Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT
ACADEMIC PRESS
J. Parallel Distrib. Comput. 63 (2003) 1164–1172
Journal of Parallel and Distributed Computing
http://www.elsevier.com/locate/jpdc

Higher dimensional hexagonal networks

Fabián García,[a] Julio Solano,[a] Ivan Stojmenovic,[b,*] and Milos Stojmenovic[b]

[a] DISCA, IIMAS, UNAM, Circuito Escolar S/N, Ciudad Universitaria, Coyoacán, México City 04510, Mexico
[b] Deptartment of Computer Science, University of Ottawa, SITE, 550 Cumberland St., Ottawa, ON, Canada K1N 6N5

Received 11 February 2003; revised 18 June 2003

Abstract

We define the higher dimensional hexagonal graphs as the generalization of a triangular plane tessellation, and consider it as a multiprocessor interconnection network. Nodes in a k-dimensional (k-D) hexagonal network are placed at the vertices of a k-D triangular tessellation, so that each node has up to $2k+2$ neighbors. In this paper, we propose a simple addressing scheme for the nodes, which leads to a straightforward formula for computing the distance between nodes and a very simple and elegant routing algorithm. The number of shortest paths between any two nodes and their description are also provided in this paper. We then derive closed formulas for the surface area (volume) of these networks, which are defined as the number of nodes located at a given distance (up to a given distance, respectively) from the origin node. The number of nodes and the network diameter under a more symmetrical border conditions are also derived. We show that a k-D hexagonal network of size t has the same degree, the same or lower diameter, and fewer nodes than a $(k+1)$-D mesh of size t. Simple embeddings between two networks are also described. That is, we show how to reduce the dimension of a mesh by removing some nodes, and converting it into a hexagonal network, while preserving the simplicity of basic data communication schemes such as routing and broadcasting.

Keywords: Interconnection networks; Hexagonal networks; Addressing; Routing

1. Introduction

Direct interconnection networks can be modeled by graphs, with nodes and edges corresponding to processors and communication links between them, respectively. A survey of these networks is given in [23]. This paper proposes a new direct interconnection network model. It also studies addressing and routing schemes for some topological properties of the new model, such as: degree (maximal number of edges from a node), node and edge symmetry and surface area (defined as the number of nodes at a given distance from the origin node).

There exist three regular plane tessellations, composed of the same kind of regular (equilateral) polygons: triangular, square, and hexagonal. They are the basis for the designs of direct interconnection networks with highly competitive overall performance. Mesh connected computers and tori (tori are meshes with added links between the first and the last processor in any row or column, that is, in any direction for the higher dimensional case) are based on regular square tessellations, and are popular and well-known models for parallel processing. Their extension, the m-ary k-cube, has been used as the underlying topology for most practical multicomputers (e.g. J-machine [17], iWarp [19], Ncube-2 [16], Cray T3E [1] and Cray T3D [12]), and has been extensively studied in the literature. The topological properties and routing algorithms for the m-ary k-cubes have been extensively studied in the past [10]. An expression for the surface area of a t-ary k-cube is provided in [2,3].

Hexagonal and honeycomb networks are based on regular triangular and hexagonal tessellations, respectively. The inconsistency in the name selection (note that a hexagonal network is not based on a hexagon, but on a triangular tessellation) is due to the duality of the two tessellations (one can be obtained from the other by joining the centers of the neighboring polygons) and the name selection taken in the past, which other authors kept afterwards.

*Corresponding author. www.site.uottowa.ca/∼ivan.
E-mail addresses: fabian@uxdea4.iimas.unam.mx (F. García), julio@uxdea4.iimas.unam.mx (J. Solano), ivan@site.uottawa.ca (I. Stojmenovic).

0743-7315/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2003.07.001

Honeycomb and hexagonal networks have been studied in a variety of contexts. They have been applied in chemistry to model benzenoid hydrocarbons [24], in image processing, in computer graphics [13], and in cellular networks [11]. The Honeycomb architecture was proposed in [22], where a suitable addressing scheme together with routing and broadcasting algorithms were investigated. Some topological properties and communication algorithms for the honeycomb network and tori have been also investigated in [6,14,15,18]. Higher dimensional honeycomb networks have been defined in [7] as a generalization of the plane honeycomb networks. Addressing, routing and broadcasting algorithms have been also proposed. The network cost, defined as the product of the network degree and its diameter, has been shown to perform better for the honeycomb network than for the mesh multiprocessor network [22,7].

The (two-dimensional) hexagonal torus has been used in the HARTS project [21]. An addressing scheme for the processors, and the corresponding routing and broadcasting algorithms for a hexagonal interconnection network have been proposed by Chen et al. [8]. The performance of hexagonal networks has been further studied in [9,20]. The approach proposed in [8] is a cumbersome addressing scheme which has lead to a page long complex routing algorithm, and similarly to a complex broadcasting scheme. Consequently, the lack of a convenient addressing scheme and the absence of elegant routing and broadcasting algorithms (basic in the design of a commercial network) has discouraged further research on this type of network. Carle and Myoupo [5] recently revisited this network and attempted to simplify the addressing, routing and broadcasting schemes given in [8] with partial success. They suggested a co-ordinate system for hexagonal networks that uses two axes, at 120° between them, which are parallel to two out of the three edge directions. Using this scheme, they have described routing and broadcasting algorithms for the network. However, their scheme exhibits asymmetry which complicates the routing algorithm (for that reason, the routing algorithm is even omitted from [5] and a reference to their technical report is given instead). Their broadcasting algorithm, on the other hand, is very elegant. A variant of this addressing scheme has been proposed by Zhang [25] in the context of a distance-based location update scheme in cellular networks. A formula for the distance between two base stations is given, and a corresponding routing algorithm is proposed. Both addressing and routing schemes in [25] have been further simplified in [11]. These schemes have been then applied in [11] to solve the distance calculation for location update and connection rerouting problems in cellular networks.

