

하이퍼-스타 그래프: 다중 컴퓨터를 위한 새로운 상호 연결망

이 형 옥[†] · 김 병 철^{††} · 임 형 석^{†††}

요 약

본 논문에서는 새로운 다중 컴퓨터의 위상으로 하이퍼-스타 그래프를 제안하고, 하이퍼-스타 그래프에 대하여 주요 망 척도인 분지수, 연결도, 확장성, 라우팅, 지름, 임베딩 등의 특징을 분석한다. 하이퍼-스타 그래프 $HS(2n, n)$ 에 에지를 한개 추가한 folded 하이퍼-스타 그래프 $FHS(2n, n)$ 은 $2C_n$ 개의 노드를 갖고 널리 알려진 상호 연결망인 하이퍼큐브에 비해 상대적으로 분지수와 지름이 작은 값을 갖는 연결망이다. Folded 하이퍼-스타 그래프 $FHS(2n, n)$ 와 하이퍼큐브 Q_n 이 같은 노드 수를 가질때 분지수 \times 지름에 대한 척도에 있어서 $FHS(2n, n)$ 은 $n^2/2$ 보다 작고, 하이퍼큐브 Q_n 은 n^2 이다. 이 결과는 folded 하이퍼-스타 그래프 $FHS(2n, n)$ 가 하이퍼큐브 Q_n 과 그의 변형들보다 연결망의 망 비용에 있어서 우수함을 의미한다. 또한 하이퍼큐브 Q_n 과 $n \times n$ 토러스를 하이퍼-스타 그래프 $HS(2n, n)$ 에 연장비율 2에 임베딩하는 방법을 보인다.

Hyper-Star Graphs: A New Interconnection Network for Multicomputer

Hyeong-Ok Lee[†] · Byung-Chul Kim^{††} · Hyeong-Seok Lim^{†††}

ABSTRACT

In this paper, we propose the hyper-star graph as a new topology for multicomputer and analyze the characteristics of its network parameters, such as degree, connectivity, scalability, routing, diameter, and embedding. Folded hyper-star graph $FHS(2n, n)$, in which an extra link is added to hyper-star graph $HS(2n, n)$, contains $2C_n$ vertices and has relatively lower degree and smaller diameter than the well-known hypercube graph. The folded hyper-star graph $FHS(2n, n)$ and the hypercube Q_n have the same number of vertices. In terms of the network cost, defined as the product of the degree and diameter, the former has less than $n^2/2$ while the latter has n^2 . It means that folded hyper-star graph $FHS(2n, n)$ is better than the hypercube Q_n and its variations with respect to network cost. In addition, we describe a method to embed hypercube Q_n and $n \times n$ torus into hyper-star $HS(2n, n)$ with dilation 2.

1. 서 론

영상처리, 기상 예보와 분석, 유체역학, 모델링과 시

* 본 논문은 정보통신부의 정보통신분야 우수학교 지원사업에 의해서 수행된 결과임.

† 준 회 원 : 전남대학교 전산통계학과 시간강사

†† 정 회 원 : 순천대학교 컴퓨터과학과 교수

††† 정 회 원 : 전남대학교 전산통계학과 교수

논문접수 : 1998년 6월 11일, 심사완료 : 1998년 10월 8일

물레이션, 인공지능, CAD/CAM 등 공학과 과학 분야의 대부분의 응용 문제들은 많은 계산을 수행하며, 동시에 실시간 처리를 필요로하므로 지금까지의 컴퓨터 보다 빠른 계산 능력을 갖는 고성능 컴퓨터에 대한 필요성이 절실히 요구된다. 이러한 요구를 만족시키기 위해서 여러개의 처리기를 사용하여 병렬적으로 작업을 처리하는 병렬 처리 컴퓨터에 대한 관심이 증가하

고 있다. 병렬 처리를 위한 컴퓨터는 크게 공유 메모리를 갖는 다중 프로세서(multiprocessor)시스템과 분산 메모리를 갖는 다중 컴퓨터(multicomputer)시스템으로 나눌 수 있다. 다중 프로세서 시스템은 공유 메모리를 가지며, 프로세서들 사이의 통신은 공유 메모리의 접근에 의해 이루어진다. 이 경우 전체 메모리 공간에 대한 접근이 쉬운 장점을 갖지만 기억장치 접근시 병목 현상으로 성능 저하를 초래할 수 있으며 확장성이 좋지 않은 단점이 있다. 다중 컴퓨터 시스템은 자신의 메모리를 갖는 프로세서들을 상호 연결망으로 연결하고, 프로세서들 사이에 통신은 상호 연결망을 통해 메시지를 교환하는 방식으로 이루어진다. 이러한 다중 컴퓨터의 연결망 구조는 전체 시스템의 성능 및 확장성에 큰 영향을 미치는 것으로 병렬 처리 컴퓨터 개발의 기반 기술 제공을 위해 그 필요성이 증가되고 있다[8].

상호 연결망은 각 프로세서들을 노드로, 프로세서들 사이에 통신 채널을 에지로 나타내는 무방향 그래프로써 표현 될 수 있다. 상호 연결망을 평가하는 망 척도는 분지수(degree), 연결도(connectivity), 확장성(scalability), 지름(diameter), 고장 허용도(fault tolerance) 및 대칭성(symetric) 등이 있다[2,4,8]. 이러한 망 척도의 정의와 의미는 다음과 같다. 연결망 G 에 속한 임의의 노드 v 의 분지수란 노드 v 에 인접한 에지의 개수이다. G 의 분지수란 $V(G)$ 에 속한 노드들의 분지수중 최대값을 말한다. 분지수는 주어진 연결망으로 병렬 컴퓨터를 설계할 때 처리기의 핀 수나 라우팅 제어 논리의 복잡도를 결정하는 요인이 된다. 따라서 분지수는 처리기의 하드웨어의 비용에 영향을 준다. 노드(에지) 연결도는 연결망을 노드 중복없는 둘 이상의 부분으로 나누기위해 제거해야할 최소 노드(에지)의 개수이다. 연결망에서 노드 연결도와 분지수가 같은 연결망을 최대 고장 허용도를 가졌다고 한다. 연결망의 지름은 연결망의 임의의 두 노드간의 최단 경로길이중 최대값으로, 이는 연결망 전체에 정보를 전파하는데 드는 지연 시간의 하한값으로 그 연결망의 처리량을 나타낸다. 연결망 G 에 속한 어떤 노드(에지)에서도 G 가 똑같이 보일 때 G 는 노드(에지) 대칭적이라 하고, 일반적으로 연결망이 재귀적 구조를 가지고 있으면 확장성이 있음을 의미한다. 위의 망 척도중 하드웨어의 비용과 관련된 분지수와 메시지의 전송 시간과 관련된 지름은 상

호간에 상관 관계를 갖고 있다. 일반적으로 연결망의 분지수를 늘리면 지름을 줄이게 되어 그 연결망에서의 처리량(throughput)을 높일 수 있는 장점이 있지만 병렬 컴퓨터를 설계할 때 처리기의 핀 수가 늘어나게 되어 하드웨어의 비용이 증가하는 단점이 있다. 분지수가 작은 연결망은 하드웨어 비용은 줄어드는 반면 메시지 전송 시간이 늘어나게 되어 연결망의 지연 시간(latency time)이나 처리량이 나빠지는 단점이 있다. 이러한 특성 때문에 상호 연결망을 비교 평가하기 위해 연결망의 분지수 \times 지름 값으로 정의되는 망 비용(network cost)[2,12,13]이 있다. 상호 연결망의 비용은 연결망이 같은 개수의 노드를 가질 때 분지수 \times 지름 값이 작은 연결망이 좋은 특성을 갖는 것으로 분석한다. 널리 알려진 상호 연결망으로는 트리(tree), Hexagonal 메쉬[3], Toroidal 메쉬[15], Diagonal 메쉬[15], Honeycomb 메쉬[13], 하이퍼큐브[11], Cube-Connected-Cycle[9], Multiply-Twisted-Cube[4], Folded Hypercube[7], star 그래프[1,10], Hierarchical Cubic Network[5], Shuffle Exchange[14] 등이 있다.

메쉬 구조는 평면 그래프로써 VLSI 회로 설계같은 분야에서 많이 이용되는 구조로 현재까지 널리 이용되고 있으며 다양한 시스템으로 상용화 되었다[13]. 낮은 차원의 메쉬는 설계하기 쉽고 알고리즘 관점에서도 매우 유용하므로 병렬 처리 컴퓨터의 연결망으로 많이 쓰이고 있다. 높은 차원의 메쉬일수록 지름이 작아지고 Bisection Width가 커지며 여러 가지 병렬 알고리즘을 빨리 수행할 수 있지만 비용이 많이 든다. 일반적인 격자 구조를 갖는 메쉬의 지름을 개선한 구조로 Hexagonal 메쉬[3], Toroidal 메쉬[15], Diagonal 메쉬[15], Honeycomb 메쉬[13] 등이 제안되었다. 특히, 망 비용(network cost)에서 Honeycomb 메쉬는 같은 노드 수를 갖는 다른 종류의 메쉬보다 우수하고 이동 전화의 기지국 위치문제, 벤젠탄화수소의 표현, 컴퓨터 그래픽, 이미지 프로세싱 등의 응용분야를 갖는 연결망이다[13]. 표 1에서는 메쉬와 그 변형된 연결망들이 같은 수의 노드를 가질때 분지수, 지름 및 연결망의 망 비용을 비교 하였다[13].

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 다른 노드 사이에 에지가 있다. 2^n 개의 노드를

〈표 1〉 메쉬와 그 변형에서 망 비용
 (Table 1) Network cost of hypercube and variation of it

	분지수	지름	망 비용
mesh-connected computer	4	$2\sqrt{n}$	$8\sqrt{n}$
hexagonal mesh	6	$1.16\sqrt{n}$	$6.93\sqrt{n}$
honeycomb mesh	3	$1.63\sqrt{n}$	$4.90\sqrt{n}$
torus	4	\sqrt{n}	$4\sqrt{n}$
hypercube	$\log n$	$\log n$	$\log^2 n$

갖는 하이퍼큐브의 분지수와 지름은 각각 n 이고, 분지수 \times 지름은 n^2 이다. 하이퍼큐브의 특성은 연결망이 노드 및 에지 대칭이고 단순한 재귀적 구조를 가지고 있으며 링, 트리, 피라미드, 메쉬 등과 같은 다른 연결망 구조들이 효율적으로 임베딩될 수 있다는 장점이 있지만 차원이 증가함에 따라 노드의 분지수 또한 그에 비례하여 증가하는 단점을 갖는다[6,11]. 이러한 단점을 극복하기 위해 하이퍼큐브에 노드나 에지를 추가함으로써 노드의 분지수가 항상 상수값을 갖는 하이퍼큐브의 변형된 형태의 새로운 연결망으로 Butterfly, Cube-Connected-Cycle[9], de Bruijn, Shuffle Exchange[14] 등이 있다. 하이퍼큐브의 또 다른 단점중의 하나는 분지수에 비해 지름과 노드간의 평균 거리가 짧지 않다는 것이다. 이것은 하이퍼큐브가 에지를 효율적으로 사용하지 못함을 의미한다. 이러한 문제를 극복하기 위해 노드와 에지의 인접 관계를 조작하여 새로운 연결망으로 Multiply-Twisted-Cube[4], Folded Hypercube[7], Hierarchical Cubic Network[5] 등이 있다.

본 논문에서는 상호 연결망의 망 비용 즉, 분지수 \times 지름 값이 하이퍼큐브보다 작은 값을 갖는 새로운 상호 연결망으로 하이퍼-스타 그래프를 제안하고 분석한다. 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 하이퍼-스타 그래프 $HS(n,k)$ 의 정의와 확장성, 연결도 및 동형사상 성질을 분석하고, 3장에서는 최단 경로 라우팅 알고리즘과 지름에 대하여 알아보고, 4장에서는 하이퍼-스타 그래프 $HS(n,k)$ 의 지름을 개선하기 위해 하이퍼-스타 그래프에 에지를 한개 추가한 folded 하이퍼-스타 그래프 $FHS(n,k)$ 를 정의하고 성질을 분석한다. 5장에서는 하이퍼큐브 Q_n 과 $n \times n$ 토러스 구조를 하이퍼-스타 그래프 $HS(n,k)$ 으로 임베딩을 분석하고 마지막으로 결론을 맺는다.

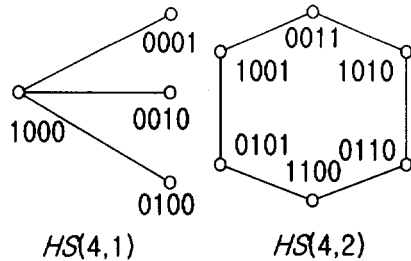
2. 하이퍼-스타 그래프의 정의와 특성

2.1 하이퍼-스타 그래프의 정의

하이퍼-스타 그래프 $HS(n,k)$ 는 이진수 "0" 또는 "1"로 구성된 n 개 비트 스트링 $s_1s_2...s_i...s_n$ 에서 k 개 비트가 "1"로 구성된 비트 스트링으로 노드를 나타내고 ($n > k$), 노드를 연결하는 에지는 심볼 s_i 과 s_j 가 보수이면서 교환된 비트 스트링을 연결하는 상호 연결망이다 ($2 \leq i \leq n$).

위의 정의에 의해 하이퍼-스타 그래프 $HS(n,k)$ 의 노드는 k 개의 "1"과 $(n-k)$ 개의 "0"에 대한 중복순열로 표현할 수 있으므로 노드 개수는 $n!/k!(n-k)!$ 이고, 이것은 nC_k 로 쓸 수 있다. 또한 분지수는 노드의 가장 좌측 심볼이 "1"이면 $n-k$ 이고, 심볼이 "0"이면 k 이다. 따라서 하이퍼-스타 그래프 $HS(n,k)$ 는 $k=n/2$ 이면 정규 연결망이고, 그렇지 않으면 비정규 연결망이다. 그림 1에 위의 정의에 따라 구성된 $HS(4,1)$ 과 $HS(4,2)$ 가 있다.

하이퍼-스타 그래프 $HS(n,k)$ 는 노드를 나타내는 비트 스트링 $s_1s_2...s_i...s_n$ 에서 심볼 $s_1="0"$ 와 $s_1="1"$ 을 갖는 노드들로 나눌 수 있으므로 이분할 그래프(bipartite graph)이다. 따라서 $HS(n,k)$ 는 짝수 길이의 싸이클만 존재한다.



