

K-ary n-cubes에서 가상채널을 사용한 적응적 웜홀 라우팅

이 성 목[†] · 김 창 수^{††}

요 약

본 논문은 k-ary n-cubes에서 다중 전송에 기반을 두고 있으며, 결합허용과 채널 경쟁을 고려한 적응적 웜홀 라우팅 알고리즘을 제안한다. 제안된 알고리즘은 k-ary n-cube에서 차수 n에 대하여 (n-1)의 결합허용을 보장하기 위해 하나의 물리적 채널당 2n개의 가상채널을 사용한다. 이것은 결합허용을 고려한 기존의 적응적 라우팅 알고리즘보다 적은 수의 가상채널을 사용한다. 카오스 시뮬레이터를 이용하여 본 논문이 제안한 적응적 라우팅 알고리즘의 교통량과 결합허용을 고려한 메시지 지연도 함께 실험하였다.

An Adaptive Wormhole Routing using Virtual Channels in K-ary n-cubes

Sung Mok Lee[†] · Chang Soo Kim^{††}

ABSTRACT

This paper is based on multicasting model in k-ary n-cubes, and proposes an adaptive wormhole routing algorithm which allows faults and channel contention. The proposed algorithm only requires $2 \times n$ virtual channels per physical channel which is proportional to the dimension n in order to allow (n-1) faults in a k-ary n-cube. This method uses smaller number of virtual channels than the previously proposed adaptive routing algorithms [5, 18]. Through a chaos simulator, we have measured message delay considering fault-tolerant as well as message traffic to our adaptive routing algorithm.

1. 서 론

단일 프로세서(single processor)로는 해결할 수 없는 여러가지 응용분야에 대한 해결책으로 다중컴퓨터(multi-computer) 구조가 도입되었다. 다중컴퓨터 구조에서는

프로세서들간의 메시지 교환을 위해 상호 통신이 요구되는데, 효율적인 통신을 위해서는 라우팅 문제가 고려되어야 한다. 이러한 라우팅 문제는 네트워크 토폴로지, 통신 유형, 스위칭 기법에 따라 다양한 기법들이 적용될 수 있는데, 라우팅과 관련해서 가장 중요한 것은 시스템의 성능 향상을 위해 통신 시간을 줄이는 것이다. 이러한 관점에서 최적의 라우팅 알고리즘이 고려되어야 하지만 이것은 NP-hard 문제로 알려져 있다[14]. 따라서, 기존의 연구들은 대부분 휴리스틱(heuristic) 접근 방법에 기반을 둔 효율적인 라

※이 논문은 1996년도 부경대학교 기성회 연구비 지원에 의한 논문임.

† 정 회 원:교보정보통신 기술개발팀

†† 정 회 원:부경대학교 전자계산학과

논문접수:1997년 3월 13일, 심사완료:1997년 10월 14일

우딩 알고리즘을 제안하는데 집중되었다[1, 5, 14, 19].

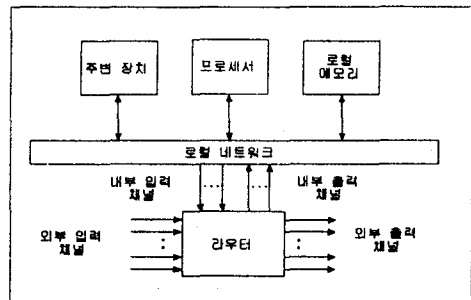
Store-and-forward 스위칭 기법을 사용한 과거 대부분의 라우팅 메카니즘은 라우팅 메카니즘 그 자체에서 교착상태(deadlock)를 방지하기 보다는 입력 버퍼와 출력 버퍼에 관련된 흐름제어(flow control) 기법을 이용하여, 교착상태를 방지하였기 때문에[2, 3], 버퍼에 대하여 비싼 추가비용이 요구된다. 그러나 최근 다중 컴퓨터에서는 wormhole 스위칭 기술을 사용함으로써 앞의 방법에 대한 단점을 상당히 줄일 수 있었으나, 교착상태에 대한 특별한 주의가 부과적으로 필요하다. 이러한 교착상태에 대한 안전성을 보장하기 위하여 Dally가 제시한 가상채널 기법을 적용할 수 있으며[2, 4, 7, 10], 이러한 가상채널의 사용은 네트워크 경쟁으로 인한 패킷 지연에도 능동적으로 대처할 수 있다. 가상채널의 수를 많이 사용할수록 패킷 지연에 대하여 효율적으로 수행할 수 있지만 이에 따른 추가 비용이 많이 들기 때문에 이들 사이에는 trade-off가 존재한다.

라우팅 알고리즘은 통신 경로가 소스 노드에서 이미 결정되어진 결정적(deterministic) 라우팅 알고리즘[19]과 채널 경쟁 혹은 노드의 고장 등을 고려하여 동적으로 경로를 할당하는 적응적(adaptive) 라우팅 알고리즘으로 구분한다[5, 8, 10, 11, 12, 15, 18, 20]. 그리고 적응적 라우팅 알고리즘은 네트워크의 상태를 고려해서 라우팅을 수행하는데, 최단 경로만을 고려해서 라우팅을 수행하는 최단 경로 라우팅[18]과 비최단 경로도 허용하는 비최단 경로 라우팅[8, 15]으로 구분한다. 결정적 라우팅 알고리즘에서는 패킷들간의 교착상태를 방지하기 위하여 라우팅 수행시 네트워크 채널 사용이 제한되어져 있기 때문에, 임의의 중간 경로에서 노드의 결합이 발생할 경우 심각한 문제를 유발시킬 수 있는 단점이 존재한다. 적응적 라우팅 알고리즘은 동적 라우팅 경로를 선택하기 위한 추가 비용(알고리즘의 복잡성, 하드웨어 추가)이 결정적 라우팅 알고리즘보다 비싼 단점을 가지고 있다.

한편, 라우팅 알고리즘은 소스 노드와 목적지 노드의 통신 관계에 따라 단일 전송(unicasting)과 다중 전송(multicasting), 그리고 방송망 전송(broadcasting)으로 구분한다. 단일 전송은 하나의 소스 노드에 대하여 단지 하나의 목적지 노드로 전송하는 방식이며, 다중 전송은 하나의 소스 노드에서 다수의 목적지 노

드로 메시지를 전송하는 방식을 말한다. 방송망 전송은 소스 노드에서 다중 컴퓨터에 있는 모든 노드로 메시지를 전송하는 방식을 말한다. 본 논문에서 채택하고 있는 다중 전송 방식은 병렬 탐색 알고리즘과 그래프 알고리즘 수행시 적용할 수 있을 뿐만 아니라, 분산 공유 메모리 시스템에서 공유 자료들의 일관성 유지 프로토콜을 위한 공유 자료의 수정에도 적용될 수 있다[16].

본 논문에서와 같이 메시지 전송 방식에 기반을 둔 다중컴퓨터에서의 일반적인 노드 구조는 (그림 1)에 나타나 있다. (그림 1)에서와 같이 노드는 프로세서와 자신의 로컬 메모리 그리고 입출력 장치들을 포함할 수 있다. 입출력 장치들은 내부 채널과 외부 채널로 구분되는데, 이들은 내부 채널을 통해서 라우터와 데이터를 주고 받으며, 외부 채널은 인접한 라우터와의 통신에 관계하며, 두 노드들 사이의 패킷 전송은 이들 라우터를 경유한다.



