

성능 향상을 위한 교차된 불완전 하이퍼큐브의 구조 및 경로배정 기법

正會員 임 화 경*, 홍 경 화**, 김 성 천*

Routing scheme and Structure of Crossed Incomplete Hypercube for Enhancing Performance

Hwa-Kyung Rim*, Kyung-Hwa Hong**, Sung-Chun Kim* *Regular Members*

*본 연구는 한국과학재단 연구비지원(과제번호: 941-0900-036-2)에 의한 것임

요 약

하이퍼큐브 구조는 두 노드간의 직경이 짧고, 노드간의 경로배정이 비교적 간단하다는 장점으로 인해 실제의 시스템에서 많이 이용되고 있다. 그러나 이 구조를 좀 더 실질적으로 만들기 위하여 직경을 줄이고 임의의 노드 갯수로 시스템을 확장할 수 있어야 한다.

본 연구에서는 노드나 링크의 추가없이 기존의 구조의 복잡도를 그대로 유지하면서 직경을 약 50% 정도 줄였던 교차된 하이퍼큐브 구조를 이용하여, 임의의 노드 갯수로 확장할 수 있는 교차된 불완전 하이퍼큐브 구조를 제안하였으며, 이를 위한 경로배정 기법을 제안하였다. 그리고 제안한 구조가 평균 경로길이 및 평균 경로배정시간 측면에서 약 30%, 교통밀도 측면에서 약 20% 향상됨을 입증하였다.

ABSTRACT

Hypercube has some advantages such as small diameter between nodes and relatively simple routing scheme, so it is one of the most popular interconnection networks used in real systems. But to make this structure more practical, it is required to reduce its diameter and to be able to expand it to any number of nodes.

In this study, we proposed a Crossed Incomplete Hypercube structure. This scheme is based on the previous Crossed Hypercube topology that reduces the diameter about 50% and has the same complexity with general hypercube and is able to expand to any number of nodes. We also proposed a fixed routing scheme for this structure. In the respect of the average path length and the average routing time, this structure shows an enhanced performance about 30% and it is enhanced about 20% in the average traffic density.

*서강대학교 전자계산학과

**한국사이베이스(주)

論文番號: 95277-0816

接受日字: 1995년 8월 16일

I. 서 론

일반적으로 MPP는 크게 SIMD 대 MIMD, Fine

grain 대 Coarse grain, 연결구조 등에 의해 분류될 수 있다. 이 중에서 연결구조는 공유메모리 대 분산메모리, 상호연결의 위상(topology)에 따라 분류될 수 있으며, 독립적인 기능을 하는 많은 모듈을 포함하면서 데이터가 수천 개의 모듈에 걸쳐 분산되어 있는 상황에서 많은 프로그램을 효율적으로 데이터와 제어 신호의 전송 부하를 줄이면서 실행하기 위해서는 처리소자간에 효율적인 상호통신체계가 필요하다. 이를 위해 MPP 시스템 내의 각각의 프로세서들은 다양한 방법으로 연결될 수 있다. 이러한 방법들 중 대표적인 구조로, 하이퍼큐브 구조는 근원지 노드에서 목적지 노드로 방문시 거치는 링크의 수 즉, 최대치의 직경이 다른 구조보다 짧으며, 노드간 경로배정 알고리즘이 비교적 간단하다는 장점을 갖기 때문에 다른 위상에 비해 우위를 점하고 있다. 따라서, 좀 더 성능을 향상시킬 수 있고 실질적인 측면에서 원하는 크기의 병렬 기계로 확장시키기 위한 연구가 다방면으로 이루어지고 있다⁽¹⁾.

본 연구에서는 일반적인 하이퍼큐브 구조 및 변형된 하이퍼큐브 구조의 단점들을 제하고 구조적 성능을 향상시키기 위하여 기존의 교차된 하이퍼큐브 구조⁽²⁾와 불완전 하이퍼큐브 구조⁽³⁾를 결합시켜 새로운 형태의 교차된 불완전 하이퍼큐브(crossed incomplete hypercube)구조를 제안하였다. 제안한 구조는 직경의 감소등의 잇점을 얻기 위해 노드 또는 링크의 추가없이 직경을 감소 하였으며 임의의 노드 수로도 시스템을 확장시킬 수 있는 능력을 제공함으로써 성능 향상에 영향을 주었다. 제안한 구조에 대한 구조와 특성 그리고 경로 배정 기법에 대하여 설명하고, 이 구조의 향상된 성능을 시뮬레이션을 통하여 입증하였다.

