ELSEVIER Discrete Applied Mathematics 83 (1998) 303–326

DISCRETE APPLIED MATHEMATICS

Optimal communication algorithms for Manhattan Street networks ☆

Emmanouel A. Varvarigos

University of California, Department of Electrical and Computer Engineering, Santa Barbara, CA 93106, USA

Received 20 April 1995; accepted 29 March 1996

Abstract

We evaluate the mean internodal distance and the saturation throughput of the Manhattan Street network in the general case, and present a simple shortest path routing algorithm for this topology. We also propose efficient routing algorithms to execute generic communication tasks, such as the total exchange and the multinode broadcast, in a Manhattan Street network of processors. The proposed algorithms execute these tasks in an optimal number of steps, while achieving full (equal to 100%) utilization of the network links. Finally, we construct edge-disjoint Hamiltonian cycles for the Manhattan Street network. © 1998 Elsevier Science B.V. All rights reserved.

1. Introduction

A useful performance measure for a multiprocessor network is its mean internodal distance, which is defined as the shortest distance between two nodes, averaged over all pairs of nodes. The mean internodal distance determines the average latency of the network for low load, and it is closely related to its saturation throughput for uniformly distributed destinations.

In addition to the latency and the throughput, an important measure of the performance of a multiprocessor network is the time it takes to execute certain prototype communication tasks in that (see [1]). Saad and Shultz [13, 14] were the first to identify a number of generic communication problems. Two communication tasks that arise frequently in numerical and other methods and are generally regarded as prototype tasks are the *multinode broadcast* (MNB) and the *total exchange* (TE) tasks. The MNB involves broadcasting a packet (the same packet) from every node to all the other nodes. It arises, e.g., in iterations of the form

$$x = f(x),$$

☆ Research supported by NSF under Grant NSF-RIA-08930554.

where each processor computes an entry (or a block of entries) of vector x. At the end of an iteration, each processor has to broadcast the updated value of the component that it computes to all other processors in order to be used at the next iteration; this is a MNB. In the TE task, each network node has to send a *personalized* (different) packet to each one of the other nodes. The TE arises, e.g., in the transposition of a matrix, when each processor stores, say, a column of the matrix. To transpose the matrix, every processor i has to send the (i,k)th entry to processor k, for all k, which is a TE.

In this paper we focus on communication problems in the Manhattan Street (abbreviated MS) network, which is two-connected regular mesh topology with unidirectional communication links. We first describe a simple shortest path algorithm to route packets between any pair of nodes. We also derive a closed-form expression for the mean internodal distance of the MS network in the general case. Previous expressions on the mean internodal distance of the MS network were restricted to the special case of a square MS network [9], or they were given in the form of conjectures verified by computer simulations (see [12] for the special case where the number of nodes across each dimension is a multiple of 4).

We also propose communication tasks to execute certain prototype communication tasks in a square MS network of processors. In particular, we present an algorithm to execute the TE task in a MS network in optimal time. In the proposed TE algorithm, packets follow shortest paths to their destination, and full (equal to 100%) utilization of the communication links is achieved. As a result, our TE algorithm is unimprovable, in the sense that using a different switching format (such as wormhole routing) would not decrease the completion time. We also present two MNB algorithms, each of which uses a different set of assumptions regarding the communication model. Both algorithms execute the MNB task in optimal time. In the course of developing the second MNB algorithm, we also show that the MS network contains two edge-disjoint Hamiltonian cycles. To the best of our knowledge, this is the first time that the TE and the MNB communication tasks are considered for the MS network. Algorithms to perform a MNB or a TE in other multiprocessor networks of interest can be found in [1,2,5,6,8,15,16]. For work on other routing aspects in MS networks see [3,7,10,11].

In our algorithms we assume that all packets have equal lengths, and they require one unit of time (or slot) for transmission over a link. All links of a node can be used simultaneously, but only one packet can travel along a link in a given direction at any one time. Thus, if more than one packets are available at a node and are scheduled to be transmitted on the same incident link of the node, then only one of these packets can be transmitted at the next time period, while the remaining packets must be stored at the node while waiting in queue. An algorithm that performs a communication task will be called optimal if its execution time is the minimum possible.

The organization of the paper is the following. In Section 2 we introduce the notation, and we derive closed-form expressions for the shortest distance between any pair of nodes in the MS network. In Section 3 we evaluate the mean internodal distance of the MS network. In Section 4 we present an algorithm to execute the total exchange task. In Sections 5 and 6 we present two algorithms for the multinode broadcast task, and describe a way to construct edge-disjoint Hamiltonian cycles in a MS network. Finally, in the appendix we resolve some technical issues that arise in the remainder of the paper.

2. Shortest paths in the Manhattan Street network

The $X \times Y$-dimensional wraparound mesh consists of XY processors arranged along the points of a two-dimensional space that have integer coordinates. There are X processors along the x-dimension and Y processors along the y-dimension, where X and Y are even numbers (note that X and Y have to be even for the network to be symmetric). Each processor has two outgoing links, one horizontal and one vertical. The horizontal links are directed eastwards on even rows and westwards on odd rows, while the vertical links are directed northwards on even columns and southwards on odd columns (see Fig. 1). Each processor is represented by a pair (x, y), with $0 \leqslant x \leqslant X - 1$ and $0 \leqslant y \leqslant Y - 1$. If we define the parity function

$$p(n) = \begin{cases} 1 & \text{if } n \text{ even,} \\ -1 & \text{if } n \text{ odd,} \end{cases}$$

then each processor (x, y) is connected by unidirectional links to processors $(x + p(x) \bmod X, y)$, and $(x, y + p(y) \bmod Y)$.

A path of length L can be represented by the pair $((x, y), \mathscr{L})$, where (x, y) is the origin of the path, and $\mathscr{L} = l_1 l_2, \ldots, l_L$ is an L-bit long binary number, called *routing tag*, whose ith bit is zero if the path crosses an horizontal link during the ith hop, and one if it crosses a vertical link. When no confusion can arise, we will refer to a path by its routing tag $\mathscr{L}$, dropping the origin.

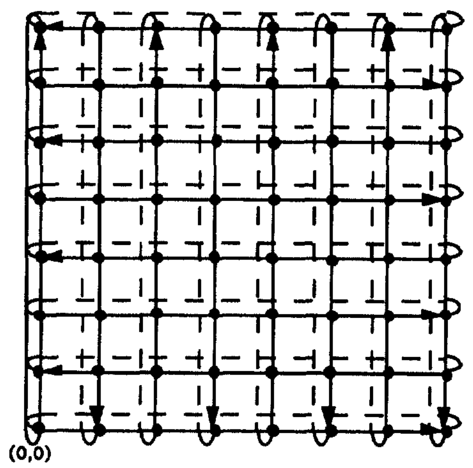

Fig. 1. An 8 × 8 Manhattan Street network.

The *relative address* $R_{x,y}(u,v)$ of node (u,v) *with respect* to node (x,y) is defined as

$$R_{x,y}(u,v) = ((u-x)p(x) \bmod X, (v-y)p(y) \bmod Y)$$
$$= \begin{cases} (u-x \bmod X,\ v-y \bmod Y), & x \text{ even},\ y \text{ even}, \\ (x-u \bmod X,\ v-y \bmod Y), & x \text{ odd},\ y \text{ even}, \\ (u-x \bmod X,\ y-v \bmod Y), & x \text{ even},\ y \text{ odd}, \\ (x-u \bmod X,\ y-v \bmod Y), & x \text{ odd},\ y \text{ odd}. \end{cases}$$

We let $\mathcal{N}$ be the set of nodes of the network. The following lemma is useful in characterizing the symmetry of the MS network.

Lemma 1. *The functions*

$$\mathcal{F} : (u,v) \to R_{x,y}(u,v)$$

and

$$\mathcal{G} : (x,y) \to R_{x,y}(u,v)$$

from $\mathcal{N}$ to $\mathcal{N}$ are one-to-one.