(그림 1) 다중 컴퓨터의 일반적인 노드 구조
(Fig. 1) The general node architecture of multicomputer

한편, 최근 많은 연구가 수행되고 있는 wormhole 스위칭 기술은 패킷을 더욱 세분화된 플릿(flit) 단위로 나누는데, 첫번째 플릿에는 목적지 노드의 주소를 그리고 다음의 모든 플릿에는 실질적인 메시지를 싣고 파이프라인 형식으로 전송한다. Wormhole 스위칭 기술과 virtual cut-through 방식의 차이점은 패킷 버퍼의 크기에 있는데, wormhole 방식에서는 플릿에 대한 버퍼의 크기만 필요하기 때문에 virtual cut-through 방식보다 비용적인 면에서 효율적이다. 그러나 wormhole 스위칭 기술은 기존의 store-and-forward와 virtual cut-through 방식보다 결합 허용에 약한 단점을 가지

고 있다. 즉, wormhole 방식에서는 하나의 패킷이 임의의 시간에 하나 이상의 채널을 사용하는데, 이러한 통신 기법으로 인해 교착상태를 방지하기 위해서는 채널의 사용을 제한하게 된다. 따라서 라우팅 수행시 채널의 사용을 제한한다는 것은 결합허용을 저하시키는 결과를 초래한다[20].

2차원 메쉬 네트워크에서 wormhole 스위칭 기술을 사용한 실제적인 시스템으로는 Intel Paragon, Symult 2010 등이 있으며, 3차원 메쉬 네트워크에서는 MIT J-machine등이 있다. 그리고 NCUBE-2/3 시스템은 하이퍼큐브 네트워크를 사용하였다[9].

본 논문의 구성은 서론에 이어, 2장에서는 본 논문과 관련된 기존의 연구에 대해서 살펴보고 3장에서는 본 논문이 제시하는 라우팅 메카니즘에 대해서 설명한다. 4장에서는 시뮬레이션 결과 및 분석에 대하여 설명하고 마지막으로 결론을 맺는다.

2. 관련 연구

현재까지 다중 컴퓨터 구조에서 대부분의 연구는 결정적 wormhole 라우팅 알고리즘과 네트워크 상태를 고려한 적응적 알고리즘에 대하여 많은 연구가 진행되어져 왔다. 그러나, 이들 대부분의 연구들은 2차원 메쉬 네트워크 혹은 2차원 토러스 네트워크 환경에 알맞은 라우팅 메카니즘들로서, 차수가 증가함에 따른 복잡한 라우팅 메카니즘들에 대해서는 많은 연구가 수행되지 않았다. 비록 [2, 12, 15]가 3차원 하이퍼큐브 네트워크 환경을 고려하였지만 이들 라우팅 알고리즘들은 노드의 수가 매우 작은 8개 혹은 16개 정도만을 고려하였다. 그러므로, 노드의 수가 증가할수록 교착상태에 대한 위험성을 가지고 있으며, 아울러 채널의 적응성은 상당히 감소하여 네트워크 지연이 상대적으로 증가하게 되는 문제점이 존재한다.

Dally와 Seitz[2]는 채널 순환 의존성 그래프로 설명되는 교착상태에 대하여 채널 순환 의존성을 제거하기 위해 가상채널에 대한 개념을 소개하였으며, 토러스 네트워크와 하이퍼큐브 네트워크상에서 라우팅 알고리즘 수행시, 교착상태를 방지하기 위한 가상채널 선택에 대한 라우팅 함수를 정의하였다.

Park과 Yhun[18]은 Barrel shifter 네트워크 상에서 지름과 같은 수의 가상채널을 사용함으로써, 적응적 라우팅 메카니즘을 제시하였다. 즉, 출발지 노드의 가상채널 0번부터 시작하여 1씩 증가하여 목적지 노드에 도착하면 가상채널의 번호와 출발지 노드간의 거리가 같게 되도록 네트워크 지름과 같은 수의 가상채널을 사용하였다. 따라서, 이러한 라우팅 메카니즘은 노드의 수가 많은 k-ary n-cubes 네트워크에 적용시킬 경우, 노드의 수에 따라 하나의 물리적 채널당 가상채널의 수는 $2 \lfloor k/2 \rfloor$ 개 필요하기 때문에 k 가 증가할수록 구현이 불가능하다.

한편, Linder와 Harden[5]은 k-ary n-cubes 네트워크 환경에서 노드의 고장과 채널에 대한 경쟁을 고려한 라우팅 메카니즘을 제시하였다. 이 방법은 하나의 물리적 네트워크를 두 개 이상의 가상 네트워크로 분할한 후, 패킷의 방향에 따라 가상 네트워크상에 존재하는 가상채널을 선택함으로써 교착상태를 방지하였다. 그러나, [5]는 교착상태와 네트워크 지연에 대한 적응성을 높이기 위해 하나의 물리적 채널당 $2^{n-1}(n+1)$ 개의 가상채널을 사용하였기 때문에, 차수가 증가할수록 사용되는 가상채널의 수는 기하급수적으로 증가하는 단점이 있다.

Chien과 Kim[8]은 메쉬 네트워크에서 결합 허용을 고려한 적응적 라우팅 알고리즘을 소개하였다. 이 방법은 메쉬 네트워크의 각 차수에 대하여 가상 네트워크를 생성하여 패킷이 하나의 가상 네트워크에서 다음 가상 네트워크로 진행하도록 가상채널을 선택하는 방법이다. 그러나, [8]은 메쉬의 차수에 상관없이 결합 허용 노드가 항상 한 개로 제한되어 있기 때문에, 이 방법을 k-ary n-cubes 네트워크에 적용할 경우 채널에 대한 심한 경쟁이 발생하는 단점이 있다.

McKinley[19]는 메세지의 다중 전송시 네트워크상에서 발생하는 지연을 줄이기 위하여 각 목적지 노드들의 방문 우선순위를 결정한 후, [2]에서 제시된 라우팅 함수(식 1)에 의하여 라우팅을 수행하였다. 비록 [19]에서 제시된 라우팅 알고리즘이 다중 전송시 발생하는 채널에 대한 경쟁을 고려하였지만, 이들 라우팅 메카니즘들은 컴파일 시간에 라우팅 경로가 모두 결정되는 결정적 라우팅 알고리즘이기 때문에 노드의 고장과 관련된 결합 허용을 보장하지 못하는 단점이 존재한다.

$$R_{BTR}(x, y) = \begin{cases} C_{dpx} & \text{if } (|\Delta| > \frac{k}{2}) \\ C_{dtx} & \text{if } (\frac{-k}{2} \leq \Delta \leq -1) \dots \dots \dots \text{(식1)} \\ C_{dhx} & \text{if } (1 \leq \Delta \leq \frac{k}{2}) \end{cases}$$

3. 라우팅 메카니즘

3.1 가정 및 정의

본 논문에서 제시한 가정과 기호는 [2]에서 제시된 것과 유사하다. 먼저 본 논문에서 사용되는 가정들은 다음과 같다.

- 1) 각 노드는 임의 길이의 메시지를 생성할 수 있으며, 패킷은 일반적으로 하나의 단일 플릿보다는 길다.
- 2) 각 메시지의 첫번째 플릿을 받아들인 큐는 또 다른 메시지의 플릿을 받아들이기 전에 메시지의 나머지 부분을 받아들인다.

정의 1: 상호연결망 I 는 방향성을 가진 연결 그래프 $I = G(N, C)$ 로 나타낸다. 여기서 N 은 정점들의 노드들 집합을 나타내며, C 는 간선들로 통신 채널들의 집합을 나타낸다. 각각의 채널 c_i 는 $cap(c_i)$ 의 용량을 가진 큐를 의미한다.

정의 2: 라우팅 함수 $R: N \times N \rightarrow C$ 는 현재 노드 n_c 와 목적지 노드 n_d 에 대하여 n_c 에서 n_d 로의 경로 상에 존재하는 즉, $R(n_c, n_d) = c_n$ 상에 있는 채널 c_n 을 매핑시킨다. 단, 채널은 자기 자신의 경로($c_c \neq c_n$)는 허용하지 않는다.