II. 제안된 교차된 불완전 하이퍼큐브 구조

1. 구조 및 특성

제안한 교차된 불완전 하이퍼큐브 구조(그림 1)는 기존의 교차된 하이퍼큐브(crossed hypercube) 구조에서 원하는 갯수의 노드를 제거시킨 것으로 기존의 불완전 하이퍼큐브 구조에서 두개의 이진 스트링 $x = x_1 x_0$ 와 $y = y_1 y_0$ 의 값, (x, y) 가 $\{(00,00), (10,10), (01,11), (11,01)\}$ 에 포함되는 노드끼리 인접하게 위치시키

로써 구성할 수 있다⁽²⁾.

정의 1. 교차된 불완전 하이퍼큐브

CI'_n 은 n 차원 교차된 하이퍼큐브에서 r 개의 노드를 제거한 교차된 불완전 하이퍼큐브 구조이다($r < 2^{n-1}$).

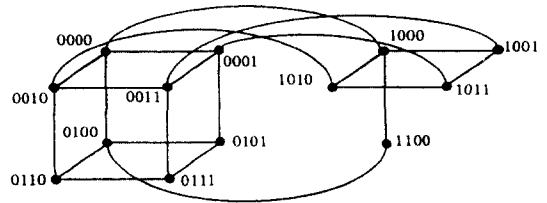


그림 1. 4차원의 교차된 불완전 하이퍼큐브 구조
Fig. 1 4-D crossed incomplete hypercube

본 연구에서 정의한 n 차원 교차된 불완전 하이퍼큐브 구조란 시스템 내의 모든 노드를 이진수로 표현하는 데에 필요한 비트 수가 n 개인 교차된 불완전 하이퍼큐브 구조를 의미한다. 이 구조에서 노드를 제거할 때에는 가장 큰 이진 주소를 갖는 노드부터 차례로 제거된다. 따라서 교차된 불완전 하이퍼큐브 구조, CI'_n 은 다음 두 개의 그룹으로 나누어질 수 있다. 즉 한 그룹은 $(n-1)$ 차원의 완전 교차된 하이퍼큐브 구조이고, 다른 한 그룹은 r 개의 노드가 제거된 $(n-1)$ 차원의 교차된 하이퍼큐브 구조이다. 따라서, CI'_n 은 $CQ_{n-1}(0)$ 와 $CQ_{n-1}(r)$ 의 결합으로 나타낼 수 있다. 여기서, $CQ_x(y)$ 는 y 개의 노드가 제거된 x 차원의 교차된 하이퍼큐브 구조를 의미한다. 그리고 이 시스템 내의 총 노드 갯수는 $2^n - r$ ($r < 2^{n-1}$)개가 되고, 총 링크 갯수는 표 1의 알고리즘에 의해 구할 수 있다.

교차된 불완전 하이퍼큐브 구조의 이러한 노드 및 링크 갯수는 불완전 하이퍼큐브 구조와 일치하므로 시스템 구성에 드는 비용 면에서는 차이가 없다고 볼 수 있다. 인접 노드의 구성은 기존의 교차된 하이퍼큐브 구조와 동일하다⁽²⁾. 즉 두 개의 이진 스트링 $x = x_1 x_0$ 와 $y = y_1 y_0$ 의 값, (x, y) 가 $\{(00,00), (10,10), (01,11), (11,01)\}$ 에 포함되면, x 와 y 는 pair-relate되었다고 말하고 $x \sim y$ 로 표기하며, 다음의 조건을 만족하면서 그 노드가 존재할 때 두 노드는 서로 이웃하게 된다. 즉, 정

표 1. 교차된 불완전 하이퍼큐브의 링크 갯수를 찾는 알고리즘

Table 1. Link counting algorithm of CI

```

□ 교차된 불완전 하이퍼큐브의 링크 갯수를 찾는 알고리즘 Algorithm D.
{r: 제거하려는 노드의 갯수, n: 교차된 불완전 하이퍼큐브의 차원, ℓ: 링크의 갯수}
ℓ: n2n-1
rℓ: 0
for i = 1 to r do
    if mod(i, 4) = 1 then
        rℓ := rℓ + (n - div(i, 4))
    else
        rℓ := rℓ + (n - (div(i, 4) + 1))
    endif
end do
ℓ := ℓ - rℓ
    
```

점 u 는 $CQ_{n-1}(0)$ 로 부터 오고 $0u_{n-2} \dots u_0$ 의 값을 가지며, 정점 v 는 $CQ_{n-1}(r)$ 로 부터 오고 $1v_{n-2} \dots v_0$ 의 값을 가질 때, (1) n 이 짝수이면, $u_{n-2} = v_{n-2}$ 이고, (2) $u_{2i+1}u_{2i} \sim v_{2i+1}v_{2i}$ ($0 \leq i < \lfloor (n-1)/2 \rfloor$)이면 u 와 v 는 CI_n^r 에서 서로 이웃한다($0 \leq r < 2^{n-1}$).

2. 경로배정 기법

이미 정의한 제안된 구조에서의 경로배정 알고리즘은 표 2와 같다. 표 2에서 밑줄친 부분이 확장된 알고리즘을 나타낸다.

경로배정 기법에서는 근원지와 목적지 노드의 상대주소를 최상위 비트부터 살펴보았을 때, 그 값이 1인 비트 위치를 $LeftmostDifferingBitIndex(u, v)$ 를 이용하여 찾아낸다. 또한 임의의 노드 u 로부터 링크 번호 dim 으로 연결된 상대노드는 $FindEndNode(u, dim)$ 를 이용하여 구할 수 있다. 그리고 두 노드 α 와 β 가 서로 pair-relate 관계에 있을 때에는 교차된 링크를 이용하고, 그렇지 않을 때에는 주어진 MOVE 테이블을 이용하여 메시지를 다음 중간 노드로 전달한다(표 3). 다음의 보조정리는 교차된 하이퍼큐브 구조의 성질에 의해 도출되었으며, 보조정리와 정리는 알고리즘 E가 정확히 동작함을 나타낸다.

보조정리 1. $n \geq 1$ 인 모든 노드 n 에 대해서, $(u_{n-1} \dots$

표 2. CI_n^r 의 경로배정 알고리즘

Table 2. Routing algorithm of CI_n^r

```

□  $CI_n^r$ 의 고정 경로배정 알고리즘
Algorithm E.
{u: source, v: destination}
ℓ := LeftmostDifferingBitIndex(u, v)
if (ℓ is odd) and ( $u_{\ell-1} \neq v_{\ell-1}$ ) then
    if there exists such a link  $dim := choice(\ell, \ell-1)$ ;
    else  $dim := \infty$ 
endif
else
    if there exists  $\alpha := u_{2k+1}u_{2k}$ ,
         $\beta := v_{2k+1}v_{2k}$  such that not( $\alpha \sim \beta$ )
        and ( $\ell > 2k + 1$ ) then
        use the leftmost such  $\alpha, \beta$  in
         $dim := choice(MOVE(\alpha, \beta))$ 
    else
         $dim := \ell$  { u and v are adjacent }
    endif
endif
n := FindEndNode(u, dim)
if ( $n \geq \#$  of nodes) or ( $dim = \infty$ ) then
     $dim := LeftmostDifferingBitIndex(u, v)$  such
    that there exists such a link endif
route to dim-neighbor
    
```

표 3. CI_n^r 의 고정 경로배정을 위한 MOVE 테이블

Table 3. MOVE table for routing of CI_n^r

α	β	00	01	10	11
00	00	ϕ	{ ℓ }	{ $2k + 1, \ell$ }	{ $2k$ }
01	01	{ $2k$ }	{ $2k + 1, \ell$ }	{ ℓ }	ϕ
10	10	{ $2k + 1, \ell$ }	{ $2k$ }	ϕ	{ ℓ }
11	11	{ ℓ }	ϕ	{ $2k$ }	{ $2k + 1, \ell$ }

$u_0, v_{n-1} \dots v_0$ 가 CI_n^r 의 간선이면 다음의 조건을 만족하는 l 이 존재하며, 그 역도 역시 성립한다.

- 1) $u_{n-1} \dots u_l = v_{n-1} \dots v_l$
- 2) $u_{l-1} \neq v_{l-1}$
- 3) $u_{l-2} = v_{l-2}$ if l is even
- 4) for $0 \leq i \leq \lfloor (l-1)/2 \rfloor$, $u_{2i+1}u_{2i} \sim v_{2i+1}v_{2i}$, and
- 5) $|u_{n-1} \dots u_0| < 2^n - r$, $|v_{n-1} \dots v_0| < 2^n - r$

위의 조건들은 최상위 비트부터 살펴볼 때, 그 값이 다른 최초의 비트 위치를 찾아내고, 그 차원의 링

크로 두 노드가 연결됨을 pair-relate를 이용하여 나타낸 것이며, 마지막으로 두 노드가 모두 교차된 불완전 하이퍼큐브 구조 내에 존재하여야만 이 조건을 만족하는 링크가 존재함을 보여준다. 이 고정 경로배정 알고리즘은 현재 노드와 목적지 노드의 주소를 최상위 비트부터 하위 비트로 살펴면서 수행된다. 처음에 현재 노드 u 와 목적지 노드 v 의 최상위 다른 값을 갖는 비트의 위치를 l -셋트시킨다. l 이 주어지면, l 의 왼쪽에 위치한 비트들에 대해서는 보조정리 1의 조건 1)에 따라 고려 대상에서 제외되고, l 의 오른쪽에 위치한 비트들에 대해서만 다음의 3가지 경우를 고려할 수 있다.

경우 1. 주어진 l 이 홀수이고 $u_{l-1} \neq v_{l-1}$ 이면, 현재 노드의 l -이웃이나 $(l-1)$ -이웃 중에서 시스템 내에 존재하는 링크를 임의로 선택한다.

경우 2. 주어진 l 이 짝수이거나, l 이 홀수이지만 $u_{l-1} = v_{l-1}$ 이면, 1보다 작은 pair-relate 관계에 있는 상위 α, β 를 찾는다. 만약 이 조건을 만족하는 α, β 가 존재하면, 두 노드는 서로 이웃하므로 현재 노드의 l -이웃을 선택한다. 그렇지 않으면, 주어진 MOVE 테이블을 이용하여 링크를 선택한다.

경우 3. 주어진 시스템이 불완전 하이퍼큐브 구조이므로 위의 경우 1과 경우 2에 의해 선택된 링크가 시스템 내에 존재하는가를 조사한다. 만약 선택된 링크가 존재하면, 현재 노드로부터 그 링크를 통해 목적지 노드나 목적지 노드로의 경로 상에 있는 중간 노드로 메시지를 보낸다. 그렇지 않은 경우에는 불완전 하이퍼큐브 구조의 경로배정 알고리즘과 같이 현재 노드로부터 링크가 존재하는 최상위 다른 값을 갖는 비트의 차원으로 메시지를 보낸다.

정리 1. CT_n^1 내의 임의의 정점 u 와 v 에 대해서, 알고리즘 E는 항상 u 로 부터 v 로의 최단경로 상에 있는 u 의 이웃노드를 발견한다.

증명) 보조정리 1과 교차된 하이퍼큐브 구조에서 이용한 몇몇 성질들을 위의 3가지 경우 별로 적용하여 살펴본다.

단계 1. 이 경우는, 전체 n 차원 하이퍼큐브 구조 중 l 의 상위 비트들에 의해 결정되는 서브큐브 내의 두

개의 $(l+1)$ 차원 서브큐브의 연결 관계를 나타내는 것으로, 하나의 $(l+1)$ 차원 서브큐브 내에 존재하는 현재 노드에서 다른 하나의 $(l+1)$ 차원 서브큐브 내의 목적 노드로 가는 가능한 경로 상에서 가장 가까운 이웃 노드는 보조정리 1에 따라 링크 l 이나 링크 $(l-1)$ 로 연결된 것이므로 이 링크가 존재하면 그 중 하나를 선택하는 것이 최적이다.

단계 2. 이 경우는 보조정리 1의 조건 1), 2), 3)에 해당하므로 만약 조건 4), 5)만 만족된다면 두 노드는 교차된 링크를 이용할 수 있다. 따라서 l 의 하위 비트 중 pair-relate 관계를 갖는 링크를 발견하면, 교차된 링크, 즉 링크 l 을 이용하는 것이 최적이다. 만약 pair-relate 관계를 갖는 링크를 발견하지 못하면, MOVE 테이블에 따라 최적의 링크를 선택한다.

단계 3. 단계 1 및 단계 2를 거쳐 선택된 링크가 시스템의 불완전 하이퍼큐브 구조의 성질 때문에 그 링크가 시스템에 존재하지 않는다면, 현재 노드에서 경로배정될 다음 노드로 갈 수 있는 경로(직진 경로 또는 우회 경로)에 대하여 링크 선택을 시작한다. 즉, 이 방법을 선택할 수 있는 링크에 대한 제한을 두는 것으로, n 보다 작은 단계 내에 성공적으로 메시지를 전달할 수 있게 된다. 특히 이 때에는 우회 채널을 사용하므로써 교차상태를 막을 수 있다. Q.E.D

다음 예를 통해 간단하게 경로배정 기법을 이해할 수 있다. 그림 2의 CT_3^1 에서 근원지 노드 010 으로부터 목적지 노드 101 로 메시지를 전송할 때, Leftmo-

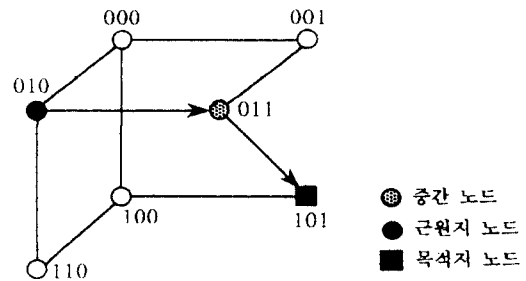


그림 2. CT_3^1 의 경로배정 예
Fig. 2 Routing example of CT_3^1

stDifferingBitIndex(010, 101)은 2 이므로 MOVE 테이블을 찾아 dim은 0이 된다. FindEndNode(010, 0)이 011로 존재하므로 메시지는 중간 노드 011로 전송된다. 마지막 단계에서는 LeftmostDifferingBitIndex(011, 101)은 2이고 11과 01은 pair-relate 관계를 이루므로 dim은 1, 즉 2가 되어 메시지는 목적지 노드인 101로 전송된다.

III. 시뮬레이션 결과 분석

제안한 구조에 대한 성능 평가는 SLAM II를 이용하여 확률에 의한 이산사건 시뮬레이션으로 수행되었다^(5,6). 시뮬레이션 환경은 다음과 같다.

① 가정: 1 사이클 (10ns), 단위시간 (10,000 시간단위), 링크당 1 사이클

② 평균 경로 길이 (average path length)

- 임의의 노드가 다른 노드와 통신시 필요로 하는 평균 거리, 즉 평균 링크 수로 본 시뮬레이션에서는 균일 메시지 분포를 가정하므로 하나의 노드에서 다른 하나의 노드로 메시지를 전송하는 평균 비율이 시스템 내의 모든 노드에 대해 동일하다.

③ 최대 경로 길이 (maximum path length)

- 임의의 노드가 다른 노드와 통신시 필요로 하는 최대 거리, 즉 최대 링크 수를 의미한다.

④ 평균 경로배정 시간 (routing time)

- 임의의 노드로부터 다른 노드로 메시지를 전송하는 데 드는 평균 사이클 수를 의미한다.

⑤ 평균 메시지 교통 밀도 (message traffic density)

- 각 노드가 평균적으로 다른 각 노드로 메시지를 전송하는 데 걸리는 단위시간 동안 시스템 내에 생성된 모든 메시지가 차지하는 링크의 수에 대한 시스템 전체의 링크 수의 비율로서, 시스템 내의 각 링크가 단위시간 동안 지게 되는 부담을 나타내며 다음 공식에 의한다.

$$\text{평균 메시지 교통 밀도} = \frac{\text{평균 경로 길이} \times \text{노드 갯수}}{\text{통신 링크 갯수}}$$

또한, 시뮬레이션 중 사용된 용어는 다음과 같다.

ICUBE: Incomplete hypercube

CCUBE: Crossed Incomplete hypercube

i-max.: incomplete hypercube's maximum value

i-avg.: incomplete hypercube's average value

c-max.: crossed incomplete hypercube's maximum value

c-avg.: crossed incomplete hypercube's average value

불완전 하이퍼큐브 구조 및 교차된 불완전 하이퍼큐브 구조에서 노드를 전혀 제거하지 않은 구조는 기존의 하이퍼큐브 구조와 교차된 하이퍼큐브 구조가 된다. 이 경우의 성능 평가는 노드가 제거된 불완전 하이퍼큐브 구조에서의 성능 평가에 참조를 위하여 행해졌다. 성능 평가 결과, 교차된 하이퍼큐브 구조가 평균 경로 길이에서 약 30%, 최대 경로 길이에 있어서 약 50%의 성능 향상을 보인 반면, 최대 경로배정 시간과 평균 경로배정 시간의 경우 교차된 하이퍼큐브 구조의 인접노드를 찾는 방식(pair-relate)의 복잡도로 인해 반전을 보인다. 결과는 그림 3과 그림 4에서 볼 수 있다.

비교적 적은 수의 노드와 많은 수의 노드를 완전 하이퍼큐브 구조에서 노드를 제거해 본 경우가 그림 5와 그림 6에서 설명된다. 이하 시뮬레이션 결과는 지속적인 프로세서의 성능 향상으로 인해 경로배정의 복잡도를 고려하지 않았다. 따라서 경로배정 시간

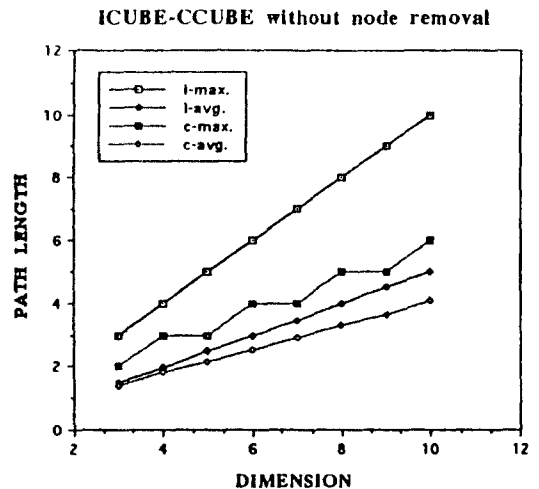


그림 3. 고정 경로배정을 이용한 하이퍼큐브와 교차된 하이퍼큐브의 평균경로길이 및 최대경로길이

Fig 3. average path length and max path length of CQ & Q for fixed length

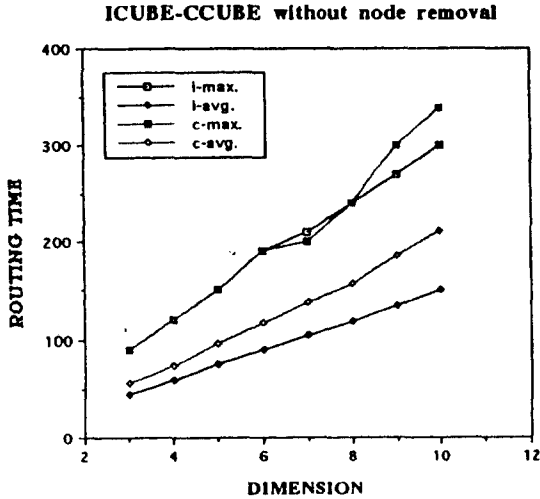


그림 4. 고정 경로배정을 이용한 하이퍼큐브와 교차된 하이퍼큐브의 평균 경로배정시간 및 최대 경로배정 시간

Fig. 4. average routing time and max routing time of CQ & Q for fixed routing

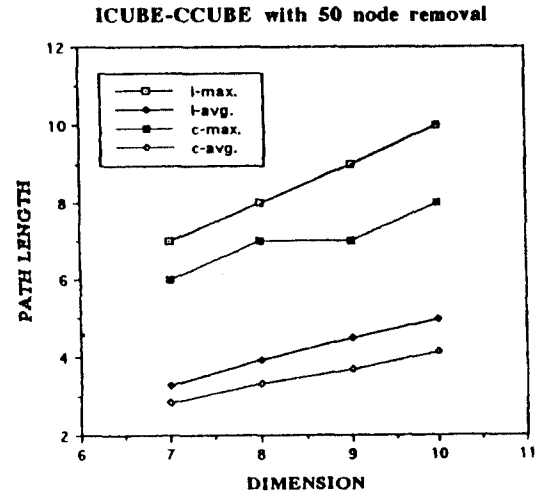


그림 6. 시스템당 50개의 노드를 제거한 불완전 하이퍼큐브와 교차된 불완전 하이퍼큐브의 고정 경로배정을 이용한 평균경로길이 및 최대 경로길이

Fig. 6 average path length and max path length of IQ & CI with removed 50 nodes for fixed routing

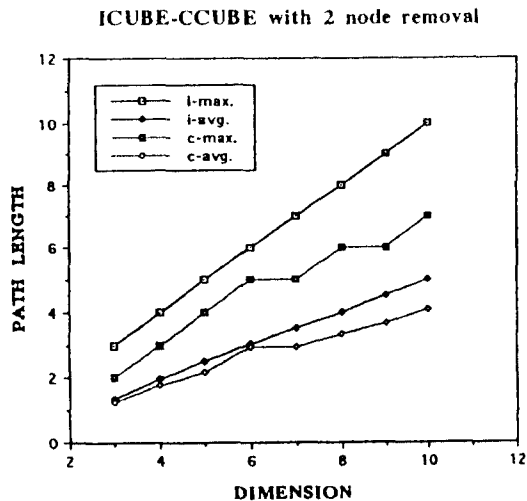


그림 5. 시스템당 두개의 노드를 제거한 불완전 하이퍼큐브와 교차된 불완전 하이퍼큐브의 고정 경로배정을 이용한 평균경로길이 및 최대 경로길이

Fig. 5 average path length and max path length of IQ & CI with removed 2 nodes for fixed routing

의 비교 결과는 경로 길이의 비교치와 동일한 형태를 보인다. 모든 경우 경로길이에 있어서 교차된 불완전 하이퍼큐브 구조가 30% 정도 우위에 있다. 그리고 차원이 높아질수록 교차된 링크를 사용할 수 있는 경우의 수가 늘어나므로 두 시스템의 성능차는 더욱 커지게 된다.

이제 비교적 적은 수의 노드와 많은 수의 노드를 완전 하이퍼큐브 구조에서 제거해 본 경우의 평균 교통 밀도 측면에서의 비교가 그림 7과 그림 8에서 보여진다. 모든 경우 평균 교통 밀도에 있어서 교차된 불완전 하이퍼큐브 구조가 약 20% 정도 낮은 수치를 보이므로 단위 시간당 시스템의 각 링크가 지게 되는 부담이 20% 줄어들게 되어 불완전 하이퍼큐브 구조에 비해 메시지 집중 현상이 적게 발생하고, 동일한 링크 부담율에 대해서는 시스템 내에 더 많은 수의 메시지를 포함할 수 있다. 그리고 교차원으로 갈수록 이용가능한 링크의 수가 2의 계승으로 늘어나기 때문에 평균 교통 밀도 역시 교차원으로 갈수록 차가 커진다.

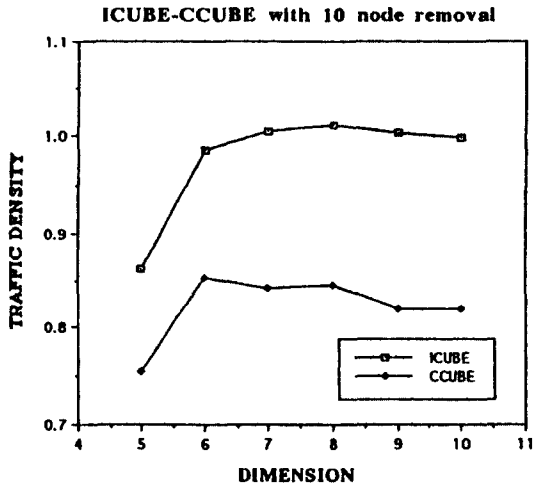


그림 7. 시스템당 10개의 노드를 제거한 불완전 하이퍼큐브와 교차된 불완전 하이퍼큐브의 고정 경로배정을 이용한 평균 교통밀도

Fig. 7 average traffic density of IQ & CI with removed 10 nodes for fixed routing

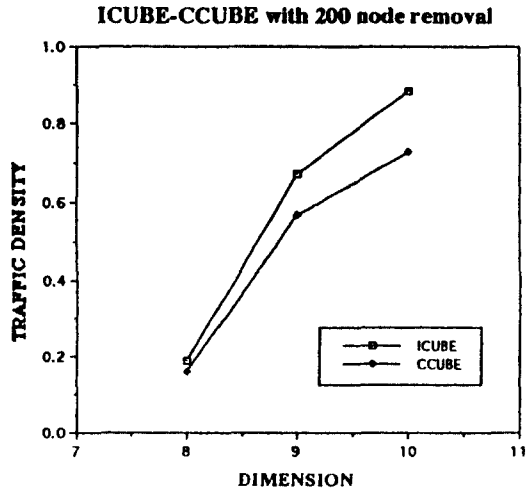


그림 8. 시스템당 200개의 노드를 제거한 불완전 하이퍼큐브와 교차된 불완전 하이퍼큐브의 고정 경로배정을 이용한 평균 교통밀도

Fig. 8 average path length and max path length of IQ & CI with removed 200 nodes for fixed routing

IV. 결 론

교차된 불완전 하이퍼큐브 구조는 일반적인 하이

퍼큐브 구조 및 다른 변형된 하이퍼큐브 구조의 단점들을 제거하는 것을 목표로 제안되었다. 즉 이 구조는 직경의 감소 등과 같은 잇점을 얻기 위해 노드나 링크를 추가시키지 않으면서도 직경의 감소가 가능하고, 임의의 갯수로도 시스템을 확장시킬 수 있는 능력을 갖는다. 이는 기존의 교차된 하이퍼큐브 구조와 불완전 하이퍼큐브 구조를 결합시킨 것으로서, 두 구조가 갖는 장점을 모두 포함한다. 따라서 교차된 불완전 하이퍼큐브 구조의 제안을 통해 프로세서 간의 연결 거리를 줄이므로써, 약 30%의 평균 경로 길이 향상과 약 20%의 평균 메시지 교통 밀도를 향상시켰다.

또한 하이퍼큐브 구조의 단점 중 하나인 2ⁿ개로 노드 수를 확장해야 한다는 점을 개선하여 임의의 갯수로도 시스템을 확장할 수 있으므로, 상당히 많은 수의 노드들을 이미 가지고 있는 대규모 병렬처리 시스템을 원하는 크기로 확장할 수 있다. 결과적으로 제안한 상호연결망 구조를 통해 보다 효율적으로 대규모 병렬처리 시스템을 구성할 수 있게 될 것이다.

참 고 문 헌

1. S.Darbha, E.W.Davis, "Network Design and Performance for a Massively Parallel SIMD System," 4th sym. on the Frontiers '92, pp.186-193, 1992.
2. K.Efe, "The Crossed Cube Architecture for Parallel Computation," IEEE Trans. on Parallel and Distributed systems, Vol.3, No.5, pp.513-524, Sep. 1992.
3. P.T.Gaughan, S.Yalamanchili, "Adaptive Routing Protocols for Hypercube Interconnection Networks," IEEE COMPUTER, pp.12-23, May 1993.
4. M.S.Chen, K.G.Shin, "Adaptive Fault-Tolerant Routing in Hypercube Multicomputers," IEEE Trans. Comput., Vol.39, No.12, pp.1406-1416, Dec. 1990.
5. A.Alan, B.Pritsker, Introduction to Simulation and SLAM II, John Wiley., 1986.
6. 이영해, 백두권, 시스템 시뮬레이션, 경문사, 1988.
7. H.P.Katseff, "Incomplete Hypercubes," IEEE Trans. Comput., Vol.37, No.5, pp.604-607, May 1988.
8. N.F.Tzeng, S.Wei, "Enhanced hypercubes," IEEE Trans. Comput., Vol.40, No.3, pp.284-294, Mar. 1991.



김 성 천(Sung-Chun Kim) 정회원
 1975년: 서울대학교 공과대학 공
 업교육학(전기전공)학사
 1976년~1977년: 동아컴퓨터(주)
 Sys. Eng.
 1977년~1978년: 스펀리유니맥
 Sales Rep.
 1979년: Wayne State Univ. 컴퓨
 터공학 석사

1982년: Wayne State Univ. 컴퓨터공학 석사
 1982년~1984년: 캘리포니아주립대 조교수
 1984년~1985년: 금성반도체(주) 책임연구원
 1986년~1989년: 서강대학교 공과대학 전자계산소 부
 소장
 1989년~1991년: 서강대학교 공과대학 전자계산학과
 학과장
 1985년~현재: 서강대학교 공과대학 전자계산학과 조
 교수(1985. 8~1987. 8), 부교수(1987. 9
 ~1992. 8), 교수(1992. 9~현재)
 1989년~현재: 한국정보과학회 병렬처리시스템 연구부
 부위원장(1989~1993), 위원장(1994~현
 재), 대한전자공학회 및 한국통신학회 분
 회지 편집위원(1991~현재, 1993~현재)
 ※주관심분야: 병렬처리시스템(Parallel Computer Ar-
 chitecture, Interconnection Network),
 Computer Network



홍 경 화(Kyung-hwa Hong) 정회원
 1989년 2월: 서강대학교 전자계
 산학과 졸업
 1994년 2월: 서강대학교대학원 전
 자계산학과 석사
 1994년~1995년 11월: 한국전자통
 신연구소(ETRI) 이
 동통신연구단 이동
 통신기술부 운용기
 술연구실

1995년 11월~현재: 한국사이베이스(주)
 ※주관심분야: 병렬처리시스템, 이동통신



임 화 경(Hwa-kyung Rim) 정회원
 1989년 2월: 홍익대학교 전자계
 산학과 졸업
 1989년~1991년: 한국건설당(주),
 교육센터
 1993년 2월: 서강대학교대학원 전
 자계산학과 석사
 1993년~현재: 서강대학교대학원
 전자계산학과 박
 사과정 재학중

※주관심분야: 병렬처리시스템, 셀룰러시스템