3차원 메쉬에 대한 완전 이진트리의 링크 충돌없는 임베딩

(Link-Disjoint Embedding of Complete Binary Trees in 3D-Meshes)

이 주 영 † 이 상 규 ††
(Ju-Young Lee) (Sang-Kyu Lee)

요 약 본 논문에서는 완전이진트리의 3차원 메쉬로의 임베딩 문제를 다룬다. 링크 충돌을 2까지 허용하면서 최적크기의 메쉬에 임베딩하는 방법은 [1]에서 다루고 있으며, [2]에서는 최적크기의 1.27배의 메쉬로 임베딩하는 방법을 보여주고 있다. 본 논문에서 제안하는 임베딩 방법은 순위차원 라우팅을 사용하며 링크 충돌이 없는 방법으로, 약 1.125배의 최적확장을 넘지 않는다. 이 임베딩 방법은 링크의 충돌 혹은, 임베딩의 확장을 최소화한다는 기준에서 볼 때 기존의 임베딩 방법에 비해 향상된 결과이다.

키워드 : 임베딩, 순위차원 라우팅, 메쉬 네트워크, 완전 이진트리, 링크충돌

Abstract In this paper, we consider the problem of embedding complete binary trees into 3-dimensional meshes. The method of embedding a complete binary tree into 3-dimensional mesh with the link congestion two is considered in [1], and the embedding in [2] shows that a complete binary tree can be embedded into a 3-dimensional mesh of expansion 1.27. The proposed embedding in this paper shows that a complete binary tree can be embedded into a 3-dimensional mesh of expansion approximately 1.125 with the link congestion one, using the dimensional ordered routing. Such method yields some improved features in terms of minimizing the link congestion or the expansion of embedding comparing to the previous results.

Key words : embedding, dimensional ordered routing, mesh interconnection networks, complete binary tree, link congestion

1. 서 론

병렬 알고리즘을 디자인하거나 실행할 때, 프로그램의 총 작업량(workload)은 동시에 병렬로 실행될 수 있는 작은 작업(task)으로 분할되어지며, 그 각각의 작업은 시스템의 처리 요소(processing element)들 간에 분산되어 진다. 이러한 작업들은 어떤 순차적인 제약조건하에 병렬 실행되어지고 실행 도중 작업들 사이에 서로 통신이 일어나게 되는데, 이런 통신에 요구되는 총 시간을 대기시간(latency)이라 부른다. 어떤 연산에서는 대기시간이 총 업무 완료 시간의 큰 장애(bottleneck)가 되기도 한다. 이러한 대기시간은 시스템의 상호연결망 형태와 적용 알고리즘의 통신 구조에 따라 좌우된다. 그러므로, 병렬 알고리즘의 통신 성능은 전적으로 시스템에 의존적이다.

병렬 컴퓨터 시스템에서 병렬 프로그램은 수행하는 가장 좋은 방법은 알고리즘의 통신 구조와 시스템의 상호연결망 사이의 정확한 일치(matching)를 찾는 일이다. 하지만, 시스템을 다른 상호연결망으로 갱신할 때마다 현재의 적용 업무에 의존적인 각 알고리즘들을 다시 개발하여야 하므로 실용적이지 못할 뿐더러 경제적이지 못하다. 시스템의 상호연결망은 유연성있는 적용업무에 의존적인 가상접속형태(virtual topology)를 지원해야 하며, 임베딩(embedding)은 시스템에 그러한 유연성을 제공한다.

임베딩 문제는 여러 가지 측면에서 연구되고 있는데,

· 본 연구는 한국과학재단 목적기초연구(과제번호 R06-2002-003-01002-0) 지원으로 수행되었음
† 종신회원 : 덕성여자대학교 전산학과 교수
jylee@namhae.duksung.ac.kr
†† 종신회원 : 숙명여자대학교 컴퓨터과학과 교수
sanglee@sookmyung.ac.kr
논문접수 : 2002년 2월 14일
심사완료 : 2003년 6월 30일

Monien과 Sudborough의 논문[3]은 임베딩에 관한 많은 결과들을 요약하고 있다. 완전이진트리의 2차원 메쉬로의 엠베딩은 [4, 5, 6], 완전이진트리의 3차원 메쉬로의 엠베딩은 [1, 2, 7]에서 보여주고 있다. 하지만, [4]에서는 3차원 메쉬 중 특별한 형태인 하이퍼큐브인 경우의 임베딩 방법을 보여주고 있다. 또한, [1]에서의 방법은 충돌을 2까지 허용하면서 최적 크기의 메쉬에 임베딩하는 방법이며, [2]에서는 링크의 충돌이 없으면서 최적크기의 1.27배의 메쉬에 임베딩하는 방법을 소개하고 있다. 본 논문에서는 링크 충돌이 없이, 순위차원 라우팅을 사용하면서 1.125배의 최적확장을 넘지 않는 임베딩 방법을 소개하고자 한다.