정의 3: 다중 전송을 위해 채널에 있는 각 큐의 노드들의 정보를 구성으로 정의할 때, 채널 c_i 에 대한 큐내의 플릿들의 수는 크기 $n(c_i)$ 로 표시된다. 만약 채널 c_i 에 대한 큐내의 첫번째 플릿이 노드 n_d 로 되어 있으면, $head(c_i) = n_d$ 가 된다. 본 논문에서는 아래의 식을 만족한다면, 구성은 정당하다고 정의한다.

$$\forall c_i \in C, n(c_i) \leq cap(c_i)$$

정의 4: 라우팅 함수 R 에 대한 교착상태 구성은 합법적인 채널 구성을 만족하는 큐가 존재하지 않

는 것이다. 즉,
 $\forall c_i \in C, (head(c_i) \neq d; \text{ and } c_j = R(c_i, n))$
 $\Rightarrow n(c_j) = cap(c_j)$

3.2 라우팅 알고리즘

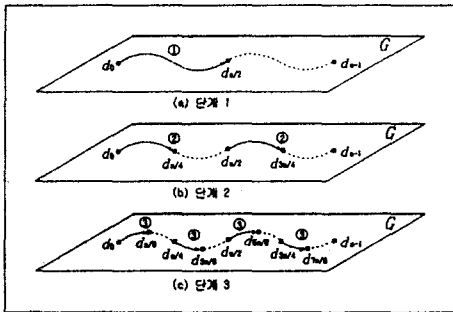
본 논문에서 제시하는 라우팅 알고리즘은 먼저 방문 우선순위 알고리즘에 의해 각 목적지 노드에 대한 방문 우선순위를 결정한 후, 이 우선순위에 따라 라우팅 함수를 적용함으로써 실제적인 라우팅이 수행된다. 본 논문에서 제시된 각 목적지 노드들에 대한 방문 우선순위 결정은 [16]에서 제시된 오일러 경로에 기반을 둔다. 다음 절에서 소개될 오일러 경로는 각 간선들에 대하여 방문 경로들이 중복 방문을 허용하지 않도록 각 노드를 방문하는 알고리즘으로서, 이러한 오일러 경로를 이용한 라우팅 알고리즘은 [3]에 의해 소개되었다.

3.2.1 오일러 경로(Eulerian ditrail)

(그림 2)와 같이 방향성 그래프 $G(V, E)$ 가 있을 때, $V(G)$ 를 모든 노드들의 집합, $E(G)$ 는 모든 간선들의 집합이라고 가정하자. 그리고 노드 $u \in V(G)$ 는 자신의 라우터를 가지는 프로세서를 나타내며, 간선 $(u, v) \in E(G)$ 과 간선 $(v, u) \in E(G)$ 는 라우터 u 와 라우터 v 를 상호 연결하는 양방향성의 링크를 나타낸다. 이때, 그래프 $G(V, E)$ 는 상호연결망으로 구성되어 있으며, 그리고 각각의 노드에 대하여 입력 차수와 출력 차수가 동일하다고 가정했을 때, 모든 노드와 간선들의 교차 순 즉, $v_0 e_1 v_1 e_2 \dots v_k - e_k v_k$ (단, $(v_{i-1}, v_i) \in E$)를 v_0 에서 v_k 까지의 디트레일(ditrail)이라고 하며, 이러한 디트레일을 포함하는 경로를 '오일러 경로'라고 한다. 각 노드에 대하여 입력 차수와 출력 차수는 방향성 그래프 G 가 대칭일 경우는 동일한데, 이때 방향성 그래프 G 에서는 모든 간선들을 한 번만 방문하여 모든 목적지 노드를 방문하도록 하는 디트레일 τ 가 존재하게 된다. 본 논문에서는 이러한 디트레일 τ 에 따라 각 노드들의 방문 우선순위를 결정한다.

노드 수가 m 개일 경우 $m = 2^k$ 를 만족하는 임의의 k 에 대하여, $2^{k-1} < m < 2^k$ 의 경우 τ 의 끝에 더미 노드를 추가시킨다. (그림 2)의 (a)는 오일러 경로 방문의 첫번째 단계로써, 소스 노드 d_0 는 τ 를 따라 노드 $d_{m/2}$ 를 방문한다. (그림 2)의 (b)는 두번째 단계로써, 소스 노

드 d_0 가 디트레일 τ 를 따라서 노드 $d_{m/4}$ 를 방문하는 동안에 노드 $d_{m/2}$ 는 노드 $d_{3m/4}$ 의 노드를 방문한다. (그림 2)의 (c)는 세번째 단계로서, 소스 노드 d_0 는 노드 $d_{m/8}$ 로 그리고 노드 $d_{m/4}$ 는 노드 $d_{3m/8}$ 로, 노드 $d_{m/2}$ 는 노드 $d_{5m/8}$ 로, 노드 $d_{3m/4}$ 는 노드 $d_{7m/8}$ 에 위치한 노드를 방문한다. 이후 단계에서는 전단계와 동일한 재귀적 방법을 사용하여 디트레일 τ 를 따라 각각의 목적지 노드들을 방문함으로써, 최소 단계수로 각각의 간선을 중복 경유하지 않고 모든 목적지 노드들을 방문할 수 있다.



(그림 2) 오일러 경로
(Fig. 2) Eulerian path

위의 오일러 경로를 라우팅 메카니즘에 적용함으로써, 네트워크간의 경쟁을 제거할 수 있는데 이에 대한 과정과 증명에 대하여 살펴보기로 한다. 다음의 기호와 기본 증명은 [19]를 따른다. 먼저, 오일러 경로에 대하여 하나의 소스 노드에서 하나의 목적지 노드를 방문하는 단일 경로(unicast) 연산을 $(u, v, P(u, v), t)$ 이라고 하자. 여기서, u 와 v 는 소스 노드와 목적지 노드를 나타내며, t 는 메시지 전송 단계((그림 2)에서 ①, $1 \leq i \leq 3$)를 나타내며, 그리고 $P(u, v)$ 는 방향성 그래프 G 내에서 메시지가 전송될 경로를 나타낸다. 그리고 멀티캐스트 집합을 $M = \{d_0, d_1, d_2, \dots, d_{m-1}\}$ 으로 가정했을 때, d_0 는 소스 노드를 그리고 d_1, d_2, \dots, d_{m-1} 들은 목적지 노드들을 나타낸다. 이때, 통신 경로들의 집합 $U_i = \{(u_1, v_1, P(u_1, v_1), t), (u_2, v_2, P(u_2, v_2), t), \dots, (u_k, v_k, P(u_k, v_k), t)\}$ 를 단일 통신의 집합이라고 가정한다.

보조정리: 목적지 노드들의 집합 M 에 대하여 단일 통

신 집합 U_1, U_2, \dots, U_k 는 다음 조건을 만족한다.

- 1) 임의의 $j(1 \leq j \leq k)$ 에 대하여, $(u, v, P(u, v), j) \in U_j$ 를 만족한다면 u 와 v 는 M 에 속한다.
- 2) $U_1 = \{(d_0, u, P(d_0, u), 1)\}$. 여기서, 어떤 $i(1 \leq i \leq m-1)$ 에 대하여 $u = d_i$ 이다.
- 3) 모든 단일 통신 $(u, v, P(u, v), t) \in U_i, 1 < t \leq k$ 에 대하여, 노드 w 의 멤버로서 $(w, u, P(w, u), j), j < t$ 를 갖는 집합 U_j 가 존재한다.
- 4) 모든 목적지 노드 $d_i(1 \leq i \leq m-1)$ 에 대하여 $1 \leq j \leq k$ 와 같은 유일한 정수 j 가 존재하며, 그리고 임의의 노드 w 에 대하여 $(w, d_i, P(w, d_i), j)$ 가 존재한다.

보조정리에서 조건 1)은 전송 메시지의 소스 노드와 목적지 노드들만이 라우팅에 직접적으로 관여하며, 다른 로컬 노드들은 단순히 라우터의 역할을 하도록 보장한다. 조건 2)는 라우팅 수행 첫번째 단계에서, 소스 노드는 목적지 노드들 중 하나의 목적지 노드로 라우팅을 수행한다. 조건 3)은 메시지를 받은 목적지 노드는 또다른 목적지 노드로 메시지를 전송하는 경우이다. 조건 4)는 모든 목적지 노드들은 메시지를 정확하게 한 번만 받아들인다.

