-2} f_u^{0}\,]\,]\,]$$

$$= [\,\prod_{v=2n-i-1}^{2n-3} r_v^{0(1)} \odot [\,f_{i-1}^{1}\,(P_\ell\ldots P_i)_{i-1}\,]\,]$$

$$= [\,\prod_{v=2n-i-1}^{2n-3} r_v^{0(1)} \odot [\,(P_\ell\ldots\overline{P_i})\,]\,]$$

$$= P_\ell\ldots\overline{P_i}X_{i-1}\ldots X_1$$

따라서 스위칭 소자($P_\ell\ldots\overline{P_i}X_{i-1}\ldots X_1$)에 연결된

메모리 모듈은

$P_\ell \ldots P_{i-1}\ \overline{P_i}\ X_{i-1}\ldots X_1X_0$ for level i, $1 \le i \le \ell$ ..... ①′

$$\left[\prod_{v=n-1}^{2n-3} r_v^{\beta(1)} \circ \left[\prod_{u=0}^{n-2} f_u^{\beta}\right]\right]$$

$= X_{n-1}\ldots X_1$

따라서 스위칭 소자($X_{n-1}\ldots X_1$)에 직접 연결된 메모리 모듈은

$X_{n-1}\ldots X_1X_0$ for level n. ..... ②′

따라서 ①≡①′ and ②≡②′ 이므로 레벨 클러스터 L_i ($1 \le i \le n$)에 있는 메모리 모듈들은 어떤 프로세서 P($P_\ell P_{\ell-1}\ldots P_0$)와도 연결이 가능하다. □

Extra Stage Cube 네트워크[12],[12]등 기존의 하드웨어를 추가하여 중복 경로를 제공하는 MIN에서는 그 네트워크의 첫번째 스테이지(input stage)와 마지막 스테이지(output stage)를 제외한 모든 스테이지에서는 중복 경로간에는 switching-disjoint한 경로를 제공한다. 그러나 Hybrid MIN에서는 하드웨어 비용을 감소시켜 구현된 MIN이므로, switching-disjoint한 완전한 중복 경로를 제공할 수 없다. 그러나 Hybrid MIN에서는 과부하 통신(heavy traffic)이 발생하는 지점에서 부터는 그 경로를 우회하여 다른 경로를 선택할 수 있는 path-disjoint한 중복경로를 제공한다. 따라서 Hybrid MIN에서의 중복경로라 함은 과부하 통신이 발생한 지점에서 부터 그 경로를 우회할 수 있는 path-disjoint한 선택 가능한 다른 대체 경로를 의미한다.

[정 리 3]

NxN Hybrid MIN은 각 프로세서당, 단지 2개의 memory module을 제외한 모든 메모리 모듈에 대해서는 2개의 중복 경로를 가지고 있다.

증명)

i) 레벨 클러스터 L_n에 대해,

임의의 한개의 프로세서 $P_\ell P_{\ell-1}\ldots P_0$는 ***[정 리 2]***에 의해 레벨 L_n에 있는 어떤 메모리 모듈과도 연결시킬 수 있다. 그리고 레벨 클러스터 L_n은 모든 메모리 모듈을 포함하고 있다. 그러므로 한개의 프로세서와 메모리 모듈사이에는 최소한 한개의 경로가 존재한다.

ii) 레벨 클러스터 L_i($1 \le i \le n-1$)에 대해,

한 프로세서 $P_\ell P_{\ell-1}\ldots P_0$에 의해 연결된 스위칭 소자 $P_\ell P_{\ell-1}\ldots P_1$는 프로세서 동일한 번호를 갖는 2개의 메모리 모듈, 즉 $P_\ell P_{\ell-1}\ldots P_10$ 와 $P_\ell P_{\ell-1}\ldots P_11$ (그 프로세서가 직접 연결된 입력 스테이지에 있는 스위칭 소자의 번호와 동일한 번호를 갖는 마지막 스테이지의 스위칭 소자에 직접 연결된 메모리 모듈로서 ***[정 의 5]***에서 정의된 레벨 i에서는 그 번호를 갖는 메모리 모듈은 속할 수 없음)을 제외한 모든 메모리 모듈들은 ***[정 리 2]***의 식 ①′에 의해 연결될 수 있다. 따라서 i)과 ii)에 의해 한개의 NxN Hybrid MIN은 각 프로세서당 단지 2개의 memory module을 제외한 모든 쌍들에 대해서 2개의 대체 경로가 존재하게 된다. □

[정 리 4]

NxN Hybrid MIN는 FAC(*Full Access Capability)*를 만족한다.

