

교차된 불완전 하이퍼큐브 구조의 적응적 경로배정 기법과 방송 기법

正會員 임 화 경*, 최 창 훈*, 김 성 천*

Adaptive routing and Broadcasting scheme for Crossed Incomplete Hypercube structure

Hwa-Kyung Rim*, Chang Hoon Choi*, Sung-Chun Kim* *Regular Members*

※본 연구는 한국과학재단 연구비지원에 의한 것임(941-0900-036-2).

요 약

교차된 불완전 하이퍼큐브 구조는 노드나 링크의 추가없이 일반적인 구조의 복잡도를 그대로 유지하면서 직경을 약 50% 정도 감소시킨 교차된 하이퍼큐브 구조를 기반으로 하여 임의의 노드의 갯수로 확장할 수 있는 향상된 구조이다. 본 연구에서는 이 구조에 적합한 오류 발생시 대처가능한 오류허용 적응적 경로배정 알고리즘과 방송 알고리즘을 제안하였으며, 이에 대한 성능 평가 실험으로 성능이 30% 향상됨을 입증하였다.

ABSTRACT

A crossed incomplete hypercube structure is based on previous crossed hypercube structure that reduces the diameter about 50% and has the same complexity with general hypercube and is able to expand to any number of nodes. In this study, we proposed a fault-tolerant adaptive routing scheme and broadcasting scheme. Fault-tolerant adaptive routing scheme is resilient to system faults. And performance evaluation is executed on the proposed structure. It is enhanced about 30%.

I. 서 론

교차된 불완전 하이퍼큐브 구조(CI: Crossed Incomplete hypercube)는 기존의 교차된 하이퍼큐브 구조(CQ: Crossed hypercube)로부터 일부 노드를 제거한 불완전한 구조를 형성함으로써 직경(diameter)과 평균 길이, 메시지 교통 밀도등의 측면에서 복잡도를 증가시

*서강대학교 전자계산학과
論文番號:95393-1117
接受日字:1995年 11月 17日

키지 않는 동시에 성능의 향상을 얻기 위하여 제안된 구조이다. 이 구조는 노드 또는 링크를 추가시키지 않으면서도 50% 가량의 직경의 감소가 가능하고, 일반적인 하이퍼큐브 구조와 같이 2의 멱수가 아닌 임의의 갯수로도 시스템을 확장시킬 수 있는 장점을 갖고 있다[1, 2, 3, 9].

따라서, 본 연구는 CI 구조에서 경로배정시 오류가 발생할 경우 이를 효율적으로 대처하여, 최소한의 경로길이를 가지면서 가능한 한 교통량을 집중시키지 않는 적응적(adaptive) 경로배정 기법을 제안하였으며, 기존의 구조와 변형된 구조들의 방송 알고리즘을 고찰하여 이 구조에 적합한 방송(broadcasting) 알고리즘을 제안하였다. 특히 방송 알고리즘은 CI 구조의 특성을 사용하여 경로배정 기법과 유사한 최단 경로 길이를 보장하며, 같은 복잡도로 한 노드에서 다른 모든 노드로의 메시지 전송이 가능하다. 그리고, 제안한 기법의 성능을 증명하기 위해 시뮬레이션 언어 SLAMII를 사용하여 이산 사건에 기반한 시뮬레이션을 수행하여 그 결과를 평가분석하였다[6].

II. 기본 정의 및 성질

CI 구조에 적합한 경로배정 기법과 방송 기법을 효과적으로 수행하기 위하여 기본 구조의 특징을 고려하여 서브그래프(subgraph) 개념을 사용하였다. 이 개념의 사용은 고차원 구조를 구성할 때 확장을 용이하게 하기 위함이며, CQ 구조를 가정하여 정의하였다. 왜냐하면, CI 구조는 CQ 구조에서 임의의 개의 노드를 제거한 형태이므로 superset 개념으로 CQ 구조에 대한 정의로 가정하여도 문제가 되지 않기 때문이다. 서브그래프의 표기는 $Sub(\alpha, \beta, G)$ 로 하며, 이는 α 와 β 를 prefix 로 갖는 그래프 G의 서브그래프를 의미한다. 또한 $Sub(\alpha, \alpha, G)$ 는 $Sub(\alpha, G)$ 와 동일한 의미를 나타내며, $CQ(n)$ 은 n차원 CQ 구조를 의미한다.