Proof. It is easy to see that $\mathcal{F}$ is one-to-one. To prove that $\mathcal{G}$ is also one-to-one, it is enough to prove that $\mathcal{G}((x_1,y_1)) = \mathcal{G}((x_2,y_2))$ implies $(x_1,y_1) = (x_2,y_2)$. Assuming $\mathcal{G}((x_1,y_1)) = \mathcal{G}((x_2,y_2))$, we have

$$(u-x_1)p(x_1) \bmod X = (u-x_2)p(x_2) \bmod X.$$

If we show that $p(x_1) = p(x_2)$ then the preceding equation will give $x_1 = x_2$. The proof that $p(x_1) = p(x_2)$ will be done by contradiction. Assume, without loss of generality, that $1 = p(x_1) \neq p(x_2) = -1$. We then have $(2u - x_1 - x_2) \bmod X = 0$, which implies (since X is even) that $(x_1 + x_2) \bmod 2 = 0$, or, equivalently, $p(x_1) = p(x_2)$. This contradicts our hypothesis. Therefore, we have $p(x_1) = p(x_2)$, from which it follows that $x_1 = x_2$. We can similarly show that $y_1 = y_2$. $\square$

The function $R_{x,y}(\cdot,\cdot)$ is an automorphism that relabels the Manhattan Street network, placing (x,y) at the origin. For any path with routing tag $\mathcal{S}$ starting at (x,y) and ending at (u,v), there is a corresponding path with routing tag $\mathcal{S}$ starting at $(0,0)$ and ending at $R_{x,y}(u,v)$. Therefore, instead of considering paths between any pair of nodes, we can focus on paths originating at node $(0,0)$, and transfer the results obtained to paths originating at any node (x,y), by using the isomorphism $R_{x,y}(\cdot,\cdot)$.

When considering paths with origin $(0,0)$ and destination (i,j), we will limit our attention to four types of paths, which we call the North–East, the South–West, the South–East, and the North–West paths. The way these paths are defined is indicated in Fig. 2. Their lengths can be found to be as follows:

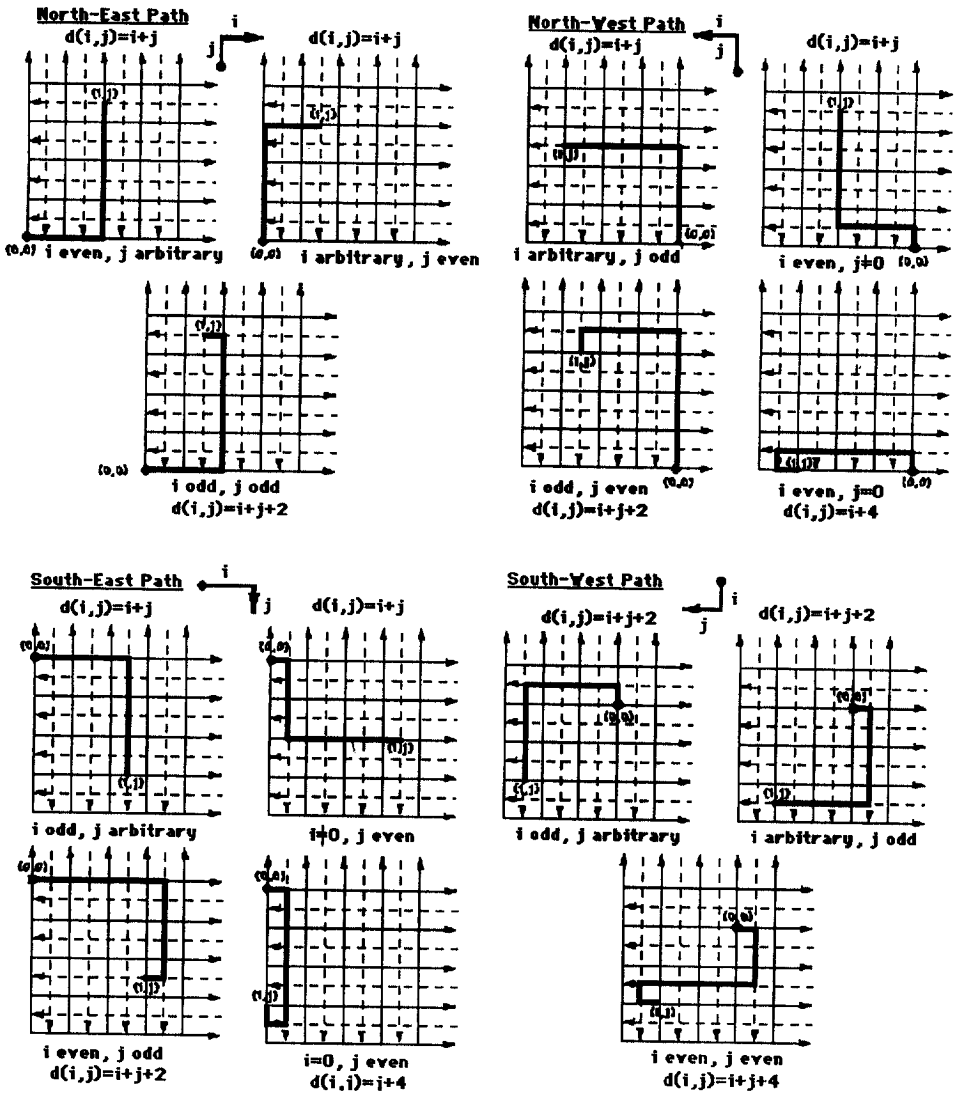

Fig. 2. The NE, SW, SE, and NW paths with origin (0,0) and destination (i,j).

North–East (NE):

$$d_{\mathrm{NE}}(i,j)=\begin{cases} i+j+2 & \text{if } i \text{ odd, } j \text{ odd,} \\ i+j & \text{otherwise.} \end{cases} \tag{1}$$

South–West (SW):

$$d_{\mathrm{SW}}(i,j)=\begin{cases} i+j+4 & \text{if } i \text{ even, } j \text{ even,} \\ i+j+2 & \text{otherwise.} \end{cases} \tag{2}$$

South–East (SE):

$$d_{\mathrm{SE}}(i,j)=\begin{cases} i+j+2 & \text{if } i \text{ even, } j \text{ odd,}\\ j+4 & \text{if } i=0,\ j \text{ even,}\\ i+j & \text{otherwise.}\end{cases} \tag{3}$$

North–West (NW):

$$d_{\mathrm{NW}}(i,j)=\begin{cases} i+j+2 & \text{if } i \text{ odd, } j \text{ even,}\\ i+4 & \text{if } i \text{ even, } j=0,\\ i+j & \text{otherwise.}\end{cases} \tag{4}$$

A packet with origin $(0,0)$ and destination (x,y) can be routed over a NE path of length $d_{\mathrm{NE}}(x,y)$, or a SW path of length $d_{\mathrm{SW}}(X-x,Y-y)$, or a SE path of length $d_{\mathrm{SE}}(x,Y-y)$, or a NW path of length $d_{\mathrm{NW}}(X-x,y)$. The shortest path distance from node $(0,0)$ to node (x,y) is

$$D_{0,0}(x,y)=\min(d_{\mathrm{NE}}(x,y), d_{\mathrm{SW}}(X-x,Y-y), d_{\mathrm{SE}}(x,Y-y), d_{\mathrm{NW}}(X-x,y)). \tag{5}$$

In order to simplify Eq. (5), we distinguish four areas, depending on whether x and y are odd or even numbers.

Case x odd, y odd: The network is partitioned into four areas (quadrants) as shown in Fig. 3(a). In area A_1 the shortest path is a NE path, in area A_2 it is a SW path, in area A_3 it is a SE path, and in area A_4 it is a NW path. On each of these areas the shortest distance $D_{0,0}(x,y)$ is given by a different expression. Using Eqs. (1)–(5) and the fact that x and y are odd numbers, we get after some algebraic manipulation that

$$\begin{aligned} D_{0,0}(x,y) &= \begin{cases} d_{\mathrm{NE}}(x,y) & \text{if } (x,y)\in A_1,\\ d_{\mathrm{SW}}(X-x,Y-y) & \text{if } (x,y)\in A_2,\\ d_{\mathrm{SE}}(x,Y-y) & \text{if } (x,y)\in A_3,\\ d_{\mathrm{NW}}(X-x,y) & \text{if } (x,y)\in A_4,\end{cases}\\ &= \begin{cases} x+y+2 & \text{if } (x,y)\in A_1,\\ X-x+Y-y+2 & \text{if } (x,y)\in A_2,\\ x+Y-y & \text{if } (x,y)\in A_3,\\ X-x+y & \text{if } (x,y)\in A_4.\end{cases} \end{aligned} \tag{6}$$

Case x even, y even: The network is again partitioned into areas A_1, A_2, A_3, and A_4, as shown in Fig. 3(b). Using Eqs. (1)–(5) and the fact that x and y are even

