

# 정이진트리, 하이퍼큐브 및 하이퍼피터슨 네트워크를 피터슨-토러스(PT) 네트워크에 임베딩

(Embedding Complete binary trees, Hypercube and Hyperpetersen Networks into Petersen-Torus(PT) Networks)

서정현<sup>†</sup> 이형옥<sup>\*\*</sup> 장문석<sup>\*\*\*</sup>  
(Junghyun Seo) (Hyeongok Lee) (Moonsuk Jang)

**요약** 본 논문에서, 정이진트리와 노드수가 증가함에 따라 분지수가 증가하는 하이퍼큐브, 하이퍼피터슨 연결망을 분지수가 고정된 피터슨-토러스(PT) 연결망에 일대일 임베딩 하였다. 일대일 임베딩은 일대다 임베딩이나 다대일 임베딩과 비교하여 프로세서의 낭비나 프로세서의 과부하의 위험이 적다. 하이퍼큐브나 하이퍼피터슨에서 개발된 알고리즘들이 피터슨-토러스에서 사용될 때 유휴 프로세서가 많지 않도록 확장율  $\approx 1$ , 연장을  $1.5n+2$  그리고 밀집율  $O(n)$ 에 일대일 임베딩 하였고, 메시지 전송길이에 영향을 받지 않는 워홀 라우팅 방식의 시스템에서 메시지의 병목현상을 피할 수 있도록 정이진트리를 PT 네트워크에 밀집율 1, 확장율  $\approx 5$  그리고 연장을  $O(n)$ 에 일대일 임베딩 하였다.

**키워드** : 피터슨-토러스, 임베딩, 하이퍼큐브, 하이퍼피터슨, 정이진트리

**Abstract** In this paper, the hypercube, hyperpetersen networks, whose degree is increasing in accordance with expansion of number of node and complete binary tree are one-to-one embedded into peterson-torus(PT) network which has fixed degree. The one-to-one embedding has less risk of overload or idle for the processor comparative to one-to-many and many-to-one embedding. For the algorithms which were developed on hypercube or hyperpetersen are used for PT network, it is one-to-one embedded at expansion  $\approx 1$ , dilation  $1.5n+2$  and link congestion  $O(n)$  not to generate large numbers of idle processor. The complete binary tree is embedded into PT network with link congestion =1, expansion  $\approx 5$  and dilation  $O(n)$  to avoid the bottleneck at the wormhole routing system which is not affected by the path length

**Key words** : petersen-torus, embedding, hypercube, hyperpetersen, complete binary tree

## 1. 서론

멀티컴퓨팅을 위한 상호연결망은 크게 동적(dynamic)

연결망과 정적(static)연결망으로 나눈다. 정적연결망은 두 프로세서 사이의 통신 링크가 수동적이므로 다른 프로세서와 다시 연결될 수 없으며, 동적연결망은 스위칭 요소에 의해 통신 링크가 다른 프로세서와 연결될 수 있다. 정적 연결망은 망을 구성하는 노드와 에지의 구성에 따라 트리 부류, 메쉬 부류, 하이퍼큐브 부류 그리고 스타그래프 부류로 나눌 수 있다. 멀티 컴퓨팅을 효과적으로 이용하기 위해서 링, 트리, 메쉬, 하이퍼큐브와 같은 공통적인 위상 구조를 토러스[1], 허니콤(honeycomb) 메쉬[2], 다이아고널(diagonal) 메쉬[1], 헥사고널(hexagonal) 메쉬[3], 하이퍼큐브[4], folded 하이퍼큐브[5], multiply-twisted-cube[6], 재귀원형군[7], 스타 그래프[8], 매크로-스타(macro-star)[9], 전위(transposition) 그래프[10], 행렬-스타(matrix-star) 그래프[11] 등의 여러 가지 연결망에 임베딩하는 문제가 연구되었다.

<sup>†</sup> 학생회원 : 순천대학교 컴퓨터공학과  
jhseo@sunchon.ac.kr

<sup>\*\*</sup> 정회원 : 순천대학교 컴퓨터교육과 교수  
oklee@sunchon.ac.kr  
(Corresponding author임)

<sup>\*\*\*</sup> 종신회원 : 순천대학교 컴퓨터공학과 교수  
jang@sunchon.ac.kr

논문접수 : 2008년 2월 27일  
심사완료 : 2008년 5월 29일

Copyright©2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 시스템 및 이론 제35권 제8호(2008.8)

새로운 상호연결망이 설계되면 연결망의 구조에 적합한 병렬 알고리즘들이 개발되는데 서로 다른 연결망간의 임베딩은 다른 연결망에서 개발된 알고리즘들을 재사용 할 수 있게 하고, 임베딩에 관여되는 두 개의 연결망의 구조에 따라 임베딩 알고리즘 작성의 난이도가 결정된다. 이진트리, 분할-정복 알고리즘, backtrack searching, branch-and-bound computation, game-tree evaluation, functional and logical programming 그리고 다양한 수 연산에 사용되므로 다른 연결망으로 임베딩하는 연구가 활발히 진행되었다. 링이나 이진트리를 메쉬, 허니콤브 메쉬, 하이퍼큐브, 스타그래프, 재귀 원형군, 하이퍼피터슨[12], 사이클릭 피터슨[13], 폴드 피터슨[14] 상호연결망에 임베딩하는 연구[12-20]가 진행되었고, 랜덤 크기의 이진트리를 정적(static)연결망에 임베딩하는 연구[21,22]도 있다. 상호연결망들 사이에 임베딩하는 연구가 있는데, 메쉬, 하이퍼큐브 그리고 스타그래프를 다른 상호연결망에 임베딩하는 연구[12,13,23]가 있고, 하이퍼큐브와 스타그래프 상호간의 임베딩에 대한 연구[24]가 있다.

정이진트리를 메쉬에 임베딩하는 연구는 다른 연결망의 임베딩에서와 마찬가지로 전통적으로 연장율을 줄이기 위해 노력해왔지만 뱀홀 라우팅 방식이 소개되고 난 뒤 밀집율을 줄이는 방향으로 연구되고 있다. [25]에서 H-트리를 이용하여 홀수 깊이의 정이진트리를 밀집율 1에 메쉬에 임베딩 하였다. [26]에서  $2^n-1$ 개의 노드를 가진 정이진트리  $T_n$ 을  $2^{\lfloor n/2 \rfloor + 1} \times (2^{\lfloor n/2 \rfloor} + 1)$ -메쉬에 밀집율 1에 임베딩하였고 [27]에서  $T_n$ 을  $2^{\lfloor n/2 \rfloor} \times (2^{\lfloor n/2 \rfloor})$ -메쉬에 밀집율 2에 임베딩하였다. [28]에서  $T_n$ 을  $2^{\lfloor n/2 \rfloor} \times (2^{\lfloor n/2 \rfloor})$ -토러스에 밀집율 1에 임베딩 하였다.

병렬처리의 다양한 알고리즘에서  $2^d$ 개의 노드를 가지는  $d$ 차원 하이퍼큐브는 시간복잡도  $d$ 를 가지는 간단한 해결책을 가지고 있다. 망의 확장성 측면에서 메쉬 부류는 바로 이웃한 프로세서를 연결하는 방법으로 적은수의 프로세서를 추가하여 확장하고 하이퍼큐브는 정확하게 같은 수의 노드를 추가하여 각각 프로세서들을 서로 연결하기 때문에 메쉬부류가 확장이 쉽다. 따라서 좋은 병렬 알고리즘을 가진 하이퍼큐브를 확장성이 용이하여 상용화가 쉬운 메쉬부류에 임베딩하는 것은 의미가 있다 [29,30]. 토러스는 분지수가 4이고 하이퍼큐브는 차원이 증가함에 따라 분지수가 증가하므로 5차원 이상의 하이퍼큐브(분지수가 4를 초과하는)는 토러스에 임베딩 할 때 밀집율과 연장율이 1을 초과하며 점점 증가한다[30]. 연구 [31]에서 6차원 하이퍼큐브를  $8 \times 8$  메쉬에 연장율 11, 밀집율 2 그리고 확장율 1에 임베딩 하였다. 연구 [29]에서는 11차원 하이퍼큐브를  $32 \times 64$  메쉬에 연장율 32 밀집율 57에 임베딩 하였고 크기가 좀 더 적은 하이

퍼큐브와 메쉬들을 임베딩 하였다. 이 두 연구는 제한된 크기의 하이퍼큐브를 메쉬에 임베딩한 결과로 임베딩 척도가 상수이다. 연구 [30]에서 이전 연구에서의 제한을 두지 않고 일반화 하였다.  $d$  차원 하이퍼큐브를 같은 노드수의 메쉬에 연장율  $2^d-2$ , 확장율 1에 임베딩 하였다.