Fig. 1 illustrates the co-ordinate system for the honeycomb network proposed by Stojmenovic [22], and adopted for 2-D hexagonal networks by García et al. [11]. In this scheme, three axes, x, y, and z, parallel to the three edge directions, and at mutual angle of 120 between any two of them are introduced. Let $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ be the three unit vectors in these axes. These three vectors are, obviously, not independent. More precisely, they are related by $\boldsymbol{i}+\boldsymbol{j}+\boldsymbol{k}=0$. However, this redundancy greatly simplifies the addressing, the distance formula, and the routing algorithm. Details for the addressing schemes for the honeycomb and the hexagonal networks can be found in [22,11]. This co-ordinate system has been generalized by Carle et al. [7] for higher dimensional honeycomb networks.

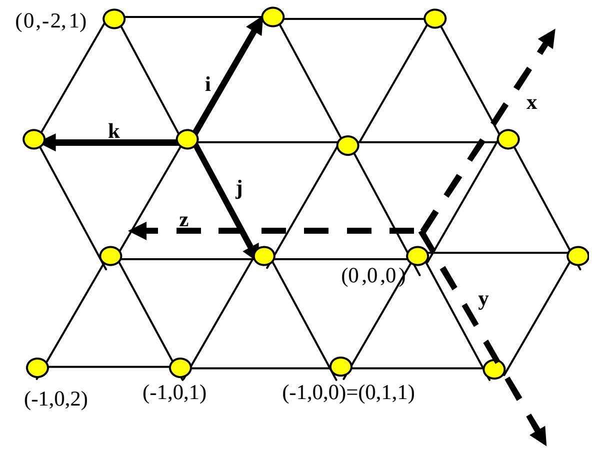

Fig. 1. Addressing scheme for hexagonal networks.

Higher dimensional mesh connected computers are straightforward generalizations of 2-D meshes since a regular square plane tessellation easily generalizes in space. That is, the higher dimensional space can be partitioned into higher dimensional cubes, or hypercubes. One such hypercube consists of all nodes whose address is $(x_1, \dots, x_k)$ where $a_i \leqslant x_i \leqslant a_i + 1$ for $1 \leqslant i \leqslant k$. In other words, each of the k co-ordinates can be chosen between two consecutive integers, giving an overall 2^k nodes for one 'cell' of the mesh. Unfortunately, such an analogy does not exist for higher dimensional triangular or its dual hexagonal tessellations, as shown in [7]. It is well known that the space can only be filled with hypercubes, but not with regular tetrahedrons. Nevertheless, it has been shown in [7] that there exist $k+1$ vectors $\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_k, \boldsymbol{X}_{k+1}$ in a k-dimensional (k-D) space such that $\boldsymbol{X}_1 + \boldsymbol{X}_2 + \cdots + \boldsymbol{X}_k + \boldsymbol{X}_{k+1} = 0$ and the dot product $\boldsymbol{X}_i \boldsymbol{X}_j = -1/k$ for any pair for distinct co-ordinates i and j (in other words, these $k+1$ vectors are fully symmetrical in space). For instance, in three dimensions, these vectors are those perpendicular to the tetrahedron faces, and their exact values are listed in [7]. The use of these $k+1$ vectors is the only common idea between this article and [7]. Subsequent details differ.

Carle and Myoupo [5] defined a 3-D hexagonal graph as a generalization of the hexagonal network in a plane. Carle [4] further generalized the network for higher dimensions. Each node in a k-D hexagonal graph is defined in [5,4] as the node with integer co-ordinates $(x_1, \ldots, x_k)$. The network has two kinds of edges. Two nodes $(x_1, \ldots, x_k)$ and $(x'_1, \ldots, x'_k)$ are connected by an edge if their addresses differ by one in exactly one direction, that is, $|x_1 - x'_1| + \cdots + |x_k - x'_k| = 1$. Therefore, higher dimensional meshes are sub-graphs of higher dimensional hexagons, as defined in [5,4] (with network borders being defined in a different manner). Some diagonal edges are added to the network as follows. Two nodes are connected by an edge when there exist exactly two co-ordinates i and j such that $(x_i - x'_i)(x_j - x'_j) = 1$ (the product is otherwise equal to 0). Therefore, each node has $k(k-1)/2$ diagonal edges. The overall number of edges at each node (that is, the network degree) is therefore quadratic with the network dimension, as opposed to being linear for higher dimensional meshes and honeycombs. This is a significant drawback for the proposed generalization [4,5]. Moreover, no formula for the distance between two nodes in the network has been given in [4,5], and the routing scheme offered is not shown to follow the shortest path between two nodes. Clearly, an adapted routing scheme should follow the shortest path between any two nodes. The addressing scheme proposed in [4,5] for the network is therefore sophisticated, has an excessive degree, and is discouraging for further study of the network.

In this paper, we suggest a variation for the generalization of the plane hexagonal graph to a higher dimensional hexagonal network. This new generalization presents a linear degree (more precisely, $2k+2$) and has a very simple addressing scheme, which leads to a straightforward formula for the distance between two nodes and to a straightforward routing algorithm, not only for one shortest path but also for all the shortest paths. Further, it allows to count and list the number of shortest paths between two nodes and to count the number of nodes at a given distance from a given node (that is, to find a closed formula for the surface area of the network). Thus, this new generalization will be shown to be a viable alternative to the well-known higher dimensional mesh connected computer, and the higher dimensional honeycomb network. We will also show embeddings and analogies between the two networks, and a simple broadcasting algorithm for the hexagonal networks. The proposed network can be physically implemented with existing technology like any other interconnection network, with processors replacing nodes and communication links added between processors according to the graph definition of higher dimensional hexagonal network.