정리 1) 모든 n에 대해서, $Sub(0, CQ(n)) \simeq CQ(n-1)$, $Sub(1, CQ(n)) \simeq CQ(n-1)$ 이다.

즉, 2차원 이상의 구조는 prefix 0, 1로 구성된 한차원 낮은 차원의 구조로 구성된다. 이것을 기반으로 하여 짝수 차원과 홀수 차원에 대한 정리를 하면 다음과 같다.

정리 2) $k \geq 1$ 일 때, $Sub(00, 10, CQ(2k)) \simeq CQ(2k-1)$ 과 $Sub(01, 11, CQ(2k)) \simeq CQ(2k-1)$ 로 구성된다.

예를들어, $k=1$ 일 때 $Sub(00, 10, CQ(2)) \simeq CQ(1)$ 과 $Sub(01, 11, CQ(2)) \simeq CQ(1)$ 로 구성되며, $k=2$ 일 때 $Sub(00, 10, CQ(4)) \simeq CQ(3)$ 과 $Sub(01, 11, CQ(4)) \simeq CQ(3)$ 으로 구성된다.

정리 3) $k \geq 1$ 일 때, $Sub(001, 111, CQ(2k+1)) \simeq CQ(2k-1)$, $Sub(011, 101, CQ(2k+1)) \simeq CQ(2k-1)$, $Sub(000, 100, CQ(2k+1)) \simeq CQ(2k-1)$, $Sub(010, 110, CQ(2k+1)) \simeq CQ(2k-1)$ 로 구성된다.

예를들어, k가 1일 때 $Sub(001, 111, CQ(3)) \simeq CQ(1)$, $Sub(011, 101, CQ(3)) \simeq CQ(1)$, $Sub(000, 100, CQ(3)) \simeq CQ(1)$, $Sub(010, 110, CQ(3)) \simeq CQ(1)$ 로 구성된다.

위와 같이 서브그래프의 개념을 사용하는 이유는 경로배정시 경로의 길이를 최소한으로 하고자 하는데 있다. 즉, 서브그래프를 한 정점으로 간주하여 경로배정을 수행하기 때문에 모든 차원의 구조가 서브그래프로 표현하면 2차원의 구조가 되므로 평균 경로 길이가 superset 측면에서 2가 된다. 이는 CI 구조의 직경을 일반적인 하이퍼큐브 구조의 직경보다 거의 반으로 줄일 수 있음을 의미한다.

1. 경로배정을 위한 상대적 위치 결정

통신을 하려는 근원지 노드(src)와 목적지 노드(dest) 간의 서브그래프 개념을 적용하여 경로배정시 이 두 노드가 동일 서브그래프내에 존재하는지 또는 이웃 서브그래프내에 존재하는지 아니면 이웃하지 않은 노드에 존재하는지를 구분하기 위하여 목적지 노드의 상대적인 위치 결정이 필요하게 된다. 이에 대한 조건은 [9]를 참조한다.

III. CI 구조의 적응적 경로배정 알고리즘

고정 경로배정(fixed routing) 알고리즘의 취약점을 개선하기 위해 도입된 적응적 경로배정 알고리즘은 다중 프로세서 구조에서 성능의 주요 구성 요소인 효율적인 프로세서간 상호통신을 위한 것으로, 오류와 병목현상에 대한 탄력있는 대처로 인해 높은 대역폭으로 프로세서간 상호통신을 가능케 해준다[7, 8].