다중 컴퓨팅에서 메시지를 전송하는 방법으로 회선 스위칭(circuit switching) 방식과 패킷 스위칭(packet switching) 방식이 있다. 회선 스위칭 방식은 메시지를 전송하고자 할 때 목적지까지 회선을 설정하여 메시지를 전송하는 동안 두 프로세서가 전용으로 사용하도록 한다. 패킷 스위칭 방식은 store-and-forward 방식, virtual cut-through 방식, 뱀홀(wormhole) 방식으로 나눌 수 있다. store-and-forward 방식은 패킷이 전송될 때 경로 상에 있는 중간 노드의 기억장치에 저장하였다가 다시 전송하는 방식으로 메시지 지연시간이 길고, 많은 기억장치를 필요로 한다. 뱀홀 방식은 하나의 패킷을 플릿(flit)이라는 작은 단위로 나누어 라우터의 지원을 받아 메시지의 가장 앞에 있는 헤더 플릿이 라우팅 할 경로를 결정하고, 나머지 플릿은 헤더 플릿의 뒤를 연속적으로 따르는 방식이다. 본 논문에서는 정이진트리를 뱀홀 방식에 적합하도록 피터슨-토러스 PT 네트워크에 임베딩한다.

상호연결망은 멀티컴퓨팅 시스템에서 프로세서간의 연결 구조를 제공하고 그 위에 작성된 병렬알고리즘의 설계에 중요한 역할을 한다. 상호연결망은 그래프  $G=(V,E)$ 로 모델링 될 수 있고, 상호연결망의 프로세서는 노드집합  $V(G)$ 로 표현되고 프로세서간의 통신링크는 에지집합  $E(G)$ 로 표현된다.  $|V(G)|$ 는 노드의 개수이다. 상호연결망  $H$ (host)에서  $G$ (guest)를 시뮬레이션 하기 위해,  $G$ 를  $H$ 에 효과적으로 임베딩  $f$  한다는 것은  $V(G)$ 를  $V(H)$ 에 효과적으로 사상하는 것이다. 임베딩을 평가하는 척도는 확장율(expansion), 부하계수(load factor), 연장율(dilation), 밀집율(congestion)이 있다[4].  $|V(H)|/|V(G)|$ 를  $f$ 의 확장율  $e$ 라고 한다.  $V(G)$ 를  $V(H)$ 에 사상할 때  $V(G)$ 의 하나의 노드가  $V(H)$ 의 하나의 노드에 사상되면 일대일 임베딩이라고 하고  $V(G)$ 의 하나의 노드가  $V(H)$ 의 여러 개의 노드에 사상되면 일대다 임베딩이라고 하고  $V(G)$ 의 여러 개의 노드가  $V(H)$ 의 하나의 노드에 사상되면 다대일 임베딩이라고 한다.  $V(G)$ 의 모든 노드가 임베딩에 참여해야 하므로 확장율  $e$ 가 1보다 적으면 다대일 임베딩만 가능하다. 다대일 임베딩에서 임베딩 함수  $f$ 에 의해서  $H$ 의 임의의 하나의 노드로 사상되는  $G$ 의 노드의 수의 최대값을  $f$ 의 부하계수  $l$  이라고 하고, 부하계수가 크면 하나의 프로세서에서 많은 작업이 이루어지므로 효율적이지 못하다. 확장율  $e$ 가 1보다 같거나 크면 일대일 임베딩이나 일대다 임베딩이

가능하다. 일대다 사상은 연장율과 프로세서의 작업량을 줄여주는 효과가 있지만 프로세서가 낭비될 위험이 있다. 본 논문에서는 프로세서 부하가 크거나 프로세서의 낭비 위험이 없는 일대일 임베딩 알고리즘을 제안한다.

본 논문에서는 정이진트리(complete binary tree), 하이퍼큐브, 하이퍼피터슨 네트워크를 PT 네트워크로 일대일 임베딩 한다. 일대일 임베딩에서는 부하계수가 1이므로 프로세서에서 작업처리 시간이 빠르다. 정이진트리는 밀집을 1에 임베딩 하였고, 이것은 라우팅 거리에 영향을 받는 store-and-forward 라우팅 방식보다 회선의 복잡도가 전송시간에 영향을 미치는 웜홀(worm hole) 라우팅방식에 더 적합하다. 하이퍼큐브와 하이퍼피터슨 네트워크는 프로세서 낭비를 없애기 위해서 확장율이 1에 근접하도록 임베딩 되었고, 망이 확장될 때 에지의 수가 증가하는 특성 때문에 연장율과 밀집율은 비례해서 증가한다. 본 논문의 구성은 2장에서 PT 네트워크, 하이퍼큐브, 하이퍼피터슨 네트워크를 소개하고, H-트리를 매쉬에서 생성하는 알고리즘을 소개한다. 3장에서 정이진트리, 하이퍼큐브, 하이퍼피터슨 네트워크를 PT 네트워크에 임베딩 하였다. 마지막으로 결론을 맺는다.

## 2. 관련연구

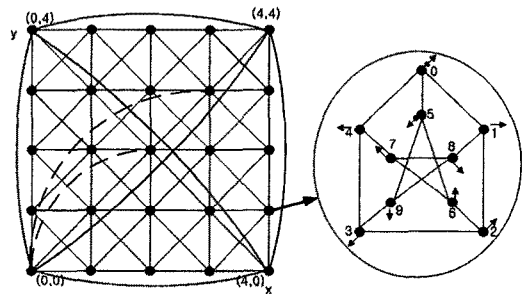
하이퍼큐브는 모든 노드의 분지수가 같은 정규 연결망이고, 노드-에지 대칭 구조를 가지고 있으며 intel Ncube, Paralex 등과 같은 병렬컴퓨팅 시스템에서 상용화되었다. 하이퍼피터슨 네트워크는 피터슨 그래프를 하이퍼큐브 방식으로 연결하여 하이퍼큐브보다 지름(diameter)과 망비용(network cost)에서 우수한 성질을 갖는 연결망이다. 이진트리는 검색 알고리즘, 정렬 알고리즘, 산술식 표현법, 디렉터리 구조 등의 다양한 문제를 위한 좋은 해결책을 얻기 위해 유용하다. 본 논문에서는 위의 하이퍼큐브, 하이퍼피터슨 그리고 정이진트리를 피터슨-토러스 PT에 임베딩 한다.

### 2.1 피터슨-토러스 $PT(m,n)$ 네트워크 정의

$PT(m,n)(m,n \geq 2)$ 은 피터슨 그래프(그림 1의 (b))를 기본모듈로 하고, 기본모듈  $m(x$ 축) $\times n(y$ 축)개를 격자점에 배치하고 에지 정의에 의해 연결한다. 피터슨-토러스 네트워크  $PT(m,n)=(V_{pt}, E_{pt})$ 이다. 본 논문에서는  $PT(m,n)$ 를 그림 1의 (a)와 같이 2차원 그래프에 사상시켜 설명한다. 단위 피터슨 그래프를 기본모듈이라고 하며, 기본모듈은  $x$ 와  $y$ 의 교차점에 있다. 기본모듈의 주소는  $(x,y)$ 로 나타내고 노드 주소는  $(x,y,p)$ 로 나타낸다. "노드  $(x,y,p)$ 는 기본모듈  $(x,y)$ 에 속한다"라고 말한다.  $x$ 는 기본모듈의  $x$ 축의 좌표,  $y$ 는 기본모듈의  $y$ 축의 좌표,  $p$ 는 기본모듈인 피터슨 그래프에 있는 노드의 주소이다. 피터슨-토러스  $PT(m,n)$ 의 노드정의는 다음과 같다.

$$V_{pt} = \{(x,y,p), 0 \leq x < m, 0 \leq y < n, 0 \leq p \leq 9\}$$

$PT(m,n)$ 의 에지는 다음과 같이 내부에지와 외부에지로 나눈다. 같은 기본모듈에 속한 노드들을 연결하는 에지를 내부에지라고 하고, 내부에지는 피터슨 그래프의 에지를 그대로 사용한다. 서로 다른 기본모듈에 있는 노드들을 연결하는 에지를 외부에지라고 하고 다음과 같이 정의한다. 아래 에지를 나타내는 수식에서 심벌 '/'는 나머지 연산자이다. ① 세로에지는  $((x,y,6), (x,(y+1)/n,9))$ 이다. ② 가로에지는  $((x,y,1), ((x+1)/m,y,4))$ 이다. ③ 사선에지는  $((x,y,2), ((x+1)/m,(y+1)/n,3))$ 이다. ④ 역 사선에지는  $((x,y,7), ((x-1+m)/m,(y+1)/n,8))$ 이다. ⑤ 지름에지는  $((x,y,0), ((x + \lfloor \frac{m}{2} \rfloor)/m, (y + \lfloor \frac{n}{2} \rfloor)/n,5))$ 과  $((x - \lfloor \frac{m}{2} \rfloor + m)/m, (y - \lfloor \frac{n}{2} \rfloor + n)/n,5))$ 이다.



