Diagonal and Toroidal Mesh Networks

K. Wendy Tang, *Member, IEEE*, and Sanjay A. Padubidri

***Abstract*—Diagonal and toroidal mesh are degree-4, point to point interconnection models suitable for connecting communication elements in parallel computers, particularly multicomputers. The two networks have a similar structure. The toroidal mesh is popular and well-studied whereas the diagonal mesh is relatively new. In this paper, we show that the diagonal mesh has a smaller diameter and a larger bisection width. It also retains advantages such as a simple rectangular structure, wirability and scalability of the toroidal mesh network. An optimal self-routing algorithm is developed for these networks. Using this algorithm and the existing routing algorithm for the toroidal mesh, we simulated and compare the performance of these two networks with $N = 35 \times 71 = 2485$, $N = 49 \times 99 = 4851$, and $N = 69 \times 139 = 9591$ nodes under a constant system with a fixed number of messages. Deflection routing is used to resolve conflicts. The effects of various deflection criteria are also investigated. We show that the diagonal mesh outperforms the toroidal mesh in all cases, and thus provides an attractive alternative to the toroidal mesh network.**

***Index Terms*—Massively parallel systems, multicomputers, interconnection networks, diameter, bisection width, and deflection routing algorithm.**

I. Introduction

EFFICIENT interconnection networks are critical to the performance of large communication networks with hundreds and thousands of communicating elements [1], [2]. Applications can be found in the design of *massively parallel computers*. Attributes of an interconnection network include *the diameter, bisection width, symmetry, wirability,* and *scalability*.

The *diameter* is the maximum of the shortest distance (hops) between any two nodes. An interconnection graph with a small diameter implies potentially a small communication delay. The *bisection width* of a network is the minimum number of wires that have to be removed to disconnect a network into two halves with identical (within one) numbers of processors [3]. It is a critical factor in determining the performance of a network because in most scientific problems, the data contained and/or computed by one half of the network are needed by the other half [3]. Therefore, it is advantageous to have networks with large bisection width so that efficient communication between the two halves of the network can be achieved. Furthermore, large bisection width also facilitates higher degrees of fault tolerance.

A *symmetric* network is also called *vertex-transitive*. Mathematically, this implies that for any two nodes a and b, there is an automorphism of the graph that maps a to b. Informally, this means that the network "looks" the same from any node. This property is useful for practical implementation of interconnection networks because every node in a symmetric network is homogeneous and the same routing algorithm can be applied to every node. A network is *wirable* implies reasonable and manageable patterns of wiring could be devised [4]. Scalability refers to the increase in wire length with the number of nodes. A *scalable* interconnection model has less than quadratic increase of wire length when compared with the number of nodes [4].

Besides the topological properties of the network, routing is an important issue of interconnection networks. The goal of routing is to send messages between any two nodes. There are two sub-problems: *path identification* and *network performance*. Path identification determines optimal (shortest) paths between any two nodes; network performance is concerned with how a network handles traffic in the presence of contention.

For path identification, it is desirable to have a *distributed, self-routing* algorithm that can identify shortest paths based only on addresses of the source and destination. Such a routing algorithm is associated with a particular topology. It provides fast, decentralized routing decisions without any storage space requirement. For network performance, computer simulations and probabilistic modeling are the tools. When two or more incoming messages at a node have the same optimal outgoing link as identified by the path-determination algorithm, conflicts are bound to occur. Priority measures are needed to resolve these conflicts; and some messages are either routed non-optimally or stored temporarily in buffers. While a path-determining algorithm is usually associated with a particular topology, the same routing algorithm can be used to evaluate performance of different networks.

Deflection routing is a popular algorithm to evaluate network performance. It is a *bufferless, dynamic* routing algorithm proposed for multicomputer networks and local and metropolitan area networks [5]–[7]. Basically, messages are sorted according to a *deflection criterion*, such as *age* or *path length*. Those with higher priorities are routed optimally while those with lower priorities are *deflected* to non-optimal links. There is no buffer and hence no buffer management at a node. This algorithm is simple and straightforward to implement.

Many interconnection topologies with different associated routing (path determination) algorithms have been proposed. Examples include the toroidal mesh, hypercube, cube-connected cycles, Moebius, DeBruijn, and Cayley networks [1], [8], [9]. Among the many existing topologies, toroidal mesh is a popular model. It is a *degree*-4, *symmetric* or

Manuscript received April 13, 1992; revised August 20, 1993.
The authors are with the Department of Electrical Engineering, SUNY at Stony Brook, Stony Brook, NY 11794-2350 USA; e-mail: wtang@sbee.sunysb.edu.
IEEE Log Number 9401104.

0018-9430/94$04.00 © 1994 IEEE

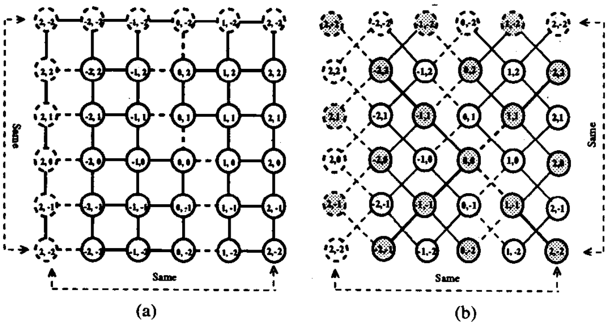


Fig. 1. $N = 5 \times 5$ toroidal (a) and diagonal (b) mesh networks.

TABLE I
A ROUTING ALGORITHM FOR TOROIDAL MESH NETWORKS

Routing between (x_1, y_1) and (x_2, y_2) in a toroidal mesh network with $N = n \times k$ nodes where n and k are odd integers.

Step 1: Evaluate $x = < x_2 - x_1 >_k$, $y = < y_2 - y_1 >_n$, where

$$< x >_n = \begin{cases} x, & \text{if } |x| \leq \frac{n-1}{2}; \\ x - n & \text{if } x > \frac{n-1}{2}; \\ x + n & \text{if } x < -\frac{n-1}{2}. \end{cases}$$

Step 2: Determine optimal directions

$$\text{If} \begin{cases} x > 0, & \text{take the } X \text{ direction;} \\ x < 0, & \text{take the } -X \text{ direction;} \end{cases} \quad \text{If} \begin{cases} y > 0, & \text{take the } Y \text{ direction;} \\ y < 0, & \text{take the } -Y \text{ direction.} \end{cases}$$

Distance between (x_1, y_1) and (x_2, y_2) is $\lfloor \frac{|x|}{2} \rfloor + \lfloor \frac{|y|}{2} \rfloor$.

vertex-transitive, *wirable*, and *scalable* network with a simple, decentralized self-routing algorithm. Recently, researchers have proposed the *diagonal mesh* as an attractive alternative to toroidal mesh networks [10]–[12]. Diagonal mesh is similar to the toroidal mesh, except that nodes in the network are diagonally-connected (Fig. 1). In other words, it is also a *degree*-4, *symmetric* or *vertex-transitive*, *wirable* and *scalable* network.

In this paper, we compare the properties and performance of *toroidal mesh* with the *diagonal mesh* networks. We first review the similarities of the two networks and then develop an optimal routing algorithm and formulate the diameter of diagonal mesh networks. We show that a diagonal mesh network has a better diameter than its toroidal counterpart. For an $N = n \times k$ network (n, k are odd integers), the toroidal mesh has a diameter $D_t = (\frac{n-1}{2} + \frac{k-1}{2})$; whereas the diameter for the diagonal mesh is $D_d = n - 1$ for $k = n$ and $D_d = \max\{n, \frac{k-1}{2}\}$ for $k > n$. In other words, $D_d = D_t$ for $k = n, n+2$. But when k is strictly greater than $n + 2$, the diagonal mesh has a smaller diameter.

The performance of the two networks in the presence of contention are then compared through computer simulations. Because of its simplicity, we use deflection routing algorithm for performance simulations. We first investigate the effects of different deflection criteria. We conclude that the *age* (the period of time a message has been introduced to the system) is the most efficient criterion. Under this criterion, an "older" message has a higher priority than a "younger" one. Self-routing algorithms are used to identify optimal outgoing links of each message for both networks. When conflicts occur, the "younger" message will be deflected to an non-optimal link. We simulate the performance of large diagonal and toroidal mesh network in a *constant* system with a fixed number of messages. The average *delay* and *throughput* of the system are observed.

This paper is organized as follows: in Section II, we review the properties and routing algorithms of toroidal mesh networks. Section III discusses the properties of diagonal mesh networks. We develop an optimal routing algorithm, propose and prove the formulation of the diameter for these networks. Network performance in the presence of contention is discussed in Section IV. This include a description on the deflection routing algorithm, various constraints of the simulations, a summary of the simulation results, and interpretation of these results. Finally, in Section V, we compare and conclude the performance of toroidal and diagonal mesh networks.