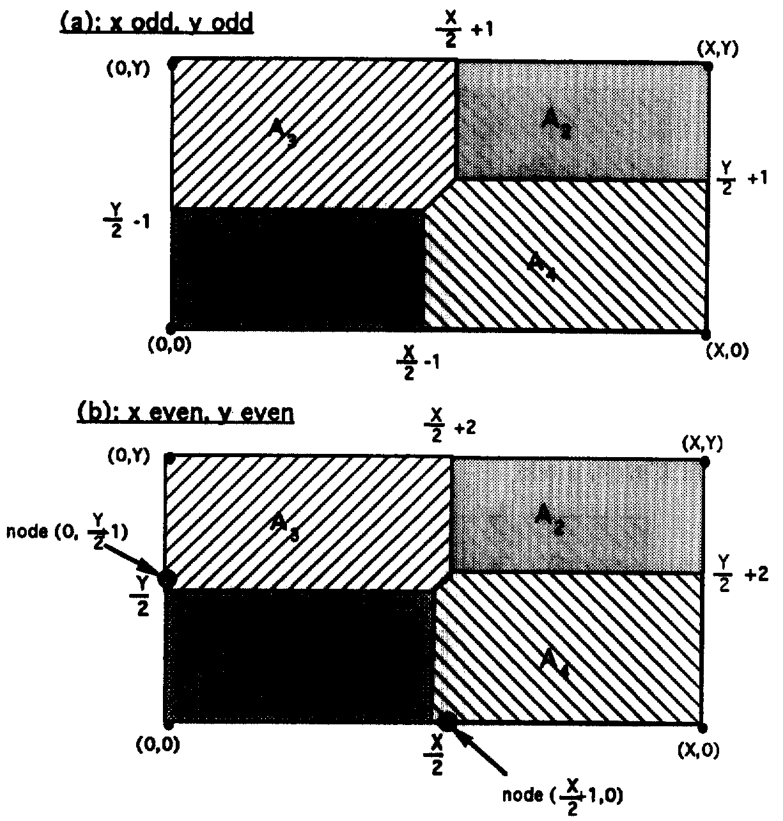

Fig. 3. (a) The areas A_i, $i=1,\dots,4$ for the case where x and y are odd numbers. Note that area A_1 also includes the nodes $((X/2)+1,0)$ and $(0,(Y/2)+1)$. (b) The corresponding areas for the case where x and y even are even numbers.

numbers, we get

$$D_{0,0}(x,y)=\begin{cases} x+y & \text{if } (x,y)\in A_1,\\ X-x+Y-y+4 & \text{if } (x,y)\in A_2,\\ x+Y-y & \text{if } (x,y)\in A_3,\ x\neq 0,\\ Y-y+4 & \text{if } (x,y)\in A_3,\ x=0,\\ X-x+y & \text{if } (x,y)\in A_4,\ y\neq 0,\\ X-x+4 & \text{if } (x,y)\in A_4,\ y=0. \end{cases} \tag{7}$$

Case x odd, y even: The network is partitioned into four areas A_1, A_2, A_3, A_4, as shown in Fig. 4(a). Using Eqs. (1)–(5) and the fact that x is odd and y is even, we

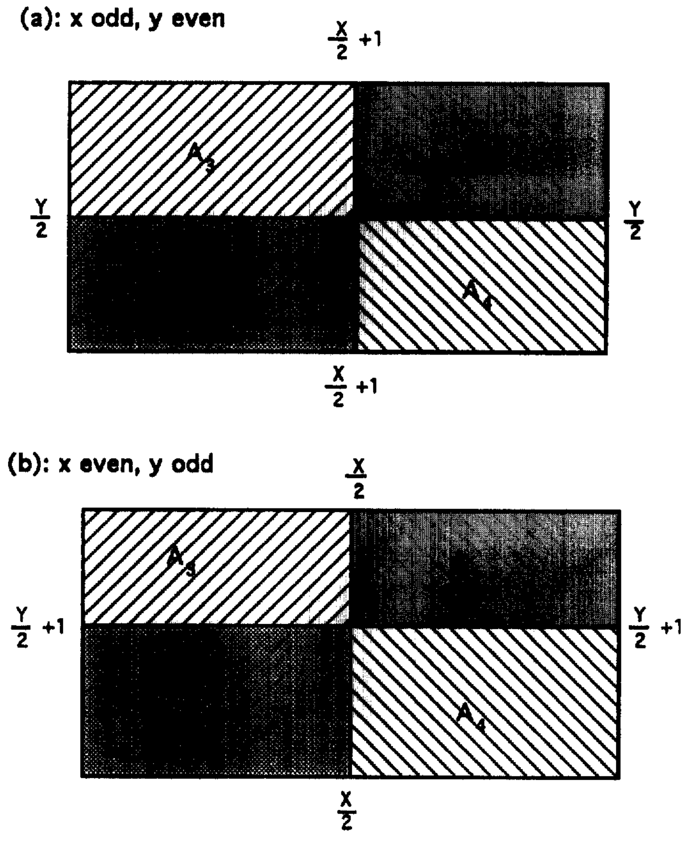

Fig. 4. (a) The areas A_i, $i = 1, \ldots, 4$ for the case where x is odd and y is even. (b) The corresponding areas for the case where x is even and y is odd.

find that

$$D_{0,0}(x,y) = \begin{cases} x+y & \text{if } (x,y) \in A_1, \\ X-x+Y-y+2 & \text{if } (x,y) \in A_2, \\ x+Y-y & \text{if } (x,y) \in A_3, \\ X-x+y+2 & \text{if } (x,y) \in A_4. \end{cases} \tag{8}$$

Case x even, y odd: The network is partitioned into areas A_1, A_2, A_3, A_4, as shown in Fig. 4(b). Using Eqs. (1)–(5) and the fact that x is even and y is odd, we obtain

$$D_{0,0}(x,y) = \begin{cases} x+y & \text{if } (x,y) \in A_1, \\ X-x+Y-y+2 & \text{if } (x,y) \in A_2, \\ x+Y-y+2 & \text{if } (x,y) \in A_3, \\ X-x+y & \text{if } (x,y) \in A_4. \end{cases} \tag{9}$$

Eqs. (6)–(9) give the shortest path distance from node $(0,0)$ to node (x,y). The shortest path distance $D_{u,v}(x,y)$ from an arbitrary node (u,v) to node (x,y) can be found from these equations using the relation

$$D_{u,v}(x,y) = D_{0,0}(R_{u,v}(x,y)).$$

These are, to the best of our knowledge, the first closed-form expression for the shortest distance between any pair of nodes of the MS network. Eqs. (6)–(9) also give implicitly the shortest paths: for example if x and y are odd numbers and (x,y) belongs to the area A_2 of Fig. 3(a), then the shortest path from node $(0,0)$ to node (x,y) will be the SW path. The next section is concerned with the determination of the mean internodal distance for general values of X and Y.

3. Mean internodal distance of the Manhattan Street network

The mean internodal distance $\bar{D}$ is defined as the average of the shortest distances over all pairs of nodes. Since the MS network is symmetric, we have

$$\bar{D} = \frac{H}{XY},$$

where H is the sum of the shortest distances of all nodes from node $(0,0)$:

$$H = \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} D_{0,0}(x,y) = \sum_{x=0,\ x\text{ odd}}^{X-1}\ \sum_{y=0,\ y\text{ odd}}^{Y-1} D_{0,0}(x,y) + \sum_{x=0,\ x\text{ even}}^{X-1}\ \sum_{y=0,\ y\text{ even}}^{Y-1} D_{0,0}(x,y)$$

$$+ \sum_{x=0,\ x\text{ odd}}^{X-1}\ \sum_{y=0,\ y\text{ even}}^{Y-1} D_{0,0}(x,y) + \sum_{x=0,\ x\text{ even}}^{X-1}\ \sum_{y=0,\ y\text{ odd}}^{Y-1} D_{0,0}(x,y). \tag{10}$$

To find H we have to substitute Eq. (9) into Eq. (10). To proceed with the calculations, we distinguish four cases:

Case $X = 4k,\ Y = 4m$: After some calculations, described in the appendix, we get

$$\sum_{x\text{ odd}}\sum_{y\text{ odd}} D_{0,0}(x,y) = 4km(1 + m + k), \tag{11}$$

$$\sum_{x\text{ even}}\sum_{y\text{ even}} D_{0,0}(x,y) = 4mk(m + k + 1) - 4, \tag{12}$$

$$\sum_{x\text{ odd}}\sum_{y\text{ even}} D_{0,0}(x,y) = 4mk(m + k + 1), \tag{13}$$

and

$$\sum_{x\text{ even}}\sum_{y\text{ odd}} D_{0,0}(x,y) = 4mk(m + k + 1). \tag{14}$$