게스트 그래프를 $G=(V_G, E_G)$라 두면, V_G는 노드(node) 집합으로 작업을 나타내며, 에지(edge)집합 E_G는 프로그램에서 작업 대 작업 통신 링크(link)를 나타낸다. 호스트 그래프 $H=(V_H, E_H)$에서 노드집합 V_H는 프로세서(processor)를 나타내고 에지집합 E_H는 시스템에서 프로세서 대 프로세서 통신 링크를 나타낸다. 게스트 그래프 G의 호스트 그래프 H로의 임베딩이란 G의 각 에지(즉, 링크)를 H에 있는 경로로 1대1 매핑(one-to-one mapping)시키는 것을 의미한다. 먼저, 임베딩 문제에서 사용되는 몇 가지 용어들을 살펴보자. 다이레이션(dilation)이란 게스트 그래프 G에서 인접한 두 노드에 대하여, 호스트 그래프 H에서 이들과 대응되는 두 노드간의 최대거리(즉, 최단 경로들의 길이 중 가장 큰 값)를 말하며, 확장(expansion)이란 호스트 그래프의 노드 수와 게스트 그래프의 노드 수의 비율(즉, $|V_H|/|V_G|$)을 말한다. 매핑에 있어서 링크 밀집도(link congestion)란 게스트 그래프 G의 에지(링크)를 호스트 그래프 H에 있는 링크들(경로)로 메핑시킬 때 중복으로 메핑되어진 링크의 최대 수를 말한다. 이 때 밀집도가 2 이상인 경우를 링크 충돌(link contention)이라 한다.

게스트 그래프 G의 호스트 그래프 H로의 매핑은 다양하게 존재한다. 임베딩의 효율성은 이러한 매핑에 의해 좌우된다. 이와 같이, 임베딩은 다른 통신 구조에 적합하게 작성된 각 병렬 알고리즘이 기존의 알고리즘을 수정하지 않고도 호스트 시스템에 시뮬레이트(simulate)될 수 있도록 해준다. 새 시스템에 의해 야기되는 통신 지연 시간의 최소치를 유지시키기 위해서는 원시 통신 구조에 대응하는 통신 경로와 작업 할당을 제공하는 임베딩이 있어야 한다.

종전의 임베딩 연구는 대부분 축적교환(store and forward) 통신 방식[7, 8]을 기반으로 한 임베딩으로, 가능한 한 다이레이션을 줄이는데 중점을 두었다. 최근 개발된 스위칭 테크닉인 회로교환스위칭(circuit-switching)과 웜홀 라우팅(wormhole routing) 방식은 다중 컴퓨터의 통신시간을 현저히 감소시키며, 통신 링크 충돌이 없는 경우, 즉, 링크 충돌이 1인 경우에는 어떤 두 노드 사이의 경로 길이가 통신 시간과는 거의 무관하다는 특성을 갖고 있다[9]. 그러므로, 이러한 테크닉을 사용하는 다중 컴퓨터 시스템의 임베딩 문제에서는 다이레이션보다 통신 링크 충돌이 더 중요한 요인이 된다.

이 논문에서는 웜홀 라우팅 방식을 적용하는 임베딩 문제를 다루는데, 링크 충돌을 최소화하는데 그 목적을 두며, 특히 완전 이진트리의 3차원 메쉬로의 임베딩 문제를 다룬다. 트리에서 루트(root)노드의 레벨(level)을 1이라 가정할 때, 트리의 높이(height)란 각 노드들의 레벨들 중 가장 큰 값으로 정의한다. 높이가 p인 완전 이진트리를 T_p라 표기하고, T_p에 있는 총 노드의 개수는 2^p-1개임을 알 수 있다. 1차원에 l_1개의 노드를, 2차원에 l_2개의 노드를, ..., d차원에 l_d 개의 노드를 가진 d-차원 메쉬를 $M(l_1, l_2, \cdots, l_d)$로 표기한다. 예를 들면, 1차원, 2차원, 3차원에 각각 두 개의 노드를 가진 3-차원 메쉬는 $M(2,2,2)$로 표기하며 큐브(cube) 형태임을 알 수 있다. 메쉬 상호연결망에서 두 노드 사이의 경로는 1개 이상 존재할 수 있으며, 경로의 선택은 어떤 라우팅 방법을 사용하느냐에 따라 달라진다. 순위차원라우팅(dimension-ordered routing)이란 메시지가 라우트되는 차원에 순서를 두어 한 번에 어떤 하나의 차원으로 라우트된 후, 적합한 주소(좌표)계산에 의해 다음 차원으로 라우트되는 방법이다. 예를 들면, 3차원 메쉬에서 노드 좌표(2, 2, 2)에서 (3, 1, 4)로의 순위차원라우팅 방법은 먼저 1차원 링크를 사용해서 (2, 2, 2)→(3, 2, 2)로 라우트된 후, 2차원 링크를 사용해서 (3, 2, 2)→(3, 1, 2)로 라우트되고, 3차원 링크를 사용하여 (3, 1, 2)→(3, 1, 3)→(3, 1, 4)로 라우트되는 경로를 말한다.

그림 1은 높이가 7인 완전 이진트리 T_7 을 이차원 메쉬 $M(15,15)$로 링크충돌없이 임베딩한 예로써, VLSI 디자인에서 잘 알려진 H-트리를 사용한 것이다. 트리에서 인접한 두 노드는 메쉬에서 같은 차원의 링크로 연결되어 있다는 것을 볼 수 있다. 즉, 순위차원라우팅을 사용한 임베딩이다.

본 논문의 다음 절에서는 완전이진트리 T_p를 링크 충돌이 없이, 순위차원 라우팅을 사용하면서 1.125배의 최적확장을 넘지 않는 임베딩 방법을 소개한다. 여기서, 높이가 k인 완전이진트리의 노드개수는 2^k-1이므로,

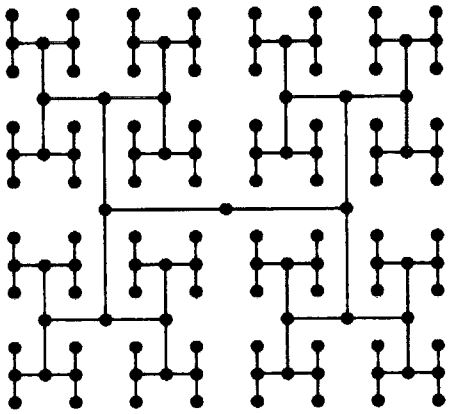

그림 1 T_7의 H 트리 배치

최적크기의 메쉬에 임베딩한다는 것(최적확장)은 노드개수가 2^k인 메쉬에 임베딩한다는 것이며, 최적확장의 1.125배라 함은 노드개수가 1.125×2^k인 메쉬에 임베딩한다는 의미이다. 마지막으로 3절에서 결론을 맺는다.

2. 임베딩 알고리즘

이 장에서는 재귀적 알고리즘을 이용하여 높이가 p인 완전 이진트리, T_p를 웜홀라우팅을 지원하는 3차원 메쉬에 통신링크의 충돌 없이 임베딩하는 방법을 소개하겠다. 본 논문에서는 2^p-1개의 노드를 갖는 높이가 p인 완전 이진트리 T_p의 노드들과 변들을 총 노드수가 $\frac{9}{8}\cdot 2^p$개를 넘지 않는 3차원 메쉬상의 노드들과 경로(path)로 지정해주는 임베딩을 E_p라 부르기로 한다.

정리 1. 높이가 p인 완전 이진트리 T_p를 순위차원 라우팅(dimension ordered routing) 방법을 사용하여 모든 통신링크의 충돌 없이 노드의 개수가 T_p의 총 노드 개수의 1.125배를 넘지 않는 3차원 메쉬에 임베딩할 수 있다. 이때 3차원 메쉬는 한 임의의 양의 정수를 k라 할 때

(1) $p\geq 9$인 경우,

p가 $3k$ 이면 $M(9\cdot 2^{k-3},\ 8\cdot 2^{k-3},\ 8\cdot 2^{k-3})$,

p가 $3k+1$이면 $M(9\cdot 2^{k-2},\ 8\cdot 2^{k-3},\ 8\cdot 2^{k-3})$,

p가 $3k+2$이면 $M(9\cdot 2^{k-2},\ 8\cdot 2^{k-2},\ 8\cdot 2^{k-3})$,

(2) $p<9$인 경우[1],

p가 $3k$ 이면 $M(2^k,\ 2^k,\ 2^k)$,

p가 $3k+1$이면 $M(2^{k+1},\ 2^k,\ 2^k)$,

p가 $3k+2$이면 $M(2^{k+1},\ 2^{k+1},\ 2^k)$이 된다.