II. TOROIDAL MESH

The toroidal mesh is a simple and popular topology. It consists of a two-dimensional grid of processing elements with wrap-around connections at edges. Consider an $N = n \times k$ toroidal mesh, where n and k are odd integers. (In general, n and k can be any integers. But in this paper, we consider the case that n and k are odd integers.) For any node (x, y), $x \in \{-\frac{k-1}{2}, \cdots, \frac{k-1}{2}\}$, $y \in \{-\frac{n-1}{2}, \cdots, \frac{n-1}{2}\}$, connections are defined as

$$(x, y) \sim \begin{matrix} (x, \langle y+1 \rangle_n), & (x, \langle y-1 \rangle_n), \\ (\langle x+1 \rangle_k, y), & (\langle x-1 \rangle_k, y), \end{matrix}$$

where $\sim$ signifies connections and

$$\langle x \rangle_k = \begin{cases} x, & \text{if } |x| \leq \frac{k-1}{2} \\ x - k, & \text{if } x > \frac{k-1}{2} \\ x + k, & \text{if } x < -\frac{k-1}{2}. \end{cases}$$

Figure 1(a) shows an $N = 5 \times 5$ toroidal mesh in Cartesian coordinates. For these networks, a self routing algorithm based on labels of the source and destination exists and is summarized in Table I. This routing algorithm is straightforward and its space complexity is of $O(1)$, independent of the size of the network. Besides a distributed routing scheme, other merits of a toroidal mesh include a simple, symmetric, rectangular structure, wirability and scalability. Note that increases in wire length of toroidal mesh is mainly of $O(N)$, except for wrap-around connections at edges.

However, drawbacks of the mesh are its large *diameter* and small *bisection width*. The diameter of an $N = n \times k$ toroidal mesh is

$$D_t = \left(\frac{n-1}{2} + \frac{k-1}{2} \right). \tag{1}$$

This relatively large diameter implies potentially long communication delay and thus hampers network performance. The bisection width of an $n \times k$ (n, k are odd and $k \geq n$) toroidal mesh is $B_t = 2n + 2$. In the center column ($x = 0$), $2 \times 2 + (n - 2)$ wires connect the two halves because 2 wires from each of the 2 boundary nodes and 1 wire from the rest of the $n - 2$ nodes. At the edge $\left(x = \frac{k-1}{2} \text{ or } x = -\frac{k-1}{2}\right)$, there are n horizontal wires connecting the two halves. These wires are indicated as dotted lines in Figure 1.

Parallel computers that use the mesh topology include ILLIAC IV, Massively Parallel Processors (MPP), Distributed Array Processors (DAP), and Wire Routing Machine (WRM) [13]. These parallel computers are used to solve many engineering and scientific problems. Examples include sorting, matrix multiplication and inversion, Fourier transformation, convolution, signal and image processing, speech recognition, and finite element analysis [14]. Besides being used to interconnect processors in a parallel computer, the toroidal mesh is also used in local and metropolitan area networks. The resultant network is called the Manhattan Street Network [15]. Both the directed and undirected cases have been studied [15], [16].

III. Diagonal (Toroidal) Mesh

Diagonal mesh networks are proposed by Arden [10]. It is similar to the toroidal case except that nodes have diagonal instead of horizontal and vertical connections. Preliminary simulations for a few specific cases have been studied by Arden and Li [11]. However, routing was accomplished by table look-up schemes. In this section, we first review the definition of diagonal mesh networks in Section III-A. A self-routing scheme based only on the addresses of source and destination nodes is then developed in Section III-B. Section III-C provides the formulation and proof of the diameter of diagonal mesh networks.

A. Definition

Diagonal mesh networks are similar to the toroidal case except that nodes have diagonal instead of horizontal and vertical connections. We consider networks with $N = n \times k$ nodes, where n and k are odd integers. Furthermore, without loss of generality, assume $k \geq n$. Figure 1(b) shows an $N = 5 \times 5$ diagonal mesh with Cartesian node labels. For any node (x,y), $x \in \{-\frac{k-1}{2}, \cdots, \frac{k-1}{2}\}$, $y \in \{-\frac{n-1}{2}, \cdots, \frac{n-1}{2}\}$. Connections are defined as

$$(x,y) \sim \begin{matrix} (\langle x+1\rangle_k, & \langle y+1\rangle_n), & (\langle x+1\rangle_k, & \langle y-1\rangle_n), \\ (\langle x-1\rangle_k, & \langle y+1\rangle_n), & (\langle x-1\rangle_k, & \langle y-1\rangle_n), \end{matrix}$$

where $\sim$ signifies connections and

$$\langle x\rangle_n = \begin{cases} x, & \text{if } |x| \leq \frac{n-1}{2}; \\ x-n, & \text{if } x > \frac{n-1}{2}; \\ x+n, & \text{if } x < -\frac{n-1}{2}. \end{cases} \tag{2}$$

As in the toroidal case, modular wrap-around connections exist at edges. For example, node $(2,2)$ connects to nodes $(-2,-2)$, $(-2,1)$, $(1,-2)$ and $(1,1)$. Strictly speaking, the *diagonal mesh* is also *toroidal*, but for simplicity, we use the name *diagonal mesh* to refer to these networks. In this paper, we consider both n and k are *odd* numbers. If n and k are both *even* integers, the graph will have two disconnected halves, the white and shaded nodes in Fig. 1(b).

The bisection width of an $n \times k$ (n, k are odd and $k \geq n$) diagonal mesh is $B_d = 4n$ because $2n$ wires at the center ($x = 0$) and $2n$ wires at the edge $\left(x = \frac{k-1}{2} \text{ or } x = -\frac{k-1}{2}\right)$ need to be removed to disconnect the graph into two equal halves (within one node). In Fig. 1, we identify these wires by dotted lines. Recall from Section II that the bisection width of a toroidal mesh is $B_t = 2n + 2$. In other words, an $n \times k$ diagonal mesh with n, k odd and $k \geq n$ always has a larger bisection width than its toroidal counterpart.

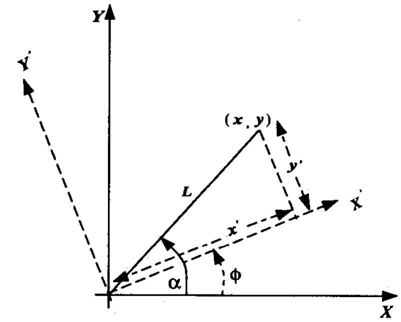


Fig. 2. Node label transformation between coordinates.

B. Routing

To establish a label-determined, self-routing algorithm, we introduce a new coordinate system X'-Y' by transforming the original coordinates X-Y through an anticlockwise rotation of 45° and scaling the axes by $\sqrt{2}$. In essence, this new coordinate frame corresponds to the diagonal connections of the network. Transformation between coordinates can be formulated mathematically.

In Fig. 2, a coordinate frame X-Y is rotated anticlockwise by an angle ϕ to form a new frame X'-Y'. Given a point (x,y) in the X-Y frame,

$$\begin{aligned} x &= L\cos\alpha \\ y &= L\sin\alpha, \end{aligned} \tag{3}$$

where L and α are the length and angle associated with the point (see Fig. 2). The coordinates of this point in the X'-Y' frame are:

$$\begin{aligned} x' &= L\cos(\alpha - \phi) \\ &= L\cos\alpha\cos\phi + L\sin\alpha\sin\phi \\ &= x\cos\phi + y\sin\phi \end{aligned}$$

$$\begin{aligned} y' &= L\sin(\alpha - \phi) \\ &= L\sin\alpha\cos\phi - L\cos\alpha\sin\phi \\ &= -x\sin\phi + y\cos\phi. \end{aligned} \tag{4}$$

In this case, $\phi = 45°$ and the X, Y axes need to be scaled by $\sqrt{2}$. In other words, the resultant transformation is:

$$\begin{aligned} x' &= \sqrt{2}(x\cos 45° + y\sin 45°) \\ &= x + y \\ y' &= \sqrt{2}(-x\sin 45° + y\cos 45°) \\ &= -x + y \end{aligned} \tag{5}$$

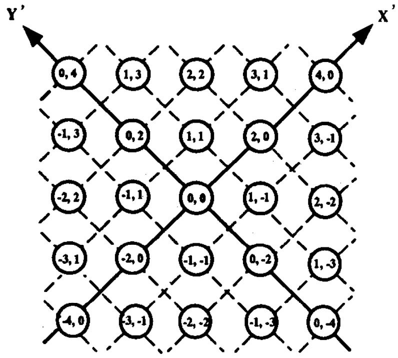


Fig. 3. An $N = 5 \times 5$ diagonal mesh network in X'-Y' coordinates.