Using Eqs. (10)–(14), we obtain

$$H = XY\left(\frac{X+Y}{4}+1\right)-4. \tag{15}$$

Case $X = 4k + 2k$, $Y = 4m + 2$: After some calculations, described in the appendix, we get

$$\sum_{x \text{ odd}}\sum_{y \text{ odd}} D_{0,0}(x,y) = 2mk(m+k+2) + 2(k+1)(m+1)(m+k+2) \\ -2k-2m-2, \tag{16}$$

$$\sum_{x \text{ even}}\sum_{y \text{ even}} D_{0,0}(x,y) = (k+1)(m+1)(m+k) \\ +4(k-1)(m-1) + (k-1)(m-1)(m+k) \\ +(k+2)m(k+m+2) + (m+2)k(m+k+2) \\ +2k+2m-6, \quad \text{for } k,m>1, \tag{17}$$

$$\sum_{x \text{ odd}}\sum_{y \text{ even}} D_{0,0}(x,y) \\ = (k+1)m(m+1) + (m+1)(k+1)^2 + 2km + k(m+1)m + mk^2 \\ +(k+1)m(m+1) + m(k+1)^2 + k(m+1)(m+2) + (m+1)k^2, \tag{18}$$

and

$$\sum_{x \text{ even}}\sum_{y \text{ odd}} D_{0,0}(x,y) \\ = (m+1)k(k+1) + (k+1)(m+1)^2 + 2km + m(k+1)k + km^2 \\ +(m+1)k(k+1) + k(m+1)^2 + m(k+1)(k+2) + (k+1)m^2. \tag{19}$$

Using Eq. (10) and Eqs. (16)–(19), we obtain

$$H = (X+Y)\frac{XY}{4} + XY - X - Y - 2 \quad \text{for } X,Y>2. \tag{20}$$

Case $X = 4k + 2$, $Y = 4m$: After some calculations, described in the appendix, we get

$$\sum_{x \text{ odd}}\sum_{y \text{ odd}} D_{0,0}(x,y) = 2mk(m+k+2) + 2(k+1) + m(m+k+1), \tag{21}$$

$$\sum_{x \text{ even}}\sum_{y \text{ even}} D_{0,0}(x,y) \\ = 2(m+k)(mk+1) + 2(m+k+1)(mk+m-1) + 4mk - 2, \tag{22}$$

$$\sum_{x \text{ odd}}\sum_{y \text{ even}} D_{0,0}(x,y) = 2km(m+k+2) + 2m(k+1)(m+k+1), \tag{23}$$

and

$$\sum_{x \text{ even}} \sum_{y \text{ odd}} D_{0,0}(x,y) = 2km(m+k+2) + 2m(k+1)(m+k+1). \tag{24}$$

Using Eq. (10) and Eqs. (21)–(24), we obtain

$$H = XY\frac{X+Y}{4} + XY - Y - 4 \quad \text{for } X>2. \tag{25}$$

Case $X=4k$, $Y=4m+2$: This case is analogous to the preceding one, with X and Y interchanged. Therefore,

$$H = XY\frac{X+Y}{4} + XY - X - 4 \quad \text{for } Y>2. \tag{26}$$

Combining Eqs. (15), (20), (25), and (26), we get that the mean internodal distance of the MS network is

$$\bar{D} = \frac{H}{XY}$$
$$= \begin{cases} \frac{X+Y}{4} + 1 - \frac{4}{XY} & \text{if } X = 0 \bmod 4,\ Y = 0 \bmod 4, \\ \frac{X+Y}{4} + 1 - \frac{1}{X} - \frac{1}{Y} - \frac{2}{XY} & \text{if } X = 2 \bmod 4,\ Y = 2 \bmod 4,\ X, Y > 2, \\ \frac{X+Y}{4} + 1 - \frac{1}{X} - \frac{4}{XY} & \text{if } X = 2 \bmod 4,\ Y = 0 \bmod 4,\ X > 2, \\ \frac{X+Y}{4} + 1 - \frac{1}{Y} - \frac{4}{XY} & \text{if } X = 0 \bmod 4,\ Y = 2 \bmod 4,\ Y > 2. \end{cases} \tag{27}$$

Eq. (27) answers an open question concerning the average internodal distance of the MS network in the *general case*. In order to verify our calculations, we ran a shortest path algorithm on all the MS networks with $4 \leqslant X, Y \leqslant 22$. The results obtained were in agreement with the closed-form expression of Eq. (27). The expression for the mean internodal distance is also in agreement with a conjecture by Rose [12] (who performed computer calculations on MS networks with up to 50 000 nodes for the case $X = 0 \bmod 4, Y = 0 \bmod 4$), and with the results obtained by Khasnabish [9] for the special case $X = Y$.

The mean internodal distance of the MS network is related to its saturation throughput for uniform traffic. Indeed, assume that packets arrive at each node with rate λ, and each of them has a single destination which is uniformly distributed over all nodes (including its origin). Each packet requires on the average at least $\bar{D}$ transmissions to arrive at its destination. Applying Little's theorem on the system consisting of the $2XY$ network links (which can be viewed as servers), we obtain $\lambda XY\bar{D} \leqslant 2(XY)$, or else $\lambda \leqslant 2/\bar{D}$.

If the links of the MS network were bidirectional (in other words, if we had an $X \times Y$ wraparound mesh with X and Y being even numbers), the mean internodal distance would be $(X+Y)/4$. This is at most one unit less than the mean internodal distance given in Eq. (27). Since the wraparound mesh has twice the number of links that the

MS network has, it is clear that the MS network makes more efficient use of the links than the wraparound mesh does. Note, however, that the saturation throughput of the wraparound mesh is approximately twice that of the MS network (because it has twice the number of links, and about the same mean internodal distance).

4. Total exchange in a square Manhattan Street network

In this section we present an optimal algorithm to execute the total exchange task in a square MS network with $X = Y = N$. We remind the reader that in the TE task, each processor has to send a separate (personalized) message to every other processor of the network. Before describing the algorithm, we introduce some notation and establish two preliminary results.

For any routing tag $\mathscr{L} = l_1 l_2 \cdots l_L$ we denote by $\overline{\mathscr{L}} = \bar{l}_1 \bar{l}_2 \cdots \bar{l}_L$ the bitwise complement of $\mathscr{L}$, where $\bar{l}_i = 1 - l_i$. In other words, a path with routing tag $\overline{\mathscr{L}}$ uses a vertical (horizontal) link during a step, whenever the path with routing tag $\mathscr{L}$ uses a horizontal (vertical, respectively) link during the same step. The following two lemmas, which are easy to prove, will be useful in designing TE algorithms for the MS network.

Lemma 2. *The path $((0,0),\mathscr{L})$ has as destination node (i,j) if and only if the path $((0,0),\overline{\mathscr{L}})$ has as destination node (j,i).*

Lemma 3. *If s and t are distinct nodes, the paths $(s,\mathscr{L}),(s,\overline{\mathscr{L}}),(t,\mathscr{L})$, and $(t,\overline{\mathscr{L}})$ use different links during a given step.*

Let $\mathscr{L}^{(i,j)}$ be the routing tag of a shortest path with origin $(0,0)$ and destination (i,j). Such a shortest path can be found, e.g., in the way described in Section 2. Our TE algorithm consists of phases $P_{i,j}$, with $i = 0,1,\ldots,N-1$, $j = 0,1,\ldots,N-1$, and $i \leqslant j$, which can be executed in any order.

Total Exchange Algorithm:

During phase $P_{i,j}$, $i = 0,1,\ldots,N-1$, $j = 0,1,\ldots,N-1$, $i<j$, each node (x,y) of the network sends a personalized packet to the nodes that have relative address (i,j) and (j,i) with respect to (x,y). The packets originating at node (x,y) follow the paths $((x,y),\mathscr{L}^{(i,j)})$ and $((x,y),\overline{\mathscr{L}^{(i,j)}})$.

During phases $P_{i,i}$, $i = 0,1,\ldots,N-1$, each node (x,y) sends a packet to the node that has relative address (i,i) with respect to (x,y) in the following way. The packet originating at node (x,y) is split into two mini packets. The first mini-packet sent by node (x,y) follows the path $((x,y),\mathscr{L}^{(i,i)})$, while the second mini-packet follows the path $((x,y),\overline{\mathscr{L}^{(i,i)}})$.