정의 5: 멀티캐스트 라우팅 수행 $I(M) = \{U_1, U_2, \dots, U_k\}$ 에서, 만약 $v = u$ 가 존재하거나 또는 임의의 노드 $w \in R_u$ (즉, 노드 w 가 노드 u 에 의해 도달될 수 있는 집합)에 대하여 $(w, v, P(w, v), j) \in U_j$ 와 같은 $j(1 \leq j \leq k)$ 가 존재한다면, 노드 v 는 노드 u 에 의해 도달될 수 있는 집합(R_u 로 표시)내에 존재한다.

[정리 1]: 멀티캐스트 $I(M)$ 에서 모든 단일 경로들 $(u, v, P(u, v), t)$ 와 $(x, y, P(x, y), r)$ (단, $t \leq r$)의 쌍이 다음 네 개의 조건중 적어도 하나의 조건을 만족하면 $I(M)$ 은 채널 경쟁이 발생하지 않는다.

- 조건 1) $x \in R_v$.
- 조건 2) $P(u, v)$ 와 $P(x, y)$ 는 간선 분리(arc-disjoint).
- 조건 3) $x = u$
- 조건 4) $0 < l$ 에 대하여, $x \in R_w$ 와 $(u, w, P(u, w), t + l) \in I(M)$.

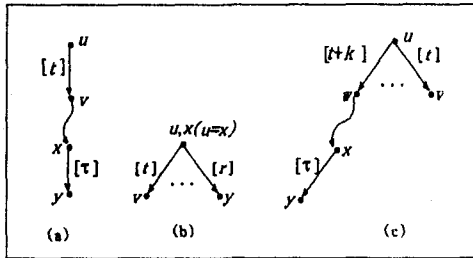
다음은 (정리 1)의 조건에 따라 채널 경쟁이 발생하지 않음을 설명한다.

조건 1)(그림 3)의 (a)와 같이, $x \in R_v$ 라면 u 에서 v 로의 단일캐스트 통신은 x 에서 y 로의 통신보다 이전에 완료되기 때문에(즉, $t \leq \tau$ 이므로 $u \in R_y$) 경쟁이 발생하지 않는다.

조건 2) 두 개의 경로가 하나의 가상채널에 대하여 서로 공유해서 사용하지 않을 경우 이들 두 경로를 간선 분리(arc-disjoint) 상태라 할 때, 메세지 전송의 두 경로 $P(u, v)$ 와 $P(x, y)$ 가 간선 분리이면 이들 두 경로에는 경쟁이 발생하지 않는다.

조건 3) u 는 어느 한 시점에서 단지 하나의 메시지만 다른 노드에 전송할 수 있다. (그림 3)의 (b)와 같이, $x = u$ 라 하더라도 조건이 $t < r$ 이기 때문에 노드 u 는 노드 y 로 메세지를 전송하기 이전에 v 로 먼저 메세지를 전송한다. 따라서, 경쟁이 발생하지 않는다.

조건 4)(그림 3)의 (c)와 같이, 노드 u 는 노드 x 의 부모가 되는 노드 w 에 메시지를 보내기 전에 노드 v 에 전송한다. 이는 노드 v 가 노드 x 전에 메세지를 받드시 수신하기 때문에 채널 경쟁이 발생하지 않는다.



(그림 3) 정리 1에 대한 조건 1, 3, 4의 예 (Fig. 3) Examples of condition 1, 3, 4 for the definition 1

3.2.2 우선순위 결정 알고리즘

라우팅 알고리즘을 수행하기 위한 이전의 단계로 각 목적지 노드들에 대한 우선순위를 결정하는데, 이러한 우선순위에 의하여 다음과 같은 장점이 존재한다. 즉, k-ary n-cubes에서 다중 메세지 전송시 교착상

태를 방지함과 동시에 (n-1)의 결합 허용을 보장하기 위하여 가상채널에 대한 갯수를 2n으로 제한할 수 있는데, 이에 대한 상세한 설명은 이 장의 후반부에 기술한다. 목적지 노드들에 대한 우선순위를 결정하기 위한 첫번째 과정으로 소스 노드와 목적지 노드들을 정렬한 후, 정렬된 결과에 따라 방문 우선순위를 결정한다.

본 논문에서는 3.2.1절에서 소개된 오일러 경로를 적용하기 위한 전단계로 소스 노드와 목적지 노드들을 먼저 (식 2)에 의하여 사전식으로 정렬한다. 이것은 라우팅 수행시 소스 노드와 소스 노드가 소유하고 있는 패킷의 복사본을 가진 다른 노드들이 단조 증가 또는 단조 감소로 중간 노드들을 방문하도록 하기 위한 것이다. 본 연구에서는 목적지 노드들의 주소 (x, y)에 대하여 y가 우선순위 값이 높은 것으로 그리고, 노드 (x1, y1)을 d1, 노드 (x2, y2)를 d2라고 가정했을 때, 다음 식이 성립한다.

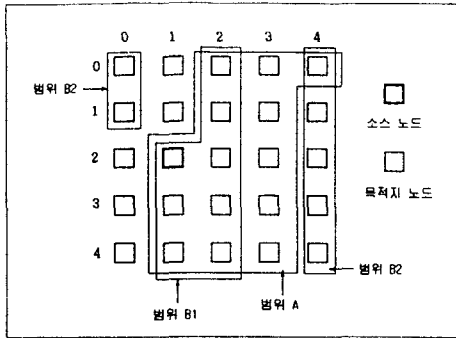
$$\begin{cases} d_2 > d_1 & \text{if } (y_2 > y_1) \text{ or } (y_2 = y_1 \text{ and } x_2 > x_1) \\ d_2 < d_1 & \text{otherwise} \dots\dots\dots \end{cases} \text{(식 2)}$$

(그림 4)는 오일러 경로에 의한 각 노드들의 방문 우선순위를 결정하는 알고리즘이다. (식 2)에 의하여 소스 노드와 목적지 노드들을 오름차순으로 정렬한 후, 소스노드 이후의 모든 노드들을 맨 좌측으로 이동시킨 노드들의 집합이 입력값이 되며, 출력값은 노드들의 방문 우선순위에 대한 집합이 된다.

```

Input : T=(ds,ds+1,ds+2,...,dm-1,d0,d1,...,dr-1)
Output : Visiting priority of nodes
Procedure :
while(left<right) (
  half=(right-left+1)/2;
  if(((half-(int)half)>0) then
    half=(int)half+1;
  else half=(int)half;
  Set Pi=(dhalf,dhalf-1,dhalf-2,...,dright); /* i is global variable */
  i+1;
  right=half-1;
)
return P;
    
```

(그림 4) 방문 우선순위 결정 알고리즘 (Fig. 4) Decision algorithm for visiting priority



(그림 5) 다중 전송시 오일러 경로의 단계 1과 단계 2의 수행 결과
 (Fig. 5) Result of step 1 and step 2 of Eulerian ditrail in multicasting

(그림 4)의 방문 우선순위 결정 알고리즘에 의하여 k-ary n-cubes에 대한 다중 메시지 전송시 교차상태와 함께 (n-1)의 결합 허용을 고려한 가상채널의 갯수를 줄일 수 있다.