증명) 이 정리는 ***[정 리 2]***과 ***[정 리 3]***에 의해 근원지 S와 목적지 D간에 최소한 1개이상의 경로가 존재하는 것을 보였으므로 사실이 명백하다. 따라서 Hybrid MIN 은 FAC를 만족시킨다. □

Ⅳ. 라우팅 알고리즘

Hybrid MIN에서는 빠르고 쉬운 라우팅을 위해서 목적지 태그(destination tag)를 사용한 자기제어 라우팅 (self-routing) 방식을 적용시킬 수 있다. Hybrid MIN에서 두쌍간의 통신을 위해 적게는 2개의 스테이지에서 부터 최대 2n-1개의 스테이지를 가지고 있기 때문에 라우팅 태그의 길이 또한 2(best case)에서 2n-1(worst case)까지 다양해 질 수 있다. 또한 Hybrid MIN은 대체 경로가 존재하므로, 비록 트래픽의 영향 때문에 생성된 라우팅 태그의 경로가 블럭(busy)될지라도 우회할 수 있는 대체 경로를 위한 라우팅 태그를 재 설정시키는 능력 또한 가지고 있다.

ROUTING TAG GENERATION ALGORITHM:

```
/*
        Source Address : Sn-1Sn-2 ... S1S0
        Destination Address : Dn-1Dn-2 ... D1D0
*/
   begin
```

```
i = n-1;
while (i > 0 and Si = Di) i = i-1;
if (i=0) then
        TAG = (n-1 0's)(Dn-1Dn-2 ... D0);
        /*짧은 경로 태그 결정 (최단거리)*/
      else
        begin
          TAG = (i-1 0's)(1)(Di-1 ...D0); /*
          짧은 경로 태그 결정 */
          if (the path is busy) TAG =
          (n-1 0's)(Dn-1Dn-2 ... D0); /* 대체
          경로 태그 결정 */
        end
end.
```

예를 들어 <그림 4>에서 근원지 $10(1010_2)$와 목적지 $8(1000_2)$를 연결시키기 위한 라우팅 태그는 위의 알고리즘에 의해 S1 ≠ D1이므로 10_2이 된다. 만약 이 경로가 이미 다른 노드의 통신을 위해 사용되고 있다면(busy), 대체 경로를 위한 라우팅 태그는 목적지 주소의 전위 부분에 3개의 0을 추가시켜서 대체 경로 설정을 할 수 있게 된다. <그림 4>에서 굵은 선은 레벨 클러스터 L_1에 있는 짧은 라우팅 경로를 나타낸다. 그리고 점선은 라우팅 태그가 0001000_2로서 대체 경로를 나타낸다.

V. 성능 분석

본 절에서는 Hybrid MIN에 대한 성능을 평가하도록 하겠다. 전통적인 MIN은 어떠한 입출력 연결로부터 그 거리가 고정되어 있어, 항상 두 쌍간의 거리가 일정한 반면, Hybrid MIN에서는 통신의 지역 참조성의 정도에 따라 그 거리는 달라지게 된다. 이때, 각 프로세서는 참조될 메모리 모듈에 대한 거리(distance)는 2에서 부터 2n-1까지 다양한 길이를 가질 수 있다. 즉 메모리 액세스 시간이 메모리 모듈의 위치에 따라 서로 다르기 때문에 이것을 NUMA 모델 시스템 환경이라 할 수 있다. 다시 말해서, 프로세서와 메모리 모듈 사이에서 통신의 빈도수가 높은 쌍들에 대해서는 메모리 액세스 시간을 줄이기 위해 짧은 거리의 링크를 부여하고, 통신 빈도수가 낮은 메모리 모듈들은 보다 통신 거리가 긴 링크를 부여하도록 하여 메모리 액세스 시간의 차이를 두는 것이다. 이를 위하여 아래와 같은 임의의 한 프로세서와 메모리 모듈사이의 통신거리에 따라 n가지의 window를 정의한다. 예를 들어 임의의 프로세서에서 window=1에 속하는 메모리 모듈들은 모두가 통신 거리가 2임을 나타낸다. 또는 window=2에 속하는 메모리 모듈들은 통신 거리가 2에서 4까지임을 나타낸다. 그리고 window=n에 속하는 메모리 모듈들은 통신거리가 2에서 최대 2n-1까지의 거리, 즉 모든 메모리 모듈을 포함하는 것을 의미한다.

