HKU CSIS Tech Report TR-97-03

# The Wall Mesh

**Guihai Chen and Francis C. M. Lau***
Department of Computer Science
The University of Hong Kong

January 1997

**Abstract**

Wall mesh is a degree-3 mesh obtained from the ordinary degree-4 mesh by deleting one degree from each node. Its node degree is reduced by 25% while its diameter is almost the same as the degree-4 mesh. It is computationally equivalent to the degree-4 mesh since a degree-4 mesh can be embedded in a wall mesh with dilation 3 and congestion 3. As a result, all existing algorithms for degree-4 meshes can be effortlessly ported to the wall mesh with only constant slowdown. A wall torus can be obtained from a wall mesh, which has the property of node symmetry. We derive the fundamental distance measures for the wall mesh and torus, and compare them with the ordinary mesh and torus.

**Index Terms**—interconnection networks, mesh-connected networks, routing, broadcasting.

## 1 Introduction

Interconnection networks is an important topic in parallel computing. Many networks have been proposed in the literature [12, 9, 7], some of which have been adopted in commercial machines, such as the mesh and the hypercube. In comparison with the hypercube, the mesh seems to have a practical advantage because of its constant degree and perfectly compact layout. In this paper, we propose a new variant of the mesh, the *wall mesh*. The wall mesh has one degree less than that of the ordinary mesh but has almost the same performance as the ordinary mesh in terms of routing distances and its ability to support various computations.

In the following, WM, WT, OM, and OT stand for wall mesh, wall torus, ordinary mesh, and ordinary torus, respectively.

Section 2 defines the WM and the WT, and presents their basic properties. From these properties, we can see that the computational power of the WM/WT is not so much affected by the deletion of a degree. Sections 3 and 4 discuss the topological properties of

*Correspondence: F. C. M. Lau, Department of Computer Science, The University of Hong Kong, Hong Kong / Email: fcmlau@cs.hku.hk / Fax: (+852) 2559 8447.

and communication algorithms for the WM, respectively, followed by Section 5 on the WT. Section 6 summarizes the comparison between WM/WT and OM/OT. Section 7 concludes the paper and proposes some further research problems.

## 2 Preliminaries

**Definition 1** *The WM$(m,n)$ is an $m \times n$ grid, where a node is represented as $(y,x)$, for $0 \leq y \leq m-1$ and $0 \leq x \leq n-1$. $(y,x)$ is an even node if $y+x$ is even; an odd node otherwise. The edges of WM$(m,n)$ are as follows.*

- $\langle (y,x),(y,x+1) \rangle$ *for* $x < n-1$*;*
- $\langle (y,x),(y+1,x) \rangle$ *for even* $(y,x)$ *and* $y < m-1$.

Fig 1(a) shows a WM(6,12). Clearly, WM$(m,n)$ is equal to OM$(m,n)$ with the following edges being cut away.

- $\langle (y,x),(y+1,x) \rangle$ for odd $(y,x)$ and $y < m-1$.

We refer to these edges as the *cut-away edges*. Cut-away edges are imaginary edges as they do not exist in the WM or WT. The number of cut-away edges in a WM$(m,n)$ is equal to $(m-1) \times n/2$, which is approximately 1/4 of the edges in the corresponding OM$(m,n)$. Note that WM is defined for any $m,n > 0$.[1]

Similarly, we can define wall torus, WT. An example is shown in Fig. 1(b), which is a WT(6,12). WT, however, is defined only for even number of rows, $m$.

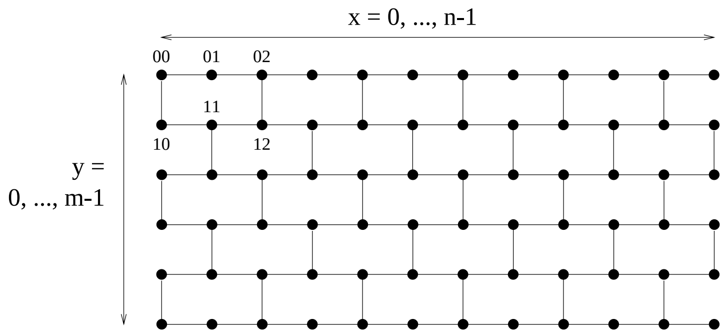
Since a cell (a brick) in a WM or WT has six nodes, except the several incomplete ones at the margins of a WM, it can be drawn as a hexagon, and the whole network would look like a honeycomb. Fig. 2 shows the result of redrawing the WT(6,12) as a honeycomb. The wall-like drawing, however, is preferred as it is in line with the fact that the WM/WT is but a variant of the OM/OT—the WM/W is the direct outcome of deleting one degree from the OM/OT.