That is, given a point (x, y) in the Cartesian frame, its node label in the new coordinate system is (x', y'), where

$$\begin{aligned} x' &= x + y, \\ y' &= -x + y. \end{aligned} \tag{6}$$

We call this node label transformation *frame transformation* and is summarized as follows:

$$(x, y) \overset{\text{frame}}{\underset{\text{transformation}}{\longrightarrow}} (\overbrace{x + y}^{x'}, \overbrace{-x + y}^{y'}). \tag{7}$$

The node labels of the 5×5 diagonal mesh after this transformation are shown in Fig. 3. Diagonal mesh networks are *symmetric* or *vertex-transitive* in graph terminology [17]. Because of this symmetry, routing between any two nodes (x_1, y_1) and (x_2, y_2) is tantamount to routing between $(0, 0)$ and $(\langle x_2 - x_1 \rangle_k, \langle y_2 - y_1 \rangle_n)$, where $\langle x \rangle_k$ is defined in (2). Therefore, it suffices to establish a routing algorithm between node $(0, 0)$ and all other nodes in the graph. In this section, we develop an *optimal* routing algorithm between $(0, 0)$ and any other node (x', y') in the X'-Y' frame. The algorithm is *optimal* because all directions ($\pm X'$, $\pm Y'$) contributing to shortest paths from $(0, 0)$ to (x', y') are identified. We first consider 1) routing between $(0, 0)$ and *even nodes* (those with even x', y' values); and then 2) routing between $(0, 0)$ and *odd nodes* (those with odd x', y' values).

When the graph is represented in the new X'-Y' coordinate system, routing between node $(0, 0)$ and the "even" nodes is similar to the toroidal case. We summarize the algorithm as follows:

Proposition 1: A routing algorithm between $(0, 0)$ and (x', y'), where x' and y' are even.

$$\begin{aligned} &\text{If } \begin{cases} x' > 0, & \text{take the } X' \text{ direction,} \\ x' < 0, & \text{take the } -X' \text{ direction;} \end{cases} \\ &\text{if } \begin{cases} y' > 0, & \text{take the } Y' \text{ direction,} \\ y' < 0, & \text{take the } -Y' \text{ direction,} \end{cases} \end{aligned} \tag{8}$$

and the distance (number of hops) between $(0, 0)$ and (x', y') is $|\frac{x'}{2}| + |\frac{y'}{2}|$.

Proof: In the X'-Y' frame, node $(0, 0)$ connects to $(2, 0)$, $(-2, 0)$, $(0, 2)$, and $(0, -2)$. Therefore, shortest paths between $(0, 0)$ and even nodes (x', y') correspond to those between $(0, 0)$ and $(\frac{x'}{2}, \frac{y'}{2})$ in a toroidal mesh. $\square$

As an example, consider routing between $(0, 0)$ and $(2, -2)$ in the X'-Y' frame. Since $x' = 2 > 0$ and $y' = -2 < 0$, both X' and $-Y'$ directions contribute to optimal paths with distance $|\frac{2}{2}| + |\frac{-2}{2}| = 2$.

However, routing between $(0, 0)$ and the "odd" nodes (those with odd x', y' values) are not as simple. Wrap-around connections need to be considered. Due to modular wrap-around connections at edges of the network, a node (x, y) has four other equivalent node labels in the X-Y frame:

$$(x, y) \cong \begin{cases} c_1 = (x, y - n), & c_3 = (x - k, y), \\ c_2 = (x, y + n), & c_4 = (x + k, y). \end{cases} \tag{9}$$

From (7), the node labels of $c_1, \cdots, c_4$ in the X'-Y' frame are:

$$\begin{aligned} c_1' &= (\overbrace{x + y}^{x'} - n, \overbrace{-x + y}^{y'} - n), & c_3' &= (\overbrace{x + y}^{x'} - k, \overbrace{-x + y}^{y'} + k), \\ c_2' &= (\overbrace{x + y}^{x'} + n, \overbrace{-x + y}^{y'} + n), & c_4' &= (\overbrace{x + y}^{x'} + k, \overbrace{-x + y}^{y'} - k). \end{aligned} \tag{10}$$

Since x', y', n and k are odd numbers, $c_1', \cdots, c_4'$ are all even nodes (their X', Y' coordinates are both even), which implies the routing algorithm (Proposition 1) established for even nodes applies. However, some of the four labels may not correspond to shortest paths between $(0, 0)$ and (x, y). From Proposition 1, the distance between $(0, 0)$ and $c_1', \cdots, c_4'$ are:

$$\begin{aligned} d_1' &= \frac{1}{2}\{|x' - n| + |y' - n|\}, & d_3' &= \frac{1}{2}\{|x' - k| + |y' + k|\}, \\ d_2' &= \frac{1}{2}\{|x' + n| + |y' + n|\}, & d_4' &= \frac{1}{2}\{|x' + k| + |y' - k|\}, \end{aligned} \tag{11}$$

where $x' = x + y$ and $y' = -x + y$. This implies that node (x, y) can be reached at distances d_1', d_2', d_3' or d_4' from node $(0, 0)$. To establish an optimal routing algorithm between $(0, 0)$ and (x, y), we need to identify c_i' $(i = 1, \cdots, 4)$ such that the corresponding distance

$$d_i' = d_{\min} = \min\{d_1', d_2', d_3', d_4'\}. \tag{12}$$

In the following propositions and corollaries, we establish a simpler expression for $d_1', \cdots, d_4'$; and identify $d_i' = d_{\min}$ for different x, y values.

Proposition 2: Let $d_1', \cdots, d_4'$ be defined as in (11),

$$\begin{aligned} d_1' &= \max\{|x|, n - y\}, & d_3' &= \max\{|y|, k - x\} = k - x, \\ d_2' &= \max\{|x|, n + y\}, & d_4' &= \max\{|y|, k + x\} = k + x. \end{aligned}$$

Proof: From (11) and (7),

$$\begin{aligned} 2d_1' &= |x + y - n| + |-x + y - n| \\ &= |x - (n - y)| + |x + (n - y)| \\ &= \begin{cases} |x| - (n - y) + |x| + (n - y), & \text{if } |x| \geq n - y; \\ -|x| + (n - y) + |x| + (n - y). & \text{if } |x| < n - y; \end{cases} \\ &= \begin{cases} 2|x|, & \text{if } |x| \geq n - y; \\ 2(n - y), & \text{if } |x| < n - y. \end{cases} \end{aligned}$$

Hence, $d_1 = \max\{|x|, (n-y)\}$. Similarly,

$$d'_2 = \max\{|x|, n+y\};$$
$$d'_3 = \max\{|y|, k-x\};$$
$$d'_4 = \max\{|y|, k+x\}.$$

Since $x \in \{-\frac{k-1}{2}, \cdots, \frac{k-1}{2}\}$, $y \in \{-\frac{n-1}{2}, \cdots, \frac{n-1}{2}\}$, and $k \geq n$,

$$k - x > k/2 \geq n/2 > |y|,$$
$$k + x > k/2 \geq n/2 > |y|.$$

Hence

$$d'_3 = k - x \quad \text{and} \quad d'_4 = k + x. \qquad \square$$

Corollary 1: Let $d'_1, \cdots, d'_4$ be defined as in (11) and $d_{\min}$ as in (12). For $|x| < n$,

$$\text{if } y > 0, \quad d'_1 < d'_2;$$
$$\text{if } y = 0, \quad d'_1 = d'_2;$$
$$\text{if } y < 0, \quad d'_2 < d'_1.$$

Proof: From Proposition 2,
If $y > 0$,
$d'_1 = \max(|x|, n - |y|)$
$d'_2 = \max(|x|, n + |y|) = n + |y|$ because $|x| < n$
$\Rightarrow d'_1 < d'_2$
If $y = 0$,
$d'_1 = d'_2 = \max(|x|, n) = n$ because $|x| < n$
If $y < 0$
$d'_1 = \max(|x|, n + |y|) = n + |y|$ because $|x| < n$
$d'_2 = \max(|x|, n - |y|)$
$\Rightarrow d'_2 < d'_1$

$\square$

Proposition 3: Let $d'_1, \cdots, d'_4$ be defined as in (11) and $d_{\min}$ as in (12). For $|x| < n$,

$$\text{if} \quad (|x| + |y| < n) \& (|x| - |y| > k - n),$$
$$d_{\min} = \begin{cases} d'_3 & \text{if } x \geq 0 \\ d'_4 & \text{if } x \leq 0 \end{cases} = k - |x|$$
$$\text{else} \quad d_{\min} = \begin{cases} d'_1 & \text{if } y \geq 0 \\ d'_2 & \text{if } y \leq 0 \end{cases} = \max(|x|, n - |y|)$$

Proof: From Proposition 2 and Corollary 1,

$$\min(d'_1, d'_2) = \begin{cases} d'_1, & \text{if } y \geq 0; \\ d'_2, & \text{if } y \leq 0; \end{cases} = \max(|x|, n - |y|).$$
$$\min(d'_3, d'_4) = \begin{cases} d'_3, & \text{if } x \geq 0; \\ d'_4, & \text{if } x \leq 0; \end{cases} = k - |x|.$$

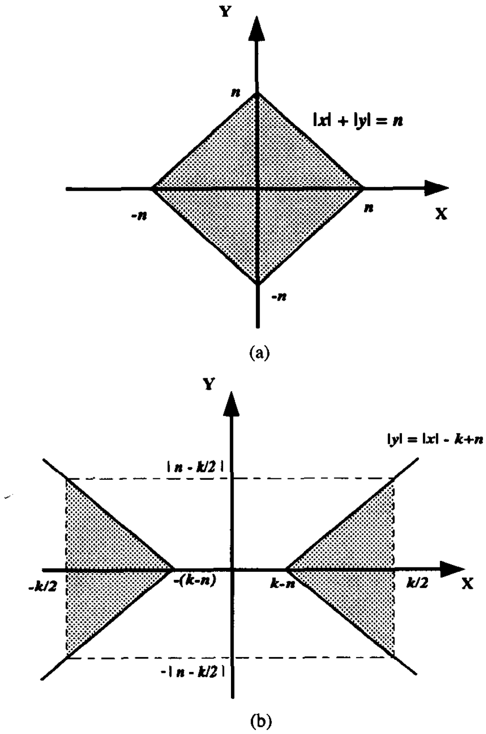


Fig. 4. Graphical inequalities.