Phase $P_{i,j}$, $i<j$, requires $D_{0,0}(i,j)$ steps to complete. If the overhead associated with the splitting of packets is small, we can assume that each mini packet involved in phase $P_{i,i}$ requires $\frac{1}{2}$ units of time for transmission over a link. Therefore, phase $P_{i,i}$ requires $D_{0,0}(i,i)/2$ steps to complete. The total duration T_{TE} of the total exchange algorithm

is given by

$$T_{TE} = \sum_{i,j,\ i<j} D(i,j) + \frac{1}{2}\sum_{i} D(i,i) = \frac{1}{2}\sum_{i,j} D(i,j) = \frac{H}{2}.$$

Using the expressions obtained in Section 3 for H for the case $X = Y = N$, we obtain

$$T_{TE} = \begin{cases} \frac{N^3}{4} + \frac{N^2}{2} - 2 & \text{if } N = 0 \bmod 4, \\ \frac{N^3}{4} + \frac{N^2}{2} - N - 1 & \text{if } N = 2 \bmod 4, \quad N > 2. \end{cases} \tag{28}$$

It can be seen that during the execution of the TE algorithm all links of the network are used 100% of the time. This combined with the fact that all packets follow shortest paths to their destinations, proves that our algorithm executes the TE task in optimal time. Since the algorithm achieves full utilization of the links of the network, which are a critical resource, it cannot be improved by using a switching format different than the store-and-forward switching assumed in this paper (e.g., by using wormhole routing [4]).

5. Multinode broadcast in a square Manhattan Street network

In the multinode broadcast task, each processor of the $N \times N$ square MS network broadcasts a packet to all other nodes of the network. Since each node has to receive $N^2 - 1$ packets and has only two incoming links, a lower bound on the time required to execute the multinode broadcast is

$$T_{\text{MNB}} \geqslant \left\lceil \frac{N^2 - 1}{2} \right\rceil = \frac{N^2}{2}, \tag{29}$$

where we used the fact that N is even. In what follows we present an algorithm to execute the MNB in time equal to the lower bound of Eq. (29). We will construct the MNB algorithm by starting with an algorithm for broadcasting from a single node (namely, from node $(0,0)$), and exploiting the symmetry of the network to obtain a MNB algorithm.

We can represent an algorithm for broadcasting a packet from node $(0,0)$ to all other nodes in m steps by a sequence of disjoint sets of directed links, $A_1, A_2, \ldots, A_m$. Each A_i is the set of links on which the packet is transmitted during the ith step. In order for the sets A_i to correspond to a broadcast from node $(0,0)$, they must satisfy some consistency requirements. In particular, if S_i (or E_i) is the set of start nodes (or end nodes, respectively) of A_i, we must have $S_1 = \{(0,0)\}$ and $S_i \subset \{(0,0)\} \cup (\bigcup_{k=1}^{i-1} E_k)$. Any node of the network must belong to $\{(0,0)\} \cup (\bigcup_{k=1}^{m} E_k)$. Finally, the set of all nodes together with the set of links $\bigcup_{k=1}^{m} A_k$ must form a spanning tree of the MS network. In the algorithm that we propose each set A_i contains only two links, one vertical and one horizontal. We will show that for sets A_i that have this property, the algorithm for broadcasting from node $(0,0)$ can be extended to a MNB algorithm.

It can be seen that an algorithm for broadcasting a packet from node (x, y) is specified by the sets

$$A_i(x,y) = \{(R_{x,y}(u,v), R_{x,y}(z,w)) \mid ((u,v),(z,w)) \in A_i\}, \quad i = 1,2,\ldots,m,$$

where $A_i(x,y)$ denotes the set of the links on which the packet is transmitted during the ith step. The following lemma shows that each set $A_i(x,y)$ contains one horizontal and one vertical link.

Lemma 4. *If $((u,v),(z,w))$ is a horizontal (vertical) link then $(R_{x,y}(u,v), R_{x,y}(z,w))$ is also a horizontal (vertical, respectively) link of the* MS *network.*

Proof. Let $((u,v),(z,w))$ be a horizontal link (the case where it is a vertical link is analogous). Then, we have $z = u + p(u) \bmod N$ and $w = v$ (we remind the reader that $p(u) = 1$ if u is even, and $p(u) = -1$ if u is odd). We also have

$$R_{x,y}(u,v) = ((u-x)p(x) \bmod N, (v-y)p(y) \bmod N),$$

and

$$\begin{aligned} R_{x,y}(z,w) &= R_{x,y}(u + p(u) \bmod N, v) \\ &= ((u-x)p(x) + p(u)p(x) \bmod N, (v-y)p(y) \bmod N) \\ &= ((u-x)p(x) + p((u-x)p(x)) \bmod N, (v-y)p(y) \bmod N). \end{aligned}$$

The preceding two equations show that $(R_{x,y}(u,v), R_{x,y}(z,w))$ is a horizontal link. □

Let (u,v) be the starting node of the horizontal link of set $A_i(0,0)$. Then, by construction, the starting node of the horizontal link of $A_i(x,y)$ is $R_{x,y}(u,v)$. Since (by Lemma 1) the function $(x,y) \to R_{x,y}(u,v)$ is one-to-one, the horizontal links in the sets $\{A_i(x,y)\}_{0 \leqslant x,y \leqslant N-1}$ have different starting nodes, and they are, therefore, distinct. Similarly, it can be proved that the vertical links in the sets $\{A_i(x,y)\}_{0 \leqslant x,y \leqslant N-1}$ are also distinct. Therefore, for a given i, the sets $\{A_i(x,y)\}_{0 \leqslant x,y \leqslant N-1}$ are disjoint.

For each node (x,y), the set $A_i(x,y)$ specifies the links on which the packet generated at this node will be transmitted during the ith step. If at a given step i, transmissions take place simultaneously on all links of the sets $\{A_i(x,y)\}_{0 \leqslant x,y \leqslant N-1}$, no packet collisions arise, and the total time taken by the multinode broadcast algorithm is equal to m.

To complete the description of the MNB algorithm, it remains to construct the sets A_i, $i = 1,2,\ldots,m$. The requirements for the sets A_i are that they should correspond to a single-node broadcast from node $(0,0)$, and each of them should have one horizontal and one vertical link. The number m of sets should be the smallest possible so that the execution time of the multinode broadcast is minimum. Since the cardinality of

$\bigcup_{i=1}^{m} A_i$ is at least $N^2 - 1$, and each set A_i contains at most two links, m has to be at least as large as $\lceil (N^2 - 1)/2 \rceil = N^2/2$, where we have used the fact that N is even. One way to construct sets A_i with the required properties for $m = N^2/2$ is described next.

Construction of the sets A_i, $i = 1, 2, \ldots, m$, **for** $m = N^2/2$.

(a) The first N sets are defined in the following way:

$$A_i = \{(i,0),(0,i)\}, \quad i = 1, 2, \ldots, N.$$

(b) We denote by E_k the ending nodes of the links of the set A_k. Suppose that we have constructed the sets $A_1, A_2, \ldots, A_{k-1}$, for some k with $N+1 \leqslant k \leqslant N^2$. The set A_k is then chosen to consist of two links, one horizontal and one vertical, whose starting nodes are in the set $\bigcup_{i=1}^{k=1} E_i$, and whose ending nodes are distinct and do not belong to the set $\bigcup_{i=1}^{k-1} E_i$ of the already covered nodes. It can be seen that such links can always be found, except for the last step. In the last step only one node remains uncovered, and therefore both the horizontal and the vertical link are chosen to lead to the same node (alternatively, the last set A_m could be chosen to consist of only one link).

Using the sets A_i constructed as described previously, we can execute the multinode broadcast in time

$$T_{\mathrm{MNB}} = \frac{N^2}{2}.$$

Comparing this equation with the lower bound of Eq. (29), we conclude that the MNB algorithm has optimal execution time. It can also be seen that all links of the network are fully utilized during the execution of the MNB algorithm, except for the last step during which only half of the links are used.

6. Hamiltonian cycles, and MNB with packet segmentation

In the MNB algorithm described in Section 5, messages were not split, and they were always transmitted as a single packet. In this section we will present a particularly simple MNB algorithm, which, however, assumes that packets can be split into two smaller packets that are routed independently and are recombined at the destination. We assume that the overhead introduced by the splitting and the recombining of packets is negligible (alternatively, we can assume that each node has to broadcast two packets to all the other nodes, so that the task we have to perform effectively corresponds to two multinode broadcasts).