(그림 1) $HS(4,1)$ 과 $HS(4,2)$ 그래프
 (Fig. 1) $HS(4,1)$ and $HS(4,2)$ graph

$HS(n,k)$ 의 비트 스트링 $s_1s_2...s_i...s_n$ 에서 심볼 s_i 과 s_j 가 보수이면서 교환된 노드의 비트 스트링 $s_1s_2...s_i...s_n$ 을 연결하는 에지를 i -차원 에지라 하자. 예를들면 $HS(4,2)$ 그래프에서 노드 심볼이 0011인 노드를 v 라 하고, 1010인 노드를 w 라 할때, 노드 v 와 w 는 $HS(4,2)$ 그래프의 에지 정의에 의해 연결되어 있고, 노드 v 와 w 를 연결하는 에지를 4-차원 에지라 한다.

2.2 하이퍼-스타 그래프의 확장성

상호 연결망의 확장성이란 노드의 개수가 적은 연결망을 이용하여 노드 개수가 많은 연결망을 쉽게 구성할 수 있음을 의미한다. 하이퍼-스타 그래프는 $HS(n-1, k-1)$ 과 $HS(n-1, k)$ 를 통하여 $HS(n, k)$ 를 생성할 수 있으며, 생성 과정은 아래와 같다($n > 2, 2 \leq k \leq \lfloor n/2 \rfloor$).

하이퍼-스타 그래프 $HS(n-1, k-1)$ 의 노드를 나타내는 비트 스트링을 $s_1s_2...s_{n-1}$ 이라 할 때, 심볼 $s_1="1"$ 인 노드 개수는 ${}_{n-1}C_{k-1}$ 개이고, 이들의 분지수는 $n-k$ 이다. 또한, 심볼 $s_1="0"$ 인 노드는 모두 ${}_{n-1}C_{k-1}$ 개이고, 노드의 분지수는 $k-1$ 이다. $HS(n-1, k)$ 의 노드를 나타내는 비트 스트링을 $s'_1s'_2...s'_n$ 이라 할 때, 심볼 $s'_1="1"$ 인 노드 개수는 모두 ${}_{n-1}C_k$ 개이고, 이들의 분지수는 $n-(k+1)$ 이다. 또한, 심볼 $s'_1="0"$ 인 노드는 모두 ${}_{n-1}C_k$ 개이고, 노드들의 분지수는 k 이다. $HS(n-1, k-1)$ 와 $HS(n-1, k)$ 를 연결하는 방법은 두 그래프의 비트 스트링에서 $s_1 = \overline{s'_1}$ 이고, $s_2...s_{n-1} = s'_2...s'_{n-1}$ 인 노드들을 서로 연결한다. 연결된 노드들에는 새로운 비트 s_n 과 s'_n 를 각각 추가한다. 이때 추가된 심볼은 $s_n="1"$ 과 $s'_n="0"$ 이다. 왜냐하면 $HS(n, k)$ 의 n 개 비트 스트링에서 심볼 "1"을 갖는 것이 k 개이므로, $HS(n-1, k-1)$ 에 추가되는 1 비트는 "1"이고, $HS(n-1, k)$ 에 추가되는 1 비트는 "0"이어야 한다. $HS(n-1, k-1)$ 의 노드에서 $s_1="0"$ 인 노드와 $HS(n-1, k)$ 의 노드에서 $s'_1="1"$ 인 노드는 각각 ${}_{n-1}C_{k-1}$ 개이다. $HS(n-1, k-1)$ 에서 가장 우측 심볼 $s_1="0"$ 인 노드들의 비트 스트링 $s_2...s_{n-1}$ 은 $n-2$ 개 비트에서 $k-1$ 개의 비트가 "1"로 구성된 모든 경우의 수 이고, $HS(n-1, k)$ 에서

가장 우측 심볼 $s'_1="1"$ 인 노드들의 비트 스트링 $s'_2...s'_n$ 은 $n-2$ 개 비트에서 $k-1$ 개의 비트가 "1"로 구성된 모든 경우의 수 이다. 그러므로 $HS(n-1, k-1)$ 와 $HS(n-1, k)$ 에서 $s_1 = \overline{s'_1}$ 이고, $s_2...s_{n-1} = s'_2...s'_{n-1}$ 인 노드들은 일-대-일로 연결되어 있으며, 연결된 각 노드들의 분지수는 k 임을 알 수 있다. $HS(n-1, k-1)$ 와 $HS(n-1, k)$ 에서 $s_1="1"$ 인 노드들과 $s'_1="0"$ 인 노드의 분지수는 k 이고, 이들 노드의 비트 스트링에 $s_n="1"$ 과 $s'_n="0"$ 비트를 추가한다. $HS(n-1, k-1)$ 와 $HS(n-1, k)$ 를 연결하여 생성한 그래프의 분지수는 k 이고, 노드수는 ${}_{n-1}C_{k-1} + {}_{n-1}C_k = {}_n C_k$ 개의 노드를 가지므로 $HS(n, k)$ 임을 알 수 있다. 그림 2는 $HS(5, 2)$ 와 $HS(5, 3)$ 을 연결하여 $HS(6, 3)$ 을 생성한 예이다.

정리 1. 하이퍼-스타 그래프 $HS(n, k)$ 는 $HS(n, n-k)$ 그래프와 동형(isomorphic)이다.

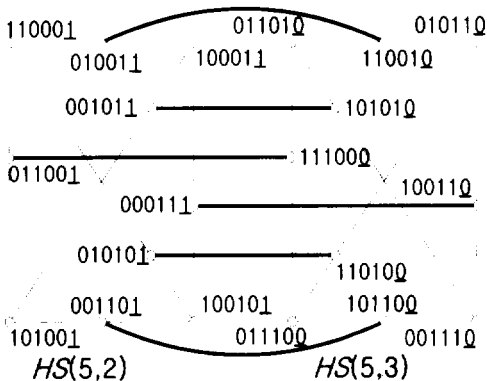
증명 그래프 G 와 H 가 동형이라는 것은 그래프 G 를 구성하는 노드의 비트 스트링을 리레이블링하여 그래프 H 를 생성할 수 있음을 의미한다. 따라서 본 증명에서는 $HS(n, k)$ 의 노드를 나타내는 이진수 "0"과 "1"의 개수를 이용하여 $HS(n, n-k)$ 의 노드를 나타낼 수 있음을 보인다.

$HS(n, k)$ 의 노드는 이진수 "0" 또는 "1"로 구성된 n 개의 비트 스트링으로 나타내고, n 개 비트에서 k 개의 비트가 "1"이고, $n-k$ 개의 비트가 "0"으로 표현된다. $HS(n, k-1)$ 의 노드는 n 개의 비트 스트링에서 $n-k$ 개의 비트가 "1"이고, k 개의 비트가 "0"으로 구성된다. $HS(n, k)$ 의 노드를 나타내는 비트 스트링을 모두 보수로 바꾸면 n 개 비트에서 "1"인 비트는 $n-k$ 개이고, "0"인 비트는 k 개를 가지므로 $HS(n, n-k)$ 의 노드 구성과 동일함을 쉽게 알 수 있다. □

$HS(n, k)$ 는 $HS(n, n-k)$ 와 동형이므로 본 논문에서 특별한 언급이 없으면 $k \leq n/2$ 인 경우이다.

2.3 노드 연결도

주어진 그래프에서 임의의 $k-1$ 개 이하의 노드가 제거되더라도 그래프가 연결되어 있고 적절한 k 개의 노드가 제거되었을 때 분리되면 그 그래프의 연결도를 k 라 한다. 노드 연결도와 분지수가 같은 연결망을 최고 장애 허용도(maximally fault tolerant)를 가졌다고 한다[1]. 이와 유사하게 에지 연결도도 정의할 수 있다. 상호 연결망에서 연결도와 에지 연결도는 각각의 노드



(그림 2) $HS(5, 2)$ 와 $HS(5, 3)$ 에 의한 $HS(6, 3)$ 구성
(Fig. 2) $HS(6, 3)$ generated by connecting $HS(5, 2)$ with $HS(5, 3)$

나 통신 링크에 고장이 발생하더라도 연결되어 있어서 계속 동작이 가능한가를 평가하는 기준이 된다. 그래프 G 의 연결도, 에지 연결도, 분지수는 각각 $\kappa(G)$, $\lambda(G)$, 그리고 $\delta(G)$ 로 하고, $\kappa(G) \leq \lambda(G) \leq \delta(G)$ 인 사실이 알려져 있다[10]. $HS(n,k)$ 의 분지수는 k 에 의해 결정되므로 $k=n/2$ 인 정규 연결망인 경우 $\kappa(HS(n,n/2)) = \lambda(HS(n,n/2)) = \delta(HS(n,n/2))$ 임을 통하여 $HS(n,n/2)$ 가 최대 고장 허용도를 가짐을 보인다.

정리 2. $\kappa(HS(n,n/2))=n/2$.

증명 $HS(n,n/2)$ 에서 $n/2 - 1$ 개의 노드를 제거해도 $HS(n,n/2)$ 가 분할되지 않음을 보인다. $HS(n,n/2)$ 의 노드를 $s(s_1s_2...s_{n/2})$ 라 할 때, $HS(n,n/2)$ 는 이분할 그래프이므로 노드의 비트 스트링에 따라서 두개의 부류 즉, 노드의 첫 번째 비트 스트링 $s_1="1"$ 인 노드와 $s_1="0"$ 인 노드로 나눌 수 있다. $HS(n,n/2)$ 의 노드 s 의 첫 번째 비트 스트링 $s_1="1"$ 인 노드들은 $s_1="0"$ 을 갖는 $n/2$ 개의 노드들과 연결되어 있고, 첫 번째 비트 스트링 $s_1="1"$ 인 노드들은 $s_1="0"$ 을 갖는 $n/2$ 개의 노드들과 연결되어 있다.

$HS(n,n/2)$ 에서 제거할 $n/2 - 1$ 개의 노드 집합을 P 이라 할 때, P 의 위치에 따라서 2가지로 나누어 $HS(n,n/2) - P$ 가 항상 연결된 그래프임을 보인다.

첫째, P 의 노드들이 $s_1="0"$ 인 경우, $HS(n,n/2)$ 의 노드 s 에 인접한 $n/2 - 1$ 개의 노드가 P 와 동일해도 적어도 한 개 노드는 $s_1="1"$ 인 노드와 연결되어 있음을 쉽게 알 수 있다. 만약 P 의 노드가 한 개의 노드와 인접하지 않고 분산된 경우에도 $HS(n,n/2) - P$ 가 항상 연결된 그래프임을 쉽게 알 수 있다. 또한 P 의 노드들이 $s_1="1"$ 인 경우도 동일하게 적용할 수 있다. 둘째, P 의 노드들이 $s_1="1"$ 인 경우와 $s_1="0"$ 인 경우로 분산된 경우, $HS(n,n/2)$ 의 모든 노드는 $n/2$ 개의 인접한 노드들이 있고, 이 노드들은 첫 번째 비트 스트링이 서로 보수 관계를 갖는 노드와 연결되어 있으므로, $HS(n,n/2)$ 에서 제거할 $n/2 - 1$ 개의 노드들이 $s_1="1"$ 인 경우와 $s_1="0"$ 인 경우로 분산되는 경우 $HS(n,n/2)$ 가 항상 연결된 그래프임을 쉽게 알 수 있다.

따라서 $HS(n,n/2)$ 에서 어떤 위치에 있는 노드 $n/2 - 1$ 개를 제거해도 $HS(n,n/2)$ 는 항상 연결되어 있으므로 $\kappa(HS(n,n/2)) \geq n/2$ 이고, $HS(n,n/2)$ 는 분지수가 $n/2$ 인 정규 그래프이므로 $\kappa(HS(n,n/2)) \leq n/2$ 이다. 그러므로 $\kappa(HS(n,n/2))=n/2$ 이다. □

3. 라우팅(routing) 알고리즘과 지름(diameter)

본 장에서는 하이퍼-스타 그래프에서의 라우팅 알고리즘을 제안하고 지름(diameter)을 분석한다.

3.1 라우팅 알고리즘

하이퍼-스타 그래프 $HS(n,k)$ 의 노드 s 와 d 에서 s 를 원시 노드라 하고 d 를 목적 노드라 할 때, 노드 s 에서 d 까지 메시지 전송을 위한 최단 경로는 s 의 비트 스트링을 d 의 비트 스트링으로 변환하는 과정으로 나타낼 수 있다. 최단 경로 설정을 위한 라우팅 알고리즘의 개요는 다음과 같다.

노드 s 의 비트 스트링을 $s_1s_2...s_n$ 이라 하고, 노드 d 의 비트 스트링을 $d_1d_2...d_n$ 이라 할 때, 노드 s 와 d 의 비트 스트링을 Exclusive-OR(심볼 \otimes)한 결과를 $r_1r_2...r_n$ 이라 하자. Exclusive-OR 성질에 의하여 심볼 $r_i="0"$ 은 노드 s 와 d 의 비트 스트링에서 i 번째 위치의 심볼이 서로 같음을 의미하고, $r_i="1"$ 은 i 번째 위치의 심볼이 서로 보수임을 의미한다($2 \leq i \leq n$). 따라서 노드 s 와 d 의 비트 스트링을 Exclusive-OR한 비트 스트링 $r_1r_2...r_n$ 에서 $r_i="1"$ 을 갖는 i 번째 위치의 심볼이 서로 같아지도록 하는 i -차원 에지의 패스가 라우팅 경로이다($2 \leq i \leq n$). 위와 같은 방법에 의한 최단 경로를 구하는 알고리즘은 다음의 Shortest_path() 함수와 같다. 이 함수에서 P_s 는 노드 s 에서 d 로 향하는 최단 경로상의 노드들의 집합을 의미하며 초기 값으로 $P_s=\{s\}$ 이다.

라우팅 알고리즘은 먼저 Shortest_path() 함수를 사용하여 원시 노드에서 목적 노드까지의 가장 짧은 경로를 구하고 이 경로를 따라 메시지를 전송하면 된다.

```

Shortest_path_1(s,d,P_s)
{
    if( s==d ) { P_s; return; }
    s'=s와 i-차원 에지에 의해 연결된 노드
    r(r_1r_2...r_i...r_n)=s(s_1s_2...s_{n/2}) ⊗ d(d_1d_2...d_{n/2})
    if( |r_i|=1 > 0 )
        if( r_i=1 ) then { P_s|s: s=s'; }
    Shortest_path_1(s,d,P_s);
}
    
```

(그림3) 최단 경로 라우팅 알고리즘
(Fig. 3) The shortest path routing algorithm

위의 라우팅 알고리즘에서 원시 노드와 목적 노드를 Exclusive-OR하여 생성한 비트 스트링 $r_1 r_2 \dots r_n$ 에서 $r_i=1$ 을 갖는 i 의 위치는 원시 노드와 목적 노드의 비트 스트링이 서로 보수임을 의미한다. 원시 노드로부터 $r_i=1$ 을 갖는 i -차원 에지를 지날 때마다 원시 노드와 목적 노드의 i 번째 심볼이 동일하게 되어 $r_i=1$ 을 갖는 i 의 개수가 하나씩 감소하므로 본 논문에서 제시하는 알고리즘은 최적 라우팅 알고리즘이다.

예를 들어 $HS(6,3)$ 에서 원시 노드 $s(000111)$ 에서 목적 노드 $d(011001)$ 까지 라우팅 경로 설정을 위해 최단 경로 알고리즘을 적용하면 다음과 같다. 먼저 노드 s 와 d 를 Exclusive-OR ($000111 \oplus 011001$)하여 $r_1 r_2 \dots r_n (= 011110)$ 을 구한다. 심볼 r_i 에서 "1"을 갖는 위치 $i=(2, 3, 4, 5)$ 이다. 노드 s 와 연결된 에지는 4, 5, 6-차원 에지이므로, 노드 s 로부터 시작하여 4, 2, 5, 3-차원 에지를 각각 지나는 라우팅 경로 000111 $\xrightarrow{4}$ 100011 $\xrightarrow{2}$ 010011 $\xrightarrow{5}$ 110001 $\xrightarrow{3}$ 011001을 설정할 수 있다.

3.2 지름

지름은 메시지가 어떤 노드에서 다른 노드까지 최단 경로로 전송될 때 지나는 통신 링크의 최대 개수를 나타낸다.

정리 3. 하이퍼-스타 그래프 $HS(n,k)$ 의 지름은 $n-1$ 이다.

증명 하이퍼-스타 그래프 $HS(n,k)$ 의 지름 $dia(HS(n, k))$ 는 최단 경로 알고리즘에 의해 비트 스트링 r_i 가 "1"을 갖는 가장 많은 i 의 개수와 같다($2 \leq i \leq n$). $HS(n, k)$ 는 k 의 조건에 따라 지름을 분류할 수 있다.

- (1) $k=1, \quad dia(HS(n,1))=2$
- (2) $k=n/2, \quad dia(HS(n,n/2))=n-1$
- (3) $k < n/2, \quad dia(HS(n,k))=2k$

경우1. $k=1$

하이퍼-스타 그래프 $HS(n,k)$ 에서 $k=1$ 인 모든 그래프는 스타 형태의 구조를 가지므로 지름은 2이다.

경우2. $k=n/2$

$k=n/2$ 인 정규 연결망에서 두 노드의 비트 스트링을 Exclusive-OR한 결과 비트 스트링 r_i 에서 r_i 가 "1"을 갖는 가장 많은 비트의 개수는 n 개이지만 $n-1$ 개의 i -차원 에지를 지나면 두 노드의 심볼이 같아지므로 지름은 $n-1$ 이다.

경우3. $k < n/2$

$k < n/2$ 인 비정규 연결망에서 r_i 가 "1"을 갖는 가장 많은 i 의 개수는 $2k$ 개로 지름은 $2k$ 이다. 이때 k 의 조건은 $k < n/2$ 이므로 경우2의 지름보다 크지는 않다.

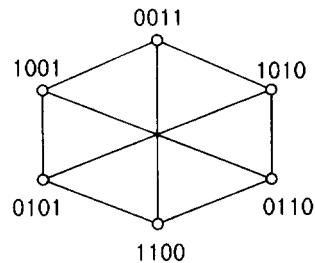
따라서 위의 3가지 경우에 의해 하이퍼-스타 그래프 $HS(n,k)$ 의 지름은 $n-1$ 이다. □

4. Folded 하이퍼-스타 그래프

본 장에서는 하이퍼-스타 그래프 $HS(n,k)$ 의 지름을 개선하기 위해 한개의 에지를 추가한 folded 하이퍼-스타 그래프 $FHS(n,k)$ 를 정의하고 성질을 분석한다. 하이퍼-스타 그래프 $HS(n,k)$ 는 k 에 따라서 다양한 분지수를 가지므로 본 논문에서는 $n=2k$ 인 정규 연결망에 대하여 한개의 에지를 추가하는 방법을 제시한다.

4.1 Folded 하이퍼-스타 그래프의 정의 및 성질

하이퍼-스타 그래프 $HS(n,k)$ 에서 $n=2k$ 으로 정규 연결망인 $HS(n,n/2)$ 에서 노드의 심볼이 보수관계인 노드들간에 한개의 에지를 추가하여 생성한 그래프를 folded 하이퍼-스타 그래프 $FHS(n,n/2)$ 로 정의한다. 그림4에 folded 하이퍼-스타 그래프 $FHS(4,2)$ 의 예가 있다.



(그림 4) $FHS(4,2)$ 그래프
(Fig. 4) $FHS(4,2)$ graph

Folded 하이퍼-스타 그래프 $FHS(n,n/2)$ 는 $HS(n,n/2)$ 에서 노드의 심볼이 보수관계인 노드간에 에지가 한개 추가 되었으므로 이분할 그래프이고, 노드 개수는 하이퍼-스타 그래프 $HS(n,n/2)$ 과 동일한 개수를 가지며, 분지수는 $HS(n,n/2)$ 의 분지수 보다 한개 더 많다. 하이퍼-스타 그래프 $HS(n,k)$ 에서 지름은 $n=2k$ 인 정규 연결망에서 노드의 심볼이 보수관계일때 $n-1$ 이지만, folded 하이퍼-스타 그래프 $FHS(n,n/2)$ 는 노드의 심볼

이 보수관계인 노드들간에 에지가 존재하므로 folded 하이퍼-스타 그래프 $FHS(n, n/2)$ 의 지름은 하이퍼-스타 그래프 $HS(n, k)$ 의 지름을 절반으로 줄인 $n/2$ 임을 쉽게 알 수 있다.

4.2 Folded 하이퍼-스타 그래프의 라우팅 알고리즘

Folded 하이퍼-스타 그래프 $FHS(n, n/2)$ 에서 라우팅은 하이퍼-스타 그래프의 라우팅과 유사하다. 단 노드의 심볼이 보수관계인 노드를 연결하는 에지를 효율적으로 이용하기 위하여 원시 노드와 목적 노드의 비트 스트링을 Exclusive-OR한 결과 비트 스트링에서 "1"의 개수가 $n/2$ 보다 큰 경우 노드의 심볼이 보수 관계를 갖는 노드를 먼저 라우팅 하도록하는 최단 경로 라우팅 알고리즘이다.

Shortest_path_2(s,d,P_s)

```

{
  if( s==d ) { Ps ; return; }
  s'=s와 i-차원(또는 c-차원) 에지에 의해 연결된 노드;
  r(r1r2...ri...rn)=s(s1s2...si...sn) ⊕ d(d1d2...di...dn);
  if(|ri=1| > n/2)
    { Ps|s; s=s'; } /* s'는 c-차원 에지로 연결 */
  if(|ri=1| > 0)
    if( ri=1 ) then { Ps|s; s=s'; } /* s'는 i-차원
    에지로 연결 */
  Shortest_path_2(s,d,Ps);
}
    
```

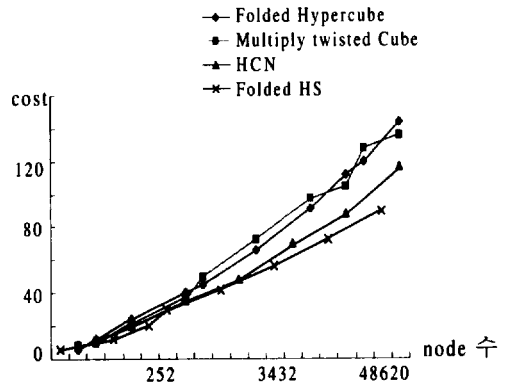
(그림 5) 최단 경로 라우팅 알고리즘
(Fig. 5) The shortest path routing algorithm

상호 연결망을 평가하는 망 척도중 하드웨어의 비용과 관련된 분지수와 그 상호 연결망에서 메시지 전송시간과 관련된 지름은 상호간에 상관 관계를 갖고 있다. 이러한 특성 때문에 망 비용[2,12,13]은 분지수 × 지름이 작은값 즉, 분지수는 작고 지름은 짧은 값을 가지면서 많은 수의 노드를 갖는 상호 연결망이 좋은 연결망이라 하겠다. 본 논문에서 제시하는 folded 하이퍼-스타 그래프 $FHS(2n, n)$ 는 분지수는 $n+1$ 이고 지름이 n 인 상호 연결망으로 분지수 × 지름은 n^2+n 이지만, 노드 개수는 $2n$ 개의 원소에서 n 개를 선택하는 조합의 가지수 즉, ${}_2nC_n$ 개 노드를 갖게되므로 같은 개수의 노드를 갖는 하이퍼큐브와 그 변형된 그래프들보

다 연결망의 비용에서 우수한 결과를 갖는다. 그림6에서는 본 논문에서 제안하는 folded 하이퍼-스타 그래프 $FHS(2n, n)$ 와 하이퍼큐브 및 그 변형된 연결망들을 노드 개수, 분지수, 지름 및 망 비용을 비교하였고, 표2에서는 연결망들이 같은 개수의 노드를 가질때 망 비용을 비교 분석하였다.

<표 2> 하이퍼큐브와 그 변형의 망 비용
<Table 2> Network cost of hypercube and variation of it

	노드 개수	분지수	지름	연결망 비용
Hypercube	2^n	n	n	n^2
Folded Hypercube	2^n	$n+1$	$\lceil \frac{n}{2} \rceil$	$\frac{n^2}{2}$
Multiply-twisted cube	2^n	n	$\lceil \frac{n+1}{2} \rceil$	$\frac{n^2}{2}$
HCN(n,n)	2^{2n}	$n+1$	$n + \lceil \frac{n}{2} \rceil + 1$	$\frac{3n^2}{2}$
Folded $HS(2n, n)$	${}_2nC_n$	n	n	n^2



(그림 6) 연결망의 노드수와 비용간의 관계
(Fig. 6) Relation of Between the number of nodes and cost of networks

5. 임베딩(Embedding)

그래프의 임베딩은 어떤 그래프가 다른 그래프 구조에 포함 혹은 어떻게 연관되어 있는지를 알아보기 위해 어떤 특정한 그래프를 다른 그래프에 사상(mapping)하는 것이다. 연결망 G가 연결망 H에 적은 비용으로 임베딩 가능 하다는것은 연결망 G에서 개발된 알고리즘을 연결망 H에서 적은 비용으로 효율적으로 이

용할 수 있기 때문에 상호 연결망에서 임베딩 문제는 매우 중요하다[6,9,10].

그래프 G 의 그래프 H 에 대한 임베딩 $f=(\varphi, \rho)$ 는 다음과 같이 정의한다. φ 는 G 의 정점 집합 $V(G)$ 를 H 의 정점 집합 $V(H)$ 에 대응시키는 함수이고, ρ 는 G 의 에지 $e=(v,w)$ 를 $\varphi(v)$ 와 $\varphi(w)$ 를 잇는 H 상의 경로로 대응시키는 함수이다. 이때 G 를 게스트(guest) 그래프라 하고 H 를 호스트(host) 그래프라 부른다. φ 가 일대일 함수인 경우는 일-대-일 임베딩이라 하고 다대일 함수인 경우는 다-대-일 임베딩이라 한다. 본 논문에서는 일-대-일 임베딩을 다루도록 한다. 임베딩의 비용을 나타내는 척도로 연장률(dilation), 밀집률(congestion), 확장률(expansion) 등이 있다. G 의 에지 e 의 연장률은 H 상에서의 경로 $\rho(e)$ 의 길이를 말하고, 임베딩 f 의 연장률은 G 의 모든 에지에 대한 연장률중 최대값이다. H 의 에지 e' 의 밀집률은 e' 를 포함하는 $\rho(e)$ 의 개수를 말한다. 임베딩 f 의 밀집률은 H 의 모든 에지의 밀집률의 최대값이다. 임베딩 f 의 확장률은 G 의 정점수에 대한 H 의 정점수의 비, 즉 $|V(H)| / |V(G)|$ 이다.

본 장에서는 하이퍼큐브 Q_n 와 $n \times n$ 토러스 구조를 하이퍼-스타 그래프 $HS(2n,n)$ 에 대한 임베딩을 분석한다. 하이퍼큐브 Q_n 을 임베딩하는 방법은 하이퍼큐브의 노드 레이블링을 확장하여 하이퍼-스타 그래프 $HS(2n, n)$ 의 비트 스트링으로 표현할 수 있음을 보이고, 하이퍼큐브 Q_n 의 에지에 의해 연결된 노드들의 비트 스트링을 통해서 연장률을 알아본다.

정리 4. 하이퍼큐브 Q_n 은 하이퍼-스타 그래프 $HS(2n,n)$ 에 연장률 2, 확장률 $2nC_n / 2^n$ 에 임베딩 가능하다.

증명 하이퍼큐브 Q_n 의 노드는 n 개의 비트 스트링 $s_1s_2...s_n$ 으로 구성되고, 에지는 노드의 심볼에서 오직 1비트 다른 노드 사이에만 존재한다. 하이퍼큐브 Q_n 에서 노드의 비트 스트링을 $2n$ 으로 확장하기 위해 하이퍼큐브의 노드 비트 스트링 $s_1s_2...s_n$ 의 보수를 $s_{n+1}s_{n+2}...s_{2n}$ 이라 하고, 노드의 레이블링이 $2n$ 으로 확장된 하이퍼큐브의 비트 스트링을 $s_1s_2...s_ns_{n+1}s_{n+2}...s_{2n}$ 이라 하자. 노드의 스트링이 확장된 하이퍼큐브 Q_n 의 비트 스트링에서 "0"과 "1"의 개수는 n 개씩이다. 왜냐하면 확장된 비트 스트링 $s_{n+1}s_{n+2}...s_{2n}$ 은 하이퍼큐브 Q_n 의 비트 스트링 $s_1s_2...s_n$ 과 보수이기 때문이다. $2n$ 비트로 확장된 하이퍼큐브 Q_n 의 노드들은 이제 2 비트 다른 노드들과

연결되어 있고, 확장된 하이퍼큐브 노드의 비트 스트링은 하이퍼-스타 그래프 $HS(2n,n)$ 의 비트 스트링의 일부분임을 쉽게 알 수 있다.

n 개의 비트 스트링으로 구성된 하이퍼큐브 Q_n 의 임의의 노드 v 와 w 가 에지로 연결되어 있고, 두 노드의 비트 스트링에서 i 번째 위치의 심볼이 서로 보수라할 때, $2n$ 비트로 확장된 하이퍼큐브 Q_n 에서 노드 v 와 w 는 연결되어 있고, 노드의 비트 스트링에서 i 와 $n+i$ 번째 위치의 심볼이 서로 보수관계임을 알 수 있다. 하이퍼큐브에서 연결된 노드 v 와 w 를 $HS(2n,n)$ 그래프의 노드로 사상하는 문제는 $HS(2n,n)$ 그래프에서 하이퍼큐브의 노드 v 와 w 의 최단 경로 즉, i 와 $(n+i)$ -차원 에지를 지나는 경로를 설정하는 문제와 같다.

따라서 하이퍼큐브 Q_n 의 비트 스트링을 $2n$ 비트로 확장했을 때 모든 에지는 하이퍼-스타 그래프 $HS(2n,n)$ 에서 경로 길이 2인 패스와 동일하므로 연장률 2, 확장률 $2nC_n / 2^n$ 에 임베딩 가능함을 알 수 있다. □

토러스와 메쉬[6] 구조는 이미지 처리 또는 대수 계산(algebra computation) 문제에 효율적으로 이용될 수 있는 상호 연결망이다[13]. 메쉬 구조의 행과 열 각각의 첫 번째와 마지막 노드들을 연결하여 생성한 구조가 토러스이므로 메쉬 구조는 토러스의 서브그래프이다. 본 논문에서는 $n \times n$ 토러스 구조를 $HS(2n,n)$ 그래프에 임베딩 가능함을 보인다. 토러스 구조를 임베딩하는 방법은 토러스의 노드 위치를 행과 열의 쌍으로 구성된 좌표값으로 할당하고, 그 값을 이진수로 변환하여 하이퍼-스타 그래프 $HS(2n,n)$ 의 노드 레이블링을 생성할 수 있음을 보인다. 그리고 토러스의 인접한 노드간의 레이블링을 통하여 임베딩 비용을 분석한다.

정리 5. $n \times n$ 토러스를 하이퍼-스타 그래프 $HS(2n,n)$ 에 연장률 2, 확장률 $2nC_n / n^2$ 에 임베딩 가능하다.

증명 격자 구조에서 행과 열 각각의 양끝 노드들을 연결한 토러스 구조의 노드는 행과 열의 좌표로 표현할 수 있다. $n \times n$ 토러스의 행의 좌표는 $2^0, 2^1, 2^2, \dots, 2^{n-1}$ 으로, 열의 좌표는 $(2^n-1)-2^{n-1}, (2^n-1)-2^{n-2}, (2^n-1)-2^{n-3}, \dots, (2^n-1)-2^0$ 으로 나타낼 수 있다. $n \times n$ 토러스에서 첫 번째 행에 위치한 모든 노드들 즉, 1행 1열부터 1행 n 열까지 위치한 노드의 좌표는 $(2^0, 2^n-2^{n-1}-1), (2^0, 2^n-2^{n-2}-1), \dots, (2^0, 2^n-2^{n-n}-1)$ 이고, 첫 번째 열에 위치한 모든 노드들 즉, 1행 1열, 2행 1열, ..., n 행 1열에 위치한 노드의 좌표는 $(2^0, 2^n-2^{n-1}-1), (2^1, 2^n-2^{n-1}-1), \dots,$

$(2^n - 1, 2^n - 2^{n-1} - 1)$ 이다. 토러스에서 같은 행에 있는 노드들의 행 좌표 값은 모두 동일하고 열 좌표 값이 다르다. 1열부터 n 열까지 좌표값은 $(2^n - 1) - 2^{n-1}, (2^n - 1) - 2^{n-2}, (2^n - 1) - 2^{n-3}, \dots, (2^n - 1) - 2^n$ 이므로, $(2^n - 1) - (2^{n-k})$ 로 나타낼 수 있다($1 \leq k \leq n$). 이러한 열 좌표값을 이진수 "0" 또는 "1"로 표현했을 때, n 개의 비트 $s_{n-1}s_{n-2} \dots s_{2n}$ 에서 오직 k 번째 비트만 "0"을 갖는다. 따라서 동일한 행에 존재하는 모든 노드들의 열 좌표값을 $s_{n-1}s_{n-2} \dots s_{2n}$ 로 나타낼 때, 임의의 두 노드는 정확히 2 비트만 서로 보수 관계임을 알 수 있다. 행의 경우와 동일한 방식으로, 같은 열에 존재하는 노드들의 열 좌표값은 모두 같고, 행 좌표값은 $2^0, 2^1, 2^2, \dots, 2^{n-1}$ 이므로 2^k 로 나타낼 수 있다($0 \leq k \leq n-1$). 이러한 행 좌표값을 n 개의 비트 스트링 $s_1s_2 \dots s_n$ 으로 표현하면 $n-k$ 번째 비트만 "1"로 표현되므로, 동일한 열에 위치하는 임의의 두 노드는 정확히 2 비트만 보수 관계임을 알 수 있다. 토러스에서 노드 레이블링 방법은 노드의 위치를 나타내는 행과 열의 비트 스트링 $s_1s_2 \dots s_n$ 과 $s_{n-1}s_{n-2} \dots s_{2n}$ 를 연결하여 비트 스트링 $s_1s_2 \dots s_{n-1}s_{n-2} \dots s_{2n}$ 을 생성한다. 이러한 방법으로 레이블링된 노드의 $2n$ 비트 스트링 $s_1s_2 \dots s_{n-1} s_{n-2} \dots s_{2n}$ 에서 "1"과 "0"이 각각 n 비트임을 알 수 있다. 토러스에서 노드를 연결하는 에지는 열과 행에 있는 에지이다. 행에 있는 에지에 의해 연결된 노드들은 비트 스트링 $s_1s_2 \dots s_{n-1}s_{n-2} \dots s_{2n}$ 에서 전위 n 비트 $s_1s_2 \dots s_n$ 는 모두 동일하고, $s_{n-1}s_{n-2} \dots s_{2n}$ 에서 k 번째 비트만 "0"이고, 나머지 $n-1$ 비트는 "1"을 가지므로 행에 있는 임의의 두 노드는 2 비트만 보수 관계이다. 열에 있는 에지에 의해 연결된 노드들은 비트 스트링 $s_1s_2 \dots s_{n-1} s_{n-2} \dots s_{2n}$ 에서 후위 n 비트 $s_{n-1}s_{n-2} \dots s_{2n}$ 는 모두 동일하고, 전위 n 비트 $s_1s_2 \dots s_n$ 에서 $n-k$ 번째 비트만 "1"이고, 나머지 $n-1$ 비트는 "0"을 가지므로 열에 있는 임의의 두 노드는 2 비트만 보수 관계이다.

따라서 $n \times n$ 토러스 구조의 노드를 $2n$ 비트의 노드 레이블링을 갖도록 확장하여 하이퍼-스타 그래프 $HS(2n, n)$ 에 연장비율 2, 확장비율 $2^n C_n / n^2$ 로 임베딩 가능함을 알 수 있다. □

6. 결 론

본 논문에서는 상호 연결망의 망 비용 즉, 분지수 \times 지름 값이 하이퍼큐브보다 작은 값을 갖는 새로운 상호 연결망으로 하이퍼-스타 그래프 $HS(n, k)$ 를 제안

하였다. $HS(n, k)$ 는 $n C_k$ 개의 노드를 갖는 상호 연결망으로 k 에 따라서 $k=n/2$ 인 정규 연결망과 $k \neq n/2$ 인 비정규 연결망으로 나눌 수 있다. 본 논문에서는 $HS(n, k)$ 의 여러가지 망 척도를 분석하였으며, 특히 $k=n/2$ 인 정규 연결망에서 노드의 심볼이 보수 관계를 갖는 노드들간에 에지를 한개 추가하여 folded 하이퍼-스타 그래프 $FHS(n, k)$ 를 정의하고 성질을 분석하였다. Folded 하이퍼-스타 그래프 $FHS(2n, n)$ 은 $2^n C_n$ 개의 노드를 갖고, 분지수 $n+1$ 과 지름 n 을 갖는 정규 연결망으로 망 비용은 n^2+n 이다. 이 결과는 folded 하이퍼-스타 그래프 $FHS(2n, n)$ 이 널리 알려진 하이퍼큐브 또는 그 변형들과 같은 수의 노드를 가질때 지름 \times 분지수에 있어서 아주 작은 값을 가짐을 의미한다. 또한 그래프의 임베딩 문제에 있어서 하이퍼큐브 Q_n 과 $n \times n$ 토러스 구조를 하이퍼-스타 그래프 $HS(2n, n)$ 에 임베딩하였다. 하이퍼큐브 Q_n 의 임베딩은 비트 스트링을 확장하는 방법을 이용하여 $HS(2n, n)$ 그래프에 연장비율 2, 확장비율 $2^n C_n / 2^n$ 에 임베딩 가능함을 보였고, $n \times n$ 토러스의 임베딩은 노드 위치를 나타내는 행과 열을 이진수로 표현하는 방법을 이용하여 $HS(2n, n)$ 그래프에 연장비율 2, 확장비율 $2^n C_n / n^2$ 에 임베딩 가능함을 보였다.

위의 결과는 정규 연결망을 갖는 folded 하이퍼-스타 그래프가 연결망의 비용에 있어서 하이퍼큐브와 그 변형된 그래프보다 우수하고, 하이퍼큐브에서 개발된 알고리즘을 적은 비용으로 효율적으로 이용할 수 있음을 의미한다.

참 고 문 헌

- [1] S. B. Akers, D. Harel, and B. Krishnamurthy, "The Star Graph: An Attractive Alternative to the n-Cube," Proc. Int'l. Conf. on Parallel Processing, pp.393-400, Aug. 1987.
- [2] L. N. Bhuyan, and D. P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," IEEE Trans. Comput., Vol.c-33, No.4, pp.323-333, Apr. 1984.
- [3] M. S. Chen, K. G. Shin and D. D. Kandlur, "Addressing, Routing and Broadcasting in Hexagonal Mesh Multiprocessors," IEEE Trans. Comput., Vol.39, No.1, pp.10-18, Jan. 1990.

[4] K. Efe, "A Variation on the Hypercube with Lower Diameter," IEEE Trans. Comput., Vol.40, No.11, pp.1312-1316, 1991.

[5] K. Ghose and K. R. Desai, "The Design and Evaluation of the Hierarchical Cubic Network," Proc. Int'l Conf. on Parallel Processing, Vol.1, pp.355-362, 1990.

[6] A. K. Gupta and S. E. Hambrusch, "Multiple Network Embeddings into Hypercubes," J. Parallel and Distributed Computing, Vol.19, pp.73-82, 1993.

[7] J. Kim and K-G. Shin, "Operationally Enhanced Folded Hypercubes," IEEE Trans. on Parallel and Distributed Systems, Vol.5, No.12, pp.1310-1316, Dec. 1994.

[8] S. Lakshminarayanan, Jung-Sing Jwo, and S. K. Dhall, "Symmetry in Interconnection Networks based on Cayley Graphs of Permutation Groups: A Survey," Parallel Computing, Vol.19, pp.361-407, 1993.

[9] F. P. Preparata and J. E. Vuillemin, "The Cube-Connected-Cycles: A Versatile Network for Parallel Computation," Commun. ACM, Vol.24, pp.300-309, 1981.

[10] S. Ranka, J-C. Wang, and N. Yeh, "Embedding Meshes on the Star Graph," J. Parallel and Distributed Computing, Vol.19, pp.131-135, 1993.

[11] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," IEEE Trans. Comput., Vol.37, pp.867-872, 1988.

[12] J. A. Stankovic, "A Perspective on Distributed Computer Systems," IEEE Trans. Comput., Vol. c-33, No.12, pp.1102-1115, December 1984.

[13] Ivan Stojmenovic, "Honeycomb Networks: Topological Properties and Communication Algorithms," IEEE Trans. Parallel and Distributed Systems, Vol.8, No.10, pp.1036-1042, October 1997.

[14] H. S. Stone, "Parallel Processing with the Perfect Shuffle," IEEE Trans. Comput., Vol.c-20, pp.153-161, 1971.

[15] K. W. Tang and S. A. Padubidri, "Diagonal and Toroidal Mesh Networks," IEEE Trans. Comput., Vol.43, No.7, July 1994.



이 형 옥

e-mail : u9698133@chonnam.chonnam.ac.kr

1994년 순천대학교 전산학과 졸업 (학사)

1996년 전남대학교 전산통계학과 졸업(석사)

1996년~1997년 순천대학교 컴퓨터교육과 조교

1996년~1998년 전남대학교 전산통계학과 박사과정 수료
 관심분야 : 병렬 및 분산처리, 그래프이론, 상호연결망, 계산이론



김 병 철

e-mail : kbc@sunchon.sunchon.ac.kr

1967년 서울대학교 수학교육과(학사)

1984년 성균대학교 자료처리학과(석사)

1995년~1997년 순천대학교 전자계산소 소장

현재 순천대학교 컴퓨터과학과 교수
 관심분야 : 이산수학, 시뮬레이션



임 형 석

e-mail : hslim@chonnam.chonnam.ac.kr

1983년 서울대학교 컴퓨터공학과 졸업(학사)

1985년 한국과학기술원 전산학과 졸업(석사)

1993년 한국과학기술원 전산학과 졸업(박사)

현재 전남대학교 전산학과 부교수
 관심분야 : 그래프이론, 암호이론, 병렬 및 분산처리