**Theorem 1** *An OM can be embedded in a WM with dilation 3 and congestion 3, and in a WT with dilation 3 and congestion 2.*

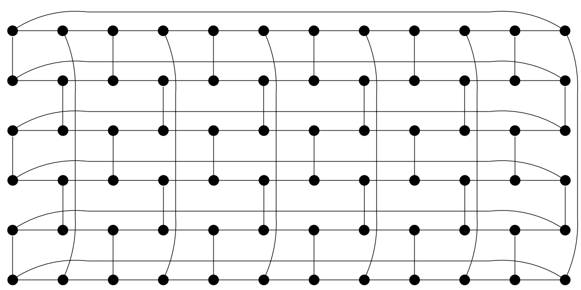
**Proof:** An embedding scheme is as shown in Fig. 3(a) and Fig. 3(b), for a WM and a WT, respectively. The dilated links of the OM, corresponding to the cut-away edges in the WT or the WM, are represented by dotted lines. It is easy to see that the dilation is 3 and the congestion is also 3 for embedding in the WM, and 3 and 2, respectively, for embedding in the WT. □

The dilation of 3 in both cases is the worst possible for any path in the OM. This is because there are no consecutive cut-away edges in a WM or WT. As a result, for any path

[1] The case of $n = 1$ is not meaningful when $m > 2$.




( a )



( b )

Figure 1: Examples of WM and WT: (a) WM(6,12) and (b) WT(6,12).

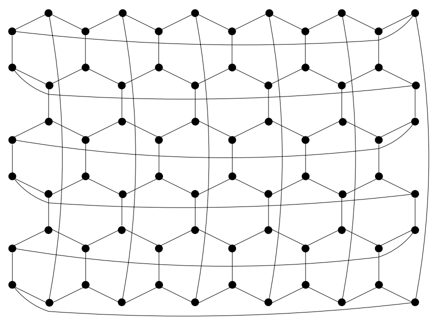
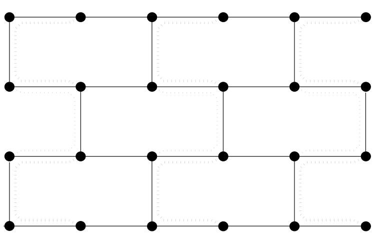
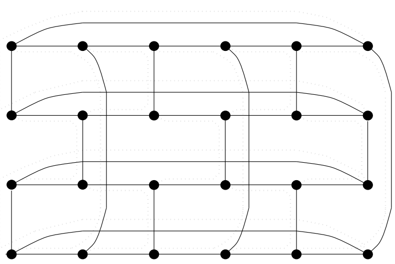


Figure 2: Honeycomb drawing of WT.



( a )



( b )

Figure 3: (a) Embedding of OM onto WM; (b) embedding of OT onto WT.

that is mapped to the WM or WT, at most half of the edges making up the path are edges that correspond to cut-away edges in the WM or WT, and hence the worst dilation (of 3) happens to a path consisting of a single edge which corresponds to a cut-away edge. On the other hand, we cannot do better than a congestion of 3 in the WM case. Consider the WM$(3,2)$, for instance; there are two cut-away edges, which are dilated to a total of six edges; these six edges have to be multiplexed on five real edges; hence a congestion of 3 in one of the five edges. The argument can be extended to any WM. The case of WT is similar.

Since the worst dilation of 3 happens only to paths that are of length 1 in the corresponding OM and only one out of four edges is a cut-away edge, it is expected that for a random path, the dilation would be much smaller than 3. In fact, as will be shown later on, the average distance in a WM (resp. WT) is only 20% (resp. 16.7%) longer than that of the corresponding OM (resp. OT).

**Corollary 1** *Any algorithms for an OM/OT can be implemented on the corresponding WM/WT with a constant slowdown.*

**Proof:** It follows from a theorem [6] by Leighton *et al.* that if a network $G$ can be embedded in another network $H$ with dilation $d$ and congestion $c$, $H$ can emulate $t$ steps of a computation running on $G$ in $O((d+c)t)$ steps.[2] □

Corollary 1 is the main motivation behind this study.

**Lemma 1** *The WT is node-symmetric.*

**Proof:** As shown in Fig. 1(b), all even nodes can be isomorphically mapped to the upper-left node $(0,0)$, and all odd nodes can be isomorphically mapped to the lower-left node $(m-1,0)$. Furthermore, node $(0,0)$ and node $(m-1,0)$ are symmetric if flipping over the WT vertically. □

## 3 Routing Paths in the WM

In this section, we give the optimal routing path for any pair of nodes, based on which we will then derive the diameter and average distance of the WM. Our aim is to show that the WM is not so much affected, in terms of these measures, by a smaller degree and fewer edges.