제안한 알고리즘은 지역적 링크 정보만을 이용한

기법으로, 모든 노드는 그 자신의 링크 상태만 알며, 각 노드가 받게 되는 메시지는 $(k, [s_1, s_2, \dots, s_k], \text{message}, \text{tag})$ 의 형태를 취한다. 여기서 k 의 초기값은 현재 노드로부터 목적지 노드까지의 해밍 거리를 의미하며, 미스라우팅될 때마다 1씩 증가하게 된다. 그리고, $[s_1, s_2, \dots, s_k]$ 는 현재 노드로부터 목적지 노드까지 가기 위해 거쳐야 하는 각 차원의 순서를 의미한다. 역시 미스라우팅 시에는 k 값의 증가와 더불어 거치는 차원을 s_k 뒤에 덧붙인다. message는 보내고자 하는 메시지를 나타내며, tag는 미스라우팅 시에 이용하게 되며, 차원의 사용 내역을 유지하는 데이터 구조이다. 그리고 시스템 내의 각 노드당 오류가 아닌 링크의 갯수는 1이 아니며, 노드 오류는 노드에 연결된 모든 링크의 오류와 동일함을 가정한다. 각 노드당 오류가 아닌 링크의 갯수를 1이 아니도록 제한하는 이유는, 정상 링크가 하나인 노드로 메시지가 전달되고, 다시 이 노드로부터 다른 노드로 메시지를 보내려 할 때 후진추적이 요구되기 때문이다. 이러한 가정을 없애기 위해서는 각 노드가 자신의 지역 링크의 정보뿐 아니라 1홉 더 떨어진 이웃 노드의 정보까지도 알아야 할 필요가 있다. 따라서 앞서 설명한 바와 같이 제안한 알고리즘은 전진추적 기법이므로 이와 같은 가정을 세운다.

정리 4) CI_n^r 내의 임의의 정점 u 와 v 에 대해, 적응적 경로배정 알고리즘은 항상 오류가 발생한 노드나 링크를 피하면서 u 로부터 v 로의 최단 경로를 따라 메시지를 전달한다(CI_n^r 는 n 차원에서 r 개의 노드를 제거한 제안한 구조를 의미한다).

증명) CQ 구조에서 이용한 몇몇 성질들을 단계별로 적용하여, 제안한 알고리즘에 들어오기 이전의 상태는 고정 경로배정 과정과 유사한 단계를 거치므로써 각 노드가 $(k, [s_1, s_2, \dots, s_k], \text{message}, \text{tag})$ 형태의 메시지를 받는다[3, 7]. 메시지를 받은 각 노드는 메시지 유형과 현재 노드 및 목적지 노드를 이용하여 다음의 행동을 결정한다.

단계 1. k 가 0이 아닐 경우, 현재 노드로부터 목적지 노드까지 가는데 거쳐야 하는 차원의 순서, 즉 메시지에 포함된 $[s_1, s_2, \dots, s_k]$ 의 차원 순서로 메시지를 보낸다. 이 때, 보내고자 하는 차원의 링크가 모두 존

재하면 이 단계에서 k 단계에 메시지를 성공적으로 보낸다. 그러나 보내고자 하는 링크에 오류가 발생하거나 불완전 하이퍼큐브 구조의 성질때문에 원하는 링크를 이용할 수 없을 경우에는 단계 2를 따른다.

단계 2. 노드가 받은 메시지의 $[s_1, s_2, \dots, s_k]$ 내에 이용 가능한 링크가 존재하지 않을 경우, $[s_1, s_2, \dots, s_k]$ 에 존재하는 현재 노드에서 블럭된 모든 차원(1, 2, ..., k)을 태그에 기록한다. 이는 이미 블럭된 차원이므로 미스라우팅 시에 이용하지 못하도록 하기 위함이며, 그렇지 않은 경우 시스템 내에 이용 가능한 링크가 존재하는 최하위 차원을 선택하여 메시지를 보내므로써 미스라우팅을 수행한다. 이 때에는 위에 설명한 바와 같이 미스라우팅 시에 행해야 하는 동작을 취한다. 따라서 미스라우팅의 횟수는 최대한 $(n-1)$ 로 제한받게 되고, 이러한 제한으로 인해 라이브록을 막을 수 있다.

단계 3. k 의 값이 0이고 현재 노드와 목적지 노드가 일치할 경우, 주어진 메시지 $(k, [s_1, s_2, \dots, s_k], \text{message}, \text{tag})$ 를 이용하여 성공적으로 메시지의 경로배정을 마친다.

위의 단계 1, 단계 2, 그리고 단계 3을 거치게 되면, 시스템 내의 오류 링크의 총합이 차원 n 보다 작은 경우만 항상 성공적으로 메시지를 라우트할 수 있고, 이는 K.G.Shin의 Algorithm A1 분석을 통해 증명된 바 있다. 그러나 제안한 구조의 경우에는 불완전 하이퍼큐브 구조의 성질상 제거되는 노드의 수가 많아지고, 이에 따라 제거되는 링크의 수도 많아지게 된다. 따라서 제거되는 링크를 오류 링크와 같은 관점으로 볼 때, 시스템에 존재하는 오류 링크의 수는 차원 n 을 훨씬 초과하게 된다. 따라서 다음의 단계 4를 추가한다.

단계 4. 이 경우에는 가상 채널의 개념을 사용하여 하나의 물리적 채널을 두 개의 가상채널로 분리하고, 단계 1, 단계 2 및 단계 3에서 사용했던 가상채널과 다른 가상채널을 이용하여 경로배정을 수행한다. 즉 현재 노드와 목적지 노드의 상대 주소 중 최상위 다른 값을 갖는 비트를 찾아 이 차원의 링크가 이용가능하면 메시지를 보낸다. 따라서 최대 n 단계 안에 메시지를 성공적으로 목적지 노드로 보낼 수 있다. Q.E.D

그림 1의 예를 통해 시스템 상태에서 근원지 노드 1001로부터 목적지 노드 0110으로 메시지를 전송하려 한다. 고정 경로배정 알고리즘에 따라 근원지 노드 1001은 (4, [3, 2, 1, 0], message, 0000)을 받고, 적응적 경로배정 알고리즘에 따라 경로배정을 수행한다. 처음 단계에서, 노드 1001로부터 3차원 및 2차원 링크는 오류이므로 1차원 링크를 따라 (3, [3, 2, 0], message, 0000)을 노드 1011로 보낸다. 그리고 두번째, 세번째, 마지막 단계에서는 차례대로 3차원, 2차원, 0차원 링크를 따라 노드 0001, 노드 0111을 거쳐 목적지 노드 0110까지 메시지가 전송된다.

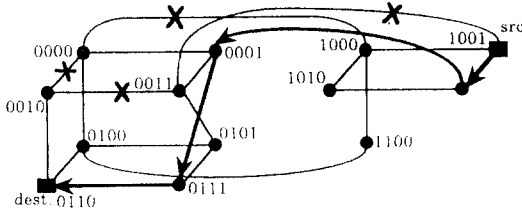


그림 1. 제안한 구조에서의 적응적 경로배정
Fig. 1 Adaptive routing scheme in proposed structure

IV. 방송 알고리즘

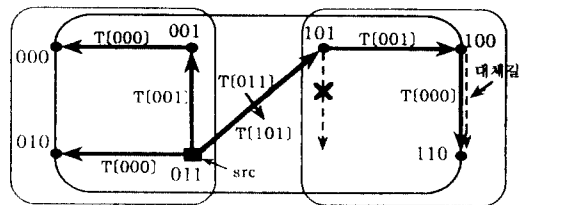
방송 알고리즘(broadcast algorithm)이란 임의의 한 근원지 노드가 동일 메시지를 자신을 제외한 모든 목적지 노드로 전송하는 통신 형태를 의미하는 것이다 [7, 8]. 제안한 방송 알고리즘은 근원지 노드에서 모든 노드로 단 한번에 중복없이 홀수 차원(CI(2k+1))에서는 $\lceil \log_2 n \rceil$ 단계로, 짝수 차원(CI(2k))에서는 $\lceil \log_2 n \rceil + 1$ 단계로 메시지를 전송 가능하다. 이 알고리즘은 교차된 하이퍼큐브 구조에서 임의의 노드가 제거된 구조에서의 방송으로 서브그래프 개념을 그대로 적용하여 방송 단계의 감소 효과를 가져올 수 있으며, 링크가 제거된 노드에서의 방송은 대체 링크를 사용함으로써 방송을 가능하게 할 수 있다. 따라서, tag 배열을 사용하고 있는데 이는 링크가 없는 대체 링크의 표시를 위해 사용된다. 또한 visit 플래그를 노드의 방송 유무를 나타내기 위해 사용한다. 이는 중복 방송을 막기 위함이다. 제안한 방송 알고리즘의 기본적으

로 사용한 요소는 다음과 같다.

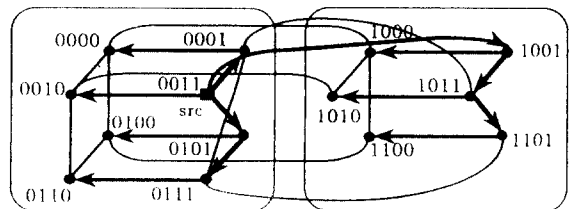
- Tag 배열 사용
- 링크가 없는 노드는 이웃 노드에서 대체 링크를 사용하여 경로배정
- 다른 차원(서브그래프 이용)으로의 경로배정시 pair 관계 고려

정리 5) 교차된 불완전 하이퍼큐브 구조의 방송 알고리즘은 짝수 차원에서는 $\lceil \log_2 n \rceil + 1$ 단계에서, 홀수 차원에서는 $\lceil \log_2 n \rceil$ 단계에서 수행완료한다.

예를들어, 노드 111이 제거된 구조를 가정한다. 홀수차원 구조의 경우 근원지 노드가 011일 때 방송 과정을 보면, 먼저 방송할 링크 0, 1, 2를 결정한다. 0 링크인 010으로 방송시 tag 배열은 000이 set되며, 1 링크는 001로 방송되고 tag 배열에는 001이 set된다. 또한 2 링크는 다른 차원으로의 방송이므로 pair 관계를 고려하여 101로 방송되며 tag 배열에는 011이 set된다. 여기까지는 없는 링크가 없으므로 다른 서브그래프로의 이동시 pair 여부만 고려하면 된다. 010 노드에서는 다른 차원으로의 방송이 가능하므로 pair를 고려한 노드 110으로 방송을 하게 된다(그림2-①). 예로 알수 있듯이 방송은 트리구조를 이루며 홀수차원



① 3차원 구조에서의 방송 예



② 4차원 구조에서의 방송 예

그림 2 교차된 불완전 하이퍼큐브구조에서의 방송알고리즘
Fig. 2 Broadcasting algorithm in CI

에서는 best case 일때 k 단계 즉, $\lceil \log_2 n \rceil$ 단계로 수행 가능하며, worst case 일 때는 $\log_2 n$ 으로 실행됨을 볼 수 있다. 그림 2의 ②의 구조는 노드 1110과 노드 1111 노드가 없는 구조이다. 근원지 노드를 0011이라 할 때 짝수 차원이기 때문에 가장 먼저 높은 차원으로 방송을 결정한 후에 홀수차원과 동일한 방송을 수행하게 된다. 즉, 근원지 노드를 포함하는 서브그래프를 기준으로 높은 차원은 1001 노드로 가는 링크가 되므로 이 링크를 결정하고 나머지는 홀수차원 알고리즘과 동일한 방송을 한다. 따라서, 짝수 차원에서는 홀수차원의 단계에서 1단계만 추가 실행하면 되므로 $\lceil \log_2 n \rceil + 1$ 단계로 수행 가능하다.

V. 성능 평가

본 연구에서 제안한 알고리즘을 성능 평가하기 위한 평가 요소는 다음과 같다.

- ① 평균경로길이(average path length): 다중 시스템에서 임의의 노드가 다른 노드와 통신할 때 필요로 하는 평균 거리
- ② 평균경로배정시간(routing time): 임의의 노드로부터 다른 노드로 메시지를 전송하는데 드는 평균 사이클 수로 한 사이클을 10ns로 가정
- ③ 평균메시지교통밀도(message traffic density):

$$\frac{\text{평균 경로 길이} \times \text{노드 갯수}}{\text{통신 링크 갯수}}$$

각 노드가 다른 노드로 메시지 전송시 걸리는 단위시간 동안 시스템내에 생성된 메시지가 차지하는 링크의 수에 대한 시스템 전체의 링크 수의 비율로 시스템내의 각 링크가 단위시간 동안의 갖는 부담 의미

5.1 성능평가 결과 및 분석

성능 평가는 SLAM II를 이용하여 확률에 의한 이산사건 시뮬레이션으로 수행되었고. 시뮬레이션 동안 시스템내에 적절한 수의 사건들을 발생시키기 위하여, 한 사이클은 10ns로 가정하고, 10,000 시간단위 동안 성능 평가를 진행하였다.