따라서 경로의 길이를 기초로 하여 형성된 window에 대한 메모리 참조 확률을 적용함으로써 통신 형태가 국부지향적(localized communicaton pattern)에서부터 균일 분포의 통신형태 (uniform communication pattern) 또는 원거리 지향적 통신 형태 (remote-request communication pattern) 까지의 모든 가능한 통신 형태를 검증할 수 있다. 따라서 각 window의 크기를 변화시키면서, 또한 각 window에서 통신, 즉 메모리 참조가 발생될 확률, α를 변화시키며, (따라서 window 밖에의 메모리 모듈 참조가 발생될 확률은 $(1-\alpha)$), Hybrid MIN에 대한 성능을 분석하였다.

i) Window W : ($1 \le W \le n$)

임의의 input-output connection에 대해, 아래와 같은 n개의 window들이 존재한다. 각 window는 ***[정의 4]***에서 정의된 n개의 레벨 클러스터 L_1, L_1, L_2 ... L_n를 포함하고 있다. 그 거리(distance)는 Hybrid MIN에서 통신을 위해 통과하여야 하는 스테이지 수이다.

W=1 : distance = 2, (L_1)
W=2 : distance = 2, 4 (L_1, L_2)
.
.
.
W=n-1 : distance = 2,4, ..., (n-1) x 2 (L_1, L_2, ... , L_{n-1})
W=n : distance = 2,4, ..., (n-1) x 2 +1 (L_1, L_2, ... , L_n)

ii) degree of locality (α)

'α'을 window W의 범위에 속하는 메모리 모듈을 참조하는 확률, 즉 지역 참조 정도(degree)를 나타낸다.

iii) 프로세서와 메모리간의 평균 거리를 평가하는 방법은 우선 균일 분포를 가정하고, 그 네트워크에서

프로세서와 메모리 쌍간의 평균 거리를 결정한다. 평균거리율(average distance ratio)은 Hybrid MIN과 동일한 네트워크 크기를 갖는 전통적인 MIN에 대한 평균 거리의 비율을 나타낸다. 그러나 이러한 균일 분포는 Hybrid MIN에 대해서는 worst case의 경우이다. Hybrid MIN의 균일 분포하에서의 평균 거리는 다음과 같이 구할 수 있다. 먼저, 한개의 프로세서에서 메모리 모듈간의 거리가 $2i(1 \le i \le n-1)$를 갖는 메모리 모듈의 갯수는 2^i개이고, 거리가 2n-1인 메모리 모듈의 갯수는 2개가 존재한다. 따라서 평균 거리는 $\{ \sum_{i=1}^{n-2} 2^i 2i + 2(2n-1) \} / N = 2n-4+(4n+2)/N$ 이다. 그리고 전통적인 MIN에서의 프로세서와 메모리 쌍간의 평균 거리는 n이므로 아래와 같은 평균 거리율에 대한 식을 산출할 수 있다.

Avg. distance ratio = $[2n-4+(4n+2)/N] / n$

iv) 지역 참조 정도 α와 window W를 고려한 평균 거리율의 평가 방법은 그 평균 거리와 그 거리가 통신의 지역 참조에 관련될 확률의 곱으로서 이루어진다. window 크기 (W=i)가 주어졌을 경우, W=i는 또한 W=1에서 부터 W=i-1의 window를 모두 포함하게 되므로, 주어진 임의의 한 프로세서가 window 크기, W=i의 내부에 있는 메모리 모듈과 통신할 수 있는 평균 거리는 $\sum_{i=1}^{W} 2^i 2i / \sum_{i=1}^{W} 2^i$이고, 이렇게 주어진 window에 속하는 메모리 참조에 대한 참조 확률 정도, α를 부과했을 경우, weighted average distance는 $\alpha \times \sum_{i=1}^{W} 2^i 2i / \sum_{i=1}^{W} 2^i$가 된다. 또한 주어진 window 밖의 메모리 참조가 발생할 확률은 $(1-\alpha)$이므로, 이에 대한 weighted average distance는 $(1-\alpha) \times \sum_{j=W+1}^{n-2} 2^j 2j + 2(2n-1) / (\sum_{j=W+1}^{n-2} 2^j + 2)$가 된다. 그리고 전통적인 MIN에서의 프로세서와 메모리 쌍간의 평균 거리는 α와 관계없이 항상 n으로 일정하다. 따라서 위의 두 경우를 모두 고려한 평균 거리율에 대한 식은 아래와 같다.

$$\text{Avg. distance ratio} = \frac{\alpha \sum_{i=1}^{W} 2^i 2i}{\sum_{i=1}^{W} 2^i} + (1-\alpha) \left\{ \frac{\sum_{j=W+1}^{n-2} 2^j 2j + 2(2n-1)}{\sum_{j=W+1}^{n-2} 2^j + 2} \right\} / n, \text{ for } 1 \le W \le n$$

v) 스위칭 소자의 수의 비율(ratio of total number of SE)은 전통적인 MIN에서 사용되는 전체의 스위칭소자 갯수에 대하여, 그와 동일한 네트워크 크기를 갖는 Hybrid MIN에서 사용되는 총 스위칭 소자의 갯수에 대한 평균율을 나타낸다. 전통적인 MIN에서 사용되는 스위칭 소자의 총 갯수는 nN/2이고, Hybrid MIN에서 사용된 스위칭 소자의 총 갯수는 2N-3 이므로 스위칭 소자의 수의 비율은

The ratio of total number of SEs = (2N-3) / (nN/2) 이다.

<그림 5>는 1024x1024 Hybrid MIN에서 지역 참조도(α) 와 window (W)의 변화에 따른 평균 거리율의 변화를 살펴 본 것이다. 다시 말해서 Hybrid MIN과 동일한 네트워크 크기를 갖는 전통적인 MIN에 대한 평균 거리의 비율을 나타낸 것이다. <그림 5>에서 window size가 크고, 지역 참조성이 낮을 경우, 다시 말해서 균일 분포의 통신 환경하에서는 Hybrid MIN의 평균거리가 기존 전통적인 MIN보다는 길어지어, 성능이 저조함을 나타내지만, 앞에서도 언급했듯이 대부분의 응용 프로그램들은 적은 수의 프로세서-메모리 쌍간의 그룹에서 대부분의 통신이 발생하게 되므로, window 크기가 작고, 지역 참조성이 높은 통신 분포 형태의 환경하에서는 Hybrid MIN이 성능이 우수해지는 것을 볼수 있다. 예를 들어 <그림 5>에서도 잘 나타나 있듯이, window크기가 1일 경우에서는 참조율이 0.5이상 부터, window 크기가 4일 경우는 참조율이 0.7 부터 기존의 MIN 보다 우수한 성능을 나타내고 있어 Hybrid MIN은 위와 같은 환경하에 매우 적합한 네트워크임을 알 수 있다. <그림 6>은 1024x1024 Hybrid MIN의 여러 window에 따라 cost-effectiveness ratio를 보인것이다. 이것은 Hybrid MIN과 동일한 네트워크 크기를 갖는 전통적인 MIN에서의 Cost-effectiveness에 대한 비율을 나타낸 것이다. 여기서 Cost-effective ratio는 그 네트워크에서 사용되는 스위칭 소자의 갯수에 평균 distance ratio를 곱한 것이다. 평균 distance ratio가 성능에 반비례하게 되므로, cost-effenceness의 값이 낮아질수록 더 좋은 cost-effectiveness를 갖게 된다. <그림 6>에서도 볼 수 있듯이, 기존 MIN보다 매우 적은 수의 스위칭 소자를 사용하고있기 때문에 지역적 참조율이 낮을 지라도, cost-effectiveness 비율이 전통적인 MIN에서 보

다 우수함을 나타내고 있다.

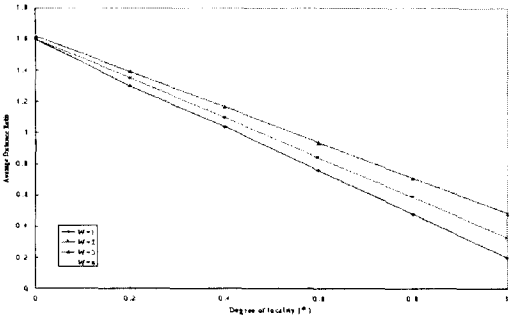


그림 5. window 크기(W) 와 지역 참조도(α) 의 변화에 따른 평균 지연 시간율
Fig. 5. Average distance ratio with variation of window size(W) and degree of locality (α).

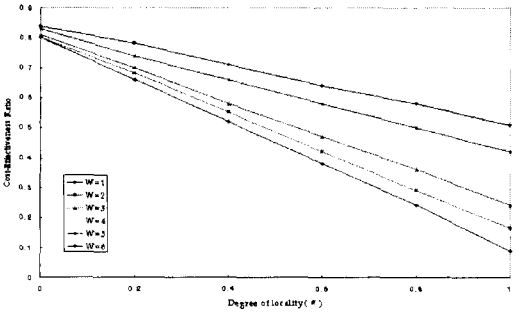
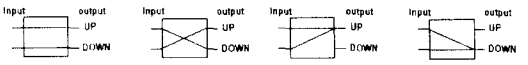


그림 6. window 크기(W) 와 지역 참조도(α)의 변화에 따른 cost-effectiveness
Fig. 6. Cost-effectiveness ratio with variation of window size(W) and degree of locality (α).

다음은 Hybrid MIN에 대한 예상되는 Bandwidth(BW)를 분석해 보도록 하자. 한개의 2x2 스위치 모듈에서 단위 시간당 각 출력 링크로 통과될 수 있는 요청율은 $1-(1-m/2)^2$이다 (여기서 m은 cycle 당 각 입력에서 발생하는 요청의 평균수) [10]. 그러나 Hybrid MIN에서 사용되고 있는 2x2 스위치 모듈은 균일 트래픽(traffic)하에서 각 출력 링크에 요청하는 비율이 다르다. 따라서 스테이지 i ($0 \le i \le n-1$))에 있는 모든 2x2스위칭 스위칭 소자에 대해, 2개의 입력 링크에 m=1의 요청율을 가정했을 경우, 단위 시간당 그 요청이 출력링크를 통과할 수 있는 요청율 M_{out}은 다음과 같이 구할 수 있다. 2x2 스위칭 소자의 상태는 다음과 같이 4가지 경우가 존재하게 된다



(a) (b) (c) (d)

그림 7. 2x2 스위칭 소자에서의 4가지 연결 상태
Fig. 7. 4 Connection states of 2x2 SE.

<그림 7>의 (a)와 (b)에서와 같이 2x2 스위칭 소자의 두 입력들이 2개의 출력 포트(port)들의 각각에 충돌없이 연결시킬 수 있는 경우는 2가지의 경우가 존재한다. 또한 Hybrid MIN에서의 스테이지 $i(0 \le i \le n-1)$에 있는 2x2 스위칭 소자의 두개의 출력 링크 Up과 Down으로 연결시킬 수 있는 목적지의 갯수는 각각 2^{i+1} 과 $N-2^{i+1}$개이다. 따라서 2x2 스위칭 소자가 서로 충돌없는 형태가 이루어 질 수 있는 확률은 $2(2^{i+1}/N \times (N-2^{i+1})/N)$이다. 그리고 <그림 7>의 (c)는 2x2 스위칭 소자에서 두개의 입력 요청이 출력 포트 UP으로의 연결을 요청한 경우이다. 이때, 2개의 입력 요청중에서 1개만이 선택되어지고, 나머지 입력 요청은 거부(drop)된다. 따라서 두개의 입력요청이 모두 동일한 출력 포트 Up에 대한 요청이면서, 이들 요청을 통과시킬 수 있는 요청 확률은 $((N-2^{i+1})/2)^2/2$ 이다. 그리고 <그림 7>의 (d)는 두개의 입력 요청이 모두 출력 포트 Down으로 향하는 요청으로서, 이들 요청을 통과시킬 수 있는 확률은 $(2^{i+1}/N)^2/2$ 이다. 따라서 단위 시간당 각 출력 링크를 통과할 수 있는 요청율, M_{out}에 대한 식은 아래와 같다.

$$M_{out} = 2\times(2^{i+1}/N\times(N-2^{i+1})/N)+((N-2^{i+1})/N)^2/2+(2^{i+1}/N)^2/2$$

여기서 2^{i+1}, $(0 < 2^{i+1} \le N)$를 고려해 보자. 이는 NxN Hybrid MIN에서 스테이지 i로 부터 도달될수 있는 목적지의 갯수를 나타낸다. 이것을 다시 정리하여 표현하면 $0 \le 2^{i+1}/N \le 1$ 된다. 여기서 $2^{i+1}/N = \alpha$라고 하자. 이는 균일 분포 통신하에서 참조율을 나타낸다. 그러나 비균일 분포(non-uniform communication)상에서 α는 스테이지 i와는 무관하게 그 출력 링크를 통해서 지역 참조할 수 있는 degree로서 표현될 수 있다. 따라서 α를 고려했을 때, 단위 시간당 각 출력 링크(Up/Down)를 통과할 수 있는 각각의 요청율은 아래와 같다.

$$M_i^{Up} = 2\alpha(1-\alpha)+(1-\alpha)^2/2 \text{ and}$$

$$M_i^{Down} = 2\alpha(1-\alpha) + \alpha^2/2.$$

아래 방정식들은 NxN Hybrid MIN의 대역폭 BW를 결정하는 것이다. 한 스테이지의 출력율은 다음 스테이지의 입력율과 동일하기 때문에 스테이지 0에서부터 시작하는 임의의 스테이지의 출력율을 순환 관계식(recursive related equation)[10]으로 표현할 수 있다. 특히 마지막 스테이지 2n-2에서의 출력율은 한 Hybrid MIN에서의 대역폭, 즉 cycle당 출력 노드로 도착한 요청의 수를 나타낸다.

스테이지 i의 Up(Down) 출력 링크에 대한 요청율을 $m_i^{Up(Down)}$라고 하자. $m_i^{Up(Down)}$은 일반적인 2x2 크로스바 스위치 네트워크에서 구해진 단위 시간당 각 출력 포트로 통과될 수 있는 요청율인 $1-(1-m/2)^2$를 이용하여 구할 수 있다. 그러나 Hybrid MIN에서는 각 스테이지에 따라, 또한 스위칭 소자의 두 출력 포트인 UP/DOWN에 따라 요청율이 서로 다르기 때문에 앞에서 구하여진 공식(equation)등을 이용하여 $m_i^{Up(Down)}$을 구할 수 있다.

$$m_j^{Up} = (1-(1-m_{j-1}^{Up}/2)^2) \times M_j^{Up}$$

$$m_j^{Down} = (1-(1-m_{j-1}^{Up}/2)^2) \times M_j^{Down} \quad \text{and} \quad m_{-1}^{Up} = m,$$
$$\text{for } 0 \le j \le n-2$$

그러므로,

$$BW = 2^n m_{2n-2}$$

여기서,

$$m_i = 1-(1-\max(m_{i-1}, m_{2n-2-i}^{Down})/2)^2,$$
$$\text{for } n \le i \le 2n-2$$

$$m_i = m_{j-1}^{Up} \text{ and } m_{-1}^{UP} = m, \text{ for } 0 \le j \le n-1.$$

<그림 8>은 위의 식을 이용하여 3가지 네트워크들에 대해서, 그의 크기 N이 증가할때 지역 참조도에 따른 예상되는 BW를 비교하여 보인 것이다. BW는 cycle당 도착된 요청의 갯수로써 측정되었다. 여기서 네트워크에 부과되는 부하(m)는 1로 하였으며, Hybrid MIN은 window 크기를 1로 하여 비교하였다. 이 window 내부에 참조 비율(α)이 낮을 경우(α=0.4이하)에서는 비교적 낮은 성능을 보이지만, α가 증가할수록 매우 우수한 성능을 나타내고 있다.

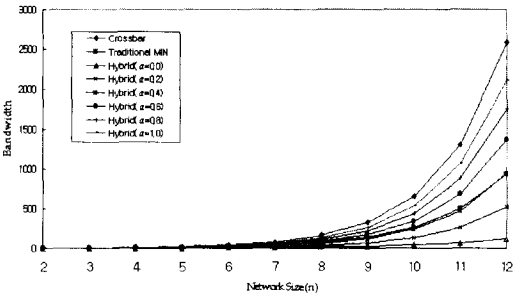


그림 8. N×N 네트워크들의 Expected BW
Fig. 8. Expected BW of N×N networks.

VI. 결 론

N^2 하드웨어 복잡도(complexity)를 갖는 크로스바 스위치 네트워크에서 MIN을 이용한 하드웨어 감소에 관한 연구는 매우 중요하고 관심을 집중시켰던 연구였다. 더우기 본 논문에서 제안된 Hybrid MIN은 전통적인 MIN에서 사용되는 스위칭 소자의 갯수($O(N\log_2 N)$)보다도 훨씬 적은 2N-3개의($O(N)$) 스위칭 소자를 가지고 FAC를 만족하면서 중복 경로를 제공할 뿐만 아니라 지역 참조성이 높은 통신은 짧은 경로를 제공할 수 있게 하여 시스템의 성능을 높일 수 있었다.

일반적으로 다중 프로세서 시스템 환경하에서의 많은 대부분의 응용 프로그램에서는 프로세서간의 통신(interprocssor communication)은 주로 프로세서-메모리들의 작은 그룹(small group)에서 발생하게 된다[1],[4]. 일반적으로 많이 그리고 자주 사용되고 있는 응용프로그램들에 대한 통신 분포를 추적기(tracer)등을 통하여 알아낼 수 있다면, 수많은 프로세서를 갖는 대형 시스템에서 각 프로세서, 메모리쌍간에 모두 동일한 길이의 연결 경로를 제공하기 보다는, 통신이 자주 발생되는 작은 그룹에 더 빠른 경로를 제공함으로써 보다 향상된 시스템 성능을 얻을 수 있다.

성능평가에서도 지역 참조도가 높은 통신의 환경에서는 기존의 MIN에서 보다 우수한 성능을 보였으며, 또한 Hybrid MIN의 cost-effectiveness함을 보였다. Hybrid MIN은 정적 네트워크인 이진 트리 위상의 장점과 동적 위상인 MIN의 장점을 결합함으로서, 지역 참조성의 활용과 적은 수의 스위칭 소자로써 대체 경로를 제공하고, 목적지 주소를 이용한 간편한 분산적 자기경로 제어 라우팅을 적용시킬 수 있을 뿐만 아니라 또한 계층버스[15]에서와 같은 트래픽의 고립화

(isolation)를 MIN에서도 적용시킬 수 있어 통신의 효율성을 높일 수 있게 되었다. 따라서 Hybrid MIN은 하드웨어 비용이 기존의 MIN보다 매우 저렴하고, 효과적인 네트워크이며, 또한 적은 수의 프로세서와 메모리로 구성된 그룹내에서의 통신의 빈도가 높게 지역화된 MPP 시스템 환경하에서 동일 크기의 기존의 전통적인 MIN 보다 우수한 상호 연결망 구조라 할 수 있다.

네트워크 크기가 256x256 이상의 대형 시스템에서는 기존 전통적인 MIN에서 사용되는 스위칭 소자의 갯수만을 비교했을 경우에 Hybrid MIN은 2배 이상의 하드웨어 절감 효과를 볼 수 있으며, 네트워크 크기가 그 이상 커질 경우 그 차이는 더욱 커지게 된다. 이와 같이 절감된 하드웨어를 이용하여 Hybrid MIN에 다중 계층(multi layer) fabric기법[7]등과 같은 연구를 적용시켜 진행시킨다면, 적은 비용으로 매우 우수한 성능을 발휘할 수 있는 MIN을 얻을 수 있을 것이다.

참 고 문 헌

[1] S.G. Abraham, and E.S. Davidson, "A Communication Model for Optimizing Hierarchical Multiprocessor System", *Int'l Conf on Parallel Processing*, pp. 467-474, 1986.

[2] C.M. Chiang, S. Bhattacharya, and L.M. Li,"Multicast in Extra-Stage Multistage Interconnection Networks", *Proc. the 6th IEEE Symp. on Parallel and Distributed Processing*, pp. 452-459, Oct. 1994.

[3] W.J. Dally, "Performance Analysis of k-ary n-cube Interconnection Networks", *IEEE Trans. Compt.*, vol. C-39, pp. 775-785, 1990.

[4] A.L. Decegama, *The Technology of Parallel Processing : Parallel Processing Architectures and VLSI hardware volume I*, Prentice-Hall International Editions, 1989.

[5] C.S. Ferner and K.Y. Lee, "Hyperbanyan Networks: A New Class of Networks for Distributed-Memory Multiprocessor", *Proc the Fourth Symposium on the Frontiers of Massively Parallel Computation*, pp. 254-261, Oct. 1992.

[6] J.R. Goodman and C.H. Sequin,"Hypertree: A multiprocessor Interconnection Topology", *IEEE Trans. Compt.*, vol. C-30, pp. 923-93,. Dec. 1981.

[7] T. Hanawa, H.Amano, and Y. Fujikawa, "Multistage Interconnection Networks with multiple outlets", *Int'l Conf on Parallel Processing*, vol. I, pp. 1-8, 1994.

[8] Kai Hwang, *Advanced Computer Architecture: Parallelism Scalability Programmability*, McGraw-Hill International Editions, 1993.

[9] K. Padmanabhan and D.H. Lawrie, "A Class of Redundant Path Multistage Interconnection Networks", *IEEE Trans. Compt.*, vol. C-32, pp. 1099-1108, Dec. 1983.

[10] J.H.Patel, "Performance of Processor-Memory Interconnections for Multiprocessors", *IEEE Trans. Compt.*, vol. C-30, pp. 771-780, Oct. 1981.

[11] M.C. Pease, "The Indirect binary n-cube microprocessor array", *IEEE Trans. Compt.*, vol. C-26, pp. 458-473. May 1977.

[12] H.J. Siegel, *Interconnection Networks for Large-scale Parallel Processing*, Lexington books, 1985.

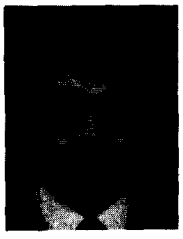
[13] S.W. Seo and T.Y. Feng, "A General Inside-Out Routing Algorithm for a Class of Rearrangeable Network", *Int'l Conf. on Parallel Processing*, pp. I 17-I20, 1994.

[14] T.H. Szymanski, "On the universality of Multipath Multistage Interconnection Networks", *J. Parallel and Distributed Computing*, vol. 7, pp. 541-569, 1989.

[15] A.W. Wilson, "Hierarchical Cache/Bus Architecture for Shared Memory Multiprocessors", *14th Int'l Symp. on Compt. Architecture*, pp. 244-252, 1987.

[16] C.L. Wu and T.Y. Feng, "On a class of Multistage Interconnection Networks", *IEEE Trans. Compt.* vol. C-29, pp. 694-702, 1980.

저 자 소 개



崔 昌 勳(正會員)

1988년 명지대학교 공과대학 전자계산학과 졸업(공학사). 1990년 서강대학교 대학원 전자계산학과 졸업(공학석사). 1990년 ~ 1991년 대우통신 근무. 1992년 ~ 현재 서강대학교 대학원 전자계산학과 박사과정 재학중. 관심분야는 parallel and distributed sysem, interconnection network, massively parallel architecture.

金 聖 天(正會員) 第 33卷 B編 第10號 參照