Define for any pair of source node $s = (y_s, x_s)$ and destination node $d = (y_d, x_d)$

$$\triangle y = |y_s - y_d|, \qquad \triangle x = |x_s - x_d|.$$

In view of the symmetric structure of a WM, there are only two cases of a routing path that need to be considered, which are as shown in Fig. 4. The rectangular region bounded

[2]The emulation algorithm [6] by Leighton *et al.* is not constructive. In practice, we can use the naive and constructive emulation algorithm with complexity $O((dc)t)$ which still yields a small constant slowdown for our problem.

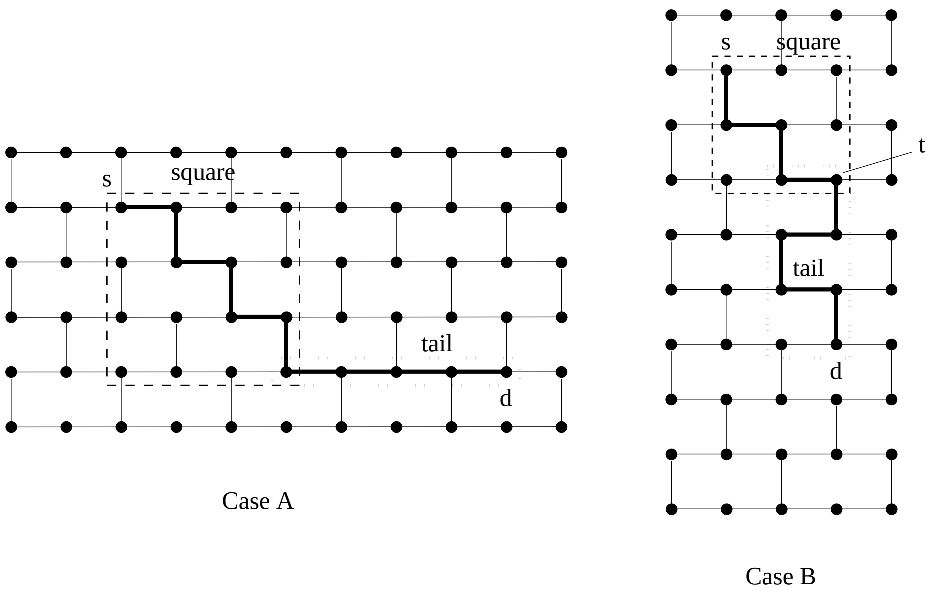



Figure 4: Two cases of optimal routing paths.

by $s$ and $d$ is called the *routing region*. We propose the following routing paths for the two cases. For case A, the routing path consists of a subpath through a square region, and then a horizontal straight-line path leading to $d$. For case B, the routing path consists of a subpath through a square region, and then a "zigzag" path along the vertical dimension to $d$. We refer to the horizontal straight-line path and the vertical zigzag path as the tail.

**Lemma 2** *The routing paths given above are optimal.*

**Proof:** The routing path for case A is clearly optimal. For case B, suppose that starting from $s$, the routing first reaches the right-hand-side edge of the rectangle at the node $t$ (see Fig. 4 for example). If the region bounded by $s$ and $t$ is an upright rectangle (*i.e.*, height > width) and if the route through this rectangle is optimal, then there must exist at least one pair of vertical edges that are consecutive; such a pair obviously does not exist in a WM. Therefore, the region is either a fat rectangle (*i.e.*, width > height) or a square. Since the remaining vertical distance (between $t$ and $d$) involves cut-away edges, the square is the optimal choice because it would leave behind a shorter remaining distance to route through. Thus, the routing path given above for case B is optimal. $\square$

**Lemma 3** *The length of the optimal path from any node $s$ to any other node $d$ is*

$$Distance(s,d) = \begin{cases} \triangle x + \triangle y & if\ \triangle x \geq \triangle y \\ 2\triangle y \pm 1 & if\ \triangle x < \triangle y \end{cases} .$$

**Proof:** The above routing path for case A has a length of $\triangle y + \triangle x$. For the routing path for case B, let $\triangle y = y_1 + y_2$, where $y_1$ is the part of $\triangle y$ from the top of the rectangle down to the point $t$, and $y_2$ the remaining distance. The subpath through the square has

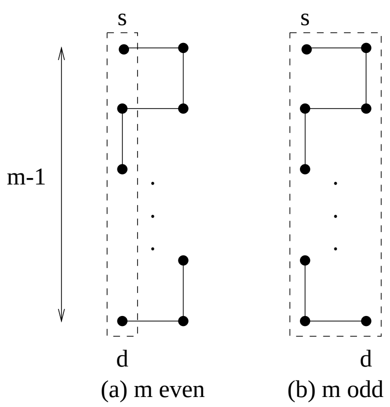



Figure 5: Two cases of a longest optimal path.

a length of $2y_1$. The remaining vertical distance includes $\lceil y_2/2 \rceil$ or $\lfloor y_2/2 \rfloor$ cut-away edges, and hence its path length is $2y_2 \pm 1$.[3] Therefore, the total length of the optimal path for case B is $2\triangle y \pm 1$. $\square$

Interestingly, when $\triangle x \geq \triangle y$, the optimal path length is the same as that of the OM; when $\triangle x < \triangle y$, the optimal path length is determined only by $\triangle y$, independent of $\triangle x$.