(a)  $PT(5,5)$  (b) 피터슨 그래프

그림 1 피터슨-토러스  $PT(5,5)$

그림 1의 (a)는 피터슨-토러스  $PT(5,5)$ 에서 기본모듈을 격자점으로 표현하였다. 기본모듈은 이웃한 기본모듈들과 완전 그래프 형태로 연결되어 있다. 가장자리를 제외한 모든 기본모듈은 지름에지를 제외한 에지들이 그려져 있으며, 기본모듈 (0,0)의 지름에지만 짧은 파선(짧은 선이 일정한 규칙으로 반복되는)으로 그려져 있다. 가장자리의 기본모듈들은 랩어라운드(wraparound) 에지가 생략되어 있으나 4개의 꼭지점에 있는 기본모듈은 몇 개의 랩어라운드 에지가 짧은 실선으로 그려져 있다. 피터슨-토러스  $PT(m,n)$ 의 기본모듈인 피터슨 그래프(그림 1의 (b))를 보면, 노드 1, 4는 가로에지, 노드 6, 9는 세로에지, 노드 2, 3은 사선에지, 노드 7, 8은 역 사선에지, 노드 0, 5는 지름에지와 결합된 노드들이다. 노드  $U$ 로부터 노드  $V$ 까지 경로가 있다고 가정하면, 그 경로에 사용된 외부에지의 수는 외부경로길이라고 하고, 내부에지의 수는 내부경로길이라고 한다.  $PT(m,n)$ 은 노드수  $10mn$ , 에지 수  $20mn$ , 연결도 4 그리고 분지수 4를 가지는 정규 그래프이다. PT에서 단순 라우팅 알고리즘, 최적 라우팅 알고리즘, 해밀톤 사이클 알고리즘이 제안되었고,

지름, bisection width가 분석되었다.  $PT(m,n)$ 는  $m=n$ 일 경우 지름이  $n+4$ 이고, bisection width가  $n^2+5n$ 이다. PT는 같은 노드수를 가지는 분지수 상수인 토러스, 허니컴브 메쉬, 다이아고날 메쉬, 헥사고날 토러스 보다 망 비용과 bisection width가 우수하다[32].

**2.2 H-트리**

홀수 깊이의 정이진트리를 2차원 메쉬에 밀집을 1에 임베딩 하는 것은 잘 알려진 H-트리 생성을 이용해서 할 수 있다. H-트리를 만드는 자세한 설명은 [25]에 있고, H-트리를 만드는 방법을 간단하게 설명하면 다음과 같다. 깊이가  $k$ 인 정이진트리를  $T_k$ 라고 하고,  $n$ 행  $m$ 열 2차원 메쉬를  $n \times m$  메쉬라고 한다.  $k$ 가 4보다 큰 홀수일 때, H-트리  $T_k$ 는 2개의  $T_{k-1}$ 트리의 루트노드를 자식 노드로 하고 4개의  $T_{k-2}$ 트리의 루트 노드를 단말 노드로 한다. 4개의  $T_{k-2}$ 트리는 재귀적인 방법으로 남은 노드들을 연결하고, 루트 노드는 메쉬의 중앙 노드에 사상한다.  $7 \times 7$  메쉬에 H-트리  $T_5$ 를 생성하는 것을 그림 2에 보였고, 이것은  $T_5$ 를  $7 \times 7$  메쉬에 임베딩 한 것과 같다.

$T_k$ 는  $2^k-1$ 개의 노드를 가지고 있으며,  $(2^{\frac{k+1}{2}} - 1) \times 2^{\frac{k+1}{2}} - 1$  메쉬에 임베딩 된다.

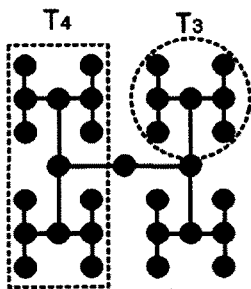


그림 2 H-트리  $T_5$

**2.3 하이퍼큐브**

$n$ -차원 하이퍼큐브  $Q_n$ 는  $2^n$ 개의 노드와  $n2^{n-1}$ 개의 에지로 구성된다. 각 노드의 주소는  $n$ -비트 이진수로 표현될 수 있고, 임의의 두 노드의 주소가 정확히 1 비트만 다를 때 그들 사이에 에지가 있다. 따라서 하이퍼큐브  $Q_n$ 는 분지수가  $n$ 인 정규 그래프이다. 하이퍼큐브  $Q_n$ 에서 위의 정의와 같이 주소를 붙이는 방법은  $n!2^n$ 이다 [4]. 하이퍼큐브  $Q_n$ 는 다음과 같은 단순한 재귀적 구조를 가지고 있으므로 확장성이 매우 좋다. 하이퍼큐브  $Q_n$ 은 그래프의 곱을 이용하여 다음과 같이 재귀적으로 정의될 수 있다.

$$Q_1=K_2,$$

$$Q_n=K_2 \times Q_{n-1} \text{ (단, } K_2 \text{는 노드가 두 개인 완전 그래프)}$$

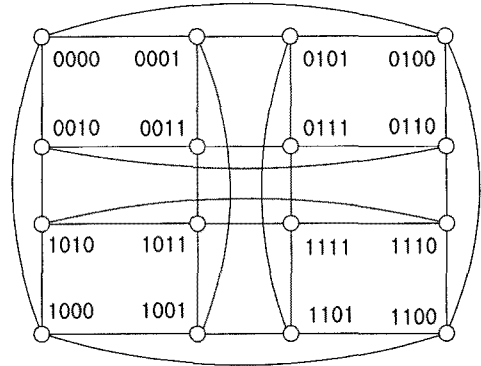


그림 3 4-차원 하이퍼큐브  $Q_4$

하이퍼큐브  $Q_n$ 에서 노드 중복이 없는 두 개의  $Q_{n-1}$ 을 찾는 방법은  $n$ 가지인데, 두 개의  $Q_{n-1}$ 의 각 노드는 일대일로 서로 연결되어 있으며 서로 대응하는 각 노드 쌍은  $i$ 번째 비트만 다르고 나머지 비트들이 같다. 따라서 하이퍼큐브  $Q_n$ 의 임의의 두 노드  $v$ 와  $w$ 가 인접할 때  $v$ 에 인접한 나머지 노드들과  $w$ 에 인접한 나머지 노드들은 각각 일대일로 연결되어 있다. 그림 3은 4차원 하이퍼큐브이다. 하이퍼큐브의 지름은  $\log_2 n$  ( $n$ =노드 수)으로 메쉬 부류 보다 큰 향상을 가져왔다. 하이퍼큐브는 노드-에지 대칭이고 이분할 그래프이다. 단순한 라우팅 알고리즘과 재귀적 성질을 가지는 그래프로서 대중적으로 연구되고, iPSC/2, iPSC/860, n-CUBE/2 시스템으로 상용화 되었다[4].

**2.4 하이퍼피터슨 HP**

하이퍼피터슨 그래프는 하이퍼큐브 유사 그래프이고 피터슨 그래프와 하이퍼큐브의 카테시안 곱(cartesian product)으로 만들 수 있다. 모든 노드에서 분지수가 같은 정규 그래프이고 노드-에지 대칭이다. 또한 높은 분지수, 높은 연결도, 적은 지름을 가지고 있다.  $j$  차원  $HP_j$ 는  $10 \times 2^{j-3}$ 개의 노드와  $5j \times 2^{j-3}$ 개의 에지로 구성되며  $HP_j=(V_{hp}, E_{hp})$ 로 나타낸다. 정점의 집합은 두 개의 투플로 이루어져 있고 정의는 다음과 같다[12].

$$V_{hp}=\{[B(u),P(i)] \mid B(u) \text{는 } j-3 \text{ 개의 2진 비트스트링}, \{0 \leq P(i) \leq 9\} \text{ 에지는 다음과 두 가지로 구분한다.}$$

피터슨 에지 :  $([B(u),P(i)], [B(u),P(j)])$ 는  $P(i)$ 와  $P(j)$ 가 피터슨 그래프로 연결되어 있다.

하이퍼큐브 에지 :  $([B(u), P(i)], [B(v), P(i)])$ 는  $B(u)$ 와  $B(v)$ 가 정확히 1 비트 다르다.

$HP_j$ 는  $Q_{j-3} \times P$  ( $j \geq 3$ )이고, 피터슨 그래프와  $j-3$ 차원

하이퍼큐브를 카테시안곱(cartesian product)하여 만든다. 즉  $Q_{j-3}$  하이퍼큐브의 각각의 노드들을 피터슨그래프로 대체한 것이다. 그림 4는 5차원 하이퍼피터슨이다. 노드의 주소는 2진 비트스트링으로 이루어진  $B(u)$ 와 한 자리 정수로 이루어진  $P(i)$ 의 조합으로 이루어져 있다. 피터슨 에지는 모두 나타냈고 그림의 복잡함을 피하기 위하여 하이퍼큐브 에지는  $P(i)$ 가 0, 3, 8일 경우만 나타내었다.  $P(i)$ 가 0, 3, 8을 제외한 다른 숫자일 경우도 0, 3, 8의 경우와 같이  $B(u)$ 가 한 비트 다르고  $P(i)$ 가 같은 노드 사이에 에지가 있다. HP의 차원은  $B(u)$ 의 비트스트링 개수 + 3이다.

