

꼬인 큐브 토러스: 3차원 꼬인 큐브에 기반한 새로운 토러스 상호연결망

김 종 석[†] · 이 형 옥^{**} · 김 성 원^{***}

요 약

본 논문은 널리 알려진 3차원 꼬인 큐브를 기반으로 하는 새로운 상호연결망 꼬인 큐브 토러스(TT) 네트워크를 제안한다. 꼬인 큐브 토러스 네트워크는 동일한 노드수를 갖는 honeycomb 토러스보다 짧은 지름을 갖고, 망 비용이 개선된 연결망이다. 본 논문에서는 꼬인 큐브 토러스의 라우팅 알고리즘을 제안하고, 지름, 망 비용, 이분할에지수, 해밀토니안 사이클을 분석한다.

키워드 : 상호연결망, 꼬인 큐브, 토러스, 라우팅, 지름, 이분할에지수, 해밀토니안 사이클

Twisted Cube Torus(TT): A New Class of Torus Interconnection Networks Based on 3-Dimensional Twisted Cube

Kim Jongseok[†] · Lee Hyeongok^{**} · Kim Sung Won^{***}

ABSTRACT

We propose a new interconnection network, called Twisted cube torus(TT) network based on well-known 3-dimensional twisted cube. Twisted cube torus network has smaller diameter and improved network cost than honeycomb torus with the same number of nodes. In this paper, we propose routing algorithm of Twisted cube torus network and analyze its diameter, network cost, bisection width and hamiltonian cycle.

Keywords : Interconnection Network, Twisted Cube, Torus, Routing, Diameter, Bisection Width, Hamiltonian Cycle

1. 서 론

대규모 병렬처리 시스템에서 슈퍼컴퓨터의 성능 향상을 위한 연구 분야로 새로운 CPU 개발, 상호연결망, 데이터 교환방식, 새로운 알고리즘 등이 있다. 슈퍼컴퓨터를 사용하기 위해서는 적게는 수십 개에서 많게는 수십~수백만 개의 프로세서를 연결하는 상호연결망(interconnection network)의 역할이 매우 중요하다. 지금까지 병렬처리 시스템을 위한 상호연결망으로 다양한 위상들이 발표되었으며, 대표적인 위상으로는 토러스, 하이퍼큐브, 스타그래프 등이 있다.

상호연결망은 프로세서의 집합, 지역메모리 그리고 프로세서간의 데이터 전송을 위한 통신링크로 구성되어 있다. 상호연결망은 그래프 $G=(V,E)$ 로 모델링될 수 있다. 각각의

프로세서 P_i 는 노드집합 V 의 원소이고 두 개의 프로세서 P_i 와 P_j 는 통신링크 (P_i,P_j) 에 의해 연결되어 있다. 상호연결망을 그래프로 모델링하면 프로세서는 노드(node)로 통신링크는 에지(edge)로 표현된다. 노드 P_i 와 이웃한 노드의 수를 그 노드의 분지수(degree)라고 한다. 상호연결망에서 분지수는 하드웨어적인 비용을 나타낸다.

상호연결망은 망을 구성하는 노드와 에지의 구성에 따라 메쉬 부류, 하이퍼큐브 부류 그리고 스타그래프 부류로 나눌 수 있다. 메쉬 부류는 토러스[5], 허니콤(honeycomb) 메쉬[12], 다이아고날(diagonal) 메쉬[13], 헥사고날(hexagonal) 메쉬[4] 등이 있으며, 하이퍼큐브 부류는 하이퍼큐브[11], 폴디드(folded) 하이퍼큐브[6], 꼬인 큐브(twisted-cube)[1], 재귀원형군[10] 등이 있으며, 스타그래프 부류는 스타그래프[2], 매크로-스타(macro-star)[15], 전위(transposition) 연결망[9], 행렬 스타(matrix star) 연결망[16] 등이 있다.

메쉬 구조는 VLSI 회로설계 분야에서 많이 이용되는 구조로 연결망의 확장이 매우 쉬우므로 현재까지 널리 이용되고 있으며 다양한 시스템으로 상용화되었다. 낮은 차원의

[†] 정 회원 : 영남대학교 정보통신공학과 연구교수
^{**} 종신회원 : 순천대학교 컴퓨터교육과 부교수(교신저자)
^{***} 종신회원 : 영남대학교 정보통신공학과 부교수
 논문접수: 2011년 2월 21일
 수정일: 1차 2011년 4월 12일
 심사완료: 2011년 4월 14일

메쉬는 설계하기 쉽고 알고리즘 관점에서도 매우 유용하므로 병렬처리 컴퓨터의 연결망으로 많이 쓰인다. 높은 차원의 메쉬 일수록 지름이 작아지고 이분할 에지수(bisection width)가 커지며 여러 가지 병렬 알고리즘을 빨리 수행할 수 있지만 비용이 많이 드는 단점이 있다.

상호연결망의 성능을 평가하는 척도는 분지수, 대칭성, 연결도, 지름, 망 비용, 이분할 에지수, 고장 지름, 고장 허용 알고리즘 그리고 임베딩 등이 있다. 망 비용은 분지수 × 지름으로 정의되며, 두 요소는 상관관계(trade-off)에 있기 때문에 망 비용을 줄이는 것은 쉽지 않다.

본 논문에서는 작은 지름과 대칭성 성질을 갖는 3차원 꼬인 큐브를 기본모듈로 하고, 분지수가 상수 4로 고정된 꼬인 큐브 토러스(TT) 연결망을 제안한다. TT 연결망은 기본모듈을 격자구조의 격자점에 배치하고, 기본모듈을 주변 기본모듈과 완전 그래프 형태로 연결한 구조이다. 본 논문의 연구결과로 TT 연결망과 동일한 노드수를 갖는 허니컴 메쉬, 허니컴 토러스, 다이아고날 메쉬, 토러스 그리고 핵사고날 토러스 연결망과 망비용 관점에서 비교한 결과를 표 2에 보이겠다. <표 2>의 비교 결과에 의해 TT 연결망이 메쉬 부류 연결망보다 우수한 연결망임을 알 수 있다.

본 논문의 구성은 2장에서 3차원 꼬인 큐브의 정의 및 기본 성질을 보이겠다. 3장에서 꼬인 큐브 토러스 연결망을 제안하고, 라우팅 알고리즘을 비롯한 다양한 위상적 망속성을 보이겠다. 또한 분지수가 고정된 메쉬 부류의 연결망과 본 논문에서 제안한 꼬인 큐브 토러스 연결망과의 망 비용을 비교하겠다. 마지막으로 결론을 맺는다.

2. 관련연구

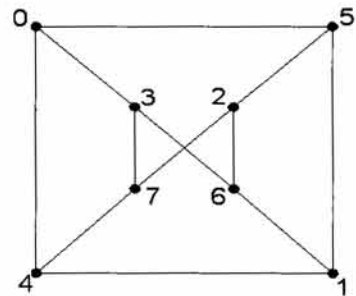
꼬인 큐브(Twisted Cube) 연결망은 하이퍼큐브 연결망의 변형된 연결망으로 하이퍼큐브와 동일한 분지수를 가지고 있으면서도 하이퍼큐브의 지름을 대략 절반 정도 개선한 우수한 연결망이다. 그래서 많은 연구자들에 의해 꼬인 큐브의 다양한 성질들 - 지름, 분지수, 라우팅 알고리즘, 최대고장허용도, 임베딩, 고장진단도 - 을 분석한 많은 연구 결과들이 발표되었다[1,3,7,8,14,17]. 3차원 꼬인 큐브 연결망(3-TC)은 8개의 노드와 12개의 에지로 구성된 연결망으로 분지수가 3이고, 지름은 2인 연결망이며, 노드 대칭이다. (그림 1)은 3차원 꼬인 큐브 연결망을 나타내고 있다. 기존에 제안된 꼬인 큐브 연결망에서는 각 노드를 이진수로 나타내고 있지만 본 논문에서는 각 노드를 십진수(0-7)로 나타내겠다.

3-TC의 노드 집합을 V 라고 나타내고, 에지 집합을 E 라고 나타내겠다. 3-TC의 정의는 다음과 같다. mod는 나머지 연산자를 나타낸다.