A *Hamiltonian cycle* is a cycle that visits each node of the network exactly once. Since the MS network has $2N^2$ links, it cannot contain more than two edge-disjoint Hamiltonian cycles. Fig. 5 illustrates a way to construct two edge-disjoint Hamiltonian cycles in the MS network.

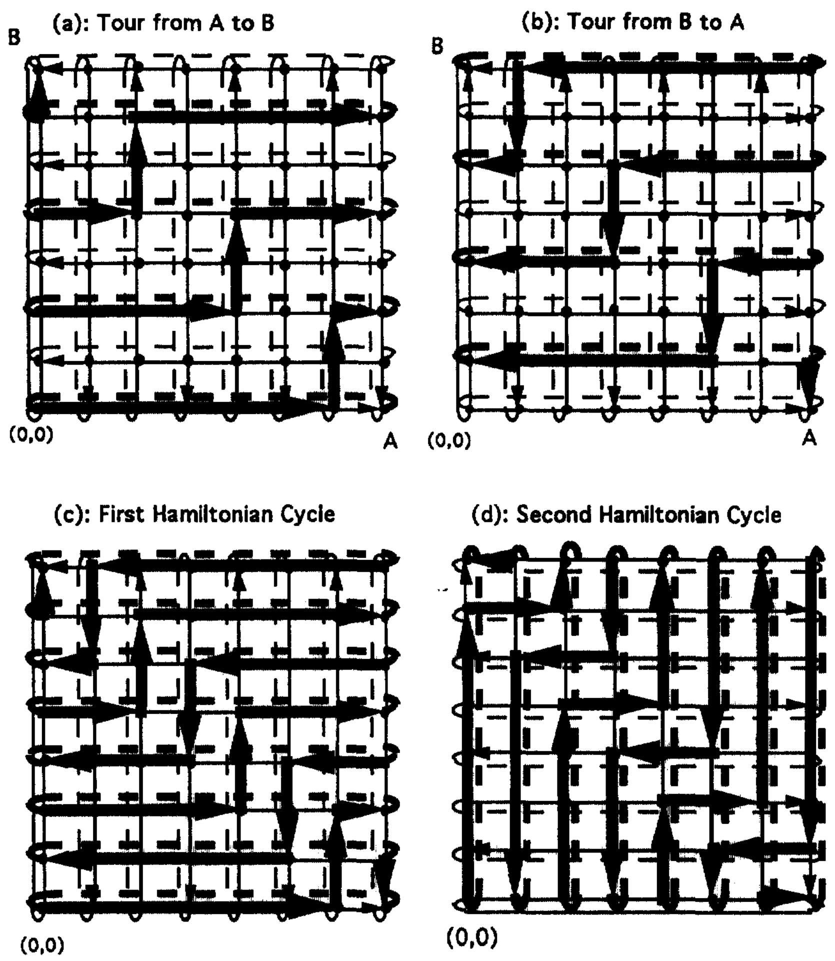

Fig. 5. The MS network contains two edge-disjoint Hamiltonian cycles. The first Hamiltonian cycle (Fig. 5(c)) is obtained by combining two tours: the one of these tours (Fig. 5(a)) uses only even rows and columns, while the other uses only odd rows and columns (Fig. 5(b)). The second Hamiltonian cycle (Fig. 5(d)) is the reflection of the first one with respect to the diagonal.

The MNB algorithm that we propose uses edge-disjoint Hamiltonian cycles. Each packet is split at the origin into two mini packets, each of which is broadcast on a different Hamiltonian cycle. The two mini packets are recombined when they reach their destination. In this way, we essentially perform two independent multinode broadcasts on two edge-disjoint unidirectional rings. The MNB is a unidirectional ring can be executed in N^2-1 steps (during a step each node on the ring simply passes to the right the packet that it received from the left during the previous step). Ignoring the overhead due to the splitting of packets, each mini packet requires 0.5 units of time to be transmitted over a link. Therefore, the time complexity of the resulting MNB algorithm for the MS network is

$$T_{\mathrm{MNB}} = \frac{N^2-1}{2}, \quad \text{(when the splitting of packets is allowed).}$$

Eq. (30) does not contradict the lower bound of Eq. (29), because the latter is not valid when the segmentation of packets is allowed (in that case the ceilings in Eq. (29) must be removed).

It is easy for each node to record which is its next node on a Hamiltonian cycle. As a result, the MNB algorithm of this section is considerably simpler to implement than the MNB algorithm of Section 5, and it may be the best choice when the overhead due to the segmentation of packets is small.

7. Conclusions

We presented simple shortest path routing algorithms for the Manhattan Street network of processors, and we derived a closed-form expression that gives the mean internodal distance of this topology in the general case. We also presented the first optimal completion time algorithms to execute the total exchange and the multinode broadcast communication tasks in the Manhattan Street network. The proposed algorithms achieve almost full utilization of the communication links, while being at the same time easy to implement.

Appendix

In this appendix we describe in detail the calculations involved in obtaining some of the equations used in Section 2.

Proof of Eq. (A.1).

$$\begin{aligned}
&\sum_{x \text{ odd}} \sum_{y \text{ odd}} D_{0,0}(x,y) \\
&\quad= \sum_{x \text{ odd},\, y \text{ odd},\, (x,y)\in A_1} [x+y+2] + \sum_{x \text{ odd},\, y \text{ odd},\, (x,y)\in A_2} [X-x+Y-y+2] \\
&\qquad+ \sum_{x \text{ odd},\, y \text{ odd},\, (x,y)\in A_3} [x+Y-y] + \sum_{x \text{ odd},\, y \text{ odd},\, (x,y)\in A_4} [X-x+y] \\
&\quad= \sum_{i=1}^{k}\sum_{j=1}^{m} [(2i-1)+(2j-1)+2] + \sum_{i=k+1}^{2k}\sum_{j=m+1}^{2m} [4k-(2i-1)+4m-(2j-1)+2] \\
&\qquad+ \sum_{i=1}^{k}\sum_{j=m+1}^{2m} [(2i-1)+4m-(2j-1)] + \sum_{i=k+1}^{2k}\sum_{j=1}^{m} [4k-(2i-1)+(2j-1)]
\end{aligned}$$

$$= 4\sum_{i=1}^{k}\sum_{j=1}^{m} 1 + 4\sum_{i=1}^{k}\sum_{j=1}^{m}[(2i-1)+(2j-1)] = 4km(1+m+k). \tag{A.1}$$

Proof of Eq. (A.2).

$$\begin{aligned}
&\sum_{x\text{ even}}\sum_{y\text{ even}} D_{0,0}(x,y)\\
&= \sum_{x\text{ even},\, y\text{ even},\, (x,y)\in A_1} [x+y] + \sum_{x\text{ even},\, y\text{ even},\, (x,y)\in A_2} [X-x+Y-y+4]\\
&\quad + 4\sum_{x\text{ even},\, (x,0)\in A_4} 1 + 4\sum_{y\text{ even},\, (0,y)\in A_3} 1\\
&\quad + \sum_{x\text{ even},\, y\text{ even},\, (x,y)\in A_3} [x+Y-y] + \sum_{x\text{ even},\, y\text{ even},\, (x,y)\in A_4} [X-x+y]\\
&= \sum_{i=0}^{k}\sum_{j=0}^{m}[2i+2j] + \sum_{i=k+1}^{2k-1}\sum_{j=m+1}^{2m-1}[4k-2i+4m-2j+4]\\
&\quad + \sum_{i=0}^{k}\sum_{j=m+1}^{2m-1}[2i+4m-2j] + \sum_{i=k+1}^{2k-1}\sum_{j=0}^{m}[4k-2i-2j] + 4(k-1) + 4(m-1)\\
&= 4(m-1)(k-1) + (k+1)m(m+1) + (m+1)k(k+1) + (k-1)m(m-1)\\
&\quad + (m-1)k(k-1) + (k+1)m(m-1) + (m-1)k(k+1) + (m+1)k(k-1)\\
&\quad + (k-1)m(m+1) + 4(k-1) + 4(m-1)\\
&= 4km^2 + 4k^2m + 4(m-1)(k-1) + 4(k-1) + 4(m-1) = 4mk(m+k+1) - 4.
\end{aligned} \tag{A.2}$$

Proof of Eq. (A.3).