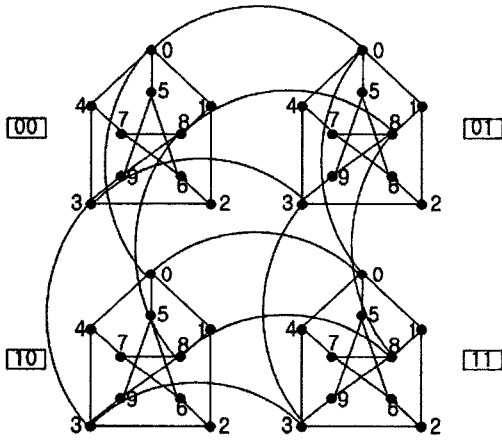


그림 4 5-차원 하이퍼피터슨  $HP_5$

$j$  차원 하이퍼피터슨은  $1.25 \times 2^j$ 개의 노드와 분지수  $j$ , 지름  $j-1$ 이다. 하이퍼피터슨은 하이퍼큐브의 성질과 매우 유사하며, 비슷한 지름과 분지수를 가질 때 노드의 수가 1.25배 많다[12].

### 3. 다른 연결망을 PT로 임베딩

그래프  $G$ 의 그래프  $H$ 에 대한 임베딩  $f$ 는 다음과 같이 정의되는 함수의 쌍  $(\theta, \rho)$ 을 말한다.  $\theta$ 는  $G$ 의 정점 집합  $V(G)$ 를  $H$ 의 정점 집합  $V(H)$ 에 대응시키는 함수이고,  $\rho$ 는  $G$ 의 에지  $e=(v,w)$ 를  $\theta(v)$ 와  $\theta(w)$ 를 잇는  $H$ 상의 경로와 대응시키는 함수이다. 확장율은  $G$ 의 정점의 개수에 대한  $H$ 의 정점의 개수의 비를 말한다. 확장율이 1보다 적은 경우  $G$ 에서 설계된 알고리즘을  $H$ 에서 효율적으로 사용하기 어렵고 확장율이 1보다 크면  $G$ 에서 설계된 알고리즘이  $H$ 에서 실행될 때  $H$ 의 모든 노드를 이용하지 못하는 비효율이 발생한다. 확장율은 1에 근접할수록 좋다. 다대일 임베딩에서 노드  $V(H)$ 의 하나의 노드에 사상되는 노드집합  $V(G)$ 의 최대 개수를 부하계수라고 하고 부하계수가 크면 프로세서의 작업처리 시

간이 증가한다. 확장율이 1보다 큰 경우 그래프  $G$ 의 하나의 노드를 그래프  $H$ 의 여러 개의 노드에 사상하는 것을 일대다 사상이라고 하고, 연장율을 줄이기 위해 사용된다. 그래프  $G$ 의 에지  $e$ 의 연장율은  $H$ 상에서의 경로  $p(e)$ 의 길이를 말하고, 임베딩  $f$ 의 연장율은  $G$ 의 모든 에지의 연장율 중 최대값이다. 연장율의 최소값은 1이며, 연장율이 크면  $G$ 에서 설계된 알고리즘이  $H$ 에서 적용될 때 연장율 만큼 메시지 전송 시간이 길어진다. 그래프  $H$ 의 에지  $e'$ 의 밀집율은  $e'$ 에 포함되는  $p(e)$ 의 개수를 말하고, 임베딩  $f$ 의 밀집율은  $H$ 의 모든 에지의 밀집율 중 최대값이다. 밀집율의 최적은 1이며, 그 연장율이 크면 전송 트래픽이 많이 발생한다[24].

#### 3.1 정이진트리(complete binary tree)를 피터슨-토러스 PT로 임베딩

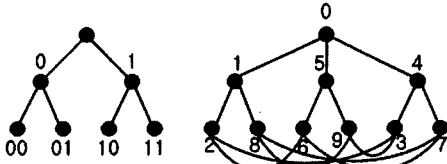
정이진트리를 PT에 임베딩 하는 것은 PT에 정이진트리를 만드는 것으로 대체한다. PT에 정이진트리를 만드는 방법은 간단하다. 그림 1의 (a)는 PT의 기본모듈을 하나의 점으로 표현하였고, 이것은 PT가 토러스(메쉬)를 서브 그래프로 가진다는 것을 보인다. 그림 1의 (a)에 H-트리를 생성하고 H-트리의 단말노드에서 H-트리의 단말노드를 루트노드로 하는 깊이 3 정이진트리를 생성한다.

깊이  $k$ 인 정이진트리  $T_k$ 는  $2^k-1$ 개의 노드와  $2^k-2$ 개의 에지로 구성된다. 루트 노드의 깊이는 1이고 단말 노드의 깊이는  $k$ 이다. 노드의 주소는 최대  $k-1$  길이의 2진 스트링으로 구성되어있고, 루트 노드는 비어있다. 에지는  $i$ 길이의 비트스트링으로 이루어진 노드  $x$ 와  $i+1$  길이의 비트 스트링으로 이루어진 노드  $xa$  사이에 있다( $a \in \{0, 1\}$ ,  $0 \leq i < k-1$ ). 루트 노드는 깊이 1이고 노드의 주소 스트링의 길이가  $i$ 인 노드의 깊이는  $i$ 이다. 깊이  $i$ 인 노드  $x$ 가 있을 때,  $x_0$ 은  $x$ 의 왼쪽 자식노드라고 하고  $x_1$ 은 오른쪽 자식노드라고 하고,  $x$ 는  $x_0$ 과  $x_1$ 의 부모 노드라고 한다.

**보조정리 1.** 깊이 3인 정이진트리(complete binary tree)  $T_3$ 을 피터슨 그래프에 밀집율 1, 확장율  $10/7$  그리고 연장율 1에 임베딩 가능하다.

**증명.** 그림 5의 (b)는 피터슨 그래프의 서브 그래프이고  $T_3$ 인 그림 5의 (a)를 포함하고 있으므로 연장율과 밀집율은 1이고 확장율은  $10/7$ 이다.  $T_3$ 의 루트노드를 피터슨 그래프의 노드 0에 사상하고,  $T_3$ 의 노드 0은 피터슨 그래프의 노드 1, 노드 00은 노드 2, 노드 01은 노드 8, 노드 1은 노드 4, 노드 10은 노드 3, 노드 11은 노드 7에 각각 사상한다.

$T_3$ 의 루트노드를 피터슨 그래프의 임의의 노드에 사상하더라도 피터슨 그래프는 노드-에지 대칭이므로 피터슨 그래프에 임베딩 가능하다. □



(a) 정이진트리  $T_3$  (b) 피터슨 그래프의 서브 그래프 그림 5 정이진트리  $T_3$ 를 피터슨 그래프에 임베딩

**정리 2.** 정이진트리(complete binary tree)  $T_{2\log_2(n+1)+1}$ 를  $PT(n,n)$ 에 밀집을 1, 확장을  $\equiv 5$  그리고 연장울  $\frac{3(n+1)}{4} + 1$ 에 임베딩 가능하다( $n=2^a-1$ ,  $a$ 는 1보다 큰 자연수).

**증명.** PT에 H-트리를 만드는 과정은 2.2절에 소개된 방법과 같고 자세히 설명하면 다음과 같다.  $r = \frac{n-1}{2}$ 라고 하고, PT의 노드  $(r,r,0)$ 을 루트노드로 한다. 왼쪽 자식노드는  $((r-1)/2, r, 5)$ 이고 오른쪽 자식노드는  $(n-(r-1)/2, r, 5)$ 이다. 루트노드의 왼쪽 자식노드의 왼쪽 자식노드는  $((r-1)/2, n-(r-1)/2, 0)$ 이고 오른쪽 자식노드는  $((r-1)/2, (r-1)/2, 0)$ 이다. 루트노드의 오른쪽 자식노드의 왼쪽 자식노드는  $(n-(r-1)/2, (r-1)/2, 0)$ 이고 오른쪽 자식노드는  $(n-(r-1)/2, n-(r-1)/2, 0)$ 이다. 이것은 정이진트리  $T_3$ 을 만든다. 4개의 단말 노드에서  $T_3$ 을 만드는 방법과 동일하게 여분의 노드가 없어질 때 까지 재귀적으로 정이진트리를 만든다. 마지막으로 모든 H-트리의 단말노드가 속한 그림 5의 (b)와 같은 PT의 기본모듈에서 노드 0을 루트노드로 하는 그림 5의 (a)와 같은  $T_3$ 을 만든다. 그림 6은  $PT(3,3)$ 에 노드  $(1,1,0)$ 을 루트노드로 한 정이진트리  $T_5$ 를 생성하였다. 크게 표시된 노드가 정이진트리의 노드들이고 큰 이진 비트스트링이 노드 주소이다.

PT에 밀집을 1에 H-트리가 생성되고, 보조정리 1에 의해 H-트리의 단말노드에서 밀집을 1에  $T_3$ 가 생성되므로 밀집율은 1이다. 보조정리 1에 의해 H-트리의 단말노드에서 연장울 1에  $T_3$ 가 생성되고, PT에 H-트리를 생성할 때 노드의 깊이에 따라 부모노드와 자식노드사이의 경로길이는 다르다. H-트리의 루트노드와 루트노드의 자식노드 사이의 경로가 가장 길고 그 깊이 연장울이다. 루트노드는  $(r,r,0)$ 에 사상 되고 왼쪽 자식노드는  $((r-1)/2, r, 5)$ , 오른쪽 자식 노드는  $(n-(r-1)/2, r, 5)$ 에 사상된다. 루트노드와 2개의 자식 노드들 사이의 외부경로길이는  $r - ((r-1)/2) = \frac{n+1}{4}$ 이고, 내부 경로길이는 루트 노드가 있는 기본모듈에서 1이고, 자식노드가 있는 기본모듈에서 2이고, 루트노드와 자식노드의 사이에 있는  $\frac{n+1}{4} - 1$ 개의 기본모듈에서 각각 2이다. 루트노드와

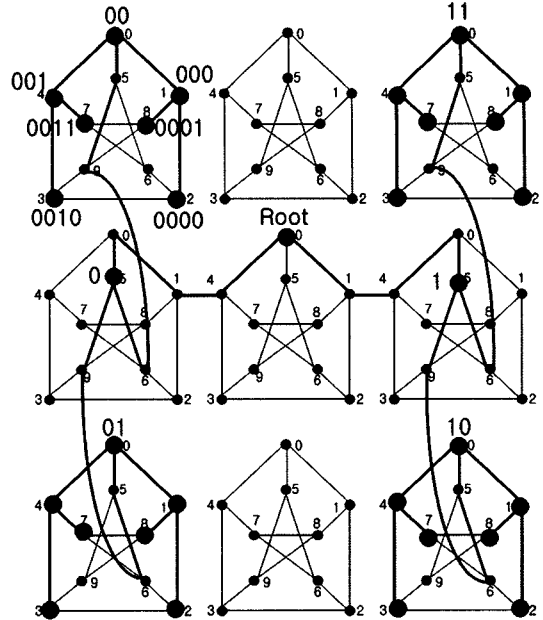


그림 6  $PT(3,3)$ 에 생성된  $T_5$

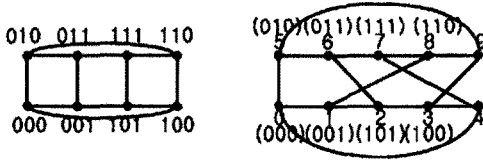
자식노드 사이의 경로길이인 연장울은  $\frac{n+1}{4} + (\frac{n+1}{4} - 1) \times 2 + 3$ 이다. 이 식을 정리하면  $\frac{3(n+1)}{4} + 1$ 이다. 정이진트리  $T_{2\log_2(n+1)+1}$ 의 노드수는  $2(n+1)^2 - 1$ 이고  $PT(n,n)$ 의 노드수는  $10n^2$ 이므로 확장율은  $10n^2 / (2n^2 + 4n + 1)$ 이다. □

**3.2 하이퍼큐브를 PT로 임베딩**

임베딩의 기본적인 전략은 하이퍼큐브를 3차원 하이퍼큐브 조각으로 나누고, 그 조각을 PT의 기본모듈에 사상하는 것이다. 3차원 하이퍼큐브를 하이퍼큐브의 기본모듈이라고 한다. 그림 8의 (a)는  $Q_5$ 를 나타낸다. 노드의 주소는 5 비트로 이루어져 있는데 편의상 LSB (least significant bit)에서부터 3자리와 나머지(그림 8의 (a)에서는 MSB(most significant bit)에서부터 2자리)는 별도로 표현하였다.

**보조정리 3.** 하이퍼큐브의 기본모듈  $Q_3$ 는 피터슨 그래프로 확장을 10/8 연장울 2 밀집율 2에 임베딩 가능하다.

**증명.** 그림 7의 (b)에서 괄호안의 이진 비트스트링은 피터슨 그래프에 사상되는 하이퍼큐브  $Q_3$ 의 주소이다. 하이퍼큐브  $Q_3$ 의 에지 (000, 100)은 피터슨 그래프의 경로 0, 4, 3으로 사상되고, 에지 (010, 110)은 경로 5, 9, 8로 사상되고, 에지 (001, 011)은 경로 1, 2, 6으로 사상되고, 에지 (101, 111)은 경로 2, 6, 7로 사상되고, 에지 (100, 110)은 경로 3, 9, 8로 사상된다. 앞 문장에



(a) 3차원 하이퍼큐브  $Q_3$  (b) 피터슨 그래프의 서브 그래프 그림 7  $Q_3$ 를 피터슨 그래프에 사상

적혀있지 않은  $Q_3$ 의 8개의 에지들은 피터슨 그래프의 에지로 사상된다. 4개의 에지는 연장율은 2이고 8개의 에지는 연장율을 1이다. 하이퍼큐브의 에지 (001, 011)과 에지 (101, 111)이 사상되는 피터슨 그래프의 경로는 피터슨 그래프의 에지 2, 6을 포함하므로 밀집율 2이다. □

**정리 4.** 하이퍼큐브  $Q_{\log_2 n^2 + 3}$ 을  $PT(n, n)$ 에 확장율 10/8, 연장율 1.5n+2 그리고 밀집율 4n에 임베딩 가능하다(n은 2의 지수 승).

**증명.** 하이퍼큐브  $Q_{\log_2 n^2 + 3}$ 의 노드수는  $8n^2$ 이고,  $PT(n, n)$ 의 노드수는  $10n^2$ 이므로 확장율은 10/8이다. 하이퍼큐브  $Q_{\log_2 n^2 + 3}$ 의 노드  $u$ 의  $i$ 번째 비트를  $u_i$  라고 한다.  $u$ 의 주소에서 LSB로부터 3번째 비트까지를 제외하고 짝수 번째 비트를 순서대로 조합한  $x(u) = u_{\log_2 n^2 + 2} \dots u_{i-2} u_i u_{i-2} \dots u_6 u_4$  이라고 하고, 홀수 번째 비트를 순서대로 조합한  $y(u) = u_{\log_2 n^2 + 3} \dots u_{i-3} u_{i-1} u_{i-1} \dots u_7 u_5$  이라고 한다( $i$ 는 짝수). 하이퍼큐브의 노드  $u$ 는  $PT$ 의 노드  $(x(u)_{(10)}, y(u)_{(10), p})$ 에 사상된다. 그림 8의 (a)에서  $x(u)=0, y(u)=0$ 인 8개의 노드들은 하이퍼큐브 기본모듈이고  $PT$ 의 기본모듈 (0,0)에 사상된다.  $x(u)=0, y(u)=1$ 은 (0,1)에  $x(u)=1, y(u)=0$ 은 (1,0)에  $x(u)=1, y(u)=1$ 은 (1,1)에 각각 사상된다. 하이퍼큐브 기본모듈을  $PT$ 의 기본모듈에 사상하는 방법은 보조정리 3의 증명에 있다.

하이퍼큐브의 에지  $(u, v)$ 에서  $u$ 와  $v$ 는 정확히 1비트 다르고,  $u_i \neq v_i$ 이다( $1 \leq i \leq \log_2 n^2 + 3$ ).  $i \leq 3$  일 경우 에지  $(u, v)$ 는 하이퍼큐브의 같은 기본모듈 안에 있는 노드들

을 연결하는 에지이고,  $i \geq 4$  일 경우 에지  $(u, v)$ 는 하이퍼큐브의 서로 다른 기본모듈 간의 노드들을 연결하는 에지이다. 하이퍼큐브의 에지를 두 가지의 경우로 나누어 분석한다.

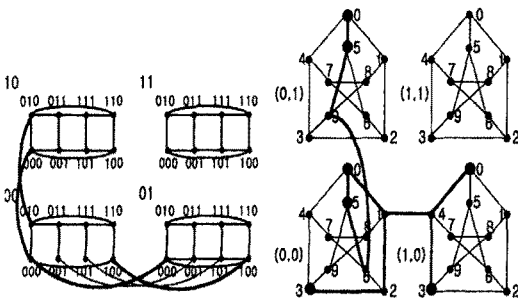
**경우 1.  $i \leq 3$  일 경우**

에지  $(u, v)$ 에서, 노드  $u$ 는  $PT$ 의 노드  $(x(u)_{(10)}, y(u)_{(10), p})$ 에 사상되고, 노드  $v$ 는  $PT$ 의 노드  $(x(v)_{(10)}, y(v)_{(10), p'})$ 에 사상된다.  $i \leq 3$  이면  $x(u)_{(10)} = x(v)_{(10)}, y(u)_{(10)} = y(v)_{(10)}, p \neq p'$ 이다.  $i \leq 3$ 의 의미는 노드  $u$ 와  $v$ 가 같은 하이퍼큐브 기본모듈에 있는 노드이고, 같은  $PT$ 의 기본모듈로 사상된다는 의미이다. 이 에지들에 대한 사상은 보조정리 3의 증명에 있다.

**경우 2.  $i \geq 4$  일 경우**

에지  $(u, v)$ 에서, 노드  $u$ 는  $PT$ 의 노드  $(x(u)_{(10)}, y(u)_{(10), p})$ 에 사상되고, 노드  $v$ 는  $PT$ 의 노드  $(x(v)_{(10)}, y(v)_{(10), p'})$ 에 사상된다.  $i \geq 4$ 의 의미는 노드  $u$ 와 노드  $v$ 가 하이퍼큐브의 서로 다른 기본모듈에 있고, 서로 다른  $PT$ 의 기본모듈로 사상된다는 의미이다.  $u_3 u_2 u_1 = v_3 v_2 v_1$ 이므로 하이퍼큐브의 기본모듈에서의 주소 스트리밍은 같다.  $x(u)$ 의  $t$  번째 비트를  $x(u)_t$ 라고 하고,  $y(u)$ 의  $t$  번째 비트를  $y(u)_t$ 라고 한다( $1 \leq t \leq \log_2 n^2 / 2$ ).

$i$ 가 홀수일 경우  $x(u)_{(10)} = x(v)_{(10)}, y(u)_{(10)} \neq y(v)_{(10)}, p = p'$ 이고, 노드  $u, v$ 는  $PT$  기본모듈의  $x$ 값이 같은 기본모듈로 사상된다.  $p = p'$ 이므로 사상된  $PT$ 의 기본모듈에서 피터슨 그래프의 주소도 같다. 에지  $(u, v)$ 에 대응하는  $PT$ 의 경로를 보면,  $p=2, 5, 7$ 일 경우  $(x(u)_{(10)}, y(u)_{(10), p}), (x(u)_{(10)}, y(u)_{(10), 6}), [(x(u)_{(10)}, y(u)_{(10), 1+9}), (x(u)_{(10)}, y(u)_{(10), 1+5}), (x(u)_{(10)}, y(u)_{(10), 1+6})], \dots, (x(v)_{(10)}, y(v)_{(10), 9}), (x(v)_{(10)}, y(v)_{(10), p'})$ 이다.  $p$ 가 2이면  $p''=3$ ,  $p$ 가 5이면  $p''=5$ ,  $p$ 가 7이면  $p''=8$ 이다. □부분은 중간경유 모듈에서 내부경로이다. 반복구간 ...은 □부분을  $|x(u)_{(10)} - x(v)_{(10)}| - 1$  만큼 반복한다. 시작노드에서 내부경로길이 1 도착노드에서 내부경로 길이 2, 중간경유모듈에서 내부경로길이  $(|x(u)_{(10)} - x(v)_{(10)}| - 1) \times 2$ 이고, 외부경로길이는  $|x(u)_{(10)} - x(v)_{(10)}| - 1 + 1$ 이다.  $p=3, 5, 8$ 일 경우  $(x(u)_{(10)}, y(u)_{(10), p}), (x(v)_{(10)}, y(v)_{(10), p'')$ ,  $(x(u)_{(10)}, y(u)_{(10), 6}), [(x(u)_{(10)}, y(u)_{(10), 1+9}), (x(u)_{(10)}, y(u)_{(10), 1+5}), (x(u)_{(10)}, y(u)_{(10), 1+6})], \dots, (x(v)_{(10)}, y(v)_{(10), 9}), (x(v)_{(10)}, y(v)_{(10), p})$ 이다.  $p$ 가 3이면  $p''=2$ ,  $p$ 가 5이면  $p''=5$ ,  $p$ 가 8이면  $p''=7$ 이다. 시작노드에서 내부경로길이 2 도착노드에서 내부경로 길이 1, 중간경유모듈에서 내부경로길이  $(|x(u)_{(10)} - x(v)_{(10)}| - 1) \times 2$ 이고, 외부경로길이는  $|x(u)_{(10)} - x(v)_{(10)}| - 1 + 1$ 이다.  $p=0, 1, 4, 6, 9$ 일 경우  $(x(u)_{(10)}, y(u)_{(10), p}), (x(v)_{(10)}, y(v)_{(10), p'')$ ,  $(x(u)_{(10)}, y(u)_{(10), 6}), [(x(u)_{(10)}, y(u)_{(10), 1+9}), (x(u)_{(10)}, y(u)_{(10), 1+5}), (x(u)_{(10)}, y(u)_{(10), 1+6})], \dots, (x(v)_{(10)}, y(v)_{(10), 9}), (x(v)_{(10)}, y(v)_{(10), p'')$ ,  $(x(v)_{(10)}, y(v)_{(10), p'')$ 이다.



(a)  $Q_5$  (b)  $PT(2,2)$   
그림 8  $Q_5$ 를  $PT(2,2)$ 에 임베딩

$y(v)_{(10),p}$ 이다.  $p$ 가 0이면  $p''=5, p'''=5$ ,  $p$ 가 1이면  $p''=2, p'''=8$ ,  $p$ 가 4이면  $p''=7, p'''=3$ ,  $p$ 가 6이면  $p''=6, p'''=5$ ,  $p$ 가 9이면  $p''=5, p'''=9$ 이다. 예지  $((x,y,p), (x,y,p'))$ 에서  $p=p'$ 일 경우 예지가 없는 것으로 간주한다. 시작노드에서 내부경로길이 2 도착노드에서 내부경로 길이 2, 중간경유모듈에서 내부경로길이  $(|x(u)_{(10)} - x(v)_{(10)}|-1) \times 2$ 이고, 외부경로길이는  $|x(u)_{(10)} - x(v)_{(10)}| - 1 + 1$ 이다.

$y(u)_t \neq y(v)_t$ 라고 하면,  $y(u)_{(10)}$ 와  $y(v)_{(10)}$ 는  $2^{t-1}$  차이가 있고  $t$ 가 가장 큰 값  $\log_2 n^2 / 2$  일 때  $n/2$  차이가 있다. 예지  $(u, v)$ 에서 노드  $u$ 가 사상된 PT의 노드  $(x(u)_{(10)}, y(u)_{(10),p})$ 와 노드  $(x(v)_{(10)}, y(v)_{(10),p})$  사이에는  $2^{t-1}$ 개의 외부예지가 있고, 최대 개수는  $n/2$ 개이다. PT의 두 노드 사이의 중간경유모듈과 노드  $(x(u)_{(10)}, y(u)_{(10),p})$ , 노드  $(x(v)_{(10)}, y(v)_{(10),p})$ 에서의 내부예지의 수는  $(n/2 - 1) \times 2 \times 2$ 이고 외부예지의 수는  $n/2$ 이므로 (내부예지의 수 + 외부예지의 수) =  $1.5n + 2$ 이다.  $i$ 가 홀수일 때 하이퍼큐브 예지  $(u, v)$ 는 PT에 연장을  $1.5n + 2$ 에 임베딩 가능하다.  $i$ 가 짝수일 경우  $x(u)_{(10)} \neq x(v)_{(10)}$ ,  $y(u)_{(10)} = y(v)_{(10)}$ ,  $p = p'$ 이고, 노드  $u, v$ 는 PT의 기본모듈의  $y$ 값이 같은 기본모듈로 사상된다.  $p = p'$ 이므로 사상된 PT의 기본모듈에서 피터슨 그래프의 주소도 같다. 증명은  $i$ 가 홀수 일 경우와 같다.

$i$ 가 홀수 일 경우 예지  $(u, v)$ 의 노드  $u$ 와  $v$ 가 사상되는 PT의 노드  $(x(u)_{(10)}, y(u)_{(10),p})$ 와 노드  $(x(v)_{(10)}, y(v)_{(10),p})$ 가 가장 먼 거리에 있을 때  $n/2$ 의 거리에 있고, 그 두 노드 사이에 있는  $n/2$ 개의 중간경유 기본모듈은 모두 예지  $((x,y,6), (x,y+1,9))$ 를 이용한다.  $i$ 가 짝수일 경우 예지  $((x,y,4), (x+1,y,1))$ 을 이용한다. 따라서 하이퍼큐브를 PT에 밀집을  $n/2 \times 8 = 4n$ 에 임베딩 가능하다. □

**정리 5.** 하이퍼큐브  $Q_{\log_2 n^2 + 4}$ 을  $PT(2n, n)$ 에 확장을  $10/8$ , 연장을  $3n + 2$  그리고 밀집을  $8n$ 에 임베딩 가능하다( $n$ 은 2의 지수 승).

**증명.**  $PT(2n, n)$ 의 노드수는  $20n^2$ 이고,  $Q_{\log_2 n^2 + 4}$ 의 노드수는  $16n^2$ 이다. 그러므로 확장을  $10/8$ 이다. 이것의 증명은 정리 4의 증명과 같다. 이 임베딩은 단지  $Q_{\log_2 n^2 + 4}$ 가  $Q_{\log_2 n^2 + 3}$ 을 2배로 확장했고,  $PT(2n, n)$ 은  $PT(n, n)$ 을  $x$ 축으로 2배만큼 확장한 것이다. 이 임베딩은  $HP_{\log_2 n^2 + 5}$ 을  $PT(2n, 2n)$ 에 임베딩 하는 것과 결과가 같다. 예지  $(u, v)$ 의 노드  $u$ 와  $v$ 가 사상되는 PT의 노드  $(x(u)_{(10)}, y(u)_{(10),p})$ 와 노드  $(x(v)_{(10)'}, y(v)_{(10)',p})$ 의 외부경로길이는 최대  $n$ 이다. 내부 경로길이는  $(n+1) \times 2$ 이므로 최대 경로길이는  $3n + 2$ 이므로 연장을  $3n + 2$ 에 임베딩 된다. 노드  $(x(u)_{(10)}, y(u)_{(10),p})$ 와 노드  $(x(v)_{(10)}, y(v)_{(10),p})$

이 가장 먼 거리에 있을 때  $n$ 의 거리에 있고, 그 두 노드를 출발노드와 목적노드로 하는 경로의 출발점과 경로 상에 있는  $n$ 개의 기본모듈은 모두 같은 외부예지를 이용한다. 따라서 밀집율은  $n \times 8$ 이다. □

**3.3 하이퍼피터슨 HP를 피터슨-토러스 PT로 임베딩**

임베딩의 기본적인 전략은 HP의 피터슨 그래프를 PT의 기본모듈 피터슨 그래프에 사상하는 것이다. HP의 노드주소  $B(u)$ 를 PT의 기본모듈  $(x,y)$ 의 주소에 사상하는 방법은 3.2절에 있는 하이퍼큐브를 PT에 임베딩 하는 방법과 같다. HP의 노드들 중 노드주소  $B(u)$ 가 같은 10개의 노드들(피터슨 그래프)을 HP의 기본모듈이라고 한다.