정의 1.

$$V = \{z \mid z \in Z_8\}$$

$$E = \{\{z_1, z_2\} \mid z_1 - z_2 \equiv 3 \text{ or } z_1 - z_2 \equiv 4 \text{ or } z_1 - z_2 \equiv 5 \pmod{8}\}$$



(그림 1) 3차원 꼬인 큐브 연결망(3-TC)

노드 U' 를 U 와 거리가 1인 노드라고 하고, 노드 U'' 를 U 와 거리가 2인 노드라고 하자. 3-TC의 지름은 2이므로 정의 1에 의해 각 노드 사이의 거리를 <표 1>과 같이 정리할 수 있다. <표 1>에 의해 노드 U' 의 개수는 3개이고, 노드 U'' 의 개수는 4개임을 알 수 있다.

<표 1> 3차원 꼬인 큐브에서 노드 사이의 거리

U	U'	U''	U	U'	U''
0	3,4,5	1,2,6,7	4	7,0,1	2,3,5,6
1	4,5,6	0,2,3,7	5	0,1,2	3,4,6,7
2	5,6,7	0,1,3,4	6	1,2,3	4,5,7,0
3	6,7,0	1,2,4,5	7	2,3,4	0,1,5,6

3-TC의 지름은 2이므로 노드 U'' 와 노드 U 사이의 경로는 노드 U' 에 의해 결정된다는 것을 <표 1>에 의해 알 수 있다. 그러므로 3-TC의 임의의 두 노드 사이의 경로를 정의 2와 같이 나타낼 수 있다.

정의 2. 3-TC의 인접한 두 노드를 U 와 U' 라고 하고 U 와 인접하지 않은 노드를 U'' 라고 하자. 인접한 두 노드 U 와 U' 의 경로는 기호 \rightarrow 를 사용하여 표현하겠다.

노드 U 와 U' 의 경로 : $U \rightarrow U'$
 노드 U 와 U'' 의 경로 : $U \rightarrow U' \rightarrow U''$

연결망의 해밀토니안(hamiltonian) 경로는 연결망을 구성하는 모든 노드를 오직 한번만 지나는 경로를 말하며, 시작 노드와 도착노드가 같은 해밀토니안 경로를 해밀토니안 사이클이라고 말한다. 해밀토니안 사이클을 가지는 연결망을 해밀토니안 연결망이라 한다. 3-TC는 해밀토니안 연결망이다. 다음의 성질 1을 통해 3-TC에 해밀토니안 사이클이 존재한다는 것을 알 수 있다. 그리고 성질 2를 통해 3-TC에 존재하는 해밀토니안 경로 구성을 알 수 있다.

성질 1. 3-TC의 임의의 노드 U 를 시작 및 도착 노드로 갖는 해밀토니안 사이클이 존재한다. 3-TC는 노드 대칭 및 에지 대칭적이므로 출발 노드 U 의 주소를 0이라 가정하면 다음과 같은 해밀토니안 사이클을 갖는다.

$$0 \rightarrow 4 \rightarrow 7 \rightarrow 2 \rightarrow 5 \rightarrow 1 \rightarrow 6 \rightarrow 3 \rightarrow 0$$

성질 2. 3-TC의 임의의 노드 U 에서 노드 V 까지 해밀토니안 경로가 존재한다($U \neq V, 0 \leq U, V \leq 7$). 3-TC는 노드 대칭 및 에지 대칭적이므로 출발 노드 U 의 주소를 0이라 가정하면 다음과 같은 해밀토니안 경로를 갖는다.

- 경우1) $V=U+1 : 0 \rightarrow 4 \rightarrow 7 \rightarrow 3 \rightarrow 6 \rightarrow 2 \rightarrow 5 \rightarrow 1$
- 경우2) $V=U+2 : 0 \rightarrow 4 \rightarrow 7 \rightarrow 3 \rightarrow 6 \rightarrow 1 \rightarrow 5 \rightarrow 2$
- 경우3) $V=U+3 : 0 \rightarrow 5 \rightarrow 2 \rightarrow 7 \rightarrow 4 \rightarrow 1 \rightarrow 6 \rightarrow 3$
- 경우4) $V=U+4 : 0 \rightarrow 3 \rightarrow 6 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 7 \rightarrow 4$
- 경우5) $V=U+5 : 0 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 1 \rightarrow 6 \rightarrow 2 \rightarrow 5$
- 경우6) $V=U+6 : 0 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 6$
- 경우7) $V=U+7 : 0 \rightarrow 3 \rightarrow 6 \rightarrow 2 \rightarrow 5 \rightarrow 1 \rightarrow 4 \rightarrow 7$

3. 꼬인 큐브 토러스 연결망

3.1 꼬인 큐브 토러스의 정의

꼬인 큐브 토러스 연결망($TT_{M,N}$)은 3-TC를 기본 모듈로 설정하여 구성된 연결망으로 3-TC를 격자점으로 $N(x$ 축) $\times M(y$ 축) 개를 배열하여 정의 3과 4에 나타낸 노드 및 에지 정의에 의해 연결되는 연결망이다. $TT_{M,N}$ 의 노드 집합을 V_{TT} 라고 나타내고, 에지 집합을 E_{TT} 라고 나타내겠다. $TT_{M,N}$ 는 기본 모듈인 3-TC를 (x,y) 로 나타낸다. 기본 모듈을 구성하는 각 단위 노드는 (x,y,z) 로 나타낸다. V_{TT} 의 정의는 다음과 같다.

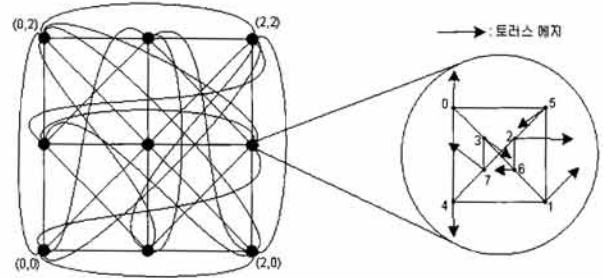
정의 3.

$$V_{TT} = \{(x, y, z) \mid x \in Z_N, y \in Z_M, z \in Z_8\}$$

$TT_{M,N}$ 의 에지는 큐브 에지($E_{TT-cube}$)와 토러스 에지($E_{TT-torus}$)로 구분한다. 토러스 에지는 기본 모듈과 기본 모듈을 연결하는 에지이고, 큐브 에지는 기본 모듈인 3-TC의 단위 노드와 단위 노드를 연결하는 에지이다. E_{TT} 의 정의는 다음과 같다.

기호 %는 나머지연산자를 나타낸다. 정의 4에 의해 구해진 x_2 가 -1이면 $x_2=N-1$ 이고, y_2 가 -1이면 $y_2=M-1$ 이며, x_2 가 N 이면 $x_2=0$ 이고, y_2 가 M 이면 $y_2=0$ 이다. 정의 4에 의해 구해진 토러스 에지의 명칭을 다음과 같이 구분하겠다. $z_1=2,6$ 이면 가로 에지, $z_1=0,4$ 이면 세로 에지, $z_1=1,5$ 이면 사선 에지,

$z_1=3,7$ 이면 역사선 에지라고 하고, $(0,y_1,z_1)$ 와 $(N-1,y_2,z_2)$ 를 연결하는 에지와 $(x_1,0,z_1)$ 와 $(x_2,M-1,z_2)$ 를 연결하는 에지를 라운드 에지라고 하겠다. (그림 2)은 기본 모듈인 3-TC를 격자점으로 표현하여 그린 $TT_{3,3}$ 이다. $TT_{3,3}$ 의 기본 모듈은 이웃하는 기본 모듈과 완전그래프 형태로 연결되어 있다. $TT_{M,N}$ 은 노드수가 $8MN$ 이고, 에지수가 $16MN$ 이며, 분지수가 4인 정규연결망(regular network)이다.