증명: 먼저 (1)에서 $p=3k$인 경우를 증명하고 이를 기본으로 나머지 부분의 증명을 보이도록 하겠다. 이 증명에서는 임의의 $k\geq 3$에 대하여 재귀적 생성방법을 이용하여 $p=3k$일 때 T_p의 $1.125\cdot 2^p$개의 노드를 갖는 3차원 메쉬 $M(l_1,l_2,l_3)$으로의 임베딩 E_p가 항상 존재할 수 있음을 보이겠다. E_p 임베딩에서 루트 노드(root node)는 r로 표시되고 임베딩 후에도 트리의 노드가 지정(embed)되지 않은 비어있는 메쉬의 노드 (empty node) 중 재귀적 생성에 필요한 세 개의 노드를 e로 표시한다.

본 논문이 제안하는 임베딩 E_p는 다음과 같은 특성을 갖는다.

(특성 1) 루트 노드는 메쉬의 안쪽에 위치한 노드 (6개의 이웃노드를 갖는 노드) 중 하나에 지정된다. 이때 r의 주소를 (r_1,r_2,r_3) 이라 하면 $(r_1,r_2,1)$과 $(l_1,r_2,1)$ 그리고 $(l_1,l_2,1)$ 노드들은 e 노드들이 된다.

(특성 2) 다음의 링크들은 임베딩 후에도 비어있다. (x는 don't care)

- (r_1,r_2,x) 노드들을 지나는 3차원 링크
- (l_1,l_2,x) 노드들을 지나는 3차원 링크
- $(l_1,x,1)$ 노드들을 지나는 2차원 링크
- $(x,r_2,1)$ 노드들을 지나는 1차원 링크
- $(x,l_2,1)$ 노드들을 지나는 1차원 링크

(특성 3) 어느 링크를 보아도 충돌이 없다.

(특성 4) 모든 경로 연결은 순위 차원 라우팅을 따른다.

위의 특성을 만족하는 임베딩 E_p의 예가 그림 2에 도식적으로 나타나 있다. 여기서 점선으로 표시된 선들은 비어 있는 링크를 의미한다.

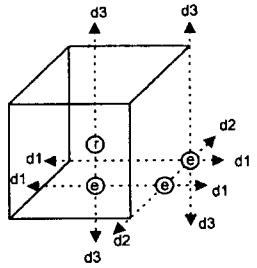

그림 2 임베딩 E_p

높이가 p ($p\geq 4$)인 T_p를 그림 3의 트리의 형태로 생각할 수 있다.

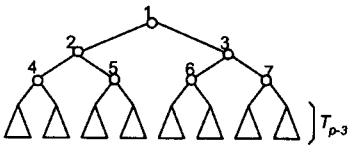

그림 3 완전 이진트리 T_P

여기서, $p=3k\geq 9$인 경우의 재귀적 알고리즘의 기본 임베딩은 그림 4와 그림 5에 나타나있다. 이는 T_9의 $M(9,8,8)$으로의 임베딩을 보여준다. 그림 3의 T_9에서 상위 세 레벨에 있는 노드들(즉, 1에서 7까지의 번호로 표시된 노드들)의 임베딩은 그림 4에 나타나 있다. 이때 r로 표기된 노드들은 여덟 개의 높이가 6인 서브트리의 루트를 의미한다.

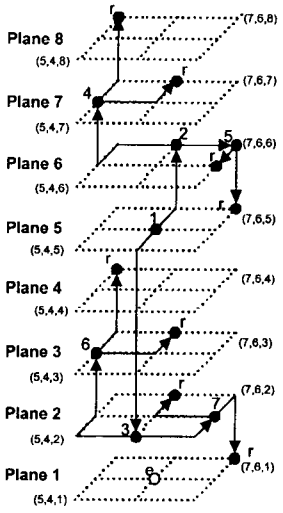

그림 4 상위 3 레벨 임베딩

여덟 개의 T_6짜리 서브트리는 2차원 평면 메쉬를 나타내는 Plane 1, Plane 2, … , Plane 8에 임베딩되는데 Plane 8과 Plane 4의 임베딩은 그림 5(a)에, Plane 7과 Plane 3의 임베딩은 그림 5(b)에, Plane 6의 임베딩은 그림 5(c)에, Plane 2의 임베딩은 그림 5(e)에, Plane 5와 Plane 1의 임베딩은 그림 5(d)에 나타나 있다. 각각의 통신링크들은 화살표로 나타내져 있고 임베딩에 쓰이지 않은 링크들은 표시하지 않았다. 링크 중 45도 좌로 기울어진 것( )들은 화살표의 시작부분에 연결된 노드와 바로 위 평면의 같은 위치에 있는 노드로의 연결을 의미하며, 이 때 두 plane 사이를 연결하는 링크인 3차원 링크를 사용한다. 그림 5(d)에서 타원으로 표시된 부분은 그 안의 모든 링크가 임베딩에 사용되지 않고 비어

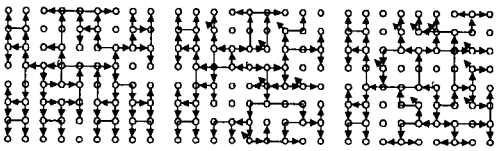

그림 5 하위 6 레벨 임베딩

있음을 나타낸다. 그림 4와 그림 5를 위에서 설명한데로 연계하여 구성해보면 위에서 언급한 4가지 특성을 모두 만족하는 E_9 임베딩이 되는 것을 쉽게 볼 수 있다.

그림 6과 그림 7은 재귀적인 방법을 사용하여 주어진 트리보다 3 레벨 큰 트리의 임베딩 방법을 나타내고 있다. 그림 3에 있는 트리 T_p는 일곱 개의 상위의 3 레벨에 있는 노드 1 - 7과 여덟 개의 T_{p-3} 서브트리의 구성으로 생각할 수 있다. 각각의 여덟 개의 서브트리가 E_{p-3}를 사용하여 그림 6과 같이 임베딩되어 있다고 하면 E_{p-3}의 비어있는 노드와 사용안한 링크들만을 이용하여 상위의 세 개 레벨에 있는 노드 1 - 노드 7 트리를 그림 7과 같이 임베딩하여 E_p를 만들 수 있다. E_9을 기본 임베딩으로 사용한다면 이와 같은 재귀적 생성 방법을 사용하여 $p=3k$인 임의의 T_p를 임베딩할 수

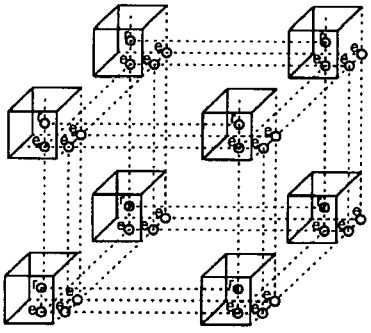

그림 6 서브메쉬 배치도

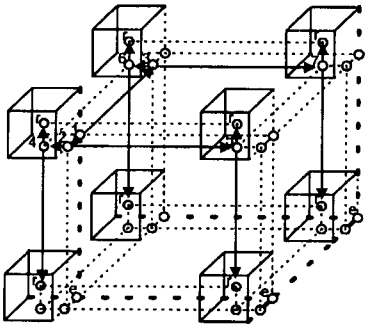

그림 7 재귀적 임베딩

있는 E_p를 찾을 수 있다. 그림 6은 여덟 개의 E_{p-3}의 배치를 보여주고 있다. 그림 6에서 여덟 개의 M $(9\cdot 2^{k-4},\ 8\cdot 2^{k-4},\ 8\cdot 2^{k-4})$ 서브메쉬 안에 r로 표시된 노드는 T_{p-3}의 루트를 말하며 e로 표시된 노드들은 앞에서 언급했던 재귀적 임베딩에 사용될 비어있는 노드를 말한다. 점선으로 표시된 선들은 비어 있는 링크를 의미하며 그림 3에서 상위 세 레벨의 일곱 개의 노드와 14개의 링크들의 임베딩은 24개의 e 노드들과 점선으로 표시된 링크들을 이용하는데 이는 그림 7에 나타나 있다. 그림 7를 보면 여덟 개의 E_{p-3}를 이용해 새로 생성된 E_p 또한 4가지의 특성을 모두 만족하고 있는 임베딩이라는 것을 볼 수 있다. 따라서 앞에서 보여준 E_9을 기본 임베딩으로 하고 이와 같은 재귀적 방법을 사용한다면 높이가 $p=3k\ \geq 9$를 만족하는 모든 T_p의 M$(9\cdot 2^{k-3},\ 8\cdot 2^{k-3},\ 8\cdot 2^{k-3})$로의 임베딩이 항상 가능함을 볼 수 있다.