Also,

$$|x| > n - |y|$$
$$\Rightarrow \min(d'_1, d'_2) = |x| < k - |x| = \min(d'_3, d'_4)$$
$$\left(\text{because } |x| \leq \frac{k-1}{2}\right)$$
$$\Rightarrow d_{\min} = \min(d'_1, d'_2)$$

$$(|x| < n - |y|) \text{ and } (n - |y| > k - |x|)$$
$$\Rightarrow \min(d'_1, d'_2) = n - |y| > k - |x| = \min(d'_3, d'_4)$$
$$\Rightarrow d_{\min} = \min(d'_3, d'_4)$$

$$(|x| < n - |y|) \text{ and } (n - |y| < k - |x|)$$
$$\Rightarrow \min(d'_1, d'_2) = n - |y| < k - |x| = \min(d'_3, d'_4)$$
$$\Rightarrow d_{\min} = \min(d'_1, d'_2)$$

Hence

$$\text{if} \quad (|x| + |y| < n) \,\&\, (|x| - |y| > k - n)$$
$$d_{\min} = \min(d'_3, d'_4)$$
$$\text{else } d_{\min} = \min(d'_1, d'_2). \qquad \square$$

Proposition 4:

$$(|x| + |y| < n) \,\&\, (|x| - |y| > k - n) \Rightarrow k < 2n$$

Proof: The shaded regions in Fig. 4(a) and (b) show the values of x and y satisfying $(|x|+|y|<n)$ and $(|x|-|y|>k-n)$, respectively. If x and y satisfy both regions, the shaded regions must overlap; i.e.,

$$\begin{aligned} & k-n<n \\ \Rightarrow \quad & k<2n. \end{aligned}$$ □

From Propositions 3 and 4, we have two useful corollaries:

Corollary 2: Let $d_1',\cdots,d_4'$ be as defined in (11) and $d_{\min}$ as in (12).

$$\begin{aligned} \text{If } (k>2n)\ \&\ (|x|<n),& \\ d_{\min} &= \min(d_1',d_2') \\ &= \begin{cases} d_1', & \text{if } y\geq 0; \\ d_2', & \text{if } y\leq 0; \end{cases} \\ &= \max(|x|, n-|y|). \end{aligned}$$

Corollary 3: Let $d_1',\cdots,d_4'$ be as defined in (11) and $d_{\min}$ as in (12):

$$\begin{aligned} \text{If } (n\leq k<2n)\ \&\ (|x|<n),& \\ d_{\min} &= \min(d_3',d_4') \\ &= \begin{cases} d_3', & \text{if } x\geq 0; \\ d_4', & \text{if } x\leq 0; \end{cases} \\ &= k-|x|. \end{aligned}$$

These two corollaries are direct consequences of Propositions 3 and 4; and the proofs are omitted. They are particularly useful for routing between $(0,0)$ and odd (x',y') where $|x|<n$. To determine optimal directions between $(0,0)$ and (x,y), where $|x|\geq n$, we present another proposition.

Proposition 5: If $|x|\geq n$, the shortest distance between $(0,0)$ and (x,y) is $d_{\min}=|x|$. Also, optimal directions from $(0,0)$ are X' and $-Y'$ if $x\geq n$; or $-X'$ and Y' if $x\leq -n$.

Proof: Case 1: $|x|=n$.

Note that $x\in\{-\frac{k-1}{2},\cdots,\frac{k-1}{2}\}$ and $y\in\{-\frac{n-1}{2},\cdots,\frac{n-1}{2}\}$. Hence, $|y|<n$ and $|x|=n\Rightarrow k>2n$. According to the frame transformation (7), node (x,y) is represented as (x',y') in the X'-Y' frame, where $x'=x+y$ and $y'=-x+y$. Since $|x|=n$ is odd, if y is also odd, x', y' are both even and Proposition 1 can be applied for routing. In this case,

$$d_{\min}=1/2(|x+y|+|-x+y|)=|x|=n, \quad \text{because } |y|<n.$$

Furthermore,

$$\begin{aligned} & x=n \\ \Rightarrow\ & x'=n+y>0 \text{ and } y'=y-n<0 \end{aligned}$$

According to Proposition 1, both X' and $-Y'$ are optimal directions. Similarly,

$$\begin{aligned} & x=-n \\ \Rightarrow\ & x'=-n+y<0 \text{ and } y'=y+n>0, \end{aligned}$$

both $-X'$ and Y' are optimal directions.

If $|x|=n$ and y is even, $x'=x+y$ and $y'=-x+y$ are odd. We consider the alternate node labels of (x,y), $c_1,\cdots,c_4$ as defined in (9). These alternate node labels are represented as $c_1',\cdots,c_4'$ in the X'-Y' frame according to (10). Using Proposition 2, for $|x|=n$, the corresponding distance between $c_1',\cdots,c_4'$ and $(0,0)$ are $d_1',\cdots,d_4'$, where

$$d_1'=\begin{cases} n & \text{if } y\geq 0 \\ n-y & \text{if } y<0 \end{cases}; \quad d_2'=\begin{cases} n+y & \text{if } y\geq 0 \\ n & \text{if } y<0 \end{cases};$$
$$d_3'=\begin{cases} k-n & \text{if } x=n \\ k+n & \text{if } x=-n \end{cases}; d_4'=\begin{cases} k+n & \text{if } x=n \\ k-n & \text{if } x=-n. \end{cases}$$

That is,

$$\min\{d_1',d_2'\}=n, \qquad \min\{d_3',d_4'\}=k-n.$$

Also,

$$\begin{aligned} & k>2n \\ \Rightarrow & k-n>n \\ \Rightarrow & d_{\min}=\min(d_1',d_2') \\ & \quad =\begin{cases} d_1'=n & \text{if } y>0 \\ d_1'=d_2'=n & \text{if } y=0 \\ d_2'=n & \text{if } y<0 \end{cases} \end{aligned}$$

If $y>0$,

$$\begin{aligned} c_1' &= (x+y-n,-x+y-n) \\ &= \begin{cases} (y,y-2n)=(>0,<0), & \text{if } x=n; \\ (y-2n,y)=(<0,>0), & \text{if } x=-n. \end{cases} \end{aligned}$$

If $y=0$, $x=n$,

$$\begin{aligned} c_1' &= (0,-2n)=(0,<0), \\ c_2' &= (2n,0)=(>0,0). \end{aligned}$$

If $y=0$, $x=-n$,

$$\begin{aligned} c_1' &= (-2n,0)=(<0,0), \\ c_2' &= (0,2n)=(0,>0). \end{aligned}$$

If $y<0$,

$$\begin{aligned} c_2' &= (x+y+n,-x+y+n) \\ &= \begin{cases} (2n-|y|,y)=(>0,<0), & \text{if } x=n; \\ (y,2n-|y|)=(<0,>0), & \text{if } x=-n. \end{cases} \end{aligned}$$

Hence, all nodes at $|x|=n$ are at distance n from $(0,0)$ with X' and $-Y'$ as optimal directions if $x=n$; or $-X'$ and Y' as optimal directions if $x=-n$.

Case 2: $|x|>n$.

We observe that the smallest path between $(0,0)$ and (x,y) where $|x|=n+i$ for some $i>0$ must go through an intermediate node at $x=n$ or $x=-n$, depending if $x>n$ or $x<-n$. In both cases, (x,y) is i hops from a node at $|x|=n$. Furthermore, from case 1, we proved that all nodes at $|x|=n$ are n hops from $(0,0)$. Hence, all nodes at $|x|=n+i$, $i\geq 0$, are $n+i$ hops from $(0,0)$ with X' and $-Y'$ as optimal directions if $x\geq n$; or $-X'$ and Y' as optimal directions if $x\leq -n$. □

To summarize, Proposition 1 determines optimal directions from $(0,0)$ to even (x',y'); Corollaries 2 and 3 provide routing between $(0,0)$ and odd (x',y') with $|x|<n$; and Proposition 5 identifies optimal paths between $(0,0)$ and any node (x,y) with $|x|\geq n$. Based on these propositions and corollaries, we summarized routing algorithms for diagonal mesh network

TABLE II
A ROUTING ALGORITHM FOR $n \times k$ DIAGONAL MESH, $n \le k \le 2n-1$

Routing between (x_1, y_1) and (x_2, y_2) in a diagonal mesh network with $N = n \times k$ nodes and $n \le k \le 2n-1$ (n, k are odd).

Step 1: Evaluate $x = \langle x_2 - x_1 \rangle_k$, $y = \langle y_2 - y_1 \rangle_n$, where

$$\langle x \rangle_n = \begin{cases} x, & \text{if } |x| \le \frac{n-1}{2}; \\ x-n & \text{if } x > \frac{n-1}{2}; \\ x+n & \text{if } x < -\frac{n-1}{2}. \end{cases}$$

Step 2: Calculate $x' = x + y$; $y' = -x + y$.

Step 3: If x', y' are odd,

$$\text{if} \quad |x| + |y| < n \text{ and } |x| - |y| > k - n$$
$$\begin{cases} x' = x' - k, & y' = y' + k, & \text{if } x > 0; \\ x' = x' + k, & y' = y' - k, & \text{if } x < 0; \end{cases}$$
$$\text{else} \quad \begin{cases} x' = x' - n, & y' = y' - n, & \text{if } y \ge 0; \\ x' = x' + n, & y' = y' + n, & \text{if } y \le 0. \end{cases}$$

Note that when $y = 0$, there are two x', y' values to be applied in Step 4.

Step 4: Determine optimal directions

$$\text{If} \begin{cases} x' > 0, & \text{take the } X' \text{ direction;} \\ x' < 0, & \text{take the } -X' \text{ direction;} \end{cases} \quad \text{If} \begin{cases} y' > 0, & \text{take the } Y' \text{ direction;} \\ y' < 0, & \text{take the } -Y' \text{ direction.} \end{cases}$$

Distance between (x_1, y_1) and (x_2, y_2) is $|\frac{x'}{2}| + |\frac{y'}{2}|$.

TABLE III
A ROUTING ALGORITHM FOR $n \times k$ DIAGONAL MESH, $k \ge 2n+1$

Routing between (x_1, y_1) and (x_2, y_2) in a diagonal mesh network with $N = n \times k$ nodes and $k \ge 2n+1$ (n, k are odd).

Evaluate $x = \langle x_2 - x_1 \rangle_k$; $y = \langle y_2 - y_1 \rangle_n$, where

$$\langle x \rangle_n = \begin{cases} x, & \text{if } |x| \le \frac{n-1}{2}; \\ x-n & \text{if } x > \frac{n-1}{2}; \\ x+n & \text{if } x < -\frac{n-1}{2}. \end{cases}$$

Case I:

If $x \ge n$, both X' and $-Y'$ are optimal.
If $x \le -n$, both $-X'$ and Y' are optimal.
Distance between (x_1, y_1) and (x_2, y_2) is $|x|$.

Case II: If $|x| < n$,

1. Calculate $x' = x + y$; $y' = -x + y$.
2. If x', y' are odd, $\quad \begin{matrix} x' = x' - n, & y' = y' - n, & \text{if } y \ge 0; \\ x' = x' + n, & y' = y' + n, & \text{if } y \le 0. \end{matrix}$

   Note that when $y = 0$, there are two x', y' values to be applied in Step 3.
3. Determine optimal directions

$$\text{If} \begin{cases} x' > 0, & \text{take the } X' \text{ direction;} \\ x' < 0, & \text{take the } -X' \text{ direction;} \end{cases} \quad \text{If} \begin{cases} y' > 0, & \text{take the } Y' \text{ direction;} \\ y' < 0, & \text{take the } -Y' \text{ direction.} \end{cases}$$

Distance between (x_1, y_1) and (x_2, y_2) is $|\frac{x'}{2}| + |\frac{y'}{2}|$.

with $N = n \times k$ nodes in Tables II and III. Table II corresponds to $n \le k < 2n$ and Table III to $k \ge 2n$. These routing algorithms identify all optimal directions and shortest path length between any two nodes for the two cases.

C. Diameter Analysis

Besides facilitating the development of routing algorithms, the propositions and corollaries described in Section III-B also allow formulation of the diameter of a diagonal mesh. We present such formulation in the following proposition:

Proposition 6: For an $N = n \times k$ diagonal mesh network, assume n, k are odd, $k \ge n$ and D_d is the diameter.

$$D_d = \begin{cases} \max(n, \frac{k-1}{2}), & \text{if } k > n; \\ n-1, & \text{if } k = n. \end{cases} \tag{13}$$

Proof: Let d be the shortest distance between node $(0,0)$ and node (x, y), where x, y are not both zero.

Case I: $k > n$.

Consider $|x| < n$.

If both x, y are even or odd, i.e., $x' = x+y$ and $y' = -x+y$ are even, the routing algorithm (Proposition 1) for even x', y' applies. In this case, according to Proposition 1,

$$d = \frac{|x+y|}{2} + \frac{|-x+y|}{2} = |x| \text{ or } |y| < n.$$

If x is even and y is odd, or vice versa, x', y' are odd. From Proposition 3,

$$d = \begin{cases} k - |x|, & \text{if } (|x| + |y| < n) \\ & \quad \text{and } (|x| - |y| > k - n); \\ \max(|x|, n - |y|), & \text{if } (|x| + |y| \ge n) \\ & \quad \text{or } (|x| - |y| \le k - n) \end{cases}$$

From Figure 4(b), the smallest $|x|$ such that the following condition:

$$(|x| - |y| > k - n)$$

holds is $|x| = k - n + 1$. In this case, $d = n - 1$. Also,

$$|x| < n \Rightarrow \max(|x|, n - |y|) \le n.$$

Hence,

$$d \le \begin{cases} n-1, & \text{if } (|x| + |y| < n) \text{ and } (|x| - |y| > k - n); \\ n, & \text{if } (|x| + |y| \ge n) \text{ or } (|x| - |y| \le k - n); \end{cases}$$

and $\quad d = n$ if $|x| \le k - n, y = 0. \quad$ (14)

Consider $|x| \ge n$. From Proposition 5,

$$d = |x| \le \frac{k-1}{2};$$

and

$$d = \frac{k-1}{2} \text{ when } |x| = \frac{k-1}{2}. \tag{15}$$

From (14) and (15),

$$D_d = \max\left(n, \frac{k-1}{2}\right), \quad \text{if } k > n.$$

Case II: $k = n$.

In this case, $x, y \in \{-\frac{n-1}{2}, \cdots, \frac{n-1}{2}\}$, i.e., $|x| + |y|$ is always $< n$.

If x, y are both even or odd, x', y' are even,

$$d = |x| \text{ or } |y| \le \frac{n-1}{2}.$$

If x is even and y is odd, or vice versa, according to Proposition 3 with $k = n$ and $|x| + |y| < n$,

$$d = \begin{cases} n - |x|, & \text{if } |x| - |y| > 0; \\ \max(|x|, n - |y|) & \text{if } |x| - |y| < 0. \end{cases}$$

$$\Rightarrow d \le \begin{cases} n-1, & \text{if } |x| - |y| > 0; \\ n-1, & \text{if } |x| - |y| < 0. \end{cases}$$

and $d = n - 1$, $\quad$ when $|x| = 1, y = 0$ or $x = 0, |y| = 1$.

Thus

$$D_d = n - 1, \quad \text{if } k = n. \qquad \square$$

Based on (1) and (13), we have the following relationship between D_t and D_d:

Proposition 7: For an $N = n \times k$ network, assume n, k are *odd*, $k \geq n$.

$$\begin{array}{ll} D_t = D_d = n - 1 & \text{for } k = n. \\ D_t = D_d = n & \text{for } k = n + 2. \\ D_t - D_d = \frac{1}{2}(k - n - 2) > 0 & \text{for } n + 2 < k \leq 2n - 1; \\ D_t - D_d = \frac{n-1}{2} > 0 & \text{for } k \geq 2n + 1. \end{array}$$

The proof of this proposition is a straightforward comparison of (13) and (1), and therefore is omitted.

D. Examples

In this section, we use three examples to verify the diameters of diagonal mesh networks. Figures 5 and 6 show the distance of all nodes from the center node $(0, 0)$ of $N = 5 \times 5$, $N = 3 \times 5$ and $N = 3 \times 9$ diagonal mesh networks. In other words, the number at the center of each node shows how many hops the particular node is away from the center node $(0, 0)$. From these figures,

$$D_d = \begin{cases} 4, & \text{when } k = n = 5; \\ 3, & \text{when } n = 3, k = 5; \\ 4, & \text{when } n = 3, k = 9. \end{cases}$$

According to (13), the diameters are:

$$D_d = \begin{cases} n - 1 = 4, & \text{if } k = n = 5; \\ n = 3, & \text{if } n = 3, k = 5; \\ \frac{k-1}{2} = 4, & \text{if } n = 3, k = 9. \end{cases}$$

Using (1), the diameters for toroidal mesh networks are:

$$D_t = \begin{cases} 4, & \text{when } k = n = 5; \\ 3, & \text{when } n = 3, k = 5; \\ 5, & \text{when } n = 3, k = 9. \end{cases}$$

Hence, Propositions 6 and 7 are true.