$$\begin{aligned}
&\sum_{x\text{ odd}}\sum_{y\text{ even}} D_{0,0}(x,y)\\
&= \sum_{x\text{ odd},\, y\text{ even},\, (x,y)\in A_1} [x+y] + \sum_{x\text{ odd},\, y\text{ even},\, (x,y)\in A_2} [X-x+Y-y+2]\\
&\quad + \sum_{x\text{ odd},\, y\text{ even},\, (x,y)\in A_3} [x+Y-y] + \sum_{x\text{ odd},\, y\text{ even},\, (x,y)\in A_4} [X-x+y+2]\\
&= \sum_{i=1}^{k+1}\sum_{j=0}^{m}[(2i-1)+2j] + \sum_{i=k+2}^{2k}\sum_{j=m+1}^{2m-1}[4k-(2i-1)+4m-2j+2]\\
&\quad + \sum_{i=1}^{k+1}\sum_{j=m+1}^{2m-1}[(2i-1)+4m-2j] + \sum_{i=k+2}^{2k}\sum_{j=0}^{m}[4k-(2i-1)+2j+2]\\
&= -(k+1)(m+1) - (k+1)(m-1) + 3(m-1)(k-1) + 3(m+1)(k-1)
\end{aligned}$$

$$
\begin{aligned}
&+(k+1)m(m+1)+(m+1)(k+1)(k+2)\\
&+(k-1)m(m-1)+(m-1)(k-1)(k-2)\\
&+(k+1)m(m-1)+(m-1)(k+1)(k+2)\\
&+(k-1)m(m+1)+(m+1)(k-1)(k-2)\\
&=4mk(m+k+1). \qquad\qquad (A.3)
\end{aligned}
$$

Proof of Eq. (A.4).

$$
\begin{aligned}
&\sum_{x\text{ even}}\sum_{y\text{ odd}} D_{0,0}(x,y)\\
&= \sum_{x\text{ even},\, y\text{ odd},\,(x,y)\in A_1} [x+y] + \sum_{x\text{ even},\, y\text{ odd},\,(x,y)\in A_2} [X-x+Y-y+2]\\
&\quad+ \sum_{x\text{ even},\, y\text{ odd},\,(x,y)\in A_3} [x+Y-y+2] + \sum_{x\text{ even},\, y\text{ odd},\,(x,y)\in A_4} [X-x+y]\\
&= \sum_{i=0}^{k}\sum_{j=1}^{m+1}[2i+(2j-1)] + \sum_{i=k+1}^{2k-1}\sum_{j=m+2}^{2m}[4k-2i+4m-(2j-1)+2]\\
&\quad+ \sum_{i=0}^{k}\sum_{j=m+2}^{2m}[2i+4m-(2j-1)+2] + \sum_{i=k+1}^{2k-1}\sum_{j=1}^{m+1}[4k-2i+(2j-1)]\\
&= 4mk(m+k+1). \qquad\qquad (A.4)
\end{aligned}
$$

Proof of Eq. (A.5).

$$
\begin{aligned}
&\sum_{x\text{ odd}}\sum_{y\text{ odd}} D_{0,0}(x,y)\\
&= \sum_{x\text{ odd},\, y\text{ odd},\,(x,y)\in A_1} [x+y+2] + \sum_{x\text{ odd},\, y\text{ odd},\,(x,y)\in A_2} [X-x+Y-y+2]\\
&\quad+ \sum_{x\text{ odd},\, y\text{ odd},\,(x,y)\in A_3} [x+Y-y]\\
&\quad+ \sum_{x\text{ odd},\, y\text{ odd},\,(x,y)\in A_4} [X-x+y] - D(2k+1,2m+1)\\
&= \sum_{i=1}^{k}\sum_{j=1}^{m}[2i+2j] + \sum_{i=k+2}^{2k+1}\sum_{j=m+2}^{2m+1}[4k+2-(2i-1)+4m+2-(2j-1)+2]\\
&\quad+ \sum_{i=1}^{k+1}\sum_{j=m+1}^{2m+1}[2i+4m+2-2j)]\\
&\quad+ \sum_{i=k+1}^{2k+1}\sum_{j=1}^{m+1}[4k-2i+2j+4] - D(2k+1,2m+1)
\end{aligned}
$$

$$
\begin{aligned}
&= km(m+1) + mk(k+1) + mk^2 + km^2 + 2km + (k+1)(m+1)^2 \\
&\quad + (m+1)(k+1)^2 + (m+1)(k+1)^2 + (k+1)(m+1)^2 - 2k - 2m - 2 \\
&= 2mk(m+k+2) + 2(k+1)(m+1)(m+k+2) - 2k - 2m - 2. \qquad \text{(A.5)}
\end{aligned}
$$

Proof of Eq. (A.6).

$$
\begin{aligned}
&\sum_{x \text{ even}} \sum_{y \text{ even}} D_{0,0}(x,y) \\
&= \sum_{x \text{ even},\, y \text{ even},\, (x,y)\in A_1} [x+y] + \sum_{x \text{ even},\, y \text{ even},\, (x,y)\in A_2} [X - x + Y - y + 4] \\
&\quad - D(2k+2, 2m+2) + 4 \sum_{x \text{ even},\ (x,0)\in A_4} 1 + 4 \sum_{y \text{ even},\ (0,y)\in A_3} 1 - 4 \\
&\quad + \sum_{x \text{ even},\, y \text{ even},\, (x,y)\in A_3} [x + Y - y] + \sum_{x \text{ even},\, y \text{ even},\, (x,y)\in A_4} [X - x + y] \\
&= \sum_{i=0}^{k} \sum_{j=0}^{m} [2i + 2j] + 4k + 4m \\
&\quad - 4 \sum_{i=k+2}^{2k} \sum_{j=m+2}^{2m} [4k - 2i + 4m - 2j + 8] + 4k + 4m - 4 \\
&\quad -2k - 2m - 2 + \sum_{i=0}^{k+1} \sum_{j=m+1}^{2m} [2i + 4m + 2 - 2j] \\
&\quad + \sum_{i=k+1}^{2k} \sum_{j=0}^{m+1} [4k + 2 - 2i + 2j] \\
&= (k+1)(m+1)(m+k) + 4(k-1)(m-1) + (k-1)(m-1)(m+k) \\
&\quad + (k+2)m(k+m+2) + (m+2)k(m+k+2) + 2k + 2m - 6, \quad \text{for } k,\ m > 1. \qquad \text{(A.6)}
\end{aligned}
$$

Proof of Eq. (A.7).

$$
\begin{aligned}
&\sum_{x \text{ odd}} \sum_{y \text{ even}} D_{0,0}(x,y) \\
&= \sum_{x \text{ odd},\, y \text{ even},\, (x,y)\in A_1} [x+y] + \sum_{x \text{ odd},\, y \text{ even},\, (x,y)\in A_2} [X - x + Y - y + 2] \\
&\quad + \sum_{x \text{ odd},\, y \text{ even},\, (x,y)\in A_3} [x + Y - y] + \sum_{x \text{ odd},\, y \text{ even},\, (x,y)\in A_4} [X - x + y + 2] \\
&= \sum_{i=1}^{k+1} \sum_{j=0}^{m} [(2i-1) + 2j]
\end{aligned}
$$

$$+\sum_{i=k+2}^{2k+1}\sum_{j=m+1}^{2m}[4k+2-(2i-1)+4m+2-2j+2]$$

$$+\sum_{i=1}^{k+1}\sum_{j=m+1}^{2m}[(2i-1)+4m+2-2j]$$

$$+\sum_{i=k+2}^{2k+1}\sum_{j=0}^{m}[4k+2-(2i-1)+2j+2]$$

$$=(k+1)m(m+1)+(m+1)(k+1)^2+2km+k(m+1)m+mk^2$$
$$+(k+1)m(m+1)+m(k+1)^2+k(m+1)(m+2)+(m+1)k^2. \qquad \text{(A.7)}$$

Proof of Eq. (A.8).

$$\sum_{x \text{ even}}\sum_{y \text{ odd}} D_{0,0}(x,y)$$
$$=\sum_{x \text{ even},\, y \text{ odd},\, (x,y)\in A_1}[x+y]+\sum_{x \text{ even},\, y \text{ odd},\, (x,y)\in A_2}[X-x+Y-y+2]$$
$$+\sum_{x \text{ even},\, y \text{ odd},\, (x,y)\in A_3}[x+Y-y+2]+\sum_{x \text{ even},\, y \text{ odd},\, (x,y)\in A_4}[X-x+y]$$
$$=\sum_{j=1}^{m+1}\sum_{i=0}^{k}[(2j-1)+2i]$$
$$+\sum_{j=m+2}^{2m+1}\sum_{i=k+1}^{2k}[4m+2-(2j-1)+4k+2-2i+2]$$
$$+\sum_{j=1}^{m+1}\sum_{i=k+1}^{2k}[(2j-1)+4k+2-2i]$$
$$+\sum_{j=m+2}^{2m+1}\sum_{i=0}^{k}[4m+2-(2j-1)+2i+2]$$
$$=(m+1)k(k+1)+(k+1)(m+1)^2+2km+m(k+1)k+km^2$$
$$+(m+1)k(k+1)+k(m+1)^2+m(k+1)(k+2)+(k+1)m^2. \qquad \text{(A.8)}$$