적응적 경로배정 기법의 시뮬레이션은 본 연구에서 제안한 기법 이외에 동일 시스템에서 비교할 수 있는 다른 적응적 기법이 존재하지 않으므로, 제안한 적응

적 기법상에서 오류 갯수에 따른 성능변화만을 관찰하였다. 시뮬레이션 대상이 되는 시스템은 다음과 같은 방식으로 생성된다. 즉, 교차된 완전 하이퍼큐브 구조에서 차원당 1/8의 비율로 노드를 제거하여 CI 구조를 생성하고, 이 시스템을 기반으로 오류 링크의 갯수와 오류 링크의 생성 비율을 변화시킨다. 시스템의 평균 경로 길이를 측정한 결과는 그림 3과 그림 4와 같다. 모든 경우에 대하여 오류의 갯수가 적을수록 평균 경로 길이가 줄어들어 좋은 성능을 보였다. 그리고, 8차원 이상에서는 오류 링크의 갯수가 시스템 전체의 링크 갯수에 대해 차지하는 비율이 비교적 적으므로 차원의 증가에 따른 별다른 차이가 없었다. 시스템의 평균 교통 밀도를 측정한 결과는 그림 5와 그림 6과 같다. 오류 갯수에 따른 그래프에서는 저차원 시스템의 경우, 오류 하나당 7~10%의 비율로 시스템의 성능이 저하됨을 알 수 있으며, 오류 갯수와 오류 링크의 생성 비율에 따른 시스템의 성능은 모든 경우에 오류의 갯수가 적을수록 좋은 성능을 보였다. 이 경우 역시 8차원 이상에서는 오류 링크의 갯수가 시스템 전체의 링크 갯수에 대해 차지하는 비율이 비교적 적으므로 차원의 증가에 따른 별다른 차이가 없다.

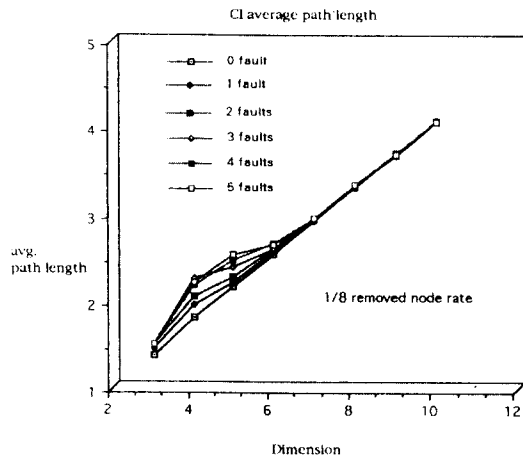


그림 3. 시스템당 1/8 비율로 노드를 제거한 CI에서 적응적 경로배정을 이용한 오류링크 갯수에 따른 평균 경로길이

Fig. 3 Average path length by the number of fault link in CI(1/8 removed node)

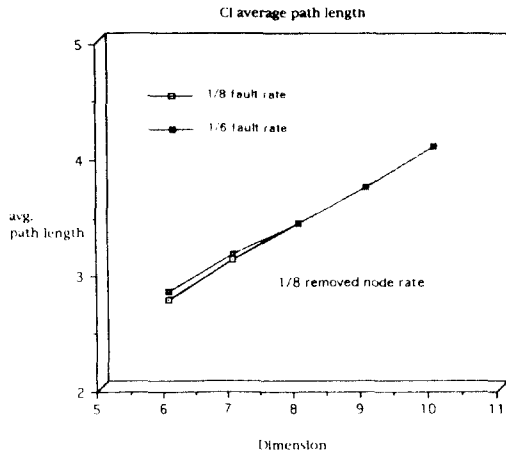


그림 4. 시스템당 1/8 비율로 노드를 제거한 CI에서 적응적 경로배정을 이용한 오류링크 비율에 따른 평균 경로길이

Fig. 4 Average path length by fault rate in CI(1/8 removed node)