**정리 6.** 하이퍼피터슨  $HP_{\log_2 n^2 + 3}$ 을 피터슨-토러스  $PT(n, n)$ 에 확장을 1, 연장을  $1.5n + 2$  그리고 밀집을  $5n$ 에 임베딩 가능하다( $n$ 은 2의 지수 승).

**증명.** 피터슨-토러스  $PT(n, n)$ 의 노드수와 하이퍼피터슨  $HP_{\log_2 n^2 + 3}$ 의 노드수는  $10n^2$ 이다. 그러므로 확장을 1이다.  $HP_{\log_2 n^2 + 3}$ 의 노드  $[B(u), P(k)]$ 에서  $B(u)$ 의  $i$ 번째 비트를  $B(u)_i$ 라고 한다.  $B(u)$ 에서 LSB로부터 홀수 번째 비트를 순서대로 조합한  $xb(u) = B(u)_{\log_2 n^2 - 1} \dots B(u)_{i-2} B(u)_i B(u)_{i-2} \dots B(u)_3 B(u)_1$ 이라고 하고, 짝수 번째 비트를 순서대로 조합한  $yb(u) = B(u)_{\log_2 n^2} \dots B(u)_{i+3} B(u)_{i+1} B(u)_{i-1} \dots B(u)_4 B(u)_2$ 이라고 한다( $i$ 는 홀수).  $HP_{\log_2 n^2 + 3}$ 의 노드  $[B(u), P(k)]$ 는  $PT(m, n)$ 의 노드  $(xb(u)_{(10)}, yb(u)_{(10)}, P(k))$ 에 사상한다. 그림 9는  $HP_7$ 과  $PT(4,4)$ 를 보였다. 두 그래프에서 하나의 점은 기본모듈이고, HP의 기본모듈에서 PT의 기본모듈로의 사상은 피터슨 그래프의 주소가 정확하게 같은 노드로 사상된다.

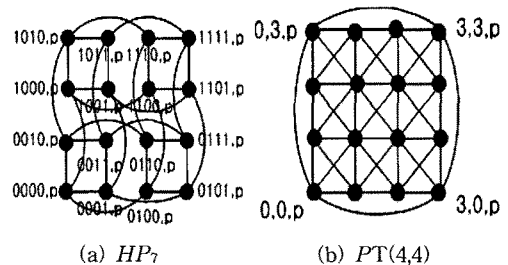


그림 9  $HP_7$ 을  $PT(4,4)$ 에 임베딩

HP의 예지는 피터슨 예지와 하이퍼큐브 예지 두 가지 종류가 있다. 예지  $([B(u), P(k)], [B(v), P(k)])$ 는 같은 피터슨 그래프 주소를 가지는 하이퍼큐브 예지이고 예지  $([B(u), P(k)], [B(u), P(o)])$ 는 같은 하이퍼 큐브 주소를 가지는 피터슨 예지이다. 피터슨 예지  $([B(u), P(k)], [B(u), P(o)])$ 는  $((xb(u)_{(10)}, yb(u)_{(10)}, P(k)), (xb(u)_{(10)}, yb(u)_{(10)},$



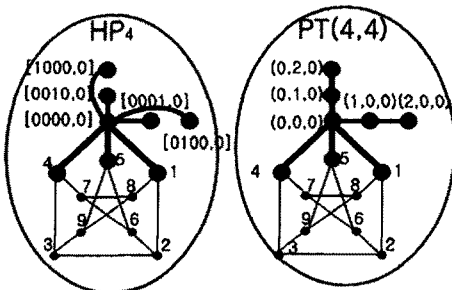
$P(o)$ 로 사상되므로 연장을 1, 밀집을 1이다.

HP의 하이퍼큐브 에지 ( $[B(u),P(k)], [B(v),P(k)]$ )에서, HP의 노드  $[B(u),P(k)]$ 는 PT의 노드  $((xb(u)_{(10)}, yb(u)_{(10)},P(k)))$ 로, HP의 노드  $[B(v),P(k)]$ 는 PT의 노드  $((xb(v)_{(10)}, yb(v)_{(10)},P(k)))$ 로 사상된다.  $B(u)$ 와  $B(v)$ 는 정확히 1 비트 다르고, 그 다른 비트가 LSB로부터 홀수 번째 비트 이면  $xb(u) \neq xb(v)$ ,  $yb(u) = yb(v)$ 이고 짝수 번째 비트이면  $xb(u) = xb(v)$ ,  $yb(u) \neq yb(v)$ 이다.  $xb(u)$ 의  $t$  번째 비트를  $xb(u)_t$ 라고 하고,  $yb(u)$ 의  $t$  번째 비트를  $yb(u)_t$ 라고 한다( $1 \leq t \leq \log_2 n^2/2$ ).  $xb(u)_t \neq xb(v)_t$ 이면  $xb(u)_{(10)}$ 와  $xb(v)_{(10)}$ 는  $2^{t-1}$ 만큼 차이가 있다.  $t$ 가 가장 큰 값  $\log_2 n/2$ 일 때  $\frac{n}{2}$ 이므로,  $((xb(u)_{(10)}, yb(u)_{(10)}, P(k)),$

$(xb(v)_{(10)}, yb(v)_{(10)}, P(k)))$ 의 외부 경로길이는 최대  $\frac{n}{2}$ 이고, 내부 경로길이는  $(\frac{n}{2} + 1) \times 2$ 이다. 두 경로길이의 합은  $1.5n + 2$ 이고 이 값이 연장율이다.  $yb(u)_t \neq yb(v)_t$ 일 경우도 마찬가지다.

HP의 에지  $(u, v)$ 의 두 노드  $u$ 와  $v$ 가 사상되는 PT의 두 노드  $(xb(u)_{(10)}, yb(u)_{(10)}, P(k))$ 와 노드  $(xb(v)_{(10)}, yb(v)_{(10)}, P(k))$ 이다.  $xb(u)_t \neq xb(v)_t$ 일 경우 두 노드는  $y$  값이 같은 기본모듈에 있고 가장 먼 거리에 있을 때  $x$  축으로  $n/2$ 의 거리에 있다. 그 경로의 출발기본모듈과  $(n/2) - 1$ 개의 중간경유 기본모듈은 모두 같은 외부에지  $((x, y, 1), (x+1, y, 4))$ 를 이용한다.  $yb(u)_t \neq yb(v)_t$ 일 경우 외부에지  $((x, y, 6), (x, y+1, 9))$ 를 이용한다. 이 에지에서 밀집율은  $n/2 \times 10$ 이다. 따라서 연장율을  $1.5n + 2$  그리고 밀집율  $5n + 2$ 에 임베딩 가능하다. □

예를 들어 그림 10의 에지  $[0000, 4], [0100, 4]$ 에서 노드  $[0000, 4]$ 는  $(0, 0, 4)$ 에 노드  $[0100, 4]$ 는 노드  $(2, 0, 4)$ 에 사상된다. 노드  $(0, 0, 4)$ 와 노드  $(2, 0, 4)$  사이의 경로는 다음과 같다.  $(0, 0, 4), (0, 0, 0), (0, 0, 1), (1, 0, 4), (1, 0, 0), (1, 0, 1), (2, 0, 1), (2, 0, 0), (2, 0, 4)$ . 2개의 외부에지와  $3 \times 2$ 개의 내부에지를 가지는 경로이며 길이는 8이다.



(a)  $HP_4$ 의 일부 (b)  $PT(4,4)$ 의 일부  
 그림 10  $HP_4$ 를  $PT(4,4)$ 에 임베딩

**정리 7.** 하이퍼피터슨  $HP_{\log_2 2n^2+3}$ 을 피터슨-토러스  $PT(2n, n)$ 에 확장율 1, 연장율  $3n+2$  그리고 밀집율  $10n+2$  에 임베딩 가능하다( $n$ 는 2의 지수 승).