(그림 2) $TT_{3,3}$

3.2 꼬인 큐브 토러스의 라우팅 알고리즘과 지름

$TT_{M,N}$ 의 라우팅 알고리즘은 출발 노드가 포함된 출발 기본 모듈에서 모듈 내부 라우팅을 하고, 목적 노드가 포함된 목적 기본 모듈로 라우팅을 한 후, 마지막으로 목적 기본 모듈 내부 라우팅을 한다. 기본 모듈 내부에 존재하는 노드 사이의 라우팅은 내부 라우팅이라 하고, 기본 모듈 사이의 라우팅은 외부 라우팅이라 부르겠다. 외부 라우팅은 어떤 기본 모듈에서 행과 열, 모두 1만큼 거리가 있는 기본 모듈로 라우팅 할 때는 가로 에지나 세로 에지를 이용하지 않고 사선 또는 역사선 에지를 이용하여 라우팅을 한다. 노드 $U(x_1,y_1,z_1)$ 를 출발 노드, 노드 $W(x_2,y_2,z_2)$ 를 도착 노드라 하자. dx 는 (x_1,y_1) 와 (x_2,y_2) 사이의 열간 거리, dy 는 (x_1,y_1) 와 (x_2,y_2) 사이의 행간 거리, dm 은 (x_1,y_1) 에서부터 사선 또는 역사선 에지를 이용하여 이동한 기본 모듈 사이의 거리, dr 은 dm 만큼 이동하여 도착한 기본 모듈로부터 (x_2,y_2) 사이의 열간 또는 행간 거리를 나타낸다. \Rightarrow 는 정의 2에 의한 내부 라우팅을, \Rightarrow 는 외부 라우팅을 표시한다. $dx=\min(|x_2-x_1|, N-|x_2-x_1|)$ 이고, $dy=\min(|y_2-y_1|, M-|y_2-y_1|)$ 이며, $dm=\min(dx,dy)$ 이고, $dr=\max(dx,dy)-\min(dx,dy)$ 이다. 라우팅 알고리

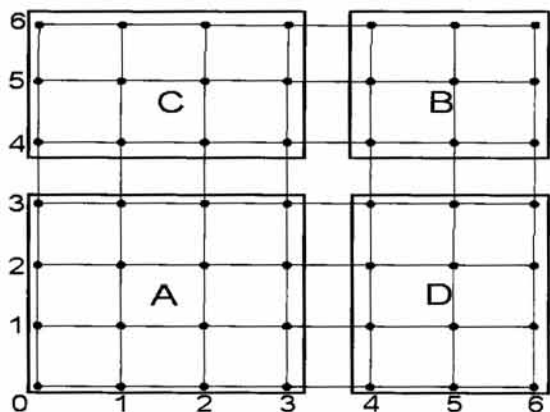
정의 4. $E_{TT} = E_{TT-cube} \cup E_{TT-torus}$

$$E_{TT-cube} = \{(x, y, z_1), (x, y, z_2) \mid (z_1, z_2) \in E\}$$

$$E_{TT-torus} = \{(x_1, y_1, z_1), (x_2, y_2, z_2) \mid x_2 = x_1 + \alpha, y_2 = y_1 + \beta, z_2 = (z_1 + 4) \% 8\}$$

$$\{(x,y,z_1), (x,y,z_2), (x_1,y_1,z_1), (x_2,y_2,z_2)\} \in V_{TT}$$

z_1	0	1	2	3	4	5	6	7
α	0	1	1	1	0	-1	-1	-1
β	1	1	0	-1	-1	-1	0	1



(그림 3) $T_{7,7}$ 의 라우팅 영역

즘 구성을 위해 dx 와 dy 에 의해 라우팅 영역을 (그림 3)과 같이 4가지 영역으로 나누겠다. 그림의 복잡성 때문에 토러스 에지 중에 가로와 세로 에지만 (그림 3)에 나타냈다.

출발 노드 U 는 기본 모듈 $(0,0)$ 에 존재한다고 가정하면 목적 노드 W 의 위치는 dx 와 dy 에 따라 다음과 같이 4가지 영역에 속한다. $dx \leq \lfloor \frac{N}{2} \rfloor$, $dy \leq \lfloor \frac{M}{2} \rfloor$ 이면 W 는 A 영역에 존재하고, $dx > \lfloor \frac{N}{2} \rfloor$, $dy > \lfloor \frac{M}{2} \rfloor$ 이면 W 는 B 영역에 존재하고, $dx \leq \lfloor \frac{N}{2} \rfloor$, $dy > \lfloor \frac{M}{2} \rfloor$ 이면, W 는 C 영역에 존재하고, $dx > \lfloor \frac{N}{2} \rfloor$, $dy \leq \lfloor \frac{M}{2} \rfloor$ 이면, W 는 D 영역에 존재한다. 라우팅 알고리즘을 구성하기 위해 모든 출발 노드 U 는 기본 모듈 $(0,0)$ 에 존재한다고 가정하겠다. 즉, $U=(0,0,z_1)$ 이고 $W=(x_2-x_1,y_2-y_1,z_2)$ 이다. W 가 존재하는 위치에 따라 라우팅 알고리즘을 다음과 같이 나타내겠다.

라우팅 알고리즘-A ($W \in A$)

```
[Step 1] 출발 기본 모듈 내부 라우팅
    If  $x_2-x_1=0$  and  $y_2-y_1=0$  then  $U \Rightarrow W$ . Stop.
    elseif  $dx-dy=dx$  then  $U \Rightarrow (0,0,2)$ . goto 3-1.
    elseif  $dy-dx=dy$  then  $U \Rightarrow (0,0,0)$ . goto 4-1
    else  $U \Rightarrow (0,0,1)$ . goto Step 2
[Step 2] 사선 에지를 이용한 외부 라우팅
     $(0,0,1) \Rightarrow (1,1,5) \Rightarrow (1,1,1) \Rightarrow (2,2,5) \Rightarrow (2,2,1) \Rightarrow (3,3,5) \Rightarrow (3,3,1) \Rightarrow \dots \Rightarrow (dm, dm, 5)$ .
    If  $dx > dy$  then goto Step 3-2.
    elseif  $dx < dy$  then goto Step 4-2.
    else goto Step 5-1. /*  $dm=x_2-x_1=y_2-y_1$  */
[Step 3] 가로 에지를 이용한 외부 라우팅
    [Step 3-1]  $(0,0,2) \Rightarrow (1,0,6) \Rightarrow (1,0,2) \Rightarrow (2,0,6) \Rightarrow (2,0,2) \Rightarrow (3,0,6) \Rightarrow (3,0,2) \Rightarrow \dots \Rightarrow (x_2-x_1, y_2-y_1, 6)$ . goto Step 5-3.
    [Step 3-2]  $(dm, dm, 5) \Rightarrow (dm, dm, 2) \Rightarrow (dm+1, dm, 6) \Rightarrow$ 
```

```
 $(dm+1, dm, 2) \Rightarrow (dm+2, dm, 6) \Rightarrow (dm+2, dm, 2) \Rightarrow \dots \Rightarrow (x_2-x_1, y_2-y_1, 6)$ . goto Step 5-3.
/*  $dm=y_2-y_1$  */
```

```
[Step 4] 세로 에지를 이용한 외부 라우팅
    [Step 4-1]  $(0,0,0) \Rightarrow (0,1,4) \Rightarrow (0,1,0) \Rightarrow (0,2,4) \Rightarrow (0,2,0) \Rightarrow (0,3,4) \Rightarrow (0,3,0) \dots \Rightarrow (x_2-x_1, y_2-y_1, 4)$ . goto Step 5-2.
    [Step 4-2]  $(dm, dm, 5) \Rightarrow (dm, dm, 0) \Rightarrow (dm, dm+1, 4) \Rightarrow (dm, dm+1, 0) \Rightarrow (dm, dm+2, 4) \Rightarrow (dm, dm+2, 0) \Rightarrow \dots \Rightarrow (x_2-x_1, y_2-y_1, 4)$ . goto Step 5-2.
    /*  $dm=x_2-x_1$  */
[Step 5] 목적 기본 모듈 내부 라우팅
    [Step 5-1]  $(x_2-x_1, y_2-y_1, 5) \Rightarrow W$ . Stop.
    [Step 5-2]  $(x_2-x_1, y_2-y_1, 4) \Rightarrow W$ . Stop.
    [Step 5-3]  $(x_2-x_1, y_2-y_1, 6) \Rightarrow W$ . Stop.
```