**Theorem 2** *The diameter of WM(m, n) is*

$$Diameter(WM(m,n)) = \begin{cases} m+n-2 & \textit{if } n > m \\ 2m-1 & \textit{if } n \leq m \end{cases} .$$

**Proof:** Consider the routing regions, of sizes $m \times 2$ and $m \times 1$ respectively, as shown in Fig. 5. Fig. 5(a) is for an even $m$, and (b) odd $m$. By Lemma 3, the length of the (optimal) path in either case is $2\triangle y + 1$ or $2m - 1$. For any WM ($n \geq 2$), such a path (of form (a) or (b)) that runs from the top row of the mesh down to the bottom row has a length of $2m - 1$—call this a *vertical-cut path.* For the case where $n > m$, the path proposed in Lemma 2 is of length $\triangle y + \triangle x$ or $m + n - 2$, by Lemma 3; hence, a vertical-cut path for this WM cannot be longer than this path since $m + n - 2 \geq 2m - 1$. For the other case, $n \leq m$, a vertical-cut path is the longest among all shortest paths. $\square$

From Theorem 2, WM$(m, n)$ has the same diameter as M$(m, n)$ if $n > m$, and has a diameter 1 unit longer than that of the M$(m, n)$ when $n = m$. This extra 1 unit should be immaterial for $n, m$ that are not trivially small. In fact, square WMs are the most likely choice for real implementation. We conclude at this point that as far as the diameter is concerned, the (square) WM and the OM are equal.

The average distance of the WM, on the other hand, is expected to be somewhat worse than that of the OM, even for the square case. The reason is that there could be a fair number of upright rectangular routing regions in a WM even when $n \geq m$.

**Theorem 3** *The average distance of the square wall mesh WM(n, n) is* $\frac{12n^2+2}{15n}$*, or approximately* $\frac{4}{5}n$*.*

[3] Here, $\pm 1$ means $-1$, $0$, or $+1$.

**Proof:** According to Lemma 3, the length of a path in the WM is longer than that in the OM only when $\triangle x < \triangle y$. In this case, the routing in the WM needs to travel through $(\triangle y - \triangle x) \pm 1$ more hops than in the mesh. We can ignore the $\pm 1$ because on average the path length for the tail segment in question is equal to $2\triangle y$.[4] Hence, the average "extra" distance traveled in the WM for a path is

$$\begin{aligned}
& P(\triangle x < \triangle y) \times (\triangle y - \triangle x) \\
= & \sum_{i=1}^{n-1} P(\triangle y = i) \sum_{j=0}^{i-1} P(\triangle x = j)(i-j) \\
= & \sum_{i=1}^{n-1} \frac{2(n-i)}{n^2} \left( \sum_{j=1}^{i-1} \frac{2(n-j)}{n^2}(i-j) + P(\triangle c = 0)i \right) \\
= & \frac{4}{n^4} \sum_{i=1}^{n-1} (n-i) \left( \sum_{j=1}^{i-1} (n-j)(i-j) + \frac{ni}{2} \right) \\
= & \frac{4}{n^4} \sum_{i=1}^{n-1} (n-i) \frac{1}{6}(i + 3ni^2 - i^3)) \\
= & \frac{4}{6n^4} \sum_{i=1}^{n-1} (ni + (3n^2-1)i^2 - 4ni^3 + i^4) \\
= & \frac{(n-1)(n+1)(2n^2+2)}{15n^3}.
\end{aligned}$$

Correcting the result by multiplying the factor $\frac{n^4}{n^4-n^2}$ which accounts for the fact that the source and the destination must differ, we obtain the average extra distance, which is $\frac{2n^2+2}{15n}$.

Thus the average distance of the WM is

$$\frac{2}{3}n + \frac{2n^2+2}{15n} = \frac{12n^2+2}{15n}$$

or approximately $\frac{4}{5}n$. $\square$

Comparing this with the OM's average distance which is $\frac{2}{3}n$, the WM has an average distance which is 20% longer than that of the WM.

## 4 Optimal Routing Algorithms for the WM

Based on the routing paths given above, a source node can compