

단위 모듈을 이용한 MIN의 점증적 설계

(Incremental Design of MIN using Unit Module)

최창훈[†] 김성천^{**}

(Chang Hoon Choi) (Sung Chun Kim)

요약 본 논문에서는 적은 비용으로 효율적인 패킷 스위칭 상호연결 네트워크를 구성할 수 있는 새로운 부류의 MIN(Multistage Interconnection Network)인 SCMIN(ShortCut MIN)을 제안한다. SCMIN은 기존 MIN에서의 스위칭 소자 갯수 보다 매우 적은 수인 $2.5N-4$ 개의 스위칭 소자만을 사용할지라도 FAC(Full Access Capability)를 만족하고, 또한 프로세서-메모리 쌍에 대해 다수개의 중복 경로를 제공할 수 있게 된다. SCMIN은 통신이 빈번하게 발생하는 프로세서-메모리 클러스터에 보다 짧은 경로를 제공하고, 또한 이들에 대한 대체 경로를 제공하여 지역화된 통신 형태의 응용 분야에 적합하도록 설계되었다. 따라서 SCMIN은 공유 메모리 다중 프로세서 시스템에서 지역화된 통신 형태를 갖는 병렬 응용 분야에 적합한 MIN으로 활용될 수 있을 것이다.

Abstract In this paper, we propose a new class of MIN(Multistage Interconnection Network) called SCMIN(ShortCut MIN) which can form a cheap and efficient packet switching interconnection network. SCMIN satisfies full access capability(FAC) and has multiple redundant paths between processor-memory pairs even though SCMIN is constructed with $2.5N-4$ SEs which is far fewer SEs than that of MINs. SCMIN can be constructed suitable for localized communication by providing the shortcut path and multiple paths inside the processor-memory cluster which has frequent data communications. Therefore, SCMIN can be used as an attractive interconnection network for parallel applications with a localized communication pattern in shared-memory multiprocessor systems.

1. 서론

다수개의 프로세서들이 상수 시간(constant time)내에 메모리 모듈을 접근하는 공유 메모리 시스템(shared memory system)에서의 상호연결 망(interconnection network)은 시스템의 성능을 좌우하는 매우 중요한 요소중에 하나이다. 여러 상호연결 망들 중에서 크로스바 스위치(crossbar switch) 네트워크는 성능은 우수하지만 시스템의 크기가 커질수록 하드웨어 비용(hardware cost)이 또한 매우 커지게 되므로 대형 컴퓨터 시스템에서의 상호연결망으로는 부적절하다. 그러나 Omega 네트워크, generalized cube 네트워크, baseline 네트워크

등[7],[8],[15]과 같은 MIN(Multistage Interconnection Network)은 크로스바 스위치 네트워크보다 적은 하드웨어 비용으로 높은 시스템 성능을 나타내어 다중 프로세서 시스템(multiprocessor system)에서의 상호연결 망으로서 많이 사용되고 있다. 본 논문에서는 위에서 언급한 단일 경로 성질을 갖는 MIN 즉, UPP MIN(Unique Path Property MIN)을 기존의 MIN이라 칭할 것이다.

기존의 $N \times N$ MIN은 $\log_2 N$ 개의 스테이지(stage)로 구성되어 있다. 이러한 MIN에서는 네트워크의 크기 N 이 증가할수록 스테이지의 수 또한 증가하게 되어 전체 입출력 쌍간의 통신 경로의 길이 또한 증가하게 된다. 그러나 병렬 처리 분야 공유 메모리 다중 프로세서 시스템 환경하에서 많은 사용자들이 사용하는 대다수의 응용 프로그램들에서는 적은 수의 프로세서-메모리 그룹(group)내에서의 통신되는 빈도수는 전체 통신양 중의 많은 부분을 차지하기 때문에, 이들 그룹에 대한 짧은

[†] 정회원 : 상주대학교 컴퓨터공학과 교수
chang@samback.sangju.ac.kr

^{**} 종신회원 : 서강대학교 컴퓨터학과 교수
ksc@arglab1.sogang.ac.kr

논문접수 : 1998년 11월 3일

심사완료 : 1999년 3월 16일

경로의 제공이 필요하다[1],[3],[8],[15]. 따라서 기존의 MIN에서는 모든 통신 쌍들간에는 스테이지 수와 동일한 거리를 항상 유지되기 때문에 지역 참조의 활용을 할 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 위치상에서 아주 적은 부분에서만 발생하게 된다. 이러한 연구는 캐쉬를 기반으로한 시스템(cache based system)의 발전을 성공적으로 이루게 되었다. 이와 유사하게 다중 프로세서 시스템 환경 하에서의 많은 대부분의 응용 프로그램에서는 프로세서간의 통신(interprocessor communication)은 주로 프로세서-메모리들의 작은 단위 크기를 갖는 그룹내에서 발생하게 된다[9],[14]. 따라서 수많은 프로세서를 갖는 대형 시스템에서 각 프로세서, 메모리 쌍간에 모두 동일한 길이의 연결 경로를 제공하기 보다는 통신이 자주 발생하는 작은 그룹에 더 빠른 경로를 제공함으로써 보다 향상된 시스템 성능을 얻을 수 있을 것이다. 이러한 프로세서들간에서 통신 분포의 지역화를 본 논문에서는 지역 참조성이라는 표현으로 사용할 것이다.

또한 지금까지 MIN의 단일 경로 성질을 해결하기 위한 다중 경로 제공 연구는 모두 추가된 하드웨어(스위칭 소자 및 스테이지 추가[2],[7],[13], 링크의 추가[2],[4],[11], 스위칭 소자의 크기 확대[29],[52], 네트워크의 복사본[10],[17])를 이용하여 중복 경로를 제공하여 왔다. 오늘날 수백에서 수만 개의 프로세서를 갖는 다중 프로세서 시스템을 구성하는데 있어 상호연결 네트워크에 대해 이렇게 추가되는 비용으로 인해 전체 시스템의 많은 비용의 상승시키는 요인이 될 것이다.

이렇게 공유 메모리 다중프로세서 시스템의 MIN에 대한 지금까지의 연구에서는 앞에서 언급된 지역 참조성을 활용할 수 있는 연구는 전혀 없었을 뿐만 아니라, 기존의 MIN에서 2×2 크기의 스위칭 소자를 기존의 MIN에서 사용되는 스위칭 소자 갯수인 $O(N \log_2 N)$ 보다 적게 사용하여 임의의 프로세서에서 모든 메모리 모듈로 연결할 수 있는 FAC를 만족시키고, 또한 기존의 MIN 보다 적은 수의 스위칭 소자를 사용하고도 대체 가능한 다수의 중복경로를 제공할 수 있는 MIN에 관한 연구는 현재 전무한 상태이다.

따라서 이러한 기존의 MIN에서의 문제점들을 해결하기 위해 본 논문에서는 새로운 형태의 MIN인 SCMIN을 제안한다. SCMIN에서는 기존의 MIN에서 사용되는 스위칭 소자의 갯수($O(N \times \log_2 N)$)에 비하여, 훨씬 적은 갯수인 단지 $O(N)$ 개의 스위칭 소자만을

사용하여 스위칭 소자의 복잡도를 줄일 수 있고, 또한 기존의 MIN 보다 훨씬 적은 수의 스위칭 소자를 사용했음에도 불구하고, FAC를 만족할 뿐만 아니라 많은 수의 중복경로를 제공하게 한다. 또한 지역 참조성을 활용할 수 있게 하여 통신 빈도수가 높은 지역 참조의 경우에는 보다 빠른 경로를 제공함으로써 자주 발생하는 통신에 대한 지연 시간을 줄일 수 있을 것이다.

2. 확장성 MIN의 설계

2.1 단위 모듈 (Unit Module)

본 절에서 정의될 UNIT module은 4개의 입력 포트와 4개의 출력 포트를 갖고 있으며, 3개의 스테이지(입력 스테이지, 중간 스테이지, 출력 스테이지)로 구성된 한 개의 네트워크이다. 여기서 입력과 출력 스테이지에서는 각각 2개의 스위칭 소자를 갖고 있으며, 중간 스테이지에서는 한 개의 스위칭 소자로 구성되어 있다. 여기서 UNIT module의 구성의 규칙을 정의하기로 하였다. UNIT module에서의 입력 스테이지 및 출력 스테이지에서는 각각 2개의 스위칭 소자로 구성되어 있기 때문에 스위칭 소자의 번호는 1-bit 2진 형태인 b_1 으로 표현될 수 있다. 따라서 입력 스테이지와 출력 스테이지에 위치한 스위칭 소자의 링크들의 2진 표현 형식은 $b_1 b_0$ 가 된다.

또한 SCMIN의 확장은 이러한 UNIT module과 아래와 같이 정의된 *BaseNet*으로 부터 설계가 이루어지게 된다. 아래의 규칙은 재귀 관계식(recurrence relation equation)으로서 SCMIN의 확장에 관련된 topology describing function을 나타낸다. 따라서 SCMIN의 제 k차 (kth)확장에 대한 규칙을 아래와 같이 정의한다.

$$SC(2^k \times 2^k) := \text{BaseNet} \parallel \text{Add_UNIT}$$

$$\text{BaseNet} := SC(2^{k-1} \times 2^{k-1})$$

$$\text{Add_UNIT} := 2^k / 4 \text{ UNIT}$$

(여기서 만약 k=2이면, 제2차 확장일 때, 제1차 확장으로서 *BaseNet*, $SC(2^1 \times 2^1)$)은 단지 한 개의 SE를 갖게 된다.)

SCMIN의 확장 과정에서 추가되는 $2^k / 4$ 개 *Add_UNIT*의 각 입력(출력) 스테이지의 스위칭 소자들은 위(top)에서 부터 시작하여 2진 형태 표현으로써 $b_\ell \dots b_1$, ($\ell = n-1$)와 같이 부여될 수 있다. 위의 식에서 operator \parallel 는 $2^k / 4$ 개 UNIT 모듈을 *BaseNet*에 연결시키기 위한 연결 동작을 의미한다. 그리고 이러한

확장 규칙들은 아래에서 정의되는 매핑 함수(mapping function) f 에서, 상단 출력 포트인 Up에 대한 연결 함수 f 와 관련이 있게 된다.

$f(i=0$ 또는 1 이며, 여기서 0 은 스위칭 소자의 입/출력의 Up 포트를 의미하며, 1 은 Down 포트를 의미함)은 UNIT module을 위한 매핑 함수라고 하자. 또한 function $f(b_i \dots b_1)_{UNIT_{to}} = (b_i)_{UNIT_u}$ 는 UNIT module의 입력(출력) 스테이지에 있는 스위칭 소자 $b_i \dots b_1$ 의 Down 출력 링크(입력 링크)는 그 UNIT module내의 중간 스테이지에 있는 스위칭 소자의 입력(출력) 링크 b_i 에 연결되어 있음을 의미한다. 또한 $f(b_i \dots b_1)_{UNIT_{to}} = (b_i \dots b_1)_{BaseNet_{to}}$ 는 UNIT module의 입력(출력) 스테이지에 있는 스위칭 소자 $b_i \dots b_1$ 의 Up 출력(입력) 링크는 BaseNet에 있는 스위칭 소자의 입력(출력) 링크 $b_i \dots b_i$ 에 연결되는 것을 의미한다.

초기 상태, 즉 확장이 이루어지지 않은 기본 단위로서의 UNIT module에 대한 표현에 있어서 입력(출력) 스테이지에 있는 스위칭 소자 b_i 의 상단 출력 링크인 Up 출력 링크(입력 링크)는 그 UNIT module 밖에 있는 다른 모듈의 스위칭 소자와 연결시키기 위한 것으로서 UNIT module내에서 입력(출력) 스테이지 스위칭 소자의 UP 링크들은 네트워크 확장에 사용될 링크들이므로 이 UNIT 모듈내의 어느 스위칭 소자에도 연결이 이루어지지 않은 상태인 것을 의미하여 $f(b_i) = outside$ 와 같이 나타낼 수 있다.

따라서 위의 두 함수 f 와 f 에 대해 SCMIN의 확장을 고려했을 경우, 즉 $f(b_i \dots b_1)_{UNIT_{to}} = (b_i \dots b_1)_{BaseNet_{to}}$ 와 $f(b_i \dots b_1)_{UNIT_{to}} = (0)_{UNIT_u}$ 으로 각각 표현될 수 있다. 또한 입력(출력) 스테이지에 있는 스위칭 소자 b_i 의 출력(입력) Up 포트는 다른 UNIT module인 BaseNet의 모듈들에 대한 연결을 지원한다. 즉 네트워크가 확장되어질 때 사용되어 질 것이다.

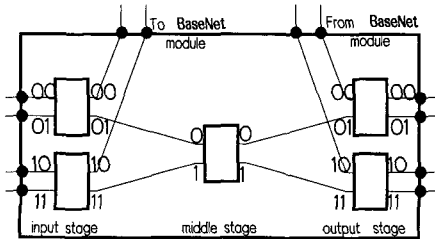


그림 1 UNIT 모듈

UNIT module은 <그림 1>과 같이 형성될 수 있다. 그리고 이렇게 형성된 UNIT module을 기초로한 확장(incremental scale)기법을 통하여 SCMIN를 설계할 수 있다.

[예 1] SCMIN의 제2차 확장 (<그림 2>
 $SC(2^2 \times 2^2) := SC(2^1 \times 2^1) \parallel \text{one UNIT}$

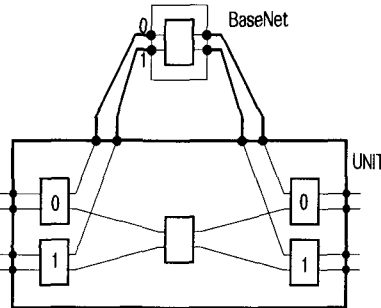


그림 2 4x4 SCMIN

[예 2] SCMIN의 제3차 확장 (<그림 3>
 $SC(2^3 \times 2^3) := SC(2^2 \times 2^2) \parallel 2 \text{ UNIT}$

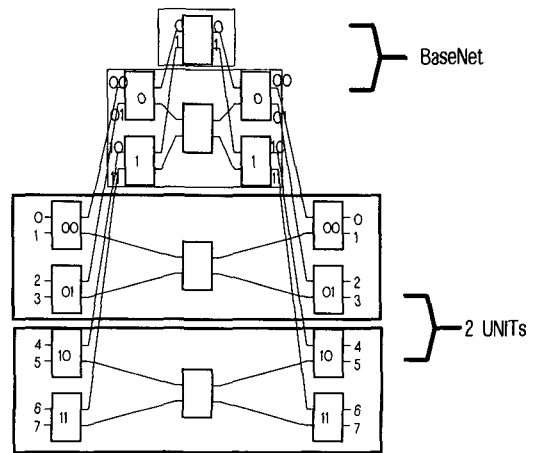


그림 3 8x8 SCMIN

2.2 NUMA(Non-Uniform Memory Access) Model

이러한 확장 법칙은 각 UNIT module들의 입력(출력) 스테이지에 있는 스위칭 소자들의 Up 링크를 통해서 이루어짐을 알 수 있다. 따라서 SCMIN에서의 n 차 확장 f_{nub}^0 에 형성 규칙은 아래와 같이 정리될 수 있다.

- Expansion n : $f_{n0}^0(b_1 \dots b_1)_{UNIT_{n0}}$
 $\leftarrow f_{n-10}^0(b_1 \dots b_2)_{UNIT_{n0}} + \text{UNIT}$
- Expansion n-1 : $f_{n-10}^0(b_1 \dots b_2)_{UNIT_{n0}}$
 $\leftarrow f_{n-20}^0(b_1 \dots b_3)_{UNIT_{n0}} + \text{UNIT}$
- ⋮
- Expansion 2 : $f_{20}^0(b_1)_{UNIT_{n0}} \leftarrow f_{10}^0(0)_{UNIT_{n0}} + \text{UNIT}$
- Expansion 1 : $f_{10}^0(0)_{UNIT_{n0}}$

위의 정의에 따르면 n차 확장까지의 상단 출력 링크로의 연결은 n-1차 확장 네트워크가 먼저 이루어져야 하며, 또한 n-1차 SCMIN을 위해서는 n-2차 확장이 먼저 선행되어야 한다. 이러한 과정은 1차 확장, 즉 SCMIN의 최소 단위인 UNIT까지 전개된다.

[정리 1]

n차 확장 UNIT module의 스위칭 소자 $b_{n-1} \dots b_1$ 에 직접 연결된 프로세서 $b_{n-1} \dots b_0$ 와 $b_{n-1} \dots b_1$ 에서 Expansion i를 통해서 연결될 수 있는 메모리 모듈은 $b_{n-1} \dots b_i X_{i-1} \dots X_0$ (여기서 $X_i=0$ 또는 1, $1 \leq i \leq n$)과 $b_{n-1} \dots \bar{b}_i X_{i-1} \dots X_0$ 이다. 즉 2^{i+1} 개의 메모리 모듈을 액세스할 수 있는 연결 경로가 존재한다.

(증명) 각 Expansion $i(1 \leq i \leq n-1)$ 의 입력 스테이지에 있는 스위칭 소자, $b_{n-1} \dots b_i$ 는 UNIT 정의에 따라 출력 스테이지에 있는 스위칭 소자들은 스위칭 소자, $b_{n-1} \dots b_i, b_{n-1} \dots \bar{b}_i$ 와 각각 연결이 이루어 질 수 있다. 그리고 이 두 스위칭 소자로 부터 연결시킬 수 있는 메모리 모듈은 $b_{n-1} \dots b_i X_{i-1} \dots X_0$ 와 $b_{n-1} \dots \bar{b}_i X_{i-1} \dots X_0$ 이다. 따라서 스위칭 소자 $b_{n-1} \dots b_1 b_0$ 는 Expansion i를 통해서 이러한 2^{i+1} 개 메모리 모듈을 액세스할 수 있는 경로가 존재한다. ■

표 1 SCMIN에서 Expansion에 따른 클러스터

Processor Cluster	$P_0 \sim P_3$	$P_4 \sim P_7$...	$P_{N-8} \sim P_{N-5}$	$P_{N-4} \sim P_{N-1}$
Cluster 1	$M_0 \sim M_3$	$M_4 \sim M_7$...	$M_{N-8} \sim M_{N-5}$	$M_{N-4} \sim M_{N-1}$
Cluster 2	$M_0 \sim M_7$...	$M_{N-8} \sim M_{N-1}$	
	⋮				
Cluster i	$M_0 \sim M_{2^{i-1}-1}$...		$M_{N-2^{i+1}} \sim M_{N-1}$	
	⋮				
Cluster n-1	$M_0 \sim M_{N-1}$				

[정리 1]을 이용하여 $N \times N$ SCMIN에서 각 프로세서는 [표 1]과 같이 n-1 종류의 클러스터를 가질수 있다.

각 클러스터내에는 4개에서 최대 N개의 메모리 모듈을 포함하고 있으며, 또한 선택 가능한 중복 경로가 2개에서 최대 n개가 존재하게 된다. 또한 이들 클러스터에 따라 프로세서 및 메모리간의 거리도 최단 거리로서 3에서 부터 $2n-1$ 개까지 다양한 길이를 가지게 된다.

[정리 2] SCMIN에서 각 프로세서는 클러스터 1에서 연결되어진 메모리 모듈들은 Expansion n차로 확장을 할 경우 전체 네트워크에서 최대 n개의 중복 경로가 존재한다.

(증명)

i) 1차 확장의 경우

1차 확장은 2×2 SCMIN으로서 단지 2×2 스위칭 소자 한 개이므로 각 프로세서 메모리당 1개의 경로만 갖게 되는 것은 명백하다.

ii) k차 확장의 경우에 계층 1을 통해 연결될 수 있는 프로세서 메모리 쌍들에 대해 k개의 중복경로가 존재한다고 가정하자.

iii) k+1차 확장의 경우

SCMIN의 확장 위상 정의에 따라 모든 SCMIN은 기본적으로 UNIT module을 갖게 된다. 따라서 k+1차 확장에서도 UNIT module을 갖게되고 이들은 각 프로세서 메모리 모듈당 1개의 경로를 갖게 된다. 더우기 UNIT module은 계층 1에 해당된다. 그리고 k+1차 확장은 SCMIN의 위상 정의상 k차 확장 네트워크를 BaseNet으로 하여 UNIT module을 연결시켜 구성된다. 그리고 ii)항에서 k차 확장의 경우 k개의 중복 경로를 갖고 있다고 가정하였고, k+1차 SCMIN의 UNIT module에서 이미 1개의 경로가 존재하므로, SCMIN의 k+1차 확장은 k+1개의 중복 경로가 존재하는 것은 명백하다. 따라서 SCMIN에서는 각 계층 1에 해당되는 프로세서 메모리쌍에 대해서 최대 n개의 중복 경로를 제공할 수 있게 된다. ■

[정리 2]에서의 증명에서 나타났듯이, SCMIN에서 각 프로세서는 각각의 계층을 통해 연결될 수 있는 메모리 모듈들에 대해서는 경로의 수가 1개씩 추가되는 것을 알 수 있고, 또한 계층 2에서 부터 새로이 추가되는 통신 가능한 메모리 수는 각 계층 번호에 대한 2의 멱승 단위로 증가하게 되므로, 각 프로세서당 계층에 따른 메모리 모듈의 갯수와 중복 경로의 수는 [표 2]와

같다.

표 2 각 클러스터에 따른 메모리 수와 중복 경로 수

Cluster	메모리 모듈 갯수	중복 경로의 수
Cluster 1	4	n
Cluster 2	4	$n-1$
Cluster 3	8	$n-2$
⋮	⋮	⋮
Cluster $n-1$	2^{n-1}	2

따라서 SCMIN에서 중복 경로의 평균 수는

$$\frac{4 \times n + \sum_{i=1}^{n-1} 2^i \times (n-i+1)}{4 + \sum_{i=1}^{n-1} 2^i} \quad \text{이다.}$$

따라서 SCMIN에서는 통신의 빈도가 높은 적은 크기의 프로세서-메모리에 대해 짧은 경로를 제공할 뿐만 아니라 많은 수의 중복 경로를 제공할 수 있게 된다.

다음은 SCMIN의 하드웨어 복잡도의 대부분을 차지하고 있는 스위칭 소자의 갯수는 초기의 $N \times N$ SCMIN에서 사용되는 스위칭 소자 보다 $2^{n-1}-1$ 개의 2×2 스위칭 소자를 추가한 것으로서, $N \times N$ SCMIN에서 사용되는 스위칭 소자의 갯수인 $2N-3$ 에 $2^{n-1}-1$ 를 더하면 된다. 그러므로 SCMIN에서 사용되는 스위칭 소자의 총 갯수는 아래의 식과 같이 구할 수 있다.

$$2N-3+2^{n-1}-1 = 2 \cdot 2^n - 3 + 2^{n-1} - 1 = 2.5N-4$$

따라서 SCMIN에서 스위칭 소자의 수는 계속해서 $O(N)$ 을 유지할 수 있고, 또한 기존의 MIN에서의 스위칭 소자의 하드웨어 비용, $O(N \log N)$ 에 비해 훨씬 적은 비용으로써 FAC를 만족할 뿐만 아니라, 네트워크의 크기의 증가에 중복 경로의 수는 증가하고, 또한 계층이 낮을 수록 많은 중복 경로를 제공할 수 있어 최대 n 개의 중복 경로를 제공할 수 있다. 또한 이들에 대해서는 짧은 경로를 제공할 수 있게 된다. [표 3]은 위와 같은 사항에 대하여 네트워크 크기에 따른 예를 보인 것이다.

3. 라우팅 전략

3.1 라우팅 태그 생성 알고리즘

SCMIN은 분산적 자기제어 라우팅(distributed self-routing) 전략을 이용하여 쉽고, 빠른 경로 선택을 지원할 수 있게 된다. SCMIN은 위상적 구성상 UNIT

표 3 스위칭 소자의 수와 평균 중복 경로 수의 비교

Network Size(n)	사용된 스위칭 소자의 갯수		평균 중복 경로 수	
	MIN	SCMIN	MIN	SCMIN
2	4	6	1	2
3	12	16	1	2.5
4	32	36	1	3
5	80	76	1	3.5
6	192	156	1	4
7	448	316	1	4.5
8	1024	630	1	5
9	2304	1276	1	5.5
10	5120	2556	1	6

module을 최소 단위의 설계 기본 module로 하고 있기 때문에 최소 길이를 갖는 경로는 이 UNIT module 내의 프로세서-메모리 쌍에 존재하게 된다. 따라서 한 프로세서에서 각각의 메모리까지의 연결을 위해서 3개의 스위칭 소자만을 통과하므로 전체 네트워크의 크기의 증가에 변화 없이 그 길이는 3으로서 항상 일정하게 유지될 수 있다. 따라서 이들 프로세서-메모리 그룹에서는 지역 참조 활용에 있어 매우 유리한 거리를 가질 수 있게 된다.

SCMIN의 프로세서-메모리 쌍들에 대해 기본적으로 계층 1에 해당되는 메모리 모듈에 대하여 최소 거리 3부터 계층 1 레벨씩 증가하는데 따라 통과하여야 하는 스위칭 소자의 수는 2씩 증가하게 된다. 따라서 최대 계층 $n-1$ 에 해당하는 메모리 모듈들에 대해서는 $2n-1$ 개의 스위칭 소자를 통과하게 된다.

따라서 이러한 모든 라우팅 경로를 지원할 수 있도록 하기 위하여 SCMIN에서 라우팅에 사용하게 될 라우팅 태그의 총 비트 수는 $2n-1$ 개으로써 아래와 같은 형태로 표현되어 진다.

t_{2n-2}	t_{2n-3}	t_{2n-4}	t_{2n-5}	t_1	t_0
------------	------------	------------	------------	-------	-------	-------

- Routing Tag -

이러한 라우팅 태그의 각 비트 $i(0 \leq i \leq 2n-2)$ 는 SCMIN의 각 스테이지 i 에 있는 스위칭 소자의 내부 연결을 위한 제어 비트이다. 예를 들어, 스테이지 i 에 있는 스위칭 소자의 입력 포트에 도착한 패킷의 라우팅

태그 비트 중에서 $t_i=0$ 이면, 그 스위칭 소자의 출력 포트중에서 Up 출력 포트에 그 패킷을 통과시키고, 그렇지 않고, 만약 $t_i=1$ 이면, 그 스위칭 소자의 출력 포트중에서 Down 출력 포트에 패킷을 전송시킨다.

그러나 항상 이러한 $2n-1$ 개의 라우팅 태그 비트를 모두 사용하지는 않는다. 일단 주어진 프로세서와 메모리에 대한 최단거리의 경로가 설정되면 그에 해당되는 라우팅 태그 비트만이 사용하게 된다. 그리고 그 경로가 블럭(block)되었거나, 혼잡할 경우 1개의 계층씩 늘려서 재 라우팅(re-routing)을 할 수 있다. 이러한 재 라우팅은 라우팅 태그 비트를 2개씩 늘려 사용함으로써 그 경로를 얻을 수 있다.

이렇게 라우팅 태그에 사용되는 비트 수가 계층에 따라 다르기 때문에 라우팅 태그 생성을 위해 주어진 한 프로세서에 대해 목적지 메모리 모듈을 포함하고 있는 계층을 결정하여야 한다. 이러한 계층별 라우팅 태그의 결정 방법은 [정리 1]에서 증명된 $n-1$ 가지의 계층에서 연결될 수 있는 메모리 모듈에 관한 이론을 기초로 하여 얻어진 방법으로 이루어질 수 있게 된다. 따라서 만약 근원지 주소가 $s_{n-1}s_{n-2}...s_0$ 이고 목적지 주소가 $d_{n-1}d_{n-2}...d_0$ 라고 하자. 그리고 이들 두개의 비트-스트링(bit-string)중에서 부분 스트링, $s_{n-1}s_{n-2}...s_2$ 와 $d_{n-1}d_{n-2}...d_2$ 을 고려하여, 이들에 대해 exclusive-OR 연산을 적용시킨다. 이에 대한 결과로서 $c_{n-2}c_{n-3}...c_1 = s_{n-1}s_{n-2}...s_2 \oplus d_{n-1}d_{n-2}...d_2$ 를 얻을 수 있다. 그리

고 비트-스트링 $c_{n-2}c_{n-3}...c_1$ 에서 MSB로 부터 시작하여 오른쪽 방향으로 LSB까지 조사하여 최초의 1이 발견될 때까지 진행한다. 즉, c_i 에서 최초의 1이 발견된다면, 이 메모리 모듈에 대한 계층은 $i+1$ 로 결정된다. 그러나 만약 그러한 c_i , 즉 모든 c_i 가 0일 경우, 그 계층은 1로 결정된다. 이는 다음 알고리즘과 같이 표현된다.

계층 결정 방법을 이용한 알고리즘으로 생성된 라우팅 태그를 이용하여 SCMIN에서 분산적 자기 경로를 제공하게 된다. 이러한 라우팅 태그를 사용할 경우 스테이지 i 에 있는 스위칭 소자로 라우팅 태그 t_{2n-2-i} 를 검사하여 이것이 0이면, 그 스위칭 소자의 출력 포트중 상단인 Up 포트로의 연결을 제공하고 만약 $t_{2n-2-i}=1$ 이면, 그 스위칭 소자의 출력 포트중 하단인 Down 포트 로 입력된 패킷을 출력할 수 있도록 연결을 제공한다.

예를 들어 <그림 4>와 같은 16×16 SCMIN에서 근원지 $0_{10}(000_2)$ 와 목적지 $3_{10}(0011_2)$ 사이를 연결할 수 있는 경로를 찾기 위해 선택될 수 있는 최단 경로를 먼저 b_3b_2 에 해당되는 근원지의 비트-스트링 00_2 과 목적지 비트-스트링 00_2 을 exclusive-OR 연산을 수행하면, 그 결과 1을 갖는 비트를 찾을 수 없기 때문에 계층 1로서 결정된다.

```

Routing Tag Generation Algorithm

/*
Src. Address :  $s_{n-1}s_{n-2} \dots s_0$ 
Dst. Address :  $d_{n-1}d_{n-2} \dots d_0$ 
*/

begin
/* the decision of class */
i = n;
C = ( $s_{n-1}s_{n-2} \dots s_2$ )  $\oplus$  ( $d_{n-1}d_{n-2} \dots d_2$ )
while ( $c_i \neq 1$  and  $i \neq 0$ ) i=i-1;
class = i;
/* routing tag generation */
j=2n-2;
for(i=1 to class)
begin
t_j = 0;
j=j-1;
end
t_j = 1;
for(i=0 to class-1) t_i = d_i;
end.
    
```

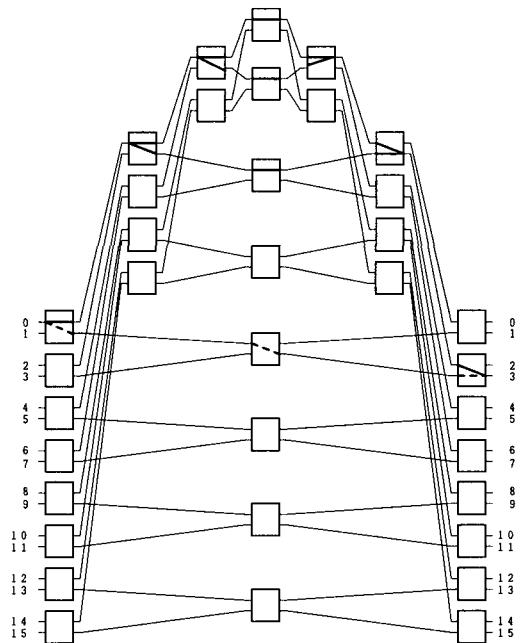


그림 4 16×16 SCMIN에서의 중복 경로들의 예

따라서 라우팅 태그 생성 알고리즘에 의해 $t_{2n-2}=1$

로 세트되고, $t_1 = d_1$ 로, 그리고 $t_0 = d_0$ 로 할당되어 그 결과 라우팅 태그는 111_2 이 된다. 이에 대한 경로의 예는 <그림 4>의 점선으로 표시되어 있다. 그리고 다른 실선들은 우회할 수 있는 대체 경로를 표현하고 있다.

4. 성능분석

4.1 네트워크 성능 파라미터 비교 분석

[표 4]는 각 상호연결 네트워크에 대한 성능 파라미터(performance parameter)를 비교한 것이다. 여기에서 HMIN[16]과 TCMIN[6]은 분산 메모리 시스템(distributed memory system)에서 지역 참조성을 활용하기 위한 MIN으로서 본 논문에서의 공유 메모리 시스템에서의 SCMIN과는 적용되는 시스템 환경이 다르다. 따라서 이러한 다른 시스템에서의 상호연결 네트워크에 대한 성능을 시뮬레이션을 통하여 비교 분석한다는 것은 의미가 없는 것으로 판단되어 성능에 큰 영향을 미칠 수 있는 몇 가지 파라미터로써 이들을 서로 비교 분석하였다. [표 4]에서 잘 나타났듯이 SCMIN은 기존의 MIN에서와 같이 네트워크 크기가 2^n 으로 증가하기 때문에 쉬운 확장성을 보장하고 또한 MIN에서의 잠정인 적응적 DTA 라우팅을 그대로 적용시킬 수 있다. 또한 HMIN이나 TCMIN은 클러스터 크기가 2^n 으로써 일정하기 때문에 적용되는 응용 프로그램의 통신 형태의 요구에 따라 그 크기를 확장 및 조정할 수 없으나, SCMIN은 클러스터 크기가 2에서부터 2^n 까지 유동적으로 변할 수 있기 때문에 응용 프로그램의 요구에 따라 클러스터 크기의 조절이 가능하다.

표 4 성능 파라미터의 비교

MIN item	HMIN	TCMIN	MIN	SCMIN
Network Size	$2^n - 2^n$	$2^n + 2^n$	2^n	2^n
Cluster Size	2^n	2^n	2^n	$2^n \sim 2^n$
Avg Distance (Unfrm. trff.)	$3n + \frac{n}{2^n}$	$2n + 1 - \frac{n+1}{2^n + 1}$	n	$\frac{2^n(2n-3)+2}{2^n - 2}$
Weighted Avg. Distance	$3n - 2an$	$2n - a(n+1)$	n	$3n + (1-a) \left\{ \frac{2^n(2n-3)-4}{2^n+2n-8} \right\}$
H/W cost	$(n + \frac{n}{2^n})2^{2n-1}$	$(n + \frac{n}{2^n})2^{2n-1}$	$\frac{n2^n}{2}$	$2.5 \times 2^n - 4$

그러나 균일 분포의 통신 형태에서는 SCMIN은 다른 MIN보다 거리가 길어지는 단점을 갖고 있음을 알 수 있다. 그러나 앞서도 언급하였듯이 대부분의 응용 프로그램들은 적은 수의 프로세서-메모리로 구성된 클러스터 내에서 많은 통신 형태를 발생하기 때문에 제안된

SCMIN은 이러한 지역 참조성이 높은 분포를 갖는 통신 형태에 매우 유리한 네트워크이다. 따라서 [표 4]에서 SCMIN에 대한 가장 평균 거리의 식에서도 잘 나타나 있듯이 지역 참조성이 높아질수록 평균거리는 네트워크 크기에 영향을 거의 받지 않는 것으로 나타나고 있다. 그리고 마지막으로 하드웨어 비용은 각 MIN에서 사용하고 있는 스위칭 소자의 수를 비교한 것으로서 SCMIN은 다른 어떤 MIN 보다도 훨씬 적은 수의 스위칭 소자의 수를 갖고 있는 것으로 나타나 하드웨어 비용 절감 또한 꽤하였다.

4.2 네트워크 조합 능력에 관한 분석

다음은 네트워크의 조합 능력(combinatorial power: CP)에 대해 분석하기로 하겠다. $N \times N$ 크기의 MIN에서 조합 능력이라고 하는 것은 N개의 총 순열(permutation)의 수에 대해서 충돌이 없이 한번에 네트워크를 통과시킬 수 있는 순열의 수에 대한 비율을 말한다. 이러한 통과 가능한 순열은 그 네트워크를 통해서 N개의 입력과 N개의 출력을 동시에 연결시킬 수 있는 능력을 설명해 주는 것이다. 만약 $CP \approx 1$ 이면, MIN에서 일부 특정한 순열만을 통과시킬 수 있다는 것을 의미한다. 따라서 2×2 크기의 스위칭 소자를 사용하는 $N \times N$ 크기의 SCMIN은 총 $2.5N - 4$ 개의 스위칭 소자를 갖고 있고, 또한 한개의 각 스위칭 소자에서 가능한 상태의 수는 2가지 상태(직렬 연결, 교환연결)라고 가정하면, SCMIN에서 총 상태의 수는 $2^{2.5N-4}$ 이다. 따라서 CP는 $2^{2.5N-4}/M!$ 이다. 이것은 동일 크기의 기존 MIN에서의 CP, $2^{nN^2}/M!$ 보다 조합 능력이 떨어지는 결과를 얻게된다. 이것은 SCMIN이 기존의 MIN보다 충돌이 없이 동시에 네트워크를 동시에 통과시킬 수 있는 순열의 수가 적다는 단점을 갖게되는 것을 말한다. 이러한 단점은 SCMIN에서 사용되는 스위칭 소자의 갯수가 기존의 MIN 보다 적게 사용함으로써 발생하는 문제이다. 따라서 위의 성능 분석과 조합능력에서 나타났듯이 적은 프로세서 그룹에 통신의 분포가 지역화되어 있고, 또한 통신 지연시간이 조합 능력보다 더욱 중요하게 다루어지는 병렬 응용 프로그램 응용 분야에서 SCMIN은 동일 크기의 기존의 MIN 보다 우수한 성능을 보이고 있어 이러한 환경에서 효율적인 상호연결 네트워크로 활용될 수 있을 것이다.

4.3 성능 분석

본 논문에서의 상호연결 네트워크는 Scientific and Engineering Software Inc.사의 SES/workbench[18]를 사용하여 이산 사건 모델링(discrete event modeling)으로 시뮬레이션(simulation)을 수행하였다. SES/

workbench 모델은 한 개 또는 그 이상의 서브모델로 구성될 수 있다. 각 모델은 확장된 방향 그래프로써 표현된다. SES/workbench 그래프(graph)의 기본적인 구성 요소(component)는 노드(node), 아크(arc), 트랜잭션(transaction), 자원(resource)들로 구성된다. 트랜잭션은 한 개의 노드에서 아크를 통해 다른 노드로의 흐름으로서 그래프에서는 나타나지 않는다. 각 트랜잭션은 수행될 process, 처리되거나 전송될 데이터, 또는 현재 진행중인 제어 신호를 나타낸다. 전형적으로 SES/workbench 모델은 많은 트랜잭션을 병렬적으로 수행시킬 수 있다. 또한 성능 통계(performance statistics)들은 각 트랜잭션에 대해 통계 수집을 할 수 있고, 결과 또한 출력할 수가 있다.

네트워크에 대한 다양한 매개 변수(parameter)들을 적용시켜 시뮬레이션 수행 시간을 10,000 시뮬레이션 시간 단위(simulation time unit)로 설정하여 시뮬레이션을 수행시켰다. 패킷 스위칭 네트워크의 성능을 분석하고 평가하기 위하여 처리율(throughput)과 지연(delay) 시간을 척도로 정하였다. 평균 처리율은 단위시간당 네트워크를 통과할 수 있는 패킷의 평균 수 즉 packet/cycle를 정의한다. 또한 평균 지연시간은 패킷이 네트워크의 입력 포트에서 출력 포트에 통과하는데 걸리는 평균시간(cycle 수)으로 정의한다. 부하(load)는 cycle당 네트워크의 입력 포트에 의해 받아들이는 총 패킷 수로 정의한다.

SCMIN의 성능을 측정하기 위해서 다음과 같은 분포를 사용한다[4],[5].

- (a) Uniform Distribution : 균등분포에서는 입력 노드 i 가 출력 포트 j 로 메시지를 보낼 확률이 모든 $i, j, i \neq j$ 에 대해 동일하다.
- (b) Sphere of Locality : 특정 영역내에 있는 노드들 간에는 더 잦은 메시지 교환이 일어날 가능성이 있다. 따라서 각 입력 노드가 radius(L)를 갖는 지역성 구(locality sphere)의 중심점이라고 가정하여, 한 노드가 그 지역성 구 내에 확률 α 를 가지고 메시지를 보내며, 또한 그 지역성 구 밖에 있는 노드로는 확률 $(1-\alpha)$ 를 가지고 메시지를 보내게 된다.
- (c) Decreasing Probability Distribution : 어느 한 노드는 메시지를 보낼 확률이 경로길이의 역함수로 정의된다. 길이가 L인 경로에 메시지를 보낼 확률 $\varphi(L)$ 은 아래와 같다.

$$\varphi(L) = \varphi(L) = ((1-D) D^L) / (1 - D^{L_{max}}) \times D.$$

여기서 L은 경로 길이이고, Lmax는 네트워크의 diameter, 그리고 D는 decay coefficient이다.

<그림 5>는 Cross-bar 네트워크와 전통적인 MIN과 제안된 네트워크인 SCMIN에 대하여, 그의 크기 N이 증가할 때 지역 참조율에 따른 예상되는 대역폭을 비교하여 보인 것이다. 대역폭은 cycle당 도착된 요청의 갯수로써 측정되었다. 여기서 네트워크에 부가되는 부하(m)는 1로 하였으며, SCMIN은 window 크기를 1로 하여 비교하였다. 이 window 내부에 참조 비율(α)이 낮은 경우($\alpha = 0.4$ 이하)에서는 비교적 낮은 성능을 보이지만, α 가 증가할수록 매우 우수한 성능을 나타내고 있다.

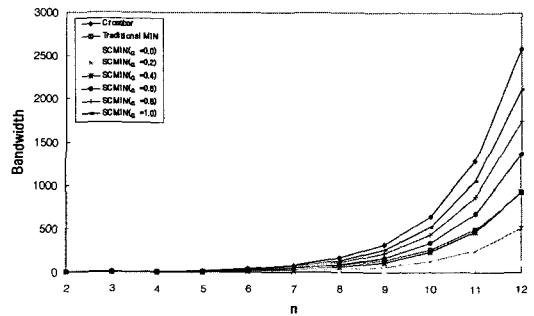


그림 5 N x N Network들에 대한 대역폭의 비교

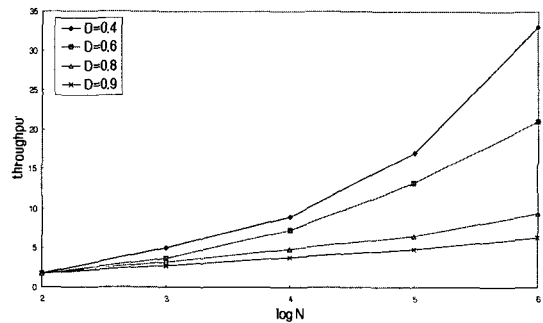


그림 6 decreasing probability distribution하에서 네트워크 크기에 따른 처리율의 변화

<그림 6>은 decreasing probability distribution에 대한 실험 결과 그래프이다. 예상대로 주어진 D값에 대해서 네트워크 크기가 증가 할 수록 처리율이 증가함을 볼 수 있다. D값이 작을 때는 패킷이 목적지 노드로 보내지는 확률이 경로 길이가 증가할 수록 감소됨을 볼 수가 있다. 그러나 D값이 증가하여 1에 근접한다는 것

은 균등분포에 가까워진다는 의미로 해석될 수 있다. 결론적으로 D값이 적을 때는 상대적 근원지로 부터 거리가 짧은 목적지로 많은 패킷이 전송하게 되어 높은 처리율을 보이게 되는 것이다.

<그림 7>은 지역구 분포(sphere of locality distribution)를 보여주는 것이다. 예상대로 window 크기가 커질 수록 처리율은 감소하게 되나, window의 크기가 작아질 수록 처리율은 커지게 된다.

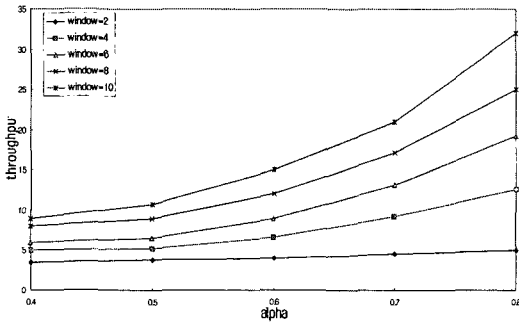


그림 7 window 크기와 지역참조 상관계수의 변화에 따른 처리율의 변화

<그림 8>는 decreasing probability distribution에 대한 그래프이다. 주어진 D값에 대해서 네트워크 값에 대해서 네트워크의 크기 증가에 따라 평균 지연 시간 또한 증가하게 된다. 그리고 D값이 증가 할 때도 지연 시간은 증가하게 됨을 볼 수 있다.

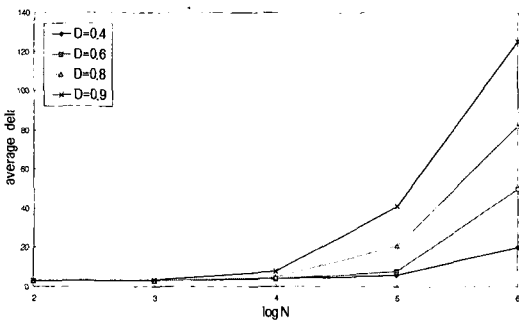


그림 8 decreasing probability distribution하에서 네트워크 크기에 따른 평균 지연시간의 변화

<그림 9>은 지역구 분포에 대한 지연 시간의 변화를 살펴 본 것이다. window 크기가 증가할 수록 평균 지연 시간은 증가하게 된다. 이것은 패킷이 도달되는 목적지간의 거리가 증가하기 때문이다. 또한 주어진

window 크기에서 α 값이 증가할 수록 평균 지연 시간이 줄어드는 것을 볼 수 있는데 이는 패킷이 지역구내에 있는 목적지로 결정될 확률이 높게 결정되기 때문이다.

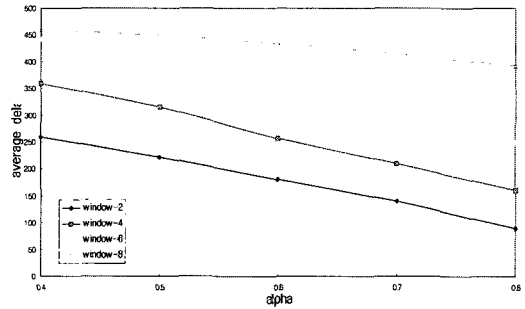


그림 9 window 크기와 지역 참조 계수의 변화에 따른 지연 시간의 변화

위와 같은 결과를 살펴볼 때, 통신 분포가 균등 분포일 때는 SCMIN의 성능이 기존의 MIN 보다 떨어지는 반면, 통신이 지역화된 환경에서는 SCMIN이 우수하게 나타났다. 일반적으로 프로세서와 메모리 모듈간의 상호 연결을 위해 MIN을 사용하고 있는 다중 프로세서 시스템에서는 대체적으로 한 프로세서가 참조하는 메모리 모듈의 수가 매우 적다. 따라서 이러한 환경하에서는 SCMIN은 매우 효과적인 네트워크로서 활용될 수 있을 것이다.

5. 결론

본 논문에서는 공유 메모리 다중 프로세서 시스템에서 지역화된 통신 형태를 갖는 병렬 응용 프로그램을 효율적으로 수행시킬 수 있도록 하기 위한 상호연결 네트워크로서 SCMIN을 제안하였다. SCMIN은 통신이 빈번하게 발생하는 클러스터 내부에 보다 빠른 경로를 제공할수 있을 뿐만 아니라 기존의 단일 경로 성질을 갖는 MIN에서 사용된 스위칭 소자의 갯수($\frac{N}{2} \times \log_2 N$)보다 매우 적은 수인 $(2.5N-4)$ 개만을 사용함에도 불구하고 다수의 다중 경로를 제공하여 오류 허용을 가능하도록 설계되었다.

본 논문에서는 SCMIN이 기존의 MIN에서 사용되는 적은 수의 스위칭 소자를 사용함에도 불구하고 임의의 한 프로세서에서 모든 메모리 모듈로 접근할 수 있는 FAC(Full Access Capability)가 만족됨을 증명하여 보였으며, 또한 지역화된 클러스터에 최대 n 개의 다중 경로를 제공할 수 있음을 증명하였다. 더우기 시스템

크기가 증가할수록, 네트워크상에서 오류 허용 및 교통의 혼잡에 대해 이를 우회할 수 있게 하는 다중 경로의 수 또한 증가함을 보였다.

SCMIN의 확장 설계는 동일 크기인 UNIT 모듈로써 점진적 확장 기법을 통해 이루어지기 때문에 미리 설치되었던 적은 시스템 크기의 SCMIN을 그대로 재사용을 할 수 있고, 사용중인 SCMIN과의 연결 방법이 매우 단순하기 때문에 실제 상용 다중 프로세서 시스템에서 쉬운 확장 구현이 이루어질 수 있을 것이다. 따라서 SCMIN은 하드웨어 비용과 성능면을 고려하였을 때 기존의 MIN보다 효율적인 네트워크라고 할 수 있다. 따라서 프로세서와 메모리 모듈로 구성된 작은 크기의 클러스터에 높은 지역화 통신 분포를 가지며, 또한 통신 지연 시간이 네트워크의 조합 능력(combinatorial power)보다 더 중요하게 되는 병렬 응용 분야에 효과적으로 사용할 수 있는 상호연결 네트워크이다.

그러나 성능 분석결과에서도 보았듯이 SCMIN에 적용하는 병렬 프로그램 응용 분야의 통신 분포가 지역화 되어 있지 않는 균일 분포 특성을 가질 경우에 기존의 MIN과 비교하였을 때 성능이 떨어지는 것으로 나타나고 있다. 또한 SCMIN에서는 기존의 MIN 보다 적은 수의 스위칭 소자를 사용하기 때문에 네트워크의 조합 능력이 떨어지게 되는 부작용이 따르고 있다. 향후 연구로서, SCMIN을 통해서 절감된 많은 수의 스위칭 소자를 이러한 단점을 보완하기 위한 활용 방안으로 연구가 이루어진다면 지역화된 통신 형태에서 뿐만 아니라 균일 분포를 갖는 병렬 프로그램 응용 분야에서도 우수한 성능을 나타낼 수 있는 MIN을 설계할 수 있을 것으로 기대된다.

참 고 문 헌

- [1] S.G. Abraham, and E.S. Davidson, "A Communication Model for Optimizing Hierarchical Multiprocessor System," In *Proc. Int'l Conf on Parallel Processing*, pp.467-474, 1986.
- [2] G.B. Adams III, D.P. Agrawal, and H.J. Siegel, "A Survey and Comparison of Fault-Tolerant Multistage Interconnection Network," *IEEE, Compt.*, pp.14-27, June, 1987.
- [3] R. Agrawal and H.V. Jagadish, "Partitioning Techniques for Large-Grained Parallelism," *IEEE Trans. Compt.*, vol. C-37, pp.1627-1634, Dec., 1988.
- [4] B.D. Alleyne and I.D. Scherson, "Expanded Delta Networks for Very Large Parallel Computer," In *Proc. Int'l Conf. on Parallel Processing*, vol. I, pp.127-131, 1992.
- [5] R.J. Baron and L. Higbie, *Computer Architecture Case Studies*, Addison-Wesley Pub., 1992.
- [6] P.T. Breznay and M.A. Lopez, "A Class of Static and Dynamic Hierarchical Interconnection Networks," In *Proc. Int'l Conf. on Parallel Processing*, vol. I, pp.59-62, 1994.
- [7] C.M. Chiang, S. Bhattacharya, and L.M. Li, "Multicast in Extra-Stage Multistage Interconnection Networks," In *Proc. the 6th IEEE Symp. on Parallel and Distributed Processing*, pp.452-459, Oct., 1994.
- [8] A.L. Decegama, *The Technology of Parallel Processing : Parallel Processing Architectures and VLSI hardware volume I*, Prentice-Hall International Editions, 1989.
- [9] M. Dubois and S.S. Thakkar, *Cache and Interconnect Architectures in Multiprocessors*, Kluwer Academic Pub., 1990.
- [10] T. Hanawa, H.Amano, and Y. Fujikawa, "Multistage Interconnection Networks with multiple outlets," In *Proc. Int'l Conf on Parallel Processing*, vol. I, pp.1-8, 1994.
- [11] M. Jurczyk and T. Schwederski, "On Partially Dilated Multistage Interconnection Networks with Uniform Traffic and Nonuniform Traffic Spot," *Proc. of 5th IEEE Symp. on Parallel and Distributed Processing*, pp.788-795, 1993.
- [12] V.P. Kumar and S.M. Reddy, "Augmented Shuffle-Exchange Multistage Interconnection Networks," *IEEE Compt.*, pp.30-40, Aug., 1987.
- [13] K.Y. Lee and W. Hegazy, "The Extra Stage Gamma Network," *IEEE Trans. Compt.*, vol. C-37, pp.1445-1450, Nov., 1988.
- [14] N. Suzuki, *Shared Memory Multiprocessing*, The MIT Press, 1992.
- [15] D.A. Patterson and J.L. Hennessy, *Computer Architecture A Quantitative Approach*, Morgan Kaufmann Pub., 1996.
- [16] G.F. Pfister et al., "The IBM Research Parallel Processor Prototype(RP3): Introduction and Architecture," In *Proc. Int'l Conf. on Parallel Processing*, pp.764-771, 1985.
- [17] D.K. Pradhan, *Fault-Tolerant Computer System Design*, Prentice-Hall PTR, 1996.
- [18] *SES/workbench Rel. 3.0*, Scientific and Engineering Software, Inc., 1995.



최 창 훈

1988년 2월 명지대학교 전자계산학과 졸업. 1990년 2월 서강대학교 대학원 전자계산학과 공학석사 취득. 1990년 1월 ~ 1990년 9월 대우통신 기술개발부 근무. 1997년 8월 서강대학교 대학원 전자계산학과 공학박사 취득. 1997년 9월 ~ 현재

국립 상주대학교 컴퓨터공학과 조교수.



김 성 천

1975년 서울대학교 학사. 1979년 Wayne State University Computer Engineering MS. 1982년 Wayne State University Computer Engineering Ph.D. 1982년 ~ 1984년 California State University Computer

Engineering 조교수. 1984년 ~ 1985년 금성반도체(주) 책임연구원. 1985년 ~ 현재 서강대학교 컴퓨터학과 교수. 관심분야는 Parallel Computer System, Interconnection Network, Process Scheduling, Faulttolerance System