In summary, the diagonal mesh network has a *smaller* diameter (when $k > n + 2$) and a *larger* bisection width than the toroidal mesh. It is also obvious that a diagonal mesh network retains all advantages such as a simple rectangular structure, wirability and scalability of the toroidal mesh. Based on these results and observations, we concluded that the diagonal mesh is potentially an attractive alternative to the toroidal mesh network. In the next section we further investigate and compare the performance of these two networks under a *constant system* with a fixed number of messages.

IV. Network Performance

In this section, we compare performance of diagonal and toroidal mesh networks using deflection routing algorithm. We first discuss deflection routing and various deflection criteria in Section IV-A, simulation results are presented in Section IV-B.

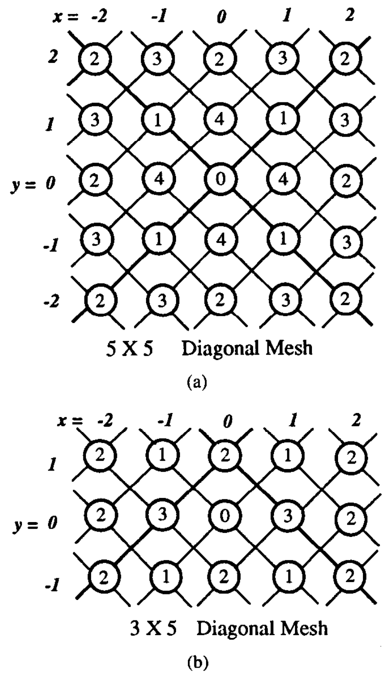


Fig. 5. Node distance of 5×5 and 3×5 diagonal mesh networks.

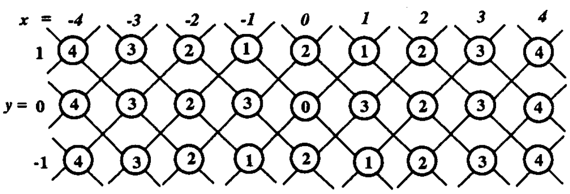


Fig. 6. Node distance of $N = 3 \times 9$ diagonal mesh network.

A. Deflection Routing

Deflection routing was first proposed in 1964 [18] under the name *hot potato routing* for communication networks. Since then, it has been used in both computer and communication networks [5], [6], [19]–[21] under the name *dynamic routing* and *deflection routing*. Popular examples include the Connection Machine [19], a massively parallel computer, and the Manhattan Street Network [6], a metropolitan area network.

Basically, messages are sorted according to a deflection criterion. Those with higher priorities are routed optimally while those with lower priorities are *deflected* to non-optimal links when conflicts for optimal links occur. The idea is for nodes to get rid of all incoming messages at each cycle. There is no buffer and hence no buffer management at a node. This routing algorithm is simple and straightforward to implement. We use this algorithm to evaluate performance of large diagonal and toroidal mesh networks.

Since both diagonal and toroidal mesh are bidirectional, degree-4 networks, there are 4 input and 4 output links at each node. We assume the network operates in a synchronous manner. At the beginning of each cycle, nodes receive incoming messages and at the end of a cycle, messages are routed to output links. At cycle 0, there are $N\text{msg} = 1, \cdots, 4$ at

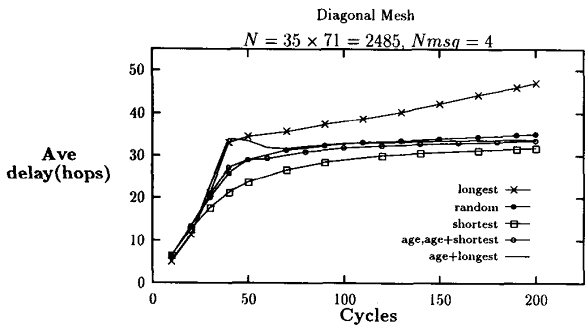


Fig. 7. Average delay for diagonal mesh with different deflection criteria.

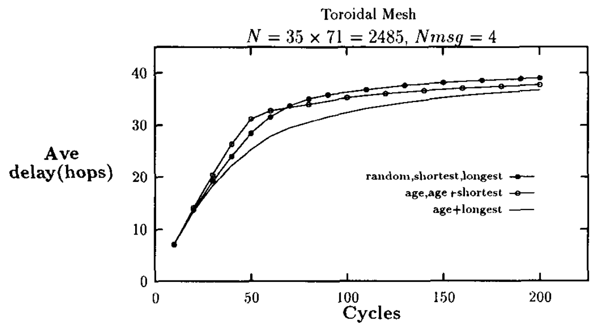


Fig. 8. Average delay for toroidal mesh with different deflection criteria.

each node. When $N\text{msg} = 1$ the system is lightly loaded and when $N\text{msg} = 4$ the system is at its full capacity because every node has only 4 bidirectional links. A pseudo-random number generator with uniform message distribution is used to generate the destination of each message. In the beginning of each subsequent cycle, the router at a node i checks the destination of all incoming messages. If the destination of a particular message is node i, then the message has reached its destination and is removed from the system. Instead, a new message is generated at node i to replace the deleted message. In other words, the total number of messages in the system remains constant, $N * N\text{msg}$. Subsequently, all incoming and new messages are sorted according to a deflection criterion. These messages are routed to output links according to their priorities. The optimal routing (path-determining) algorithms summarized in Tables I–III are used to determine the optimal out-going links for each message.

For simplicity, we assume a two-phase scheduling algorithm. In the first phase, the router goes through all messages in the input links according to their priorities. If a message has only one optimal output link which is not occupied by a message with a higher priority, the message is assigned to the optimal link. If a message has more than one optimal output link, the router chooses an available optimal link arbitrarily. In the case that there are no unoccupied optimal link, the message is left in the input links until all messages have been through the first phase. In the second phase, all messages left in the input links are routed arbitrarily to available output links. This two-phase scheme is simple to implement but is only sub-optimal because a message $m1$ may have more than one optimal link and one of which, say l, may be the sole option for another message $m2$ at a lower priority. This two-phase scheme may assign $m1$ to l and introduce unnecessary deflection for $m2$. A more sophisticated scheduling scheme can be developed to improve the performance but at a higher time complexity.

For deflection routing, proper choice of a deflection criterion is important to network performance. Inappropriate deflection criteria may cause *livelock* situation, in which a message is trapped in the system indefinitely. We have investigated six deflection criteria.

1) *random*: messages are routed in an arbitrary order;
2) *age*: an "older" message has a higher priority;
3) *shortest*: a "shorter" message has a higher priority;
4) *longest*: a "longer" message has a higher priority;
5) *age + shortest*: an "older" message has a higher priority but for messages with the same "age", a "shorter" message has a higher priority; and
6) *age + longest*: an "older" message has a higher priority but for messages with the same "age", a "longer" message has a higher priority.

An "older" message refers to one that has been introduced into the system earlier; a "shorter"/"longer" message is one that is a shorter/longer distance (number of hops) from its final destination. The "age" of a message is the number of cycles that a message has been in the system. In the next section, we present our simulation results for these criteria.

B. Simulation Results

For each network with a constant number of messages ($N \times N\text{msg}$), we observe the *average delay*, *maximum delay* and *throughput* of the system for a number of cycles. Average/maximum delay is the average/maximum path length (hops) in the system; and throughput is the average number of messages reached destination per cycle.

We first investigate effects of the six different deflection criteria by comparing results from toroidal and diagonal mesh networks with $N = 35 \times 71 = 2485$ nodes. We then compare the performance of the two networks using the same deflection criterion and $N = 35 \times 71 = 2485$, $N = 49 \times 99 = 4851$ and $N = 69 \times 139 = 9591$ nodes. From (1) and (13), the diameters for these networks are:

$$\begin{aligned} D_t &= 52, \quad D_d = 35 \quad \text{for } N = 35 \times 71 \\ D_t &= 63, \quad D_d = 49 \quad \text{for } N = 49 \times 99 \\ D_t &= 103, \quad D_d = 69 \quad \text{for } N = 69 \times 139 \end{aligned} \tag{16}$$

Deflection Criteria: Figures 7–10 show the effects of the six deflection criteria listed in Section IV-A for networks with $N = 2485$ nodes and $N\text{msg} = 4$. Since $N\text{msg} = 4$, the network is heavily loaded and a proper choice of the deflection criterion is critical.