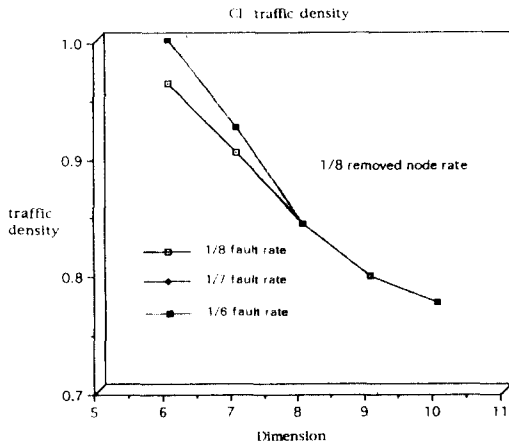


그림 5. 시스템당 1/8 비율로 노드를 제거한 CI에서 적응적 경로배정을 이용한 오류링크갯수에 따른 평균 교통밀도

Fig. 5 Average traffic density by the number of fault link in CI(1/8 removed node rate)

VI. 결 론

일반적으로 하이퍼큐브 구조로 부터 일부 노드를 제거한 불완전 하이퍼큐브 구조의 직경 및 평균 경로

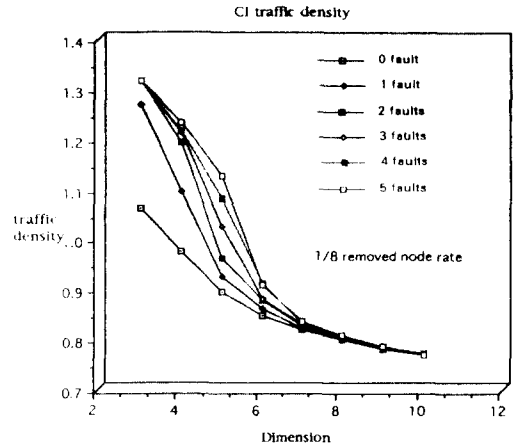


그림 6. 시스템당 1/8 비율로 노드를 제거한 CI에서 적응적 경로배정을 이용한 오류링크비율에 따른 평균 교통밀도

Fig. 6 Average traffic density by fault rate in CI(1/8 removed node rate)

길이, 메시지 교통 밀도등의 측면에서 구조의 복잡도를 증가시키지 않는 동시에 성능 향상을 얻기위해 시도된 구조가 교차된 불완전 하이퍼큐브 구조이다. 교차된 불완전 하이퍼큐브 구조를 위해 고정 경로배정 기법은 이미 개발하였으며, 성능 평가 결과 약 30%의 성능 향상을 보인 바 있다[9]. 이 구조를 위해 제안된 오류허용을 위한 적응적 경로배정 기법 역시 노드당 허용할 수 있는 오류 링크 갯수의 제한을 따른다면, N 개의 오류링크에 대해 성공적으로 수행됨을 시뮬레이션을 통해 알 수 있었다. 시뮬레이션은 오류 노드 및 링크가 동적으로 시스템에 발생하는 것을 전제로 한 결과, 6차원 이하의 시스템에서는 오류 링크 하나당 7~10%의 비율로 성능이 저하됨을 보았다. 또한 교차된 하이퍼큐브 구조에서 임의의 노드가 제거된 구조에서의 방송으로 서브그래프 개념을 적용하여 근원지 노드에서 모든 노드로 단 한번에 중복없이 홀수 차원(CI(2k+1))에서는 $\lceil \log_2 n \rceil$ 단계로, 짝수 차원(CI(2k))에서는 $\lceil \log_2 n \rceil + 1$ 단계로 메시지를 전송 가능하였다.

참 고 문 헌

1. R. Michael Hord, Parallel Supercomputing in SIMD Architectures, CRC press, 1990.