예를 들어, (그림 5)와 같이 각각의 노드가 단방향성 채널로 연결된 5-ary 2-cubes 네트워크로 가정하였을 때, 다중 전송의 특별한 경우로 단일 소스 노드에 대해 모든 노드가 목적지가 될 경우를 고려함으로써 쉽게 설명된다. (그림 4)의 방문 우선순위 결정 알고리즘을 적용한 오일러 경로의 첫번째 단계에서 단일 경로 $(d_{(1,2)}, d_{(4,0)}, P(d_{(1,2)}, d_{(4,0)}), 1)$ 에 의해 범위 A가 결정되며, 두번째 단계에서는 두 개의 단일 경로 $(d_{(1,2)}, d_{(2,4)}, P(d_{(1,2)}, d_{(2,4)}), 2)$ 와 $(d_{(1,2)}, d_{(0,1)}, P(d_{(1,2)}, d_{(0,1)}), 2)$ 에 의하여 범위 B1과 범위 B2가 각각 결정된다. 그리고 세번째 단계 이후 $(C_i, D_i, \dots (단, 1 \leq i \leq 5))$ 는 나타내지 않았으나, 전단계와 동일하게 적용된다. 범위 A 내에서 결정되는 두번째 단계 이후의 모든 범위들 (B_1, C_1, \dots) 은 wrap around 간선을 사용하지 않고 모든 메시지를 전송할 수 있다. 이와함께 범위 B2는 차수 0 혹은 1에 대하여 최대 한 번의 wrap around 간선에 의하여 형성되며, B2 이후의 범위(예를들어, C2)에 있는 모든 단일 경로들은 더 이상의 wrap around 간선을 사용하지 않고 메시지를 전송할 수 있다. 결과적으로, 범위 A에 의해 생성되는 경로와 범위 B1과 B2에 의해 생성되는 경로들은 모두 정리 1에 의하여 채널 경쟁이 발생하지 않는다.

한편, (그림 5)에서 네트워크 구조가 양방향성 채널로 구성되어 있다고 가정하더라도 동일하게 적용할 수 있다. 즉, 단일 경로 $(d_{(1,2)}, d_{(4,0)}, P(d_{(1,2)}, d_{(4,0)}), 1)$ 는 그림과 같이 wrap around 채널을 이용하게 된다. 이때, 소스 노드 (1,2)의 출력 프레임이 해제되었을 때는 즉시 (그림 5)의 범위 B1을 결정하는 두번째 목적지 노드 (2,4)로 메시지를 전송할 수 있으며, 이때의 경로 $(d_{(1,2)}, d_{(2,4)}, P(d_{(1,2)}, d_{(2,4)}), 2)$ 는 wrap around 채널을 사용하지 않는다. 그리고, 경로 $(d_{(1,2)}, d_{(4,0)}, P(d_{(1,2)}, d_{(4,0)}), 1)$ 와 간선을 중복해서 사용할 가능성이 있는 경로 $(d_{(1,2)}, d_{(0,1)}, P(d_{(1,2)}, d_{(0,1)}), 2)$ 는 경로 $(d_{(1,2)}, d_{(4,0)}, P(d_{(1,2)}, d_{(4,0)}), 1)$ 이 종료해야만 가능하기 때문에 양방향성 채널에서도 간선 분리(arc-disjoint) 상태를 알 수 있다. 이상과 같은 결과에 의하여 임의의 시간(혹은 단계) t에서 간선을 중복 사용하는 경로들은 존재하지 않는다. 즉, 간선 분리(arc-disjoint)가 되어서 채널에 대한 경쟁이 발생하지 않기 때문에 결합허용만을 고려한 가상채널을 구성할 수 있다. 본 연구에서는 양방향성의 채널로 구성된 k-ary n-cubes 네트워크에서 (n-1)의 결합허용도를 보장하기 위해서는 하나의 물리적 채널당 2개의 가상채널을 사용한다.

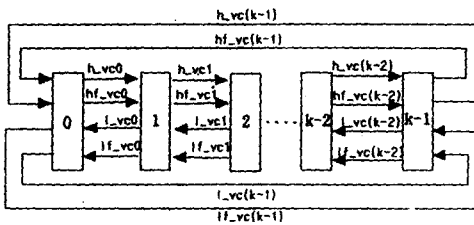
3.2.3 라우팅 함수

본 논문에서의 라우팅 알고리즘은 3.2.2절에서 소개된 방문 우선순위 결정 알고리즘에 의해 결정된 각 노드들이 정의 2에서 기술된 라우팅 함수 $(R(n_c, n_d) = C_n)$ 즉, 현재 노드 n_c 와 목적지 노드 n_d 에 대하여 n_c 에서 n_d 로의 경로상에 존재하는 채널 C_n 을 선택함으로써 라우팅을 수행하게 된다.

네트워크를 단방향성 채널로 연결할 경우 양방향성 채널로 구성할 때보다 하드웨어 구현시 단순하다는 잇점이 있는 반면 메시지 전송시 많은 오버헤드를 감수해야 하는 단점이 있다. 그러나 양방향성 채널로 구성된 네트워크의 경우에는 하드웨어 설계시 다소 복잡하지만 데이터 지역성(locality)으로 인하여 메시지 지연을 상당히 줄일 수 있는 장점이 있다. 즉, 인접하는 노드 A와 노드 B에 대하여 노드 A에서 노드 B로 메시지를 전송할 경우 노드 B는 노드 A로 메시지를 재전송할 확률이 상당히 높기 때문에 단방향성의 채널로 인한 메시지 지연을 최소화 할 수 있다.

본 논문에서는 양방향성 채널에 대하여 하나의 물

리적 채널당 4개의 가상채널을 사용하였다. (그림 6)은 임의의 한 차수에 대한 채널의 구성도를 나타내고 있다. (그림 6)에서 가상채널 h_vc 와 l_vc 는 오일러 경로에 의해 결정된 가상채널이고, 가상채널 hf_vc 와 lf_vc 는 임의의 경로에서 결합이 발생했을 경우 사용되는 가상채널이다. 즉, 정상적인 경로에 대해서는 h_vc 혹은 l_vc 를 따라 라우팅을 수행하며, 결합이 발생한 노드를 만난 이후부터는 hf_vc 혹은 lf_vc 를 따라 라우팅을 수행한다.



(그림 6) 가상채널 구성도
(Fig. 6) Configuration of virtual channels

한편, 라우팅 함수는 이들 4개의 가상채널을 선택하는 것으로 다음과 같이 정의할 수 있다. 여기서, $\delta_d(x)$ 는 소스 노드 x 의 주소, $\delta_d(y)$ 는 목적지 노드 y 의 주소, $\delta = \delta_d(y) - \delta_d(x)$ 는 각 차수에 대한 변위로 가정한다. 그리고 가상채널 $C_{M,uc}^n$ 은 현재 노드의 주소 n 과 출력 가상채널 vc 로 나타낸다. (그림 7)은 4개의 가상채널 선택에 대한 라우팅 함수를 나타낸 것이다.

(그림 9)는 결합 허용을 고려한 5-ary 2-cubes 네트워크의 양방향성 채널로 소스 노드를 (0,0), 목적지 노드들의 집합을 $\{(4,0), (1,2), (2,1), (2,3), (4,3), (3,2)\}$, 그리고 결합이 발생한 노드를 (2,0)이라고 가정하자. 이때, 3.2.2절의 식 2)에 기초한 방문 우선순위 결정 알고리즘의 입력 집합 $T = \{(0,0), (1,2), (2,1), (2,3), (3,2), (4,0), (4,3)\}$ 가 결정되며, 알고리즘에 의하여 다음과 같이 세 단계로 우선순위가 결정된다.

- 단계 1: (0,0) → (3,2)
- 단계 2: (0,0) → (2,1), (3,2) → (4,3)
- 단계 3: (0,0) → (1,2), (2,1) → (2,3), (3,2) → (4,0)

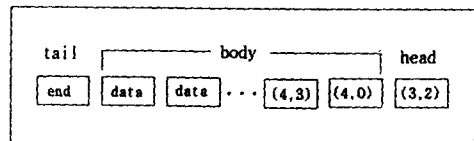
이후로 이들 각각의 단계는 앞에서 정의된 라우팅 함수를 따라 각각의 목적지 노드로 패킷을 전송하게 된다. 이때, 본 논문에서는 뱀홀 스위칭 기법을 사용하

기 때문에 하나의 패킷은 여러 개의 플릿(flit)으로 구성되는데, 플릿 구성은 (그림 8)과 같이 플릿의 몸체(body) 부분에 재귀적인 오일러 경로 공식을 적용하기 위한 다른 목적지 노드들의 주소를 포함시키고 있다. 즉, 오일러 경로의 첫 번째 단계를 수행한 결과로 목적지 노드 (3,2)가 소스 노드 (0,0)의 메세지 복사본을 가지기 때문에 목적지 노드 (3,2)는 노드 (4,0)과 (4,3)을 가지고 다시 재귀적으로 오일러 경로를 적용시킬 수 있다. 마지막 플릿에는 패킷의 끝을 나타내는 정보와 오류 정정을 위해 필요한 정보를 포함하게 된다.