Proof of Eq. (A.9).

$$\sum_{x \text{ odd}}\sum_{y \text{ odd}} D_{0,0}(x,y)$$
$$=\sum_{x \text{ odd},\, y \text{ odd},\, (x,y)\in A_1}[x+y+2]+\sum_{x \text{ odd},\, y \text{ odd},\, (x,y)\in A_2}[X-x+Y-y+2]$$

$$+ \sum_{x \text{ odd}, y \text{ odd}, (x,y)\in A_3} [x + Y - y] + \sum_{x \text{ odd}, y \text{ odd}, (x,y)\in A_4} [X - x + y]$$

$$= \sum_{i=1}^{k} \sum_{j=1}^{m} [(2i-1) + (2j-1) + 2]$$

$$+ \sum_{i=k+2}^{2k+1} \sum_{j=m+1}^{2m} [4k + 2 - (2i-1) + 4m - (2j-1) + 2]$$

$$+ \sum_{i=1}^{k+1} \sum_{j=m+1}^{2m} [(2i-1) + 4m - (2j-1)]$$

$$+ \sum_{i=k+1}^{2k+1} \sum_{j=1}^{m} [4k + 2 - (2i-1) + (2j-1)]$$

$$= 2mk(m + k + 2) + 2(k+1)m(m + k + 1). \qquad \text{(A.9)}$$

Proof of Eq. (A.10).

$$\sum_{x \text{ even}} \sum_{y \text{ even}} D(x,y)$$

$$= \sum_{x \text{ even}, y \text{ even}, (x,y)\in A_1} [x + y] + \sum_{x \text{ even}, y \text{ even}, (x,y)\in A_2} [X - x + Y - y + 4]$$

$$+ \sum_{x \text{ even}, y \text{ even}, (x,y)\in A_3} [x + Y - y] + \sum_{x \text{ even}, y \text{ even}, (x,y)\in A_4} [X - x + y]$$

$$+ 4 \sum_{x \text{ even}, (x,0)\in A_4} 1 + 4 \sum_{y \text{ even}, (0,y)\in A_3} 1 - 2$$

$$= \sum_{i=0}^{k} \sum_{j=0}^{m} [2i + 2j] + \sum_{i=k+2}^{2k} \sum_{j=m+1}^{2m-1} [4k + 2 - 2i + 4m - 2j + 4]$$

$$+ \sum_{i=0}^{k+1} \sum_{j=m+1}^{2m-1} [(2i + 4m - 2j]$$

$$+ \sum_{i=k+1}^{2k} \sum_{j=0}^{m} [4k + 2 - 2i + 2j] + 4k + 4(m-1) - 2$$

$$= 2(m+k)(mk+1) + 2(m + k + 1)(mk + m - 1) + 4mk - 2. \qquad \text{(A.10)}$$

Proof of Eq. (A.11).

$$\sum_{x \text{ odd}} \sum_{y \text{ even}} D_{0,0}(x,y)$$

$$= \sum_{x \text{ odd}, y \text{ even}, (x,y)\in A_1} [x + y] + \sum_{x \text{ odd}, y \text{ even}, (x,y)\in A_2} [X - x + Y - y + 2]$$

$$+ \sum_{x \text{ odd}, y \text{ even}, (x,y)\in A_3} [x+Y-y] + \sum_{x \text{ odd}, y \text{ even}, (x,y)\in A_4} [X-x+y+2]$$

$$= \sum_{i=1}^{k+1}\sum_{j=0}^{m}[(2i-1)+2j]$$

$$+ \sum_{i=k+2}^{2k+1}\sum_{j=m+1}^{2m-1}[4k+2-(2i-1)+4m-2j+2]$$

$$+ \sum_{i=1}^{k+1}\sum_{j=m+1}^{2m-1}[(2i-1)+4m-2j]$$

$$+ \sum_{i=k+2}^{2k+1}\sum_{j=0}^{m}[4k+2-(2i-1)+2j+2]$$

$$= 2km(m+k+2)+2m(k+1)(m+k+1). \qquad \text{(A.11)}$$

Proof of Eq. (A.12).

$$\sum_{x \text{ even}}\sum_{y \text{ odd}} D_{0,0}(x,y)$$

$$= \sum_{x \text{ even}, y \text{ odd}, (x,y)\in A_1} [x+y] + \sum_{x \text{ even}, y \text{ odd}, (x,y)\in A_2} [X-x+Y-y+2]$$

$$+ \sum_{x \text{ even}, y \text{ odd}, (x,y)\in A_3} [x+Y-y+2] + \sum_{x \text{ even}, y \text{ odd}, (x,y)\in A_4} [X-x+y]$$

$$= \sum_{i=0}^{k}\sum_{j=1}^{m+1}[2i+(2j-1)]$$

$$+ \sum_{i=k+1}^{2k}\sum_{j=m+2}^{2m}[4k+2-2i+4m-(2j-1)+2]$$

$$+ \sum_{i=0}^{k}\sum_{j=m+2}^{2m}[2i+4m-(2j-1)+2] + \sum_{i=k+1}^{2k}\sum_{j=1}^{m+1}[4k+2-2i+(2j-1)]$$

$$= 2km(m+k+2)+2m(k+1)(m+k+1). \qquad \text{(A.12)}$$

References

[1] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[2] D.P. Bertsekas, C. Ozveren, G.D. Stamoulis, P. Tseng, J.N. Tsitsiklis, Optimal communication algorithms for hypercubes, J. Parallel Distrib. Comput. 11 (1991) 263-275.

[3] J.T. Brassil, Deflection routing in certain regular networks, Ph.D. Thesis, University of California Dan Diego, 1991.

[4] W.J. Dally, Network and processor architecture for message-driven computers, in: R. Suaya, G. Birtwistle (Eds.), VLSI and Parallel Computation, Morgan Kaufmann Publishers, San Mateo, CA, 1990, pp. 140–222.
[5] A. Edelman, Optimal matrix transposition and bit reversal on hypercubes: all-to-all personalised communication, J. Parallel Distrib. Comput. 11 (1991) 328–331.
[6] P. Fraigniaud, Complexity analysis of broadcasting in hypercubes with restricted communication capabilities, J. Parallel Distrib. Comput. 16 (1992) 15–26.
[7] A.G. Greenberg, J. Goodman, Sharp approximate models of deflection routing in mesh networks, IEEE Trans. Commun. COM-41(1) (1993) 210–223.
[8] S.L. Johnsson, C.T. Ho, Optimum broadcasting and personalized communication in hypercubes, IEEE Trans. Comput. C-38 (1989) 1249–1268.
[9] B. Khasnabish, Topological properties of Manhattan Street networks, Electron. Lett. 25 (1989) 1388–1389.
[10] N.F. Maxemchuk, Routing in the Manhattan Street network, IEEE Trnas. Commun. COM-35(5) (1987) 503–512.
[11] N.F. Maxemchuk, Comparison of deflection and store-and-forward techniques in the Manhattan Street and shuffle-exchange networks, INFOCOM'89, vol. 3, April 1989, pp. 800–809.
[12] C. Rose, Mean internodal distance in regular and random multihop networks, IEEE Trans. Commun. COM-40(5) (1992) 1310–1318.
[13] Y. Saad, M.H. Schultz, Data communication in hypercubes, J. Parallel Distrib. Comput. (1989) 115–135.
[14] Y. Saad, M.H. Schultz, Data communication in parallel architectures, Parallel Comput. 11 (1989) 131–150.
[15] E.A. Varvarigos, D.P. Bertsekas, Communication algorithms for isotropic tasks in hypercubes and wraparound meshes, Parallel Comput. 18 (1992) 1233–1257.
[16] E.A. Varvarigos, D.P. Bertsekas, Multinode broadcast in hypercubes and rings with randomly distributed length of packets, IEEE Trans. Parallel Distrib. Systems 4(2) (1993).