여기서 임베딩의 확장성을 보자. 메쉬의 총 노드 개수는 $9\cdot 2^{k-3}\times 8\cdot 2^{k-3}\times 8\cdot 2^{k-3}$이므로 이를 계산하면 $9\cdot 2^{3k-3}$이 된다. 그리고, $p=3k$이므로 확장성은 $\frac{9\cdot 2^{3k-3}}{2^{3k}-1}\simeq\frac{9}{8}=1.125$가 된다. 다음은 나머지 두 경우인 트리의 높이가 $p=3k+1\geq 9$인 경우와 $p=3k+2\geq 9$인 경우를 증명하는데 그 기본적인 방법은 $p=3k\ \geq 9$인 경우와 같다. $p=3k+1\geq 9$인 경우의 재귀적 생성방법은 $p=3k\ \geq 9$의 경우에 보여준 재귀적 방법에 기본 임베딩을 E_9 대신 E_{10}을 사용하고 $p=3k+2\ \geq 9$인 경우에는 기본 임베딩을 E_{11}을 사용하면 되는데 이 두개의 기본 임베딩인 E_{10}과 E_{11}은 E_9을 이용하여 간단히 만들 수 있다. E_9을 자세히 보면 그림 8(a)의 형태를 하고 있음을 알 수 있다. 이는 Plane 5와 Plane 1에 그림 5(d) 임베딩을 똑같이 사용했기 때문임을 알 수 있다. $p=3k+1\ \geq 9$ 인 경우와 $p=3k+2\ \geq 9$ 인 경우의 기본 임베딩인 E_{10}과 E_{11}은 그림 8(b)와 그림 8(c)에 나타나 있다. 기본 임베딩 E_9 두 개를 그림 8(b)에 나타난 것처럼 배치하고 e 노드들과 비어있던 링크를 사용해 연결하면 앞에서 언급한 4가지 특성을 만족하는 E_{10}이 됨을 볼 수 있다. E_{11}의 경우도 같은 방법으로 E_9 4개를 연결하여 만들어지는 것이 그림 8(c)에 설명되어 있다. E_{10}과 E_{11}은 모두 임베딩 조건 4가지를 만족하므로 그림 6과 그림 7의 재귀적 방법의 여덟 개의 서브트리를 위한 기본 임베딩으로 사용한다면 높이가

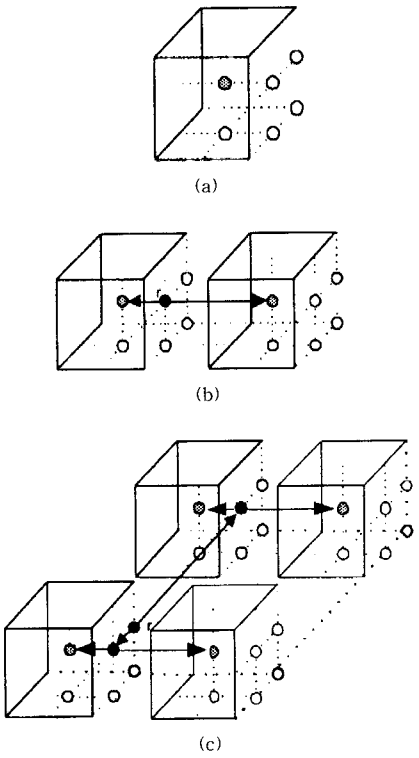

그림 8 기본 임베딩들

$p=3k+1\geq 9$를 만족하는 T_p의 $M(9\cdot 2^{k-2},8\cdot 2^{k-3},\ 8\cdot 2^{k-3})$로의 임베딩이, $p=3k+2\ \geq 9$를 만족하는 T_p의 M$(9\cdot 2^{k-2},\ 8\cdot 2^{k-2},\ 8\cdot 2^{k-3})$로의 임베딩이 항상 가능하다는 것을 볼 수 있다.