For average delay, the criterion "*longest*" has an unbounded delay. This phenomenon is a result of the non-optimal two-phase scheduling scheme and the topological properties of diagonal mesh network. Our study indicates that, for diagonal mesh networks, most messages have two optimal outgoing links but those that are at a distance D_d (the diameter) from

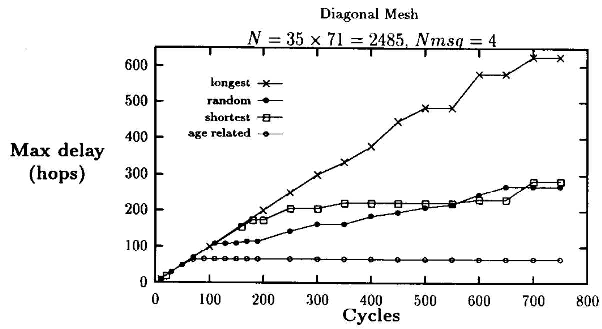


Fig. 9.- Maximum delay for diagonal mesh with different deflection criteria.

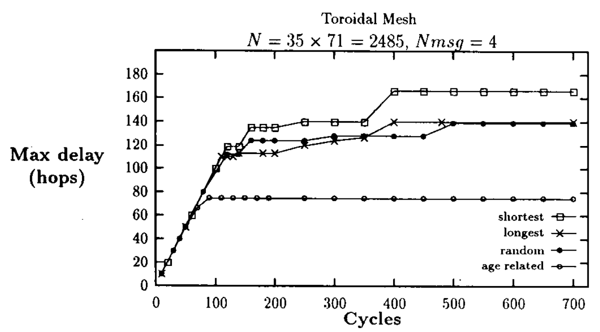


Fig. 10. Maximum delay for toroidal mesh with different deflection criteria.

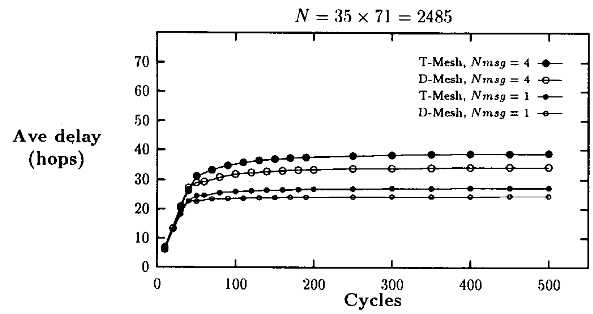


Fig. 11. Average delay for diagonal and toroidal mesh networks.

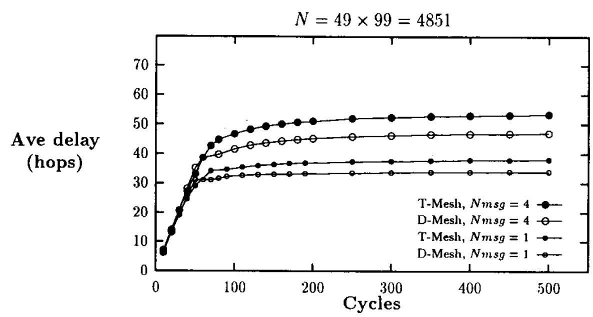


Fig. 12. Average delay for diagonal and toroidal mesh networks.

its destination have an average of 2.5 number of links. This property and the fact that two-phase scheduling is sub-optimal imply that more unnecessary deflections are introduced when longer messages have higher priorities and thus resulting in the unbounded average delay.

For the same reason, the criterion "*shortest*" demonstrates the smallest average delay. The is because by giving a shorter message (with fewer optimal options) a higher priority, unnecessary deflections are minimized. However, the maximum delay (Fig. 9) for "*shortest*" is among the highest. This is because a message's age is not considered and a long message may be trapped in the system. On the contrary, when a message's age is part of the deflection criterion, the maximum delay saturates after a certain number of cycles (Fig. 9). Therefore, *livelock* problem or high maximum delay can be avoided only if *age* is part of the deflection criterion.

For average delay (Fig. 7), the curve corresponding to *age* + *longest* follows that of *longest* initially when all messages have the same age; and later traces that of *age* when messages began to have different ages. Furthermore, the maximum delay among *age*, *age+shortest* and *age* + *longest* are almost indifferentiable. In fact, for clarity, Fig. 10 combines all the age related criteria into one curve. The effects of these criteria for larger networks ($N = 4851, 9591$) have also been investigated and again, their differences are diminutive. We therefore concluded that *age* alone is a simple and efficient criterion for diagonal mesh networks of these sizes.

Figures 8 and 10 show the average and maximum delay for a toroidal mesh with $N = 35 \times 71 = 2485$ nodes and $N\text{msg} = 4$. Again, when the age of a message is not part of the deflection criterion, maximum delay is much higher. Also, the average delay for *age*, *age* + *shortest* and *age* + *longest* is close after the system saturates. We therefore also concluded that *age* alone is a sufficient criterion for toroidal mesh networks of these sizes.

It is worth noting that the curve for criterian "*longest*" does not experience the unbounded increase as in the diagonal case. This is because, unlike the diagonal mesh, the majority of messages have 2 optimal links regardless of their distance from destinations. In other words, by giving a longer message a higher priority will not introduce additional unnecessary deflections to a shorter message.

Performance Comparisons: Figures 11–13 show the *average delay* for diagonal and toroidal mesh networks with $N = 2485, 4851, 9591$ nodes and $N\text{msg} = 1, 4$ using *age* as the deflection criterion. Note that $N\text{msg} = 1$ corresponds to a network with very light load whereas $N\text{msg} = 4$ implies the network is fully loaded.

We observed that the average delay saturates after certain number of cycles. This result is consistent with our constant system model. Under this model, there is a fixed number of messages, $N \times N\text{msg}$, in the system and their destinations are uniformly distributed. Intuitively, there should be a characteristic average associated with each network and network load. This average path length should be bigger for larger networks and network loads. These arguments are confirmed by our simulations. Furthermore, the diagonal mesh network always has a smaller average path length at saturation. Such difference between the two networks also increases with the network size and network load. We have also investiaged the maximum delay for these networks. We found that they depicted similar

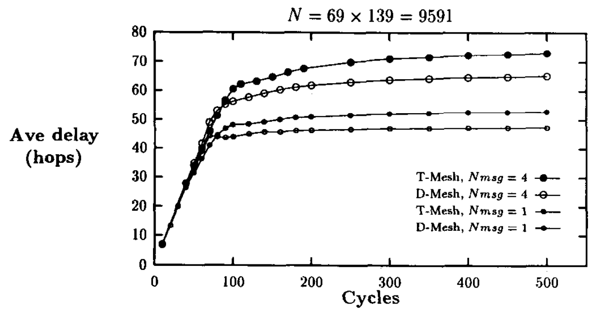


Fig. 13. Average delay for diagonal and toroidal mesh networks.

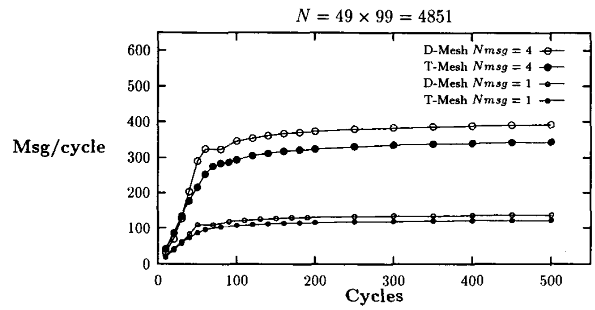


Fig. 15. Throughput for diagonal and toroidal mesh networks.

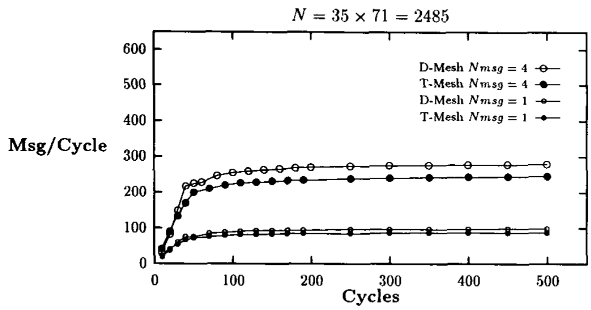


Fig. 14. Throughput for diagonal and toroidal mesh networks.

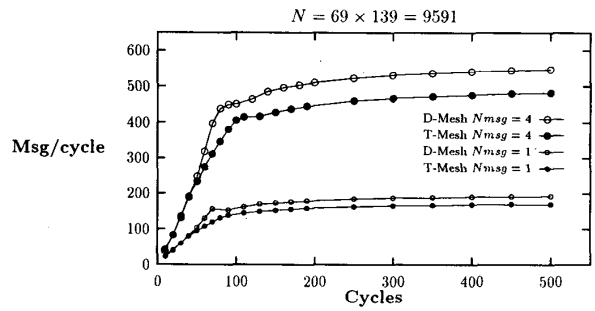


Fig. 16. Throughput for diagonal and toroidal mesh networks.

behavior as the average delay except that the magnitude of the saturated maximum delay is about twice that of the average delay.

Figures 14–16 show the *throughput* for diagonal and toroidal mesh networks with $N = 2485, 4851, 9591$ nodes and $N\text{msg} = 1, 4$ using *age* as the deflection criterion. Again, due to the constant number of messages, the throughput of the system saturates after a certain number of cycles. This saturation occurs later for larger networks and networks with heavier loads. Similar to the average delay, the diagonal mesh network always have a higher (better) throughput and this difference in performance increases with network size and network load.

V. Conclusion

The toroidal mesh is a popular and well-studied network. It is a *symmetric, wirable* and *scalable* network with an optimal self-routing algorithm. However, its drawbacks include a relatively large *diameter* and a small *bisection width.* For an $N = n \times k$ toroidal mesh with n, k odd and $k \geq n$, the diameter and bisection width are $D_t = (\frac{n-1}{2} + \frac{k-1}{2})$ and $B_t = 2n + 2$. These drawbacks imply potentially long communication delay and thus hamper network performance.

The diagonal mesh is similar to the popular toroidal mesh, except that the nodes are diagonally connected. In other words, it is also a degree-4, point to point interconnection model suitable for connecting communication elements in parallel computers, particularly multicomputers. Furthermore, it retains advantages such as *symmetry, wirability,* and *scalability* of the toroidal mesh.

In this paper, we developed an optimal, self-routing algorithm, and proposed and proved an analytic formula for the diameter of diagonal mesh networks. We showed that for an $N = n \times k$ diagonal mesh with n, k odd and $k \geq n$, the diameter is: $D_d = n - 1$ for $k = n$ and $D_d = \max(n, \frac{k-1}{2})$ for $k > n$. In other words, $D_d = D_t$ for $k = n, n+2$. But when k is strictly greater than $n + 2$, the diagonal mesh has a smaller diameter and thus a potentially smaller communication delay. We also showed that the bisection bandwidth of the corresponding diagonal mesh network is $B_d = 4n$, an improvement over the toroidal mesh. These topological properties show that the diagonal mesh network has a potentially better performance than the toroidal mesh. This result is further strengthened by our computer simulations.

We have simulated and compared the performance of diagonal and toroidal mesh networks in the presence of contention. For both diagonal and toroidal mesh, we considered networks with $N = 35 \times 71 = 2485$, $N = 49 \times 99 = 4851$, and $N = 69 \times 139 = 9591$ nodes. We assume communication is achieved in a synchronous manner, in which every node receives incoming or new messages at the beginning of a cycle and routes messages to output links at the end of a cycle. At cycle 0, every node has $N\text{msg} = \{1, \cdots, 4\}$ to be routed. When a message reaches its destination i, a new message is generated at i to replace the deleted message. In other words, the network is a *constant system* with a fixed number of messages $N \times N\text{msg}$.

To evaluate the performance of the network, we use the *deflection routing*, a dynamic and bufferless routing algorithm popular for both computer and communication networks.

There is a deflection criterion that determines the priority of messages. When conflicts for the same optimal outgoing links occur among messages, those with lower priorities are *deflected* to non-optimal out-going links. There is no buffer and hence no buffer management at a node. However, a proper deflection criterion is critical to the performance of the network.

Using networks with $N = 35 \times 71 = 2485$, we showed that the *age*, the number of cycles a message has been in the system, is a simple and efficient deflection criterion. We then use this deflection criterion to simulate the performance of networks with $N = 35 \times 71 = 2485$, $N = 49 \times 99 = 4851$, and $N = 69 \times 139 = 9591$ nodes and $N\mathrm{msg} = 1, 4$. We observe the *average delay*, *throughput* and the *maximum delay* of the system. Due to the constant system model, the performance of these networks saturates after a certain number of cycles. In all cases, the diagonal mesh outperforms the toroidal mesh. Furthermore, the difference in performance increases with the network size and network loads. Based on these results, we concluded that a diagonal mesh network, particularly one with $N = n \times k$ nodes and $k > n + 2$, is an attractive alternative to its toroidal counterpart.

REFERENCES

[1] T. Y. Feng, "A survey of interconnection networks," *Computer*, vol. 14, pp. 12–27, Dec. 1981.

[2] L. D. Wittie, "Communication structures for large networks of microcomputers," *IEEE Trans. Comput.*, vol. 30, no. 4, pp. 264–273, Apr. 1981.

[3] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures; Arrays, Trees, Hypercubes*. San Mateo, CA: Morgan Kaufmann, 1992.

[4] C. D. Thompson, "Area-time complexity," in *Proc. Eleventh Annu. ACM Symp. The Theory of Computing*, Apr. 1979, pp. 81–88.

[5] M. Ben-Ayed, "Dynamic routing for regular direct computer networks," Ph.D. thesis, Elec. Eng. Dep., College of Eng. and Appl. Sci., Univ. of Rochester, Rochester, NY, 1990.

[6] N. F. Maxemchuk, "Routing in the Manhattan street network," *IEEE Trans. Commun.*, vol. 35, no. 5, pp. 503–512, May 1987.

[7] ——, "Comparison of deflection and store-and-forward techniques in the Manhattan street and shuffle-exchange networks," in *Proc. IEEE INFOCOM'89*, 1989, pp. 800–809.

[8] J. C. Bermond and C. Delorme, "Strategies for interconnection networks: Some methods from graph theory," *J. Parallel and Distrib. Computing*, vol. 3, pp. 433–449, 1986.

[9] K. W. Tang, "Dense symmetric interconnection networks," Ph.D. thesis, Elec. Eng. Dep., College of Eng. and Appl. Sci., Univ. of Rochester, Rochester, NY, 1991.

[10] B. W. Arden, private communication, 1991.

[11] B. W. Arden and F. Li, "Simulation of interconnections for massively parallel systems," Tech. Rep., Dep. of Elec. Eng., Univ. of Rochester, Rochester, NY, 1991.

[12] K. W. Tang and S. A. Padubidri, "Routing and diameter analysis of diagonal mesh networks," In *Proc. Int. Conf. Parallel Processing*, St. Charles, IL, Aug. 17–21 1992, pp. I-143–150.

[13] G. S. Almasi and A. Gottlieb, *Highly Parallel Computing*. New York: Benjamin/Cummings, 1989.

[14] D. Parkinson and J. Litt, Eds., *Massively Parallel Computing with the DAP*,. Cambridge, MA: MIT Press, 1990.

[15] N. F. Maxemchuk, "Regular mesh topologies in local and metropolitan area networks," *AT&T Tech. J.*, vol. 64, no. 7, pp. 1659–1685, Sept. 1985.

[16] F. Borgonovo and E. Cadarin, "Routing in the bidirectional Manhattan network," in *Proc. Third . Conf. Data Commun. Syst. and their Perform.*, Rio de Janerio, June 1987, pp. 181–189.

[17] K. W. Tang and B. W. Arden, "Vertex-transitivity and routing for Cayley graphs in GCR representations," in *Proc. 1992 Symp. Appl. Computing*, Kansas City, MO, Mar. 1–3, 1992, pp. 1180–1187.

[18] P. Barnes, "On distributed communication networks," *IEEE Trans. Commun. Syst.*, vol. 12, pp. 1–9, 1964.

[19] W. D. Hillis, *The Connection Machine*. Cambridge, MA: MIT Press, 1985.

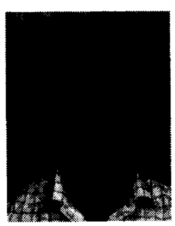
[20] A. Krishna and B. Hajek, "Performance of shuffle-like switching networks with deflection," in *Proc. IEEE INFOCOM'90*, San Francisco, CA, June 1990, pp. 473–480.

[21] M. Decina, P. Giacoazzi, and A. Pattavina, "Shuffle interconnection networks with deflection routing for ATM switching: The closed-loop shuffleout," in *Proc. IEEE INFOCOM'91*, Bal Harbour, FL, Apr. 1991, pp. 1253–1263.

K. Wendy Tang (S'89–M'90–S'91–M'91) received her B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Rochester, Rochester, NY in 1986, 1988, and 1991, respectively.

Since 1991, she has been an Assistant Professor at the Department of Electrical Enginnering in the State University of New York at Stony Brook. Her current research interests include parallel and neural computing, interconnection networks, communication networks, and robotics.

Dr. Tang is a member of the IEEE Computer Society, the Association for Computing Machinery, the Society for Industrial and Applied Mathematics, the International Neural Network Society, and the Society of Women Engineers.



Sanjay Padubidri received the B.Engg. degree from Bangalore University, India, in 1990 and the M.S. degree from SUNY at Stony Brook in 1993, both in electrical engineering.

He is currently a Ph.D. student in the Department of Computer Science, SUNY at Stony Brook. His current research interests are in the area of parallel processing architectures, gigabit networks and compilation techniques for parallel machines.