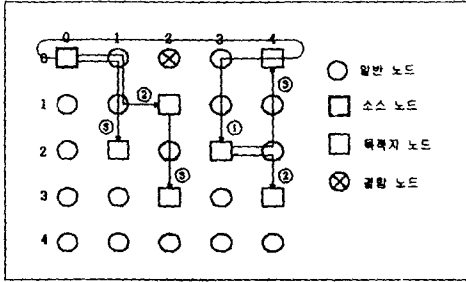
```

if (delta > k/2) then
  /* wrap around edges를 사용할 경우 */
  if (modk(δ_d(x))=0) then
    /* 현재 노드의 주소가 k-1일 경우 */
    if (δ_d(k-1) != fault node) then use C_{M,uc}^{k-1}
    /* k-1번째에 존재하는 노드와 l_uc번째의 가상채널 사용 */
    else ( select ∃min_node which isn't a current dim and shortest path )
    if (min_node is a positive path) then δ_d(min_node)
    /* 선택된 인접 노드의 주소가 현재 노드의 주소보다 클 경우 */
    if (modk(δ)=0) then use C_{M,uc}^{k-1}
    /* k-1 번째에 존재하는 노드와 hf_uc번째의 가상채널 사용 */
    else use C_{M,uc}^{δ_d(min_node)}
    /* 다른 차수에 존재하는 임의의 노드를 hf_uc번째의 가상채널을 사용 */
    else ( it's also applied to a negative path as similar to positive path )
  )
else use C_{M,uc}^{δ_d(x)}
/* 현재 노드와 lf_vc번째의 가상채널 사용 */
else ( it's also applied to an else-condition as similar to if-condition )
    
```

(그림 7) 라우팅 함수
(Fig. 7) Routing function



(그림 8) 플릿의 구성(단계 1)
(Fig. 8) Configuration of flit(step 1)



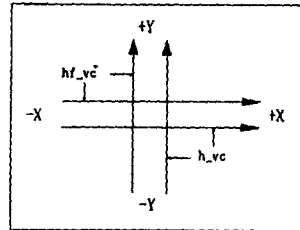
(그림 9) 결함 노드를 포함한 다중 전송의 수행 단계
(Fig. 9) Execution steps of multicasting with a fault-node

라우팅 함수에 의한 단계 1은 헤드 플릿의 목적지 노드 (3,2)의 x좌표와 소스 노드 (0,0)의 x좌표에 대한 변위(delta=3)가 k(k=5/2)보다 크기 때문에 모든 플릿들은 wrap around로 전송된다. 이후 모든 단계의 플릿들은 라우팅 함수에 의해 정상적으로 전송된다. 그러나, 단계 2에서 소스 노드 (0,0)에서 목적지 노드 (2,1)로 전송할 경우 결함 노드 (2,0)을 만나게 된다. 이때, 현재의 로컬 노드 (1,0)에서 진행중인 차수(0차수)와 다른 차수(1차수)에 있는 노드 (1,1)과 (1,4) 중에서 최단 경로에 존재하는 노드 (1,1)을 연결하는 가상채널 hf_vc를 선택하게 된다. (그림 9)는 양방향성 채널로 구성된 5-ary 2-cubes 네트워크에서 다중 전송 결과를 나타내고 있다.

[정리 2] 본 논문에서 제시한 결합허용 적응적 라우팅 알고리즘은 k-ary n-cubes 네트워크상에서 교착상태가 발생하지 않는다.

[증명] 본 논문의 결합허용 적응적 라우팅 알고리즘이 교착상태가 발생하지 않음을 증명하기 위해 라우팅시 사용된 가상채널들이 정의 2의 조건 2에서 정의된 간선 분리(arc-disjoint) 상태임을 보임으로써 증명한다. 그리고 증명의 단순성을 위해서 차수 n을 2로 고정시킨다. (그림 10)은 교착상태 방지를 위해 가상채널의 상태를 나타낸 것으로 패킷의 경로가 (-x, +x, P(-x, +x), a)일때 노드 +x 중에 있는 임의의 한 노드가 결합이 발생했다고 가정하자. 이때, 본 논문의 라우팅 함수에 의하여 경로 (-x, +x, P(-x, +x), a)은 일시적으로 단 한번의 경로 (-x, +y, P

(-x, +y), a) 또는 (-x, -y, P(-x, -y), a)을 따르며, 이후에는 다시 본래의 경로 (-x, +x, P(-x, +x), a)을 수행한다. 비록, 경로 (-x, +y, P(-x, +y), a) 또는 (-x, -y, P(-x, -y), a)가 진행 중일때, 단계를 서로 달리하는 또다른 경로 (-x, +y, P(-x, +y), b) 또는 (-x, -y, P(-x, -y), b)가 존재할지라도 경로 (-x, +y, P(-x, +y), a)과 (-x, -y, P(-x, -y), a)은 가상채널 hf_vc 또는 lf_vc를 사용하고, 경로 (-x, +y, P(-x, +y), b)과 (-x, -y, P(-x, -y), b)은 h_vc 또는 l_vc를 사용하기 때문에 간선 분리 상태가 된다. 그리고, 경로 (-x, +y, P(-x, +y), a)과 (-x, -y, P(-x, -y), a)은 이후에 계속해서 본래의 경로 (-x, +x, P(-x, +x), a)을 따르기 때문에 교착상태는 발생하지 않고 오일러 경로를 그대로 유지한다.



(그림 10) 교착상태 방지를 위한 가상채널의 상태
(Fig. 10) State of virtual channels for deadlock prevention

[정리 3] 본 논문에서의 결합허용 적응적 라우팅 알고리즘은 k-ary n-cubes 네트워크상에서 라이브락(livelock)이 발생하지 않는다.

[증명] 앞의 (그림 10)에서 +x 중에 있는 노드가 결합이 발생하였다고 가정하자. 이때, 정리 2의 증명에서 경로 (-x, +x, P(-x, +x), a)은 일시적으로 단 한번의 경로 (-x, +y, P(-x, +y), a) 또는 (-x, -y, P(-x, -y), a)을 따르며, 이후에는 다시 본래의 경로 (-x, +x, P(-x, +x), a)을 수행한다. 이것은 정리 1의 조건 1에서 조건 4까지의 증명에 의하여 최단 경로를 유지하며 간선 분리 상태를 따르기 때문에 언젠가는 목적지 노드에 도달하게 된다. 따라서, 본 논문에서의 결합허용 라우팅 알고리즘은 라이브락이 발

생하지 않는다.

4. 시뮬레이션 결과 및 분석

본 장에서는 본 논문에서 제안한 라우팅 알고리즘의 성능 평가를 위해 시뮬레이션 환경을 기술하고 성능 측정 요소별로 시뮬레이션 결과를 비교 분석한다.

4.1 시뮬레이션 환경

시뮬레이션 환경으로는 AVIION-5500 + 시스템에서 카오스(chaos) 시뮬레이터(워싱턴 대학에서 개발)를 사용하였으며, gcc 컴파일러로 컴파일하였다. 그러나 본 논문에서는 카오스 시뮬레이터가 제공하는 기본적인 기능을 이용하여 본 논문이 제시한 알고리즘에 적합하도록 수정하였다. 다음은 카오스 시뮬레이터의 특징을 나타낸 것이다.

- ① 기본적으로 k-ary n-cubes의 상호연결성을 제공하며, 가상채널 기법을 적용할 수 있다.
- ② Wormhole 라우팅과 virtual cut-through 라우팅 기법을 수행한다.
- ③ 패킷관련 정보와 목적지 노드들에 대한 정보를 파일 형식으로 사용자가 직접 입력할 수 있으며, 패킷관련 전송 정보를 파일로 출력 가능하다.
- ④ 여러 가지 트래픽 정보를 입력할 수 있다.
- ⑤ 시뮬레이션 과정이 사이클 단위로 측정되어지며, 소프트웨어에만 의존하기 때문에 하드웨어에 구애를 받지 않는다.

본 논문에서 제시된 라우팅 알고리즘을 시뮬레이션하기 위해 다음과 같은 속성을 가지도록 추가하였다.

- ① 라우팅 알고리즘은 wormhole 스위칭 기법을 사용하였으며, 멀티캐스팅 통신을 한다.
- ② 네트워크는 k-ary 2-cubes로 구성되어 있으며, 각 노드들은 양방향성 채널로 연결되어 있다.
- ③ 임의의 소스 노드에서 목적지 노드로 메시지 전송시 1 fault를 만족하는 다수의 결함 노드가 존재할 수 있으며, 이들 결함 노드는 무작위로 발생한다.
- ④ 입출력 프레임은 각각 20바이트로 설정하였다.
- ⑤ 메시지의 길이는 하나의 패킷에 대해 고정된 길이를 가지며, 하나의 패킷은 20개의 플릿으로 구성되어 있다.

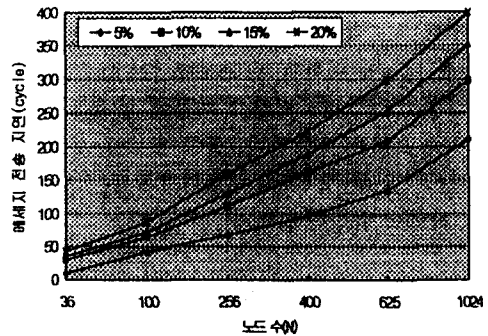
- ⑥ 각 노드에서 헤드 플릿을 처리하는데 소요되는 시간은 2사이클로 가정하였으며, 전송하는데 소요되는 시간은 1사이클로 가정하였다.
- ⑦ 소스 노드와 목적지 노드들은 무작위로 추출하였다.

4.2 시뮬레이션 결과

본 절에서는 본 논문에서 제시한 적응적 라우팅 알고리즘의 시뮬레이션 결과에 대하여 기술하고자 한다. 먼저 다양한 네트워크 환경하에서 목적지 노드들의 수를 변화시키면서 그에 따른 평균 메시지 지연 시간을 비교하였으며, 그에 따른 교통량에 대하여 시뮬레이션 하였다.

메시지 지연은 소스 노드의 큐에서 생성된 모든 플릿에 대하여 마지막 플릿이 목적지 노드로 전송될 때까지 소요되는 네트워크 사이클의 수로써 측정된다.

(그림 11)은 목적지 노드의 수를 5~20%로 변화시켰을 때의 메시지 전송지연에 대한 시뮬레이션 결과를 나타내고 있다. 그림에서 노드의 수에 비례하여 완만한 기울기를 나타내고 있다.

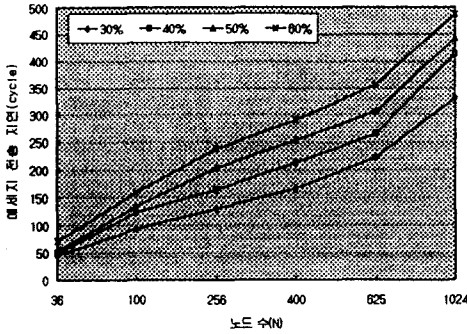


(그림 11) 목적지 노드 수가 전체 네트워크의 5~20% 일 때 메시지 전송 지연

(Fig. 11) Message delay when destination nodes are 5~20% of networks

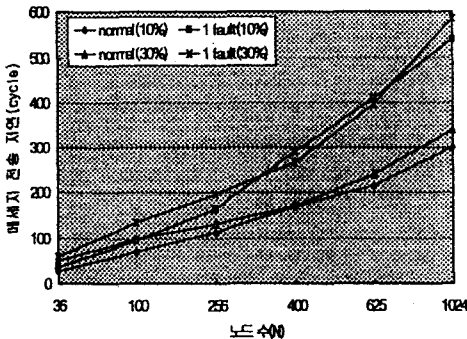
(그림 12)는 목적지 노드 수를 전체 네트워크의 30~60%로 증가시켰을 경우 메시지 지연 시간을 측정 한 것으로 목적지 노드 수를 60%까지 증가시켜도 전체 메시지 지연 시간은 전체적으로 완만한 기울기를 나타냄을 알 수 있다. 즉, 목적지 노드를 50% 이상 증가시켜도 전체 메시지 지연 시간은 본 논문에서 이

용한 오일러 경로에 의하여 네트워크 경쟁에 거의 영향을 받지 않음을 알 수 있다.



(그림 12) 목적지 노드 수가 전체 네트워크의 30~60%일 때 메시지 전송 지연
 (Fig. 12) Message delay when destination nodes are 30~60% of networks

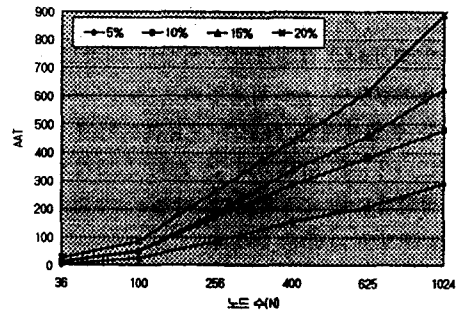
(그림 13)은 다양한 네트워크 환경에서 목적지 노드의 수가 전체 네트워크의 10~30%일 때, 정상적인 라우팅 수행 시간과 1개 노드의 결함을 고려한 라우팅 수행 시간을 비교하였다. 라우팅 수행시 결함이 발생한 노드에 대하여 본 논문에서는 다른 차수에 있는 인접노드로 라우팅하기 때문에 일시적인 디라우팅(derouting)을 수행한다. 결함 노드를 고려할 경우 본 논문의 k-ary 2-cubes 네트워크에서는 최소 9사이클의 지연이 발생한다.



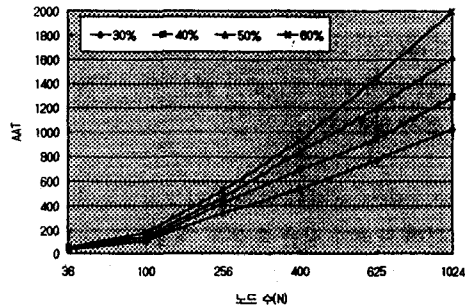
(그림 13) 목적지 노드가 10와 30% 일 때 정상적인 라우팅과 결함 노드를 고려한 라우팅 수행 결과
 (Fig. 13) Result of normal and fault routing when destination nodes are 10% and 30% of networks

다중 전송에 사용된 채널의 전체 수를 교통량으로 환산할 때, 평균 추가 교통량(AAT: Average Additional Traffic)은 아래와 같이 전체 교통량에서 목적지 노드 수를 제외한 교통량으로 정의[6]할 수 있다.

$$AAT = \frac{\text{다중 전송에 사용된 전체 채널수} - \text{목적지 노드의 수}}{\text{전체 교통량}}$$



(그림 14) 목적지 노드 수가 전체 네트워크의 5~20% 일 때 AAT
 (Fig. 14) AAT when destination nodes are 5~20% of networks



(그림 15) 목적지 노드 수가 전체 네트워크의 30~60%일 때 AAT
 (Fig. 15) AAT when destination nodes are 30~60% of networks

(그림 14)와 (그림 15)는 다양한 네트워크 환경에서 목적지 노드의 갯수 변화에 따른 각각의 AAT를 시물레이션하였다. (그림 14)와 (그림 15)는 목적지 노드의 수가 5~60%일 때, 라우팅 수행 결과에 대한 AAT

를 나타내고 있다.

(그림 14)와 (그림 15)에서 목적지 노드의 수가 증가할수록 AAT가 큰 폭으로 증가함을 알 수 있는데, 이것은 오일러 경로에 의해 채널에 대한 경쟁이 발생하지 않도록 중간 노드를 선택함으로써 추가적으로 요구되는 교통량 때문이다. 즉, 오일러 경로를 사용함으로써 교통량은 증가하더라도 채널에 대한 경쟁이 제거되어 전체 메시지 전송 시간은 줄일 수 있는 장점이 있다.

5. 결 론

본 논문은 k-ary n-cubes 시스템에서 다중 전송에 기반을 둔 라우팅 메카니즘에 대해서 연구하였다. 본 논문이 제시하는 라우팅 메카니즘은 최근 많은 연구가 진행중일 뿐만 아니라 확장성이 우수한 k-ary n-cubes 망을 모델로 하였으며, 메시지 전송 시간이 중간 노드의 길이에 영향을 거의 받지 않는 뮌홀 스위칭 기법을 적용하였다.

본 논문의 라우팅 메카니즘은 기존의 결정적 라우팅 알고리즘에서 고려할 수 없는 네트워크 지연과 노드의 고장 등을 고려하였는데, 네트워크 지연과 관련하여 채널에 대한 경쟁을 제거하기 위하여 오일러 경로 공식을 이용하였으며, k-ary n-cubes 망에서 차수 n 에 대하여 $n-1$ 의 결합허용을 보장하기 위해 하나의 물리적 채널당 $2n$ 개의 가상채널을 사용하였다. 이것은 Chien과 Kim[20]이 제시한 적응적 라우팅 알고리즘보다 높은 결합허용도를 보장함과 동시에 Linder와 Harden[5]이 제시한 적응적 라우팅 알고리즘보다 적은 가상채널을 사용함으로써 하드웨어 비용이 적게 들고 현실적으로 구현이 가능하다는 장점을 가지고 있다. 그리고 시뮬레이션을 통하여 본 논문이 제시한 적응적 라우팅 알고리즘에 대한 메시지 전송 지연과 평균 교통량을 측정하였다.

참 고 문 헌

- [1] P Merlin and P. J. Schweitzer, "Deadlock avoidance in Store-and-Forward Networks-I:Store-and-Forward Deadlock," IEEE Trans. on Parall. and Distri. Systems, vol. COM-28, no. 3, Mar 1980, pp. 345-354.
- [2] W. J. Dally and C. L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Inter-connection Networks," IEEE Trans. Comput., vol. C-36, no. 5, May 1987. pp. 547-553.
- [3] W. Welch, "Message-driven solver for Euler fluid dynamics equations," in Proc. 1988 ACM Southeast Regional Conf., Apr 1988.
- [4] W. J. Dally, "Performance Analysis of k-ary n-cube Interconnection Networks," Inter-connection Networks for High-Performance Parallel Comp.,1989, pp. 423-433.
- [5] D. H. Linder and J. C. Harden, "An Adaptive and Fault Tolerant Wormhole Routing Strategy for k-ary n-cubes," IEEE Trans. Comput., vol. 40, no. 1, Jan 1991. pp. 2-12.
- [6] X. Lin, P. K. Mckingley and L. M. Ni, "Deadlock-Free Multicast Wormhole Routing in 2D-Mesh Multicomputers," Tech. Report MSU-CPS-ACS-33, Feb. 5, 1991.
- [7] W. J. Dally, "Virtual-Channel Flow Control," IEEE Trans. on Parall. and Distri. Systems, vol. 3, no. 2, Mar 1992, pp. 194-205.
- [8] A. A. Chien and J. H. Kim, "Planar Adaptive Routing:Low-Cost Adaptive Networks for Multiprocessors," Proc. 19th Ann. Int'l Symp. Computer Architecture, May. 1992. pp. 268-277.
- [9] L. M. Ni, P. K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," IEEE Computer, Feb 1993, pp. 62-76.
- [10] W. J. Dally and H. Aoki, "Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels," IEEE Trans. on Parallel and Distri. Systems, vol. 4, no. 4, Apr 1993. pp. 466-475.
- [11] X. Lin, P. K. McKinley, and A. Esfahanian, "Adaptive Multicast Wormhole Routing in 2D Mesh Multicomputers," Proc. 1993 Parall. Archit. and Lang. Europe Conf., pp. 228-241.
- [12] X. Lin, A. Esfahanian, P. K. McKinley and A. Burago, "Adaptive Wormhole Routing in Hyper-

cube Multicomputer," Proceeding of the fifth IEEE Symp. on Parall. Distri. Proc., Dec 1993, pp. 72-79.

[13] X. Lin, P. K. McKinley, L. M. Ni, "Deadlock-Free Multicast Wormhole Routing in 2-D Mesh Multicomputers," IEEE Trans. on Parall. and Distri. Systems, vol. 5, no. 8, Aug 1994, pp. 793-804.

[14] S. Lee and J. Kim, "Path Selection for Communicating Tasks in a Wormhole-Routed Multicomputer," 1994 Int'l Conf. on Parallel Processing, pp. III-172~175.

[15] G. D. Pifarre, L. Gravano, G. Denicolay, and L. C. Sanz, "Adaptive Deadlock-and Livelock-Free Routing in the Hypercube Network," IEEE Trans. on Parall. and Distri. Systems, vol. 5, no. 11, Nov 1994. pp. 1121-1139.

[16] P. K. McKinley, H. Xu, A. Esfahanian, L. M. Ni, "Unicast-Based Multicast Communication in Wormhole-Routed Networks," IEEE Trans. on Parall. and Distri. Systems, vol. 5, no. 12, Dec 1994, pp. 1252-1265.

[17] R. Cypher and L. Gravano, "Storage-Efficient, Deadlock-Free Packet Routing Algorithms for Torus Networks," IEEE Trans. on Parall. and Distri. Systems, vol. 43, no. 12, Dec 1994, pp. 1376-1385.

[18] J. H. Park, H. S. Yoon and S. R. Maeng, "Performance Evaluation of Fully-Adaptive Minimal Routing in Wormhole-Routed Recursive-Circulant Networks," journal of KISS(A):Computer Systems and Theory, vol. 22, no. 11, Nov 1995, pp. 1638-1646.

[19] D. F. Robinson, P. K. McKinley and H. C. Cheng, "Optimal Multicast Communication in Wormhole-Routed Torus Networks," IEEE Trans. on Parall. and Distri. Systems, vol. 6, no. 10, Oct 1995, pp. 1029-1042.

[20] C. J. Glass and L. M. Ni, "Fault-Tolerant Wormhole Routing in Meshes without Virtual Channels," IEEE Trans. on Parall. and Distri. Systems, vol.

7, no. 6, Jun. 1996. pp. 620-636.



김 창 수

1984년 울산공과대학 전자계산학과(공학사)
 1986년 중앙대학교 대학원 전자계산학과(이학석사)
 1991년 중앙대학교 대학원 전자계산학과(공학박사)
 1992년~1996년 7월 부산수산대학교 전자계산학과 전임강사, 조교수
 1994년~1996년 국립수산진흥원 겸임연구원
 1996년 7월~현재 부경대학교 전자계산학과 조교수
 관심분야: 분산 및 병렬처리, 실시간 시스템, 결합하용 시스템, 인터넷 통신 응용 등



이 성 목

1995년 부경대학교 전자계산학과(이학사)
 1997년 부경대학교 대학원 전자계산학과(이학석사)
 1997년 1월~현재 (주)교보정보통신 S/W제품연구소 기술개발팀
 관심분야: 분산 및 병렬처리, 실시간 시스템, 통신시스템 응용 등