이로써 정리 1의 (1) $p\geq 9$인 경우에 대한 증명을 보였다. $p<9$인 (2)의 경우는 [4]의 그림 4-6에 나와 있는 임베딩을 사용하면 된다. 이 방법은 edge-disjoint하며, 확장이 $\frac{2^{3k}}{2^{3k}-1}\ \simeq 1$인 임베딩이다. 따라서, 임의의 높이 p를 가진 완전 이진트리에 대해 정리 1의 조건을 만족하는 임베딩이 항상 존재함을 알 수 있다. ■

3. 결 론

본 논문에서는 간단한 재귀적 알고리즘을 이용하여 높이가 p인 완전 이진트리, T_p를 웜홀 라우팅을 지원하는 3차원 메쉬에 통신링크의 충돌없이 임베딩하는 방법을 제안했다. 이때 3차원 메쉬의 노드 확장 정도의 범위를 최적의 1.125배를 넘지 않음을 보장한다. 이는 완전 이진트리의 통신형태를 갖는 분할 정복방식의 알고

리즘을 병렬 컴퓨터에서 실행시킬 때 작업들의 프로세서 분배에 직접 응용될 수 있는 방법으로 기존의 방법보다 간단하면서도 보다 최적에 접근시키는 결과라 하겠다. 따라서 많은 병렬처리 프로그램에 유용하게 사용될 수 있으리라 생각된다.

참 고 문 헌

[1] S. K Lee and H. A. Choi, "Link Disjoint Embedding of Complete Binary Trees in Meshes", Networks Journal, Vol. 30, No. 4, pp 283 292, 1997.

[2] S. M. Park, S. K. Lee and B. H. Moon, "Link Disjoint Embedding of Complete Binary Trees into 3D Meshes with Dimension Ordered Routing", 한국정보과학회 병렬처리연구회 1998년 춘계 학술발표회, pp. 85 96, 1998.

[3] B. Monien and H. Sudborough, "Embedding One Interconnection Network in Another,'' preprint, 1992.

[4] A. Gibbons and M. Patterson, "Dense Edge Disjoint Embedding of Binary Trees in the Mesh", Proc. of 4th Annual ACM Symposium on Parallel Algorithms and Architecture, pp. 257 263, 1992.

[5] U. K Lee and H. A. Choi, "Embedding of Complete Binary Trees into Meshes with Row Column Routing", IEEE Transaction on Parallel and Distributed Systems, Vol. 7, No. 5, pp.493-497, 1996, May.

[6] J. D. Ullman, "Computational Aspects of VLSI", Computer Science Press, 11, Taft Court, Rockville, Maryland 20850, U.S.A. 1984.

[7] J. E. Brandenburg and D. S. Scott, "Embeddings of Communication Trees and Grids into Hypercubes," Technical Report, Intel Scientific Computers, 1985.

[8] S. N. Bhatt and S. S. Cosmadakis, "The Complexity of Minimizing Wire Lengths for VLSI Layouts," Information Processing Letters, Vol. 25, pp 263 267, 1987.

[9] L. M. Ni and P. K. McKinley, "A Survey of Routing Techniques in Wormhole Networks," IEEE Computers, pp. 62 76, Feb. 1993

이 주 영

1984년 이화여자대학교 수학과 졸(학사) 1991년 The George Washington Unvi. Dept. of Electrical Engineering and Computer Science(공학석사). 1996년 The George Washington Univ. Dept. of Electrical Engineering and Computer Science(공학박사). 1996년~현재 덕성여자대학교 전산학과 조교수. 관심분야는 알고리즘, 병렬분산처리, 그래프 이론, 무선 통신

이 상 규

1989년 University of Southern California Dept. of Computer Science(공학사). 1991년 George Washington University Dept. of Electrical Engineering and Computer Science. 1995년 George Washington Univ. Dept. of Electrical Engineering and Computer Science(공학박사). 1995년~1996년 George Washington Univ. Dept of Electrical Engineering & Computer Science 박사후과정. 1997~현재 숙명여자대학교 컴퓨터과학과 부교수. 관심분야는 무선 통신, 모바일 컴퓨팅, 병렬/분산처리, 고성능 컴퓨팅 및 통신