라우팅 알고리즘-B ($W \in B$)

```
[Step 1] 출발 기본 모듈 내부 라우팅
     $U \Rightarrow (0,0,5)$ . goto Step 2
[Step 2] 사선 에지를 이용한 외부 라우팅
     $(0,0,5) \Rightarrow (N-1, M-1, 1) \Rightarrow (N-1, M-1, 5) \Rightarrow (N-2, M-2, 1) \Rightarrow (N-2, M-2, 5) \Rightarrow (N-3, M-3, 1) \Rightarrow (N-3, M-3, 5) \Rightarrow \dots \Rightarrow (N-dm, M-dm, 1)$ .
    If  $dx > dy$  then goto Step 3.
    elseif  $dx < dy$  then goto Step 4-2.
    else goto Step 5-1.
    /*  $N-dm=x_2-x_1$ ,  $M-dm=y_2-y_1$  */
[Step 3] 가로 에지를 이용한 외부 라우팅
     $(N-dm, M-dm, 1) \Rightarrow (N-dm, M-dm, 6) \Rightarrow (N-dm-1, M-dm, 2) \Rightarrow (N-dm-1, M-dm, 6) \Rightarrow (N-dm-2, M-dm, 2) \Rightarrow (N-dm-2, M-dm, 6) \Rightarrow \dots \Rightarrow (x_2-x_1, y_2-y_1, 2)$ . goto Step 5-3. /*  $M-dm=y_2-y_1$  */
[Step 4] 세로 에지를 이용한 외부 라우팅
     $(N-dm, M-dm, 1) \Rightarrow (N-dm, M-dm, 4) \Rightarrow (N-dm, M-dm-1, 0) \Rightarrow (N-dm, M-dm-1, 4) \Rightarrow (N-dm, M-dm-2, 0) \Rightarrow (N-dm, M-dm-2, 4) \Rightarrow \dots \Rightarrow (x_2-x_1, y_2-y_1, 0)$ . goto Step 5-2. /*  $N-dm=x_2-x_1$  */
[Step 5] 목적 기본 모듈 내부 라우팅
    [Step 5-1]  $(x_2-x_1, y_2-y_1, 1) \Rightarrow W$ . Stop.
    [Step 5-2]  $(x_2-x_1, y_2-y_1, 0) \Rightarrow W$ . Stop.
    [Step 5-3]  $(x_2-x_1, y_2-y_1, 6) \Rightarrow W$ . Stop.
```

라우팅 알고리즘-C ($W \in C$)

```
[Step 1] 출발 기본 모듈 내부 라우팅
    If  $dy-dx=dy$  then  $U \Rightarrow (0,0,4)$ . goto 4-1
    else  $U \Rightarrow (0,0,3)$ . goto Step 2
[Step 2] 역사선 에지를 이용한 외부 라우팅
     $(0,0,3) \Rightarrow (1, M-1, 7) \Rightarrow (1, M-1, 3) \Rightarrow (2, M-2, 7) \Rightarrow (2, M-2, 3) \Rightarrow (3, M-3, 7) \Rightarrow (3, M-3, 3) \Rightarrow \dots \Rightarrow$ 
```

$(dm, M-dm, 7)$.

If $dx > dy$ then goto Step 3.

elseif $dx < dy$ then goto Step 4-2.

else goto Step 5-1.

/* $dx = x_2 - x_1, M - dm = y_2 - y_1$ */

[Step 3] 가로 에지를 이용한 외부 라우팅

$(dm, M-dm, 7) \Rightarrow (dm, M-dm, 6) \Rightarrow (dm+1, M-dm, 2) \Rightarrow$
 $(dm+1, M-dm, 6) \Rightarrow (dm+2, M-dm, 2) \Rightarrow (dm+2, M-dm, 6)$
 $\Rightarrow \dots \Rightarrow (x_2 - x_1, y_2 - y_1, 2)$. goto Step 5-3.

/* $M - dm = y_2 - y_1$ */

[Step 4] 세로 에지를 이용한 외부 라우팅

[Step 4-1] $(0, 0, 4) \Rightarrow (0, 1, 0) \Rightarrow (0, 1, 4) \Rightarrow (0, 2, 0) \Rightarrow (0, 2, 4) \Rightarrow$
 $(0, 3, 0) \Rightarrow (0, 3, 4) \Rightarrow \dots \Rightarrow (x_2 - x_1, y_2 - y_1, 0)$.
 goto Step 5-2.

[Step 4-2] $(dm, M-dm, 5) \Rightarrow (dm, M-dm, 0) \Rightarrow$
 $(dm, M-dm-1, 4) \Rightarrow (dm, M-dm-1, 0) \Rightarrow$
 $(dm, M-dm-2, 4) \Rightarrow (dm, M-dm-2, 0) \Rightarrow \dots \Rightarrow$
 $(x_2 - x_1, y_2 - y_1, 4)$. goto Step 5-2.

/* $dm = x_2 - x_1$ */

[Step 5] 목적 기본 모듈 내부 라우팅

[Step 5-1] $(x_2 - x_1, y_2 - y_1, 7) \Rightarrow W$. Stop.

[Step 5-2] $(x_2 - x_1, y_2 - y_1, 0) \Rightarrow W$. Stop.

[Step 5-3] $(x_2 - x_1, y_2 - y_1, 2) \Rightarrow W$. Stop.

라우팅 알고리즘-D ($W \in D$)

[Step 1] 출발 기본 모듈 내부 라우팅

If $dx = dy$ then $U \Rightarrow (0, 0, 6)$. goto 3-1.

else $U \Rightarrow (0, 0, 7)$. goto Step 2

[Step 2] 역사선 에지를 이용한 외부 라우팅

$(0, 0, 7) \Rightarrow (N-1, 1, 3) \Rightarrow (N-1, 1, 7) \Rightarrow (N-2, 2, 3) \Rightarrow (N-2, 2, 7) \Rightarrow$
 $(N-3, 3, 3) \Rightarrow (N-3, 3, 7) \Rightarrow \dots \Rightarrow (N-dm, dm, 3)$.

If $dx > dy$ then goto Step 3-2.

elseif $dx < dy$ then goto Step 4.

else goto Step 5-1.

/* $N - dm = x_2 - x_1, dm = y_2 - y_1$ */

[Step 3] 가로 에지를 이용한 외부 라우팅

[Step 3-1] $(0, 0, 6) \Rightarrow (N-1, 0, 2) \Rightarrow (N-1, 0, 6) \Rightarrow (N-2, 0, 2) \Rightarrow$
 $(N-2, 0, 6) \Rightarrow (N-3, 0, 2) \Rightarrow (N-3, 0, 6) \Rightarrow \dots \Rightarrow$
 $(x_2 - x_1, y_2 - y_1, 2)$. goto Step 5-3.

[Step 3-2] $(N-dm, dm, 3) \Rightarrow (N-dm, dm, 6) \Rightarrow$
 $(N-dm-1, dm, 2) \Rightarrow (N-dm-1, dm, 6) \Rightarrow$
 $(N-dm-2, dm, 2) \Rightarrow (N-dm-2, dm, 6) \Rightarrow \dots \Rightarrow$
 $(x_2 - x_1, y_2 - y_1, 2)$. goto Step 5-3.

/* $dm = y_2 - y_1$ */

[Step 4] 세로 에지를 이용한 외부 라우팅

$(N-dm, dm, 3) \Rightarrow (N-dm, dm, 0) \Rightarrow (N-dm, dm+1, 4) \Rightarrow$
 $(N-dm, dm+1, 0) \Rightarrow (N-dm, dm+2, 4) \Rightarrow (N-dm, dm+2, 0)$
 $\Rightarrow \dots \Rightarrow (x_2 - x_1, y_2 - y_1, 4)$. goto Step 5-2.

/* $N - dm = x_2 - x_1$ */

[Step 5] 목적 기본 모듈 내부 라우팅

[Step 5-1] $(x_2 - x_1, y_2 - y_1, 3) \Rightarrow W$. Stop.

[Step 5-2] $(x_2 - x_1, y_2 - y_1, 4) \Rightarrow W$. Stop.

[Step 5-3] $(x_2 - x_1, y_2 - y_1, 2) \Rightarrow W$. Stop.

소정리 1. 기본 모듈 (0,0)과 x축으로 가장 먼 거리에 위치한 기본 모듈은 $(\lfloor \frac{N}{2} \rfloor, 0)$ 이고, y축으로 가장 먼 거리에 위치한 기본 모듈은 $(0, \lfloor \frac{M}{2} \rfloor)$ 이므로 두 모듈 사이의 최대 거리는 $M > N$ 이면 $\lfloor \frac{M}{2} \rfloor$ 이고, $M < N$ 이면 $\lfloor \frac{N}{2} \rfloor$ 이다.

제안된 라우팅 알고리즘에 의해 $TT_{M,N}$ 의 지름을 구하겠다. 출발 기본 모듈과 목적 기본 모듈을 제외한 라우팅 경로에 존재하는 기본 모듈을 중간경유 기본 모듈이라고 하겠다. [Step 1]에 적용된 내부 라우팅 거리는 최대 2이다. [Step 2],[Step 3],[Step 4]에 존재하는 중간경유 기본 모듈의 개수는 $\text{Max}(dx-1, dy-1)$ 이다. 중간경유 기본 모듈의 내부 라우팅 거리는 모두 1임을 <표 1>에 의해 알 수 있다. 그러므로 중간경유 기본 모듈의 내부 라우팅 거리의 합은 $\text{Max}(dx-1, dy-1)$ 이고, 출발 기본 모듈과 목적 기본 모듈 사이의 외부 라우팅 거리의 합은 $\text{Max}(dx, dy)$ 이다. 소정리 1에 의해 dx의 최대값은 $\lfloor \frac{N}{2} \rfloor$ 이고, dy의 최대값은 $\lfloor \frac{M}{2} \rfloor$ 이다. [Step 5]에 적용된 내부 라우팅 거리는 최대 2이다. 그러므로 $TT_{M,N}$ 의 지름은 다음과 같다.

$$\begin{aligned} \text{정리 1. } TT_{M,N} \text{의 지름} &= 2 + \text{Max}(\lfloor \frac{N}{2} \rfloor - 1, \lfloor \frac{M}{2} \rfloor - 1) + \text{Max}(\lfloor \frac{N}{2} \rfloor, \lfloor \frac{M}{2} \rfloor) + 2 \\ &= \text{Max}(2 \lfloor \frac{N}{2} \rfloor, 2 \lfloor \frac{M}{2} \rfloor) + 3 \end{aligned}$$

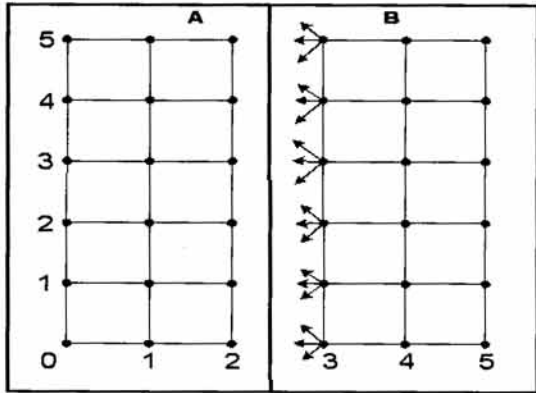
3.3 이분할에지수

이분할에지수(bisection width)는 연결망을 노드수가 같은 두개의 망으로 분리하기 위해 제거해야 할 에지의 최소 개수이다. 분리된 두 개의 망은 노드수가 1차이 보다 같거나 적어야 한다. 이분할에지수가 적으면 보다 적은 에지 고장 만으로도 망이 분리되어 제 역할을 할 수 없으므로 이분할 에지수는 클수록 좋다. $TT_{M,N}$ 의 이분할에지수는 $N \geq M$ 인 경우와 $N \leq M$ 인 경우로 나누어서 구해야만 하지만 두 경우의 결과가 유사하기 때문에 본 장에서는 $N \geq M$ 인 경우에 대하여서 구하도록 한다.

정리 2. $TT_{M,N}$ 의 이분할에지수는 N이 짝수인 경우 $6M$ 이고, 홀수인 경우 $6M+6(M=홀수)$, $6M+2(M=짝수)$ 이다.

증명. N이 짝수인 경우와 홀수인 경우로 나누어서 증명 하겠다.

경우 1) N 이 짝수인 경우 : 기본 모듈 $(0,i)$ 와 기본 모듈 $(N-1,i)$ 를 연결하는 토러스 에지를 모두 제거하고, 기본 모듈 $(\frac{N}{2}-1,i)$ 와 기본 모듈 $(\frac{N}{2},i)$ 를 연결하는 토러스 에지를 모두 제거하면 동일한 노드수를 갖는 두 개의 연결망으로 분리된다($0 \leq i \leq M-1$). 하나의 기본 모듈과 인접한 하나의 기본 모듈을 연결하는 토러스 에지수는 3(사선, 가로, 역사선 에지)이다. i 는 M 개이므로 기본 모듈 $(0,i)$ 와 기본 모듈 $(N-1,i)$ 를 연결하는 토러스 에지수는 $3M$ 이고, 기본 모듈 $(\frac{N}{2}-1,i)$ 와 기본 모듈 $(\frac{N}{2},i)$ 를 연결하는 토러스 에지수도 $3M$ 이므로 두 개의 연결망으로 분리할 이분할에지수는 $6M$ 이다. (그림 4)는 $TT_{7,6}$ 의 이분할에지수를 나타냈다. (그림 4)에서 기본 모듈 $(0,i)$ 와 기본 모듈 $(N-1,i)$ 를 연결하는 $3M$ 개의 제거해야할 토러스 에지는 표시하지 않았고($0 \leq i \leq M-1$), 제거해야할 나머지 토러스 에지는 화살표로 표시하였다. $TT_{6,6}$ 의 이분할에지수는 36임을 알 수 있다.

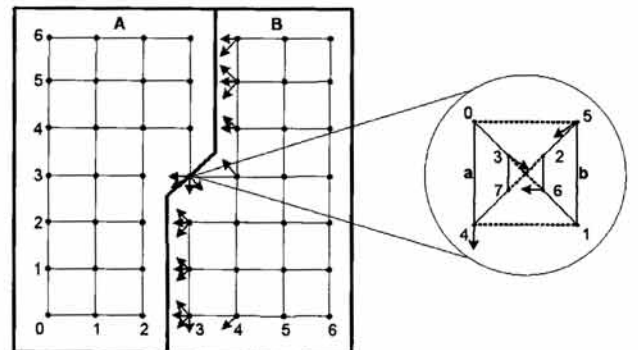


(그림 4) $TT_{6,6}$ 의 이분할에지수

경우 2) N 이 홀수인 경우 : M 이 짝수인 경우와 홀수인 경우로 나누어서 증명하겠다.

경우 2-1) M 이 홀수인 경우 : 기본 모듈의 개수가 홀수이므로 기본 모듈 $(\lfloor \frac{N}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ 을 구성하는 4개의 에지를 제거하여 두 개의 그룹으로 분리한다. 노드 0,3,4,7이 포함된 그룹을 a, 노드 1,2,5,6이 포함된 그룹을 b라고 하자. 그리고 $(\lfloor \frac{N}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ 을 제외한 나머지 기본 모듈들을 다음과 같이 두 개의 연결망 A,B에 포함시키겠다. $i > \lfloor \frac{M}{2} \rfloor$ 인 경우, 기본 모듈 (j,i) 는 연결망 A에 포함되고, 기본 모듈 (k,i) 는 연결망 B에 포함된다($0 \leq j \leq \lfloor \frac{N}{2} \rfloor, \lfloor \frac{N}{2} \rfloor < k < N$). $i \leq \lfloor \frac{M}{2} \rfloor$ 인 경우, 기본 모듈 (j',i) 는 연결망 A에 포함되고, 기본 모듈 (k',i) 는 연결망 B에 포함된다($0 \leq j' < \lfloor \frac{N}{2} \rfloor, \lfloor \frac{N}{2} \rfloor \leq k' < N$). 그룹 a는 연결망 A에 포함

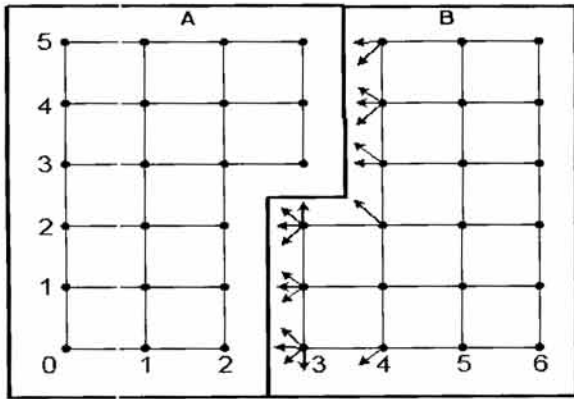
되고, 그룹 b는 연결망 B에 포함된다. $(\lfloor \frac{N}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ 을 a, b 그룹으로 나누기 위해 제거해야할 에지수는 4개(0과 5, 1과 4, 2와 7, 3과 6을 연결하는 에지)이다. a와 B를 연결하는 토러스 에지수는 2개이고, b와 A를 연결하는 토러스 에지수는 2개이다. 그러므로 $(\lfloor \frac{N}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ 에서 제거해야할 에지수는 8개이다. B에 포함된 모듈 $(\lfloor \frac{N}{2} \rfloor + 1, M-1)$ 에서 제거해야할 에지수는 2개이고, $(\lfloor \frac{N}{2} \rfloor + 1, \lfloor \frac{M}{2} \rfloor + 1)$ 에서 제거해야할 에지수는 2개이다. $(\lfloor \frac{N}{2} \rfloor + 1, \lfloor \frac{M}{2} \rfloor)$ 에서 제거해야할 에지수는 1개이고, $(\lfloor \frac{N}{2} \rfloor + 1, 0)$ 에서 제거해야할 에지수는 1개이다. $(\lfloor \frac{N}{2} \rfloor, 0)$ 에서 제거해야할 에지수는 4개이며, A와 B 연결망을 연결하고 있는 나머지 기본 모듈에서 제거해야할 에지수는 3개이다. 그러므로 이 경우의 이분할에지수는 $6M+6$ 이다. (그림 5)는 $TT_{7,7}$ 의 이분할에지수를 나타냈다. (그림 5)에서 기본 모듈 $(0,i)$ 와 기본 모듈 $(N-1,i)$ 를 연결하는 $3M$ 개의 제거해야할 토러스 에지는 표시하지 않았고($0 \leq i \leq M-1$), a, b 그룹으로 나누기 위해 제거해야할 에지는 점선으로 표시하였으며, 제거해야할 나머지 토러스 에지는 화살표로 표시하였다. $TT_{7,7}$ 의 이분할에지수는 48임을 알 수 있다.



(그림 5) $TT_{7,7}$ 의 이분할에지수

경우 2-2) M 이 짝수인 경우 : 연결망 $TT_{M,N}$ 을 다음과 같이 두 개의 연결망으로 분리한다. $i \geq \frac{M}{2}$ 인 경우, 기본 모듈 (j,i) 는 연결망 A에 포함되고, 기본 모듈 (k,i) 는 연결망 B에 포함된다($0 \leq j \leq \frac{N}{2}, \frac{N}{2} < k < N$). $i < \frac{M}{2}$ 인 경우, 기본 모듈 (j',i) 는 연결망 A에 포함되고, 기본 모듈 (k',i) 는 연결망 B에 포함된다($0 \leq j' < \frac{N}{2}, \frac{N}{2} \leq k' < N$). B에 포함된 모듈 $(\frac{N}{2} + 1, M-1)$ 에서 제거해야할 에지수는 2개이고, $(\frac{N}{2} + 1, \frac{M}{2})$ 에

서 제거해야할 에지수는 2개이다. $(\frac{N}{2}+1, \frac{M}{2}-1)$ 에서 제거해야할 에지수는 1개이고, $(\frac{N}{2}+1, 0)$ 에서 제거해야할 에지수는 1개이다. $(\frac{N}{2}, 0)$ 에서 제거해야할 에지수는 4개이며, A와 B 연결망을 연결하고 있는 나머지 기본 모듈에서 제거해야할 에지수는 3개이다. 그러므로 이 경우의 이분할에지수는 $6M+2$ 이다. (그림 6)은 $TT_{7,6}$ 의 이분할에지수를 나타냈다. (그림 6)에서 기본 모듈 $(0, i)$ 와 기본 모듈 $(N-1, i)$ 를 연결하는 $3M$ 개의 제거해야할 토러스 에지는 표시하지 않았고 $(0 \leq i \leq M-1)$, 제거해야할 나머지 토러스 에지는 화살표로 표시하였다. $TT_{7,6}$ 의 이분할에지수는 38임을 알 수 있다.



(그림 6) $TT_{7,6}$ 의 이분할에지수

본 증명에서 제안한 이분할에지수가 최소임을 다음과 같이 보이겠다.

연결망 $TT_{M,N}$ 을 두 개의 연결망 A, B로 나누는 가장 쉬운 방법은 경우 1에서 제안한 x축(또는 y축)을 기준으로 연결되어 있는 토러스 에지를 모두 제거하는 방법이다. 이 경우 기본 모듈 당 제거되는 에지의 개수는 3이다. 만약 기본 모듈에서 제거되는 에지의 개수가 3 미만이면 연결망 $TT_{M,N}$ 은 연결망 A, B로 분리되지 않고 하나의 연결망으로 연결되어 있음을 그림 4에 의해 쉽게 알 수 있다. 즉, 이 경우 연결망 $TT_{M,N}$ 을 두 개의 연결망 A, B로 나누기 위해 기본 모듈 당 제거해야 할 최소 에지의 개수는 3임을 알 수 있다. 그러므로 이 경우 연결망 $TT_{M,N}$ 을 두 개의 연결망으로 분리하기 위한 이분할에지수는 경우 1에서 제안한 $6M$ 임을 알 수 있다.

연결망 $TT_{M,N}$ 을 두 개의 연결망 A, B로 나누기 위해 (그림 6)과 같이 x축(또는 y축)을 기준으로 연결되어 있는 토러스 에지를 모두 제거하고, y축(또는 x축)을 기준으로 연결되어 있는 특정한 기본 모듈의 토러스 에지를 제거하는 경우를 살펴보자. y축(또는 x축)을 기준으로 연결되어 있는 특정한 기본 모듈을 이하 특정한 기본 모듈이라고 하겠다. 이 경우 특정한 기본 모듈의 토러스 에지의 개수는 1임을 알 수 있다. (그림 6)에서 제거해야할 특정한 기본 모듈의 토러

스 에지는 $\{(3,2,0)(3,3,4)\}$ 와 $\{(3,5,0)(3,0,4)\}$ 이다. 즉, 특정한 기본 모듈의 개수가 증가하면, 증가하는 특정한 기본 모듈의 개수에 의해 제거해야할 토러스 에지의 개수는 1씩 증가한다는 것을 알 수 있다. 그러므로 특정한 기본 모듈의 개수가 적을수록 이분할에지수도 줄어든다는 것을 알 수 있다. 연결망 $TT_{M,N}$ 을 두 개의 연결망으로 나누기 위해 필요한 특정한 기본 모듈의 개수는 항상 짝수임을 알 수 있다. 경우 2-2)에서 N이 홀수이므로 연결망 $TT_{M,N}$ 을 두 개의 연결망으로 나누기 위해서는 최소한 2개의 특정한 기본 모듈이 필요하다. x축(또는 y축)을 기준으로 연결되어 있는 토러스 에지의 개수는 $6M$ 이고, 2개의 특정한 기본 모듈에서 제거해야할 토러스 에지의 개수는 2이므로 제안한 이분할에지수 $6M+2$ 가 최소임을 알 수 있다.

연결망 $TT_{M,N}$ 을 두 개의 연결망 A, B로 나누기 위해 단위노드를 연결하는 에지를 제거하는 방법을 생각해 보겠다. (그림 4)에서 연결망 A에 속해있는 노드 $a=(2,3,z_1)$ 를 연결망 B에 속해 있다고 가정하고, 연결망 B에 속해있는 노드 $b=(3,1,z_2)$ 를 연결망 A에 속해 있다고 가정하자. 노드 a의 토러스 에지의 연결 상태에 의해 두 가지 경우로 나누어서 제거해야 할 에지의 개수를 알아보겠다. 먼저, 노드 a가 토러스 에지에 의해 연결망 B와 연결되어 있는 경우를 살펴보자. 이 경우 노드 a가 포함되어 있는 기본 모듈과 노드 a를 연결하고 있는 3개의 큐브 에지를 제거해야 한다. 그리고 a가 포함되어 있는 기본 모듈과 연결망 B를 연결하고 있는 2개의 토러스 에지를 제거해야 한다. 두 번째 경우는 노드 a가 토러스 에지에 의해 연결망 A와 연결되어 있는 경우를 살펴보자. 이 경우 노드 a가 포함되어 있는 기본 모듈과 노드 a를 연결하고 있는 3개의 큐브 에지를 제거해야 한다. 그리고 a가 포함되어 있는 기본 모듈과 연결망 B를 연결하고 있는 3개의 토러스 에지를 제거해야 한다. 노드 a가 토러스 에지에 의해 연결망 B와 연결되어 있고, 노드 b가 토러스 에지에 의해 연결망 A와 연결되어 있을 때 이분할에지수가 최소값을 가지므로 이 경우의 이분할에지수는 $3M+3(M-2)+5+5=6M+4$ 이다.

연결망 $TT_{M,N}$ 을 두 개의 연결망 A, B로 나누기 위해 하나의 기본 모듈 내부에 있는 두 개의 단위노드를 연결하는 에지를 제거하는 방법을 생각해 보겠다. (그림 4)에서 연결망 A에 속해있는 노드 $a=(2,3,z_1)$ 와 $a'=(2,3,z'_1)$ 를 연결망 B에 속해 있다고 가정하고, 연결망 B에 속해있는 노드 $b=(3,1,z_2)$ 와 $b'=(3,1,z'_2)$ 를 연결망 A에 속해 있다고 가정하자. 그러면 두 노드 a와 a'의 연결 상태와 두 노드와 연결망 B의 연결 상태에 의해 제거해야 할 에지의 개수가 달라진다. 제거해야할 에지의 개수가 최소가 되는 경우는 두 노드 a와 a'가 인접해 있고, 두 노드 모두 연결망 B와 토러스 에지에 의해 연결되어 있는 경우인데 이 경우는 연결망 $TT_{M,N}$ 의 정의에 의해 존재하지 않으므로, 두 노드 a와 a'가 인접해 있고, 노드 a는 연결망 B와 노드 a'는 연결망 A와 토러스 에지에 의해 연결되어 있는 경우이다. 이 경우 제거해야할 에지는 토러스 에지 3개와 큐브 에지 4개이다. 그러므로 이분할에지수는 $3M+3(M-2)+7+7=6M+8$ 이다.

연결망 $TT_{M,N}$ 을 두 개의 연결망 A, B로 나누기 위해 하나의 기본 모듈 내부에 있는 세 개의 단위노드를 연결하는 에지를 제거하는 방법을 생각해 보겠다. (그림 4)에서 연결망 A에 속해있는 노드 $a=(2,3,z_1)$ 와 $a'=(2,3,z'_1)$ 와 $a''=(2,3,z''_1)$ 를 연결망 B에 속해 있다고 가정하고, 연결망 B에 속해있는 노드 $b=(3,1,z_2)$ 를 연결망 A에 속해 있다고 가정하자. 그러면 세 노드 a와 a'와 a''의 연결 상태와 두 노드와 연결망 B의 연결 상태에 의해 제거해야 할 에지의 개수가 달라진다. 제거해야할 에지의 개수가 최소가 되는 경우는 두 노드 a와 a'와 a''가 인접해 있고, 세 노드 모두 연결망 B와 토러스 에지에 의해 연결되어 있는 경우인데 이 경우는 연결망 $TT_{M,N}$ 의 정의에 의해 존재하지 않으므로, 세 노드 a와 a'와 a''가 인접해 있고, 노드 a와 a''는 연결망 B와 노드 a'는 연결망 A와 토러스 에지에 의해 연결되어 있는 경우이다. 이 경우 제거해야할 에지는 토러스 에지 2개와 큐브 에지 5개이다. 그러므로 이분할에지수는 $3M+3(M-2)+7+5=6M+6$ 이다.

연결망 $TT_{M,N}$ 을 두 개의 연결망 A, B로 나누기 위해 하나의 기본 모듈 내부에 있는 세 개의 단위노드를 연결하는 에지를 제거하는 방법은 경우 2-1에서 제안한 방법이 제거해야할 에지의 개수를 최소가 되는 경우이다. 그러므로 이 경우 이분할에지수는 $3M+3(M-2)+8+8=6M+10$ 이다.

이상의 경우에 의해 연결망을 분리할 때 분리되는 단위노드가 적을수록 이분할에지수가 최소가 된다는 것을 알 수 있고, x축을 기준으로 연결망을 분리할 때 y축을 기준으로 제거되는 토러스 에지가 적을수록 이분할에지수가 최소가 된다는 것을 알 수 있다. 그러므로 본 증명에서 제안한 이분할에지수는 최소임을 알 수 있다.

3.4 해밀토니안 사이클

연결망에 해밀토니안 경로나 해밀토니안 사이클이 존재하면 링이나 선형배열을 쉽게 구현할 수 있어서 병렬처리에 유용하게 쓰이는 파이프라인으로 활용된다. 정리 3에서 연결망 $TT_{M,N}$ 에 해밀토니안 사이클이 존재함을 보이겠다.

정리 3. $TT_{M,N}$ 에는 해밀토니안 사이클이 존재한다.

증명. $TT_{M,N}$ 의 기본 모듈인 3-TC에 해밀토니안 경로가 존재함을 성질 2에서 증명했다. 증명을 위해 3-TC에 적용된 해밀토니안 경로는 \leftrightarrow 라고 표시하고, $=>$ 는 인접한 두 기본 모듈을 연결하는 토러스 에지를 나타낸다. 예를 들어, $(0,0,1)\leftrightarrow(0,0,4)$ 는 $(0,0,1)$ 로부터 출발하여 기본 모듈 $(0,0)$ 의 모든 노드들을 경유하여 $(0,0,4)$ 에 도착하는 해밀토니안 경로를 나타낸다. M 이 짝수인 경우와 홀수인 경우로 나누어서 증명하겠다.

경우 1) M 이 짝수인 경우:

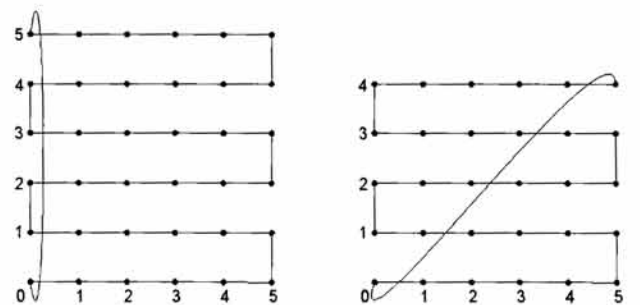
$(0,0,4)\leftrightarrow(0,0,2)=>(1,0,6)\leftrightarrow(1,0,2)=>(2,0,6)\leftrightarrow(2,0,2)=>\dots=>(N-1,0,6)\leftrightarrow(N-1,0,0)=>(N-1,1,4)\leftrightarrow(N-1,1,6)=>(N-2,1,2)\leftrightarrow(N-2,1,6)=>(N-3,1,2)\leftrightarrow(N-2,1,6)=>\dots=>(0,1,2)\leftrightarrow(0,1,0)=>(0,2,4)\leftrightarrow(0,2,2)=>(1,2,6)\leftrightarrow(1,2,2)=>(2,2,6)\leftrightarrow(2,2,2)=>\dots=>(N-1,2,6)\leftrightarrow$

$(N-1,2,0)=>\dots=>(N-1,M-1,4)\leftrightarrow(N-1,M-1,6)=>(N-2,M-1,2)\leftrightarrow(N-2,M-1,6)=>(N-3,M-1,2)\leftrightarrow(N-2,M-1,6)=>\dots=>(0,M-1,2)\leftrightarrow(0,M-1,0)=>(0,0,4)$

경우 2) M 이 홀수인 경우:

$(0,0,5)\leftrightarrow(0,0,2)=>(1,0,6)\leftrightarrow(1,0,2)=>(2,0,6)\leftrightarrow(2,0,2)=>\dots=>(N-1,0,6)\leftrightarrow(N-1,0,0)=>(N-1,1,4)\leftrightarrow(N-1,1,6)=>(N-2,1,2)\leftrightarrow(N-2,1,6)=>(N-3,1,2)\leftrightarrow(N-2,1,6)=>\dots=>(0,1,2)\leftrightarrow(0,1,0)=>(0,2,4)\leftrightarrow(0,2,2)=>(1,2,6)\leftrightarrow(1,2,2)=>(2,2,6)\leftrightarrow(2,2,2)=>\dots=>(N-1,2,6)\leftrightarrow(N-1,2,0)=>\dots=>(0,M-1,4)\leftrightarrow(0,M-1,2)=>(1,M-1,6)\leftrightarrow(1,M-1,2)=>(2,M-1,6)\leftrightarrow(2,M-1,2)=>\dots=>(N-1,M-1,6)\leftrightarrow(N-1,M-1,1)=>(0,0,5)$

(그림 7)은 $TT_{6,6}$ 과 $TT_{6,5}$ 의 해밀토니안 사이클을 나타낸다.



(a) $TT_{6,6}$ 의 해밀토니안 사이클 (b) $TT_{6,5}$ 의 해밀토니안 사이클
(그림 7) 해밀토니안 사이클의 예

3.5 TT 연결망과 메쉬부류 연결망들 사이의 망비용 비교

지금까지의 연구 결과를 바탕으로 본 논문에서 제안한 TT 연결망이 병렬 처리를 위한 대규모 시스템 구현에 적합하다는 것을 보이기 위해 기존에 제안된 메쉬 부류 연결망들보다 망비용 측면에서 우수하다는 것을 <표 2>를 통해 보이겠다(K =노드수). $M=N$ 인 경우 $T_{M,N}$ 의 지름은 정리 1에 의해 N 이 짝수인 경우 $N+3$ 이고, 홀수인 경우 $2 \lfloor \frac{N}{2} \rfloor + 3$ 임을 알 수 있다. $M=N$ 이고 N 이 짝수인 경우의 $T_{M,N}$ 의 지름을 선택하여 망비용을 비교하겠다. $M=N$ 이므로 $T_{M,N}$ 의 노드수는 $8N^2$ 이다. K 가 노드수를 나타내므로 N 에 대하여 정리하면 다음과 같다.

$$K = 8N^2$$

$$N^2 = \frac{K}{8}$$

$$N = \sqrt{\frac{K}{8}}$$

$$N = \sqrt{\frac{K}{2^2 \times 2}}$$

$$N = \frac{1}{2} \sqrt{\frac{K}{2}}$$

$$N = \frac{1}{2} \sqrt{\frac{1}{2}} \times \sqrt{K} \quad /* \sqrt{\frac{1}{2}} \approx 0.7 */$$

$$N = \frac{1}{2} \times 0.7 \sqrt{K}$$

$$N = 0.35 \sqrt{K}$$

그러므로 $T_{M,N}$ 의 지름= $N+3=0.35 \sqrt{K}+3$ 임을 알 수 있다. 망비용=지름×분지수이므로 $T_{M,N}$ 의 망비용은 $1.4 \sqrt{K}+12$ 이다.

〈표 2〉 TT 연결망과 메쉬부류 연결망들 사이의 망비용 비교

연결망	망척도	분지수	지름	망비용
허니컴 메쉬		3	$1.63 \sqrt{K}$	$4.9 \sqrt{K}$
다이아고널 배위		4	$\sqrt{K-1}$	$4 \sqrt{K-1}$
토러스		4	\sqrt{K}	$4 \sqrt{K}$
헥사고널 토러스		6	$0.58 \sqrt{K}$	$3.46 \sqrt{K}$
허니컴 토러스		3	$0.81 \sqrt{K}$	$2.45 \sqrt{K}$
TT 연결망		4	$0.35 \sqrt{K}+3$	$1.4 \sqrt{K}+12$

4. 결 론

본 논문에서는 분지수가 상수인 다이아고널 메쉬, 토러스, 허니컴 메쉬, 허니컴토러스, 헥사고널 토러스보다 지름, 망비용이 더 우수한 새로운 연결망 TT 연결망을 제안하였다. TT 연결망은 동일한 노드수를 가질 때 다이아고널 메쉬나 토러스에 비해 지름이 대략 1/2 정도이고, 헥사고널 토러스보다 망 비용이 대략 1/2가량 정도이다. 특히 동일한 노드수 (K)를 가질 때 허니컴 토러스의 지름은 $0.81 \sqrt{K}$, TT 연결망의 지름은 $0.35 \sqrt{K}+3$ 이고, 허니컴 토러스의 망 비용은 $2.45 \sqrt{K}$, TT 연결망의 망 비용은 $1.4 \sqrt{K}+12$ 으로 널리 알려진 허니컴 토러스보다 TT 연결망이 지름과 망 비용에서 개선된 값을 가진다. TT 연결망은 기본모듈을 단일노드로 보면 망 전체가 격자구조를 갖는다는 장점이 있어서 병렬 처리를 위한 대규모 시스템 구현에 매우 적합한 연결망임을 알 수 있다.

향후 본 논문에서 제안한 TT 연결망과 기존에 제안된 메쉬 부류의 연결망과 임베딩을 분석하는 연구가 이루어질 수 있으며, 고장 허용 라우팅 알고리즘과 최단 경로 길이를 갖는 병렬 경로를 구성하는 것 등이 연구과제로써 가치가 있을 것이다.

참 고 문 헌

[1] S. Abraham and K. Padmanabhan, "The Twisted Cube Topology for Multiprocessors: A Study in Network

Asymmetry," J. Parallel Distrib. Comput. Vol.13, No.1, pp.104-110, 1991.

[2] S. B. Akers, D. Harel and B. Krishnamurthy, "The Star Graph: an Attractive Alternative to The N-cube," Proc. Int'l Conf. Parallel Processing, pp.393-400, 1987.

[3] C.-P. Chang, J.-N. Wang and L.-H. Hsu, "Topological Properties of Twisted Cube," Information Sciences, Vol.113, pp.147-167, 1999.

[4] M. S. Chen and K. G. Shin, "Addressing, Routing, and Broadcasting in Hexagonal Mesh Multiprocessors", IEEE Trans. Computers, Vol.39, No.1 pp.10-18, 1990.

[5] W. Dally and C. Seitz, "The Torus Routing Chip," Distributed Computing, Vol.1, pp.187-196, 1986.

[6] A. El-Amawy and S. Latifi, "Properties and Performances of Folded Hypercubes," IEEE Trans. Parallel Distrib. Syst., Vol.2, No.1, pp.31-42, 1991.

[7] J. Fan and X. Lin, "The t/k-Diagnosability of the BC Graphs," IEEE Trans. Computers, Vol.54, No.2, pp.176-184, 2005.

[8] C.-J. Lai and C.-H. Tsai, "Embedding a Family of Meshes into Twisted Cubes," Information Processing Letters, Vol.108, No.5, pp.326-330, 2008.

[9] S. Latifi and P. K. Srimani, "Transposition Networks as a Class of Fault-Tolerant Robust Networks," IEEE Trans. Computers, Vol.45, No.2, pp.230-238, 1996.

[10] J-H. Park, "Circulant Graphs and Their Application to Communication Networks," Ph.D. Thesis, Dept. of Computer Science, KAIST, Taejon Korea, 1992.

[11] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," IEEE Trans. Computers, Vol.37, No.7, pp.867-872, 1988.

[12] I. Stojmenovic, "Honeycomb Network: Topological Properties and Communication Algorithms," IEEE Trans. Parallel Distrib. Syst., Vol.8, No.10, pp.1036-1042, 1997.

[13] K. W. Tang and S. A. Padubidri, "Diagonal and toroidal Mesh Networks", IEEE Trans. Computers, Vol.43, No.7 pp.815-826, 1994.

[14] M.-C. Yang, T.-K. Li, J. J. M. Tan and L.-H. Hsu, "On Embedding Cycles into Faulty Twisted Cubes," Information Sciences, Vol.176, No.6, pp.676-690, 2006.

[15] 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, pp.290-297, 1996.

[16] 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, 2005.

[17] 김숙연, "크기 $2^n \times 2^n$ 인 메쉬의 꼬인 큐브에 대한 임베딩," 정보처리학회논문지A, Vol.16-A, No.4, pp.223-226, 2009.



김 종 석

e-mail : rockhee7@gmail.com

1995년 순천대학교 전자계산학과(이학사)
2001년 순천대학교 컴퓨터과학과(이학석사)
2004년 순천대학교 컴퓨터과학과(이학박사)
2005년~2008년 오클라호마 주립대학교
컴퓨터과학과 박사후과정

2008년~현 재 영남대학교 정보통신공학과 연구교수
관심분야: 병렬 및 분산처리, 계산이론, 알고리즘, 네트워크
설계 및 분석



김 성 원

e-mail : swon@ynu.ac.kr

1990년 서울대학교 제어계측공학과(공학사)
1992년 서울대학교 제어계측공학과
(공학석사)
2002년 서울대학교 전기컴퓨터공학부
(공학박사)

2005년~현 재 영남대학교 정보통신공학과 부교수
관심분야: 무선 네트워크, 모바일 네트워크, 임베디드시스템



이 형 옥

e-mail : oklee@sunchon.ac.kr

1994년 순천대학교 전산학과(이학사)
1996년 전남대학교 전산통계학과(이학석사)
1999년 전남대학교 전산통계학과(이학박사)
1999년~2002년 한국정보사회진흥원
(선임연구원)

2006년~2007년 University of Texas at Dallas 교환교수
2002년~현 재 순천대학교 컴퓨터교육과 부교수
관심분야: 병렬 및 분산처리, 계산이론, 알고리즘, 네트워크
설계 및 보안