2. An addressing scheme for higher dimensional hexagonal networks

The goal of this paper is to describe a higher dimensional hexagonal network as a generalization of 2-D one, preserving desirable properties such as low diameter, small degree and symmetries. We observe that, starting from an origin node, all nodes of a 2-D hexagonal network are obtained by adding unit size vectors along three symmetrically positioned directions in the plane. We will therefore extend this construction to k-D space. Starting from the origin, unit size vectors will be added and will lead to new nodes along $k+1$ symmetrically positioned vectors (in both positive or negative orientations along these vectors), so that each node has at most $2k+2$ neighbors. Two nodes are neighbors in such graph if and only if they differ by one of such $2k+2$ unit size vectors (in vector sense). While such construction is intuitively clear, the addressing of nodes in such networks is not obvious. Our main contribution is to propose a co-ordinate system that can be used to assign *id*s to the nodes in a higher dimensional hexagonal network. We will then show that the distance between any two nodes can be computed easily if the proposed node *id* assignment scheme is utilized. Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k, \boldsymbol{X}_{k+1}$ be $k+1$ (unit) vectors in a k-D space such that $\boldsymbol{X}_1 + \boldsymbol{X}_2 + \cdots + \boldsymbol{X}_k + \boldsymbol{X}_{k+1} = 0$ (the dot product property is not needed in the sequel). The nodes in a k-D hexagonal network can be defined as follows.

Definition 1. Choose any node as the origin and assign $(0, 0, \ldots, 0)$ ($k+1$ zeros) as its address. For any other node A on the network, if there is a path from the origin to node A, and the path has altogether $|a_i|$ units of vector $\text{sign}(a_i)\boldsymbol{X}_i$ (that is, $\boldsymbol{X}_i$ for $a_i > 0$ and $-\boldsymbol{X}_i$ otherwise), $1 \leqslant i \leqslant k+1$, then an address for node A is $(a_1, \ldots, a_{k+1}) = a_1\boldsymbol{x}_1 + \cdots + A_{k+1}\boldsymbol{X}_{k+1}$.

Clearly, more than one $(k+1)$-tuple point corresponds to the same node. For example, in Fig. 1, $(-1{,}0{,}0) = (0{,}1{,}1)$ since $-\boldsymbol{i} = \boldsymbol{j} + \boldsymbol{k}$ and thus the two paths, the direct one, and using the two other sides of the equilateral triangle, end in the same point. In general, we have $(a_1, \ldots, a_{k+1}) = (a'_1, \ldots, a'_{k+1}) \Leftrightarrow a_1\boldsymbol{X}_1 + \cdots + a_{k+1}\boldsymbol{X}_{k+1} = a'_1\boldsymbol{X}_1 + \cdots + a'_{k+1}\boldsymbol{X}_{k+1}$. In particular, if $(a_1, \ldots, a_{k+1}) = (a'_1, \ldots, a'_{k+1})$, then there exists an integer r such that $a'_i = a_i + r$, $1 \leqslant i \leqslant k+1$. Combining those two facts, we have the following. If $(a_1, \ldots, a_{k+1})$ is an address for node A, then all possible addresses for node A are of the form $(a_1 + r, \ldots, a_{k+1} + r)$ for any integer r. Starting from the nonunique addressing, we shall now find a way to arrive at a unique node address. We will first define the *shortest path form* for the address, and then will define the *distinguished shortest path form* for the node address.

Definition 2. An address $(a_1, \ldots, a_{k+1})$ for node A is of the shortest path form if there is a path from the origin to node A, consisting of $|a_i|$ units of either vector $\boldsymbol{X}_i$, (for $a_i > 0$) or vector $-\boldsymbol{X}_i$ (for $a_i < 0$), $1 \leqslant i \leqslant k+1$, and the path has the shortest possible length.

Corollary 1. *The distance between two nodes A and B, is*$|a_1| + \cdots + |a_{k+1}|$, *where* $B - A = (a_1, \ldots, a_{k+1})$ *is in the shortest path form. Therefore the shortest path form* $(a_1, \ldots, a_{k+1})$ *minimizes* $|a_1| + \cdots + |a_{k+1}|$.

We shall now investigate the uniqueness of the shortest path form. Let *np*, *nn*, and *nz* denote the number of positive, negative and zero co-ordinates in a shortest path form $(a_1, \ldots, a_{k+1})$, respectively. Clearly $np + nn + nz = k + 1$.

Theorem 1. *If k is an even number then* $(a_1, \ldots, a_{k+1})$ *is in a shortest path form if and only if* $nz \geqslant 1$, $np \leqslant k/2$, $nn \leqslant k/2$. *The shortest path form is also the unique shortest path form. The number of shortest paths between two nodes A and B with* $B - A = (a_1, \ldots, a_{k+1})$ *in the shortest path form is* $(|a_1| + \cdots + |a_{k+1}|)!/(|a_1|! \ldots |a_{k+1}|!)$.

Proof. We will prove the theorem by contradiction. Assume that $(a_1, \ldots, a_{k+1})$ is in a shortest path form, and $np > k/2$. Without loss of generality, we assume that $a_i > 0$, $1 \leqslant i \leqslant np$. Since $a_1\boldsymbol{X}_1 + \cdots + a_{k+1}\boldsymbol{X}_{k+1} = a_1\boldsymbol{X}_1 + \cdots + a_{k+1}\boldsymbol{X}_{k+1} - (\boldsymbol{X}_1 + \cdots + \boldsymbol{X}_{k+1}) = (a_1 - 1)\boldsymbol{X}_1 + \cdots + (a_{k+1} - 1)\boldsymbol{X}_{k+1}$, $(a_1 - 1, \ldots, a_{k+1} - 1)$ is another address for node A. The length of the path corresponding to $(a_1 - 1, \ldots, a_{k+1} - 1)$ is $|a_1 - 1| + \cdots + |a_{k+1} - 1| \leqslant |a_1| - 1 + \cdots + |a_{np}| - 1 + |a_{np+1}| + 1 + \cdots + |a_{k+1}| + 1 = |a_1| + \cdots + |a_{k+1}| - np + (k + 1 - np) < |a_1| + \cdots + |a_{k+1}|$ since $k + 1 < 2np$. It means that the path corresponding to $(a_1 - 1, \ldots, a_{k+1} - 1)$ is shorter than the shortest path, a contradiction. Therefore we proved that $np \leqslant k/2$, and similarly $nn \leqslant k/2$. Then $nz = k + 1 - np - nn \geqslant k + 1 - 2k/2 = 1$. Assume now that $nz \geqslant 1$, $np \leqslant k/2$, $nn \leqslant k/2$ is satisfied. Then for node $(a_1 + r, \ldots, a_{k+1} + r)$ and $r > 0$ (the proof for $r < 0$ is similar) we get $|a_1 + r| + \cdots + |a_{k+1} + r| \geqslant |a_1| + \cdots + |a_{k+1}| + r(np + nz - nn) > |a_1| + \cdots + |a_{k+1}|$(since $np + nz - nn = k + 1 - 2nn > 0$), thus $(a_1 + r, \ldots, a_{k+1} + r)$ is not in the shortest path form. □

However, the shortest path form is not always unique for k odd. For example, for $k = 3$, (4,4,0,0) = (3,3,−1,−1) = (2,2,−2,−2) = (1,1,−3,−3) = (0,0,−4,−4), and all these representations have the shortest path length 8. Using similar arguments as in the proof of Theorem 1, we can prove the following corollaries.

Corollary 2. *A node address* $(a_1, \ldots, a_{k+1})$ *is in the shortest path form if and only if* $np \leqslant (k+1)/2$ *and* $nn \leqslant (k+1)/2$. *This is valid for both cases of k being an odd or an even number.*

Corollary 3. *If k is an odd number then a shortest path form* $(a_1, \ldots, a_{k+1})$ *is unique if and only if* $nz > |np - nn|$.

We shall now define the median for an address $(a_1, \ldots, a_{k+1})$ (not necessarily in the shortest path form). Let $(b_1, \ldots, b_{k+1})$ be the permutation of elements $(a_1, \ldots, a_{k+1})$ in sorted order, that is $b_1 \leqslant \cdots \leqslant b_{k+1}$. The median is any integer m which satisfies $b_{(k+1)/2} \leqslant m \leqslant b_{(k+1)/2+1}$. If $(a_1, \ldots, a_{k+1})$ is in the shortest path form then an alternative definition of median can be given as follows. Let *mn* and *mp* be the maximal negative and minimal positive elements of sequence $(a_1, \ldots, a_{k+1})$, respectively; if there are less than $(k+1)/2$ negative (positive) elements then *mn* (*mp*, respectively) is set to 0. The median is any integer m which satisfies $mn \leqslant m \leqslant mp$.

Corollary 4. *If* $(a_1, \ldots, a_{k+1})$ *is an address for node A then* $(a_1 - m, \ldots, a_{k+1} - m)$ *is an address for node A in a shortest path form for any median m.*

Corollary 5. *The number of shortest paths between two nodes A and B with* $B - A = (a_1, \ldots, a_{k+1})$ *in a shortest path form is* $\sum_{m=mn}^{mp} (|a_1 - m| + \cdots + |a_{k+1} - m|)!/(|a_1 - m|! \ldots |a_{k+1} - m|!)$.

Theorem 2. The address of each node can be uniquely represented in the *distinguished shortest path form* $(a_1, \ldots, a_{k+1})$, where $np \leqslant \lfloor (k+1)/2 \rfloor$, $nn \leqslant \lfloor k/2 \rfloor$, and $nz \geqslant 1$.

Proof. For k even the theorem is equivalent to Theorem 1. If k is odd then Corollary 4 can be applied for m=mn, leading to a (unique) distinguished shortest path representation. □

3. Routing in higher dimensional hexagonal networks

We shall now describe a corresponding routing algorithm from a source S to a destination D, where $D - S = (a_1, \ldots, a_{k+1})$ is in a shortest path form. The algorithm reduces the path length at each step. The routing algorithm is quite simple and can be described as follows:

Algorithm ***route***(S, D) (*$D - S = (a_1, \ldots, a_{k+1})$ is in a shortest path form *) {
  For $i = 1$ **to** $k + 1$ **do**
    For $j = 1$ **to** $|a_i|$ **do** Send message $|a_i|$ times in direction sign(a_i) $\boldsymbol{X}_i$ }

The algorithm may be easily modified to offer flexibility in selecting one of the possible shortest paths, whose number is given in Corollary 5. At each step, the message can be sent along any edge that will shorten the destination distance. These edges are easy to detect. Such flexible routing is important in case of congestion in a corresponding interconnection network. The level of congestion at each node corresponds to the amount of traffic at that node (for instance, the queue length in interconnection networks). Various heuristics for selecting a path based on the network conditions can be derived from the analysis of congestion, but this is beyond the scope of this paper. Note that the algorithm assumes the availability of intermediate nodes in the network. Given some boundary conditions for the nodes, the algorithm might need to be modified to avoid missing nodes.

4. The surface area of higher dimensional hexagonal networks

We shall now count the number of nodes at a given distance n from the origin node (0,…,0). Because of the symmetry, the same number is obtained from any node, subject to the network border conditions that can reduce the count. The surface area at distance n is equal to the number of nodes $(a_1, \ldots, a_{k+1})$ in the unique shortest path representation which satisfy $|a_1| + \cdots + |a_{k+1}| = n$ (Corollary 1). Let $C(p,q) = p!/(q!(p-q)!)$ be the binomial coefficient. In order to make the expression clearer, let $np\max = \min(k+1-nz, \lfloor (k+1)/2 \rfloor)$ and $np\min = \max(0, \lfloor (k+3)/2 \rfloor - nz)$.

Theorem 3. *The surface area of a k-D hexagonal network at distance n is*

$$\sum_{nz=1}^{k} \{C(k+1, nz)C(n-1, k-nz) \times \sum_{np=npmin}^{npmax} C(k+1-nz, np)\}.$$

Proof. The number of elements nz in sequence $(a_1, \ldots, a_{k+1})$ which are equal to zero can be between 1 and k, according to Theorem 2, and the count is considered separately for each possible value of nz. There are $C(k+1, nz)$ ways to choose nz zero positions in vector $(a_1, \ldots, a_{k+1})$. The remaining elements in the vector are $\neq 0$. Integer compositions of n into s parts are representations of n as sums of s positive integers, called parts; that is, $n = u_1 + \cdots + u_s, u_i > 0,\ 1 \leqslant i \leqslant s$. It is well-known that the number of such compositions is $C(n-1, s-1)$. In our case $s = k+1-nz$. Each of these nonzero parts can be either a positive or a negative number, and the numbers np and nn of the positive and negative numbers must be chosen in accordance to Theorem 2. Since $nn = k+1-np-nz$, it suffices to find the bounds for np. It is bounded above by both $k+1-nz$ and $\lfloor (k+1)/2 \rfloor)$, thus its maximum is $npmax$=min$(k+1-nz, \lfloor (k+1)/2 \rfloor)$. Since $nn \leqslant \lfloor k/2 \rfloor$, it follows that $np = k+1-nz-np \geqslant k+1-nz-\lfloor k/2 \rfloor = \lfloor (k+3)/2 \rfloor - nz$. It is also bounded below by 0, therefore $npmin$=max(0, $\lfloor (k+3)/2 \rfloor - nz$). Among $k+1-nz$ nonzero parts, np parts are selected to be positive, others are negative. The theorem then follows. □

Table 1
Surface area of a k-D hexagonal network at distance n

k	n = 1	2	3	4	5	6	7
1	2	2	2	2	2	2	2
2	6	12	18	24	30	36	42
3	8	26	56	98	152	218	296
4	10	50	150	340	650	1110	1750
5	12	72	272	762	1752	3512	6372
6	14	98	462	1596	4410	10,374	21,658
7	16	128	680	2722	8679	23,331	55,073
8	18	162	978	4482	16,470	50,718	135,702
9	20	200	1340	6800	27,752	94,940	281,360

Table 1 gives the surface areas of k-D hexagonal networks at distance n for small values of k and n. The volume of the network can be then calculated as the summation over distances up to n, applied on formula in Theorem 3. The network can be bounded using the maximum distance n from the origin as a criterion for a node to belong to the network. The diameter of the network is obviously $2n$, which is the distance between nodes $(n, 0, \ldots, 0)$ and $(-n, 0, \ldots, 0)$. The network degree $2k+2$ and diameter $2n$ can be considered as a function of the volume, and be compared with similar analysis for the honeycomb and mesh connected computers, in order to compare network costs. The closed formulas for comparison seem difficult to obtain, so one can rely on computer data to compare small size networks and derive conclusions.

5. A border with better connectivity

A k-D hexagonal network can be defined with the origin as a center and with all nodes up to certain distance from the origin as part of the network. However, using such border definitions, some nodes have only one neighbor. For example, node $(n, 0, \ldots, 0)$ is connected only to node $(n-1, 0, \ldots, 0)$ (for $k > 2$). We shall define now a ‘friendlier’ border condition. Let a k-

D hexagonal network of size t be defined as the set of nodes whose unique shortest path form $(a_1, \ldots, a_{k+1})$ satisfies $|a_i| \leqslant t$, $1 \leqslant i \leqslant k+1$.

Theorem 4. *The diameter of a k-D hexagonal network of size t is* $4t\lfloor (k+1)/2 \rfloor$.

Proof. The distance between nodes $(-t, \ldots, -t,\ t, \ldots, t)$ and $(t, \ldots, t,\ -t, \ldots, -t)$ (with $\lfloor (k+1)/2 \rfloor$ positive and negative parts in each, and with an additional component equal to 0 at the end of both for k even) is $4t\lfloor (k+1)/2 \rfloor$. The distance cannot be larger since for the k even $\geqslant 1$ part in both vectors must be 0. $\square$

Theorem 5. *The number of nodes in a k-D hexagonal network of size t is*

$$1 + \sum_{nz=1}^{k} \{C(k+1, nz) t^{k+1-nz} \sum_{np=npmin}^{npmax} C(k+1-nz, np).$$

Proof. The count is considered separately for each nz. There are $C(k+1, nz)$ ways to choose nz zero positions in vector $(a_1, \ldots, a_{k+1})$. Then $1 \leqslant |a_i| \leqslant t$ for each remaining nonzero part. There are t^{k+1-nz} such variations. Next, np out of $k+1-nz$ positions have positive parts, and the remaining parts are negative. There are $C(k+1-nz, np)$ such choices of np positions. Finally, np is between $npmin$=max$(0, \lfloor (k+3)/2 \rfloor - nz)$ and $npmax$=min$(k+1-nz, \lfloor (k+1)/2 \rfloor)$. $\square$

Table 2 gives volumes of k-D hexagonal networks of size t for small values of k and t. For comparison, note that a similar volume for a k-D cube of size t is $(2t+1)^k$. When the volumes of hexagonal and cubic networks are divided, for the same values of t and k, the ratio appears to be around 2, and appears to be slowly increasing with increasing values of t and k. This means that hexagonal networks are approximately twice denser than cubic networks.

As an example, for $k=3$ and $t=1$, the 39 nodes in the 3-D hexagonal network of size 1 are the following: (0,0,0,0), (0,0,0,1), (0,0,0,−1), (0,0,1,0), (0,0,−1,0), (0,1,0,0), (0,−1,0,0), (1,0,0,0), (−1,0,0,0), (0,0,1,1), (0,0,1,−1), (0,0,−1,1), (0,1,0,1), (0,1,0,−1), (0,−1,0,1), (1,0,0,1), (1,0,0,−1), (−1,0,0,1), (0,1,1,0), (0,1,−1,0), (0,−1,1,0), (1,0,1,0), (1,0,−1,0), (−1,0,1,0), (1,1,0,0), (1,−1,0,0), (−1,1,0,0), (1,1,−1,0), (1,−1,1,0), (−1,1,1,0), (1,1,0,−1), (1,−1,0,1), (−1,1,0,1), (1,0,1,−1), (1,0,−1,1), (−1,0,1,1), (0,1,1,−1), (0,1,−1,1), (0,−1,1,1). Fig. 2 shows the network with all the edges (the figure is not a projection of the corresponding 3-D object since the real projection results in a dense graph with limited clarity due to all the points being concentrated near each other).

Table 2
Volume of k-D hexagonal networks of size t

k	t = 1	2	3	4	5	6
1	3	5	7	9	11	13
2	13	37	73	121	181	253
3	39	185	511	1089	1991	3289
4	141	1141	4441	12,201	27,301	53,341
5	423	5705	31,087	109,809	300,311	693,433
6	1429	32,845	252,169	1.1E6	3.8E6	1.0E7
7	4254	163,361	1.7E6	1.0E7	4.2E7	1.3E8
8	13,981	911,845	1.4E7	1.0E8	5.2E8	2.0E9
9	41,898	4.5E6	9.6E7	9.3E8	5.7E9	2.6E10

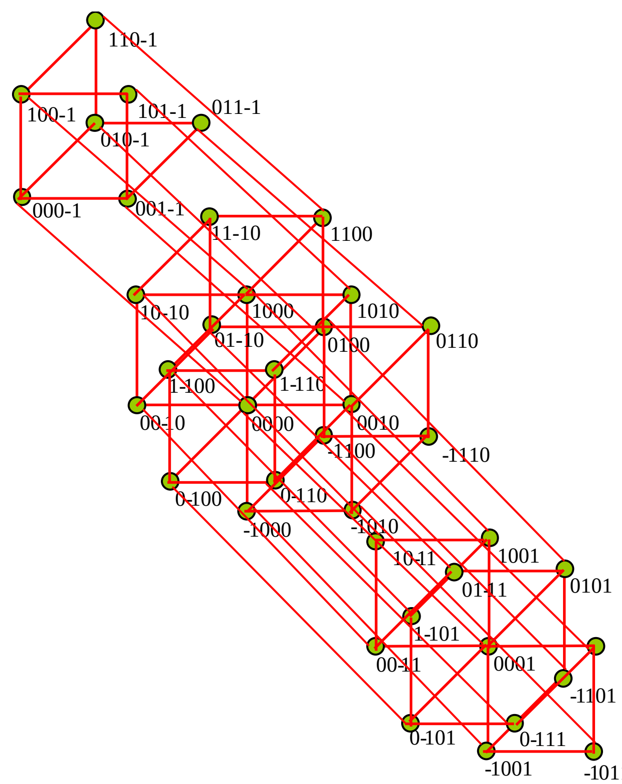

Fig. 2. Three-dimensional hexagonal network of size 1.

6. Embeddings between hexagonal and mesh networks

We will now elaborate on the similarity of hexagonal and mesh connected networks, that is, networks based on a triangular and a square tessellation, respectively. First, observe that the degree of a k-D hexagonal network is $2k+2$, which is the same as degree of a $(k+1)$-D mesh connected computer. Let a $(k+1)$-D mesh connected computer of size t be defined as the set of nodes $(a_1, \ldots, a_{k+1})$ satisfying $|a_i| \leqslant t, 1 \leqslant i \leqslant k+1$. The diameter of the network is $2(k+1)t$, which is the distance between two corner nodes $(-t, \ldots, -t)$

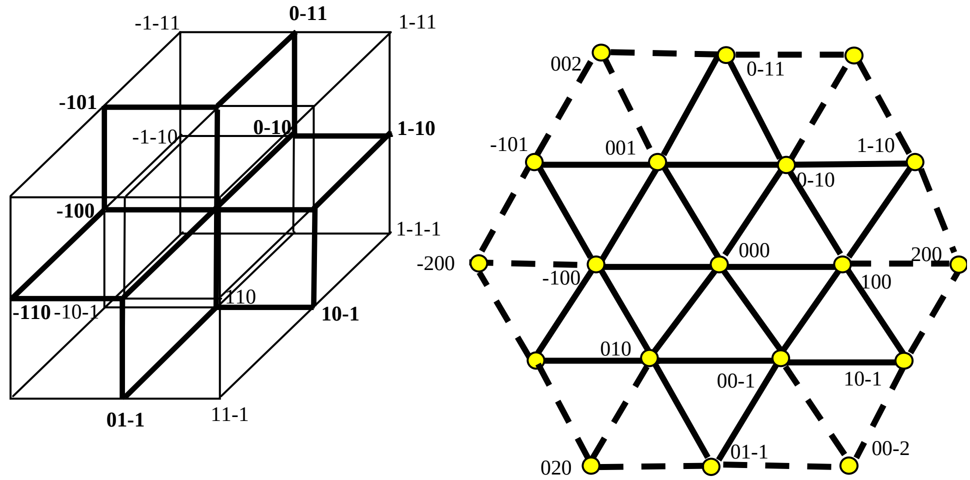

Fig. 3. Embedding of a k-D hexagonal network into a $(k+1)$-D mesh connected computer.

and $(t, \ldots, t)$. On the other hand, the diameter of a k-D hexagonal network of size t is $4t\lfloor (k+1)/2 \rfloor$ (Theorem 4). For k odd the diameters are the same, while for k even they are lower than the diameter of the corresponding mesh.

The mapping between the two networks is straightforward, since we use the same addressing scheme, even the same border conditions (when a border with better connectivity is used). The only difference is that the k-D hexagonal network addresses have additional restrictions, according to Theorem 2. Therefore the number of nodes in a k-D hexagonal network of size t is less than the number of nodes in a $(k+1)$-D mesh connected computer of size t. Moreover, the distances in a hexagonal network are preserved or reduced, since some nodes that are not neighbors in the mesh can become neighbors in hexagonal network (e.g. nodes (0,0,1) and (0,$-$1,0)). The mapping is illustrated in Fig. 3, which shows 27 nodes of a 3-D mesh of size 1, with nodes and edges belonging to the corresponding 2-D hexagonal network of size 1 marked in bold. The 13 nodes of the later network are also drawn separately, using bold edges. Note that the remaining nodes of the mesh network have also their maps, in a hexagonal network either of size t, or size $2t$. For example, in Fig. 3, the full projection of a mesh of size 3 is a hexagonal networks with all nodes at a distance of at most three from the origin (additional edges are marked in dashed lines in Fig. 3, and there are six new nodes where these edges originate from). Several nodes of the mesh can map into the same node of hexagonal network (e.g. nodes (1,1,1), (0,0,0) and ($-$1,$-$1,$-$1) all map to (0,0,0)). The six corner nodes of the mesh network are mapped into six nodes outside of size 1 hexagonal network (connected by dashed lines to the rest of the network).

The described mapping leads to two embeddings between the two networks. The embedding from a $(k+1)$-D mesh of size t to a k-D hexagonal network of size $2t$ has an optimal dilation (where dilation is the maximum ratio of distances between two nodes in the host network and their corresponding nodes in the guest network). Neighboring nodes in the hexagonal network may either remain neighbors in the mesh, or be at a distance k (thus the dilation in this direction is not optimal). The later case appears when a node, after increasing one co-ordinate by 1, does not satisfy the criterion in Theorem 2 (this may happen only when a zero co-ordinate becomes 1). It can be shown that, when one subsequently reduces all co-ordinates by 1, the new address satisfies the criterion, and therefore the k co-ordinates of the two neighboring nodes in the hexagonal network differ by 1, and one co-ordinate is the same, leading to a distance k between the two nodes in the mesh. For example, node (0,0,1) has a neighbor (1,0,1) in the mesh, and later is addressed as (0,$-$1,0) in the hexagonal network, with a distance $k=2$ between the two nodes (0,0,1) and (0,$-$1,0) within the mesh network. The case of reducing one co-ordinate by one is symmetric to this one.

7. Broadcasting in hexagonal networks

The mapping between mesh and hexagonal networks may be used to describe some basic data communication schemes for a hexagonal network, following an analogous design for the mesh-connected computers. As an illustration, we will design a broadcasting algorithm. In a broadcasting task, one node sends the same message to all the nodes in the network. In an efficient broadcasting algorithm, each node receives the message exactly once.

In a mesh-connected computer, broadcasting is performed by 'spreading' the information within nodes sharing the same co-ordinates as origin node except the first co-ordinate, then all these flood nodes sharing all co-ordinates except the first two using the second dimension for distribution, etc. The same algorithm can be applied for hexagonal networks, with several changes. First, the last co-ordinate in a hexagonal addressing is not needed. Next, the borders of a hexagonal network require some changes for the 'spreading' algorithms in each dimension. Let us illustrate a broadcasting algorithm on the hexagonal network in Fig. 3. Suppose that node (0,1,0) is the source node. The message is first broadcasted along the first dimension, and nodes (–1,1,0), (0,0,−1) = (1,1,0), (1,0,−1) = (2,1,0) receive it. Then each of these nodes forwards the message in a similar way using the second co-ordinate. All nodes except (−2,0,0), (−1,0,1) = (−2,−1,0), and (0,0,2) = (−2,−2,0) receive the message. These nodes did not receive the message since node (−2,1,0) that was 'responsible' for them after the first dimension is outside of the borders of the hexagonal network. In order to fix this problem, each message should carry the direction and distance 'traveled' in each dimension, so that the path along each dimension can be extended when new nodes are discovered. In the example discussed, node (−1,0,0) discovers that its neighbor, node (−2,0,0) did not receive the message since it was not previously extended beyond –1 in the first dimension. Node (−2,0,0), upon receiving the message, forwards it to two other nodes, (−1,0,1) and (0,0,2), to complete the broadcast.

In order to simplify the algorithm description, let us label each link at a given node A as follows. If $B - A = (0, \ldots, 0, 1, 0, \ldots, 0)$ (the ith co-ordinate is equal to one) then B is a neighbor on link i. If $B - A = (0, \ldots, 0, -1, 0, \ldots, 0)$ (the ith co-ordinate is equal to -1) then B is a neighbor on link $-i$. In addition to the message, each node also receives an integer vector $j=(j_1, j_2, \ldots, j_k)$ which indicates the borders met by the copy of the message on its way to a given node. In the example considered, node (−1,1,0) recognizes that it is a border node along the first dimension, and will forward vector (−1,0,0) where –1 refers to the border met on link −1. Similarly, node (1,0,−1) will forward vector (1,0,0). The vector elements are always 0, 1 or –1. The algorithm can be formally described as follows.

Algorithm ***broadcast***(S) (* $S= (a_1,\ldots,a_{k+1})$ is in a shortest path form *) {

S forwards to its two neighbors on links 1 and -1, which receive $j= (0,0,\ldots,0)$; if there is no neighbor on link 1, S sets $j=(1,0,\ldots,0)$, similarly to $j=(-1,0,\ldots,0)$ if there is no neighbor on link -1;

For each node A receiving the message **do** {

Let the message be received on link i or $-i$ and let the received vector be $j=(j_1,j_2,\ldots,j_k)$;

If A has a neighbor on the same link as the received message (that is, another neighbor in the same direction as the incoming message)

then A forwards the message to that neighbor, with the same vector j **else** it changes the i-th coordinate to 1 or -1, where the sign is the sign of the incoming link;

For $m=1$ to $i-1$ **do**

If $j_m=\pm 1$ and A has a neighbor on link $\pm m$

then A forwards the message on link $\pm m$ and changes the m-th coordinate in j to 0 for further forwarding.

For $m=i+1$ to k **do**

A forwards the message and vector j to its two neighbors on links m and $-m$, if they exist (otherwise, it changes the m-th coordinate in j to ± 1, respectively);

}

8. Conclusion

It is an interesting open problem to extend our addressing and routing schemes and define a higher dimensional hexagonal tori as an alternative to the popular t-ary k-cubes. A similar open problem is to define a higher dimensional honeycomb tori, extending the work done in [7].

There are a number of topological properties and data communication algorithms that need to be investigated for the proposed network before a final conclusion can be made. This paper certainly provides a promising starting point. In particular, the prefix computation, the Hamiltonian path and the disjoint path problems, the embeddings with other networks, the bisection width and the fault tolerance, among others. Given some border conditions, it is also interesting to design formulas for ranking and unranking the nodes, that is matching the introduced addresses with the addresses $1, 2, \ldots, n$, where n is the number of nodes in the network.

Acknowledgments

This research is supported by CONACYT Project 37017-A and NSERC grants.

References

[1] E. Anderson, J. Brooks, C. Grass, S. Scott, Performance of the Cray T3E multiprocessor, Proceedings of the Supercomputing Conference, 1997.

[2] M.M. Bae, B. Bose, Resource placement in torus-based networks, IEEE Trans. Comput. 46 (10) (1997) 1083–1092.

[3] B. Bose, B. Broeg, Y. Kwon, A. Ashir, Lee distance and topological properties of k-ary n-cubes, IEEE Trans. Comput. 44 (1995) 1021–1030.

[4] J. Carle, Étude des propriétés des réseaux d'interconnexion de type nid d'abeilles, Ph.D.Thesis, Laboratoire de Recherche en Informatique d'Amiens LARIA, Université de Picardie, Jules Verne, Amiens, France, December 2000.

[5] J. Carle, J.F. Myoupo, Topological properties and optimal routing algorithms for three dimensional hexagonal networks, Proceedings of the High Performance Computing in the Asia-Pacific Region HPC-Asia, Vol. I, Beijing, China, 2000, pp. 116–121.

[6] J. Carle, J.F. Myoupo, D. Seme, All-to-all broadcasting algorithms on honeycomb networks and applications, Parallel Process. Lett. 9 (4) (1999) 539–550.

[7] J. Carle, J.F. Myoupo, I. Stojmenovic, Higher dimensional honeycomb networks, J. Interconnect. Networks 2(4) (December 2001) 391–420.

[8] M.S. Chen, K.G. Shin, D.D. Kandlur, Addressing, routing and broadcasting in hexagonal mesh multiprocessors, IEEE Trans. Comput. 39 (1) (1990) 10–18.

[9] J.W. Dolter, P. Ramanathan, K.G. Shin, Performance analysis of virtual cut-through switching in HARTS: a hexagonal mesh multicomputer, IEEE Trans. Comput. 40 (6) (1991) 669–680.

[10] J. Duato, S. Yalamanchili, L. Ni, Interconnection Networks: An Engineering Approach, IEEE Computer Society Press, Silver Spring, MD, 1997.

[11] F. García, I. Stojmenovic, J. Zhang, Addressing and routing in hexagonal networks with applications for location update and connection rerouting in cellular networks, IEEE Trans. Parallel Distrib. Systems 13(9) (September 2002) 963–971.

[12] R.E. Kessler, J.L. Schwarzmeier, Cray T3D: A new dimension for Cray Research, in: CompCon, Spring 1993, pp. 176–182.

[13] L.N. Laster, J. Sandor, Computer graphics on hexagonal grid, Comput. Graph. 8 (1984) 401–409.

[14] G.M. Megson, X. Liu, X. Yang, Fault-tolerant ring embedding in a honeycomb torus with node failures, Parallel Process. Lett. 9 (4) (1999) 551–562.

[15] G.M. Megson, X. Yang, X. Liu, Honeycomb tori are Hamiltonian, Inform. Process. Lett. 72 (1999) 99–103.

[16] N-Cube Company, NCUBE-2 Processor Manual, N-Cube Company, 1990.

[17] M. Noakes, W.J. Dally, System design of the J-machine, Proceedings of the Advanced Research in VLSI, MIT Press, Cambridge, MA, 1990, pp. 179–192.

[18] B. Parhami, D.-M. Kwai, A unified formulation of honeycomb and diamond networks, IEEE Trans. Parallel Distrib. Systems 12(1) (January 2001) 74–79.

[19] C. Peterson, et al., iWarp: a 100-MOPS VLIW microprocessor for multicomputers, IEEE Micro 11 (2) (1991) 26–37.

[20] B. Robic, J. Silc, High performance computing on a honeycomb architecture, Proceeedings of the International ACPC Parallel Computation Conference, Lecture Notes in Computer Science, Austria, 1993.

[21] K.G. Shin, HARTS: a distributed real-time architecture, IEEE Comput. 24 (5) (1991) 25–35.

[22] I. Stojmenovic, Honeycomb networks: topological properties and communication algorithms, IEEE Trans. Parallel Distrib. Systems 8(10) (October 1997) 1036–1042.

[23] I. Stojmenovic, Direct interconnection networks, in: A.Y. Zomaya (Ed.), Parallel and Distributed Computing Handbook, McGraw-Hill, Inc., Tokyo, 1996, pp. 537–567.

[24] R. Tosic, D. Masulovic, I. Stojmenovic, J. Brunvoll, B.N. Cyvin, S.J. Cyvin, Enumeration of polyhex hydrocarbons up to $h = 17$, J. Chem. Inform. Comput. Sci. 35 (1995) 181–187.

[25] J. Zhang, A cell ID assignment scheme and its applications, Proceedings of the Workshop on Wireless Networks and Mobile Computing, IEEE International Conference on Parallel Processing, Toronto, Canada, August 2000, pp. 507–512.