**증명.** 피터슨-토러스  $PT(2n, n)$ 의 노드수와 하이퍼피터슨  $HP_{\log_2 2n^2+3}$ 의 노드수는  $20n^2$ 이다. 그러므로 확장율 1이다. 노드의 사상은 정리 6의 증명과 같다. 이 임베딩은 단지  $HP_j$ 가  $Q_{j-3}$  그래프와 피터슨 그래프의 카터션 곱으로 이루어진 것인데  $j-3$ 이 홀수일 때를 말하고, PT는  $x$ 축이  $y$ 축보다 정확히 2배만큼 큰 그래프이다. 이 임베딩은  $PT(2n, 2n)$ 을  $HP_{\log_2 4n^2+3}$ 에 임베딩하는 것과 같다.  $((xb(u)_{(10)}, yb(u)_{(10)}, p), (xb(v)_{(10)'}, yb(v)_{(10)'} p))$ 의 외부 경로길이는 최대  $n$ 이다. 내부 경로길이는  $(n+1) \times 2$ 이므로 최대 경로길이는  $3n+2$ 이므로 연장율  $3n+2$ 에 임베딩 된다. 에지  $([B(u),P(k)], [B(v),P(k)])$ 를 가지는 피터슨 그래프의 10개의 노드들은 모두 같은 외부경로를 이용하므로 외부경로에서의 밀집율은  $10n$ 이다. 경로의 마지막 노드에서의 밀집율이 2이므로 밀집율  $10n+2$ 에 임베딩 된다. 따라서 연장율  $3n+2$  그리고 밀집율  $10n+2$ 에 임베딩 가능하다. □

4. 결론

상호연결망 사이의 임베딩은 설계된 병렬 알고리즘을 재사용 할 수 있게 하는 의미 있는 작업이다. 본 논문에선 널리 알려진 정이진트리와 상용화된 하이퍼큐브 그리고 하이퍼피터슨 네트워크를 PT 네트워크로 임베딩 하였다. 정이진트리  $T(2\log_2(n+1)+1)$ 를  $PT(n, n)$ 에 밀집율 1, 확장율  $\approx 5$  그리고 연장율  $\frac{3(n+1)}{4} + 1$ 에 임베딩 하였다. 하이퍼큐브  $Q_{\log_2 2n^2+3}$ 을  $PT(n, n)$ 에 확장율 10/8, 연장율  $1.5n+2$  그리고 밀집율  $4n$ 에 임베딩 하였고, 하이퍼 피터슨  $HP_{\log_2 2n^2+3}$ 을  $PT(n, n)$ 에 확장율 1, 연장율  $1.5n+2$  그리고 밀집율  $5n$ 에 임베딩 하였다. 역으로 PT 네트워크에서 개발된 알고리즘을 다른 연결망에서 사용할 수 있도록 PT 네트워크를 다른 연결망에 임베딩 하는 것은 더 연구해볼만한 가치가 있다.

참고 문헌

[1] K. W. Tang and S. A. Padubidri, "Diagonal and toroidal Mesh Networks," IEEE trans. comput., Vol.43, No.7, pp. 815-826, Jul 1994.  
 [2] I. Stojmenovic, "Honeycomb Network: Topological Properties and Communication Algorithms," IEEE trans. on Parallel and Distributed Systems, Vol.8, No.10, pp. 1036-1042, Oct 1997.  
 [3] M. S. Chen and K. G. Shin, "Addressing, Routing, and Broadcasting in Hexagonal Mesh Multipro-

- cessors," IEEE trans. comput., Vol.39, No.1, pp. 10-18, Jan 1990.
- [4] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," IEEE Trans. Comput., Vol.37, No.7, pp. 867-872, July 1988.
- [5] A. El-Amawy and S. Latifi, "Properties and Performances of Folded Hypercubes," IEEE Trans. Parallel and Distributed System, Vol.2, No.1, pp. 31-42, 1991.
- [6] K. Efe, "A Variation on the Hypercube with Lower Diameter," IEEE Trans. comput., Vol.40, No.11, pp. 1312-1316, 1991.
- [7] J-H. Park, "Circulant Graphs and Their Application to Communication Networks," Ph.D. Thesis, Dept. of Computer Science, KAIST, Taejon Korea, 1992.
- [8] S. B. Akers, D. Harel, and B. Krishnamurthy, "The Star Graph: An Attractive Alternative to the N-Cube," Proc. of the Int. Conf. Parallel Processing, pp. 393-400, 1987.
- [9] C-H Yeh and E. Varvarigos, "Macro-Star Networks: Efficient Low-Degree Alternatives to Star Graphs for Large-Scale Parallel Architectures," Frontier'96, Symp. on the Frontiers of Massively Parallel Computation, Oct. 1996.
- [10] S. Latifi and P. K. Srimani, "Transposition Networks as a Class of Fault-Tolerant Robust Networks," IEEE Trans. Comput., Vol.45, No.2, pp. 230-238, Feb. 1996.
- [11] H. O. Lee, J. S Kim, K. W. Park and J. H. Seo, "Matrix Star Graphs: A New Interconnection Network Based on Matrix Operations," Lecture Note in Computer Science : ACSAC 2005, LNCS 3740, pp. 478-487, Oct. 2005.
- [12] S.K. Das and A.K. Banerjee. Hyper Petersen network: Yet another hypercube-like topology. In Proceedings of the 4th Symposium on the Frontiers of Massively Parallel Computation (Frontiers' 92), pages 270-277, McLean, Virginia, USA, October 1992. Computer Society Press.
- [13] C. H. Yeh and B. Parhami, "Routing and Embeddings in Cyclic Petersen Networks: An Efficient Extension of the Petersen Graph," 1999 International Conference on Parallel Processing (ICPP'99), pp. 258-265, 1999.
- [14] Ohring and S. K. DAS, "Folded Petersen cube networks: new competitors for the hypercubes," IEEE Transactions on Parallel and Distributed Systems, Vol.7, No.2, pp. 151-168, Feb 1996.
- [15] S. K. Lee and H. A. Choi, "Embedding of Complete Binary Trees into Meshes with Row-Column Routing," IEEE Transactions on Parallel and Distributed Systems, Vol.7, No.5, pp. 493-497, MAY 1996.
- [16] C. H Kim, J. Choi and H.S. Lim, "Embedding Full Ternary Trees into Recursive Circulants," EurAsia-ICT2002, LNCS 2510, pp. 874-882, 2002.
- [17] S. Y. Hsieh, "Embedding of Cycles in the Faulty Hypercube," ACSAC2005, LNCS 3740, pp. 229-235, 2005.
- [18] P. F. Cortese and G. D. Battista, "On Embedding a Cycle in a Plane Graph," GD2005, LNCS 3843, pp. 49-60, 2005.
- [19] S. Bezrukov, "Embedding complete trees into the Hypercube," Discrete Applied Mathematics, Vol. 100, No.2-3, pp. 101-119, 2001.
- [20] J. H. Park, H. S. Lim and H. C. Kim, "Embedding Starlike Trees into Hypercube-Like Interconnection Networks," ISPA 2006 Ws, LNCS 4331, pp 301-310, 2006.
- [21] K. Li, "On the Performance of Randomized Embedding of Reproduction Trees in Static Networks," International Journal of Parallel Programming, Vol.31, No.5, pp. 393-406, OCT 2003.
- [22] K. Li, "Comparative Performance Evaluation of a Random-Walk-Based Algorithm for Embedding Dynamically Evolving Trees in Hypercubic Networks," Computation and Concurrency: Practice and Experience, Vol.15, 2003.
- [23] S. K. Das and A. K. Banerjee, "Embeddings into Hyper Petersen networks: yet another hypercube-like interconnection Topology," VLSI Design, Vol.2, No.4, pp. 335-351, 1995.
- [24] S. Bettayeb and B. Cong and M. Girou and I. H. Sudborough, "Embedding Star Networks into Hypercubes," IEEE trans. comput., Vol.45, No.2, pp. 186-194, Feb. 1996.
- [25] J.D. Ullman, Computational Aspects of VLSI. Rockville, md.: Computer Science Press, 1984.
- [26] D. Gorden, "Efficient embeddings of binary trees in VLSI arrays," IEEE Trans. Comput., Vol.C-36, No.9, pp. 1009-1018, 1987.
- [27] P. Zienicke, "Embeddings of treelike graphs into 2-dimension meshes," Graph-Theoretic Concepts in Computer Science, LNCS, Vol.484, pp. 182-192, 1991.
- [28] A. Matsubayashi and R. Takasu, "Minimum Congestion Embedding of Complete Binary Tree into Tori," IEICE Trans. Fundamentals, Vol. E83-A, No.9, Sep. 2000.
- [29] T. M Stricker "Supporting the hypercube programming model on mesh architectures," Proc. of the fourth annual ACM symposium on Parallel algorithms and architectures, pp. 148-157, 1992.
- [30] A. Gonzalez, V. Garcia and L. D. Cerio, "Executing Algorithm with Hypercube Topology on Torus Multicomputers," IEEE Transactions on Parallel and Distributed Systems, Vol.6, No.8, pp. 803-814, Feb 1995.
- [31] S. Matic, "Emulation of Hypercube Architecture on Nearest-Neighbor Mesh-Connected Processing Elements," IEEE Transaction on computer. Vol.39, pp. 698-700, May 1990.

- [32] 서정현, 이형욱, 장문석, "멀티컴퓨팅 시스템을 위한 피터슨-토러스(PT) 네트워크", 정보과학회 논문지 제 35권 6호, 2008년 6월.

서 정 현

정보과학회논문지 : 시스템 및 이론  
제 35 권 제 6 호 참조

이 형 욱

정보과학회논문지 : 시스템 및 이론  
제 35 권 제 6 호 참조

장 문 석

정보과학회논문지 : 시스템 및 이론  
제 35 권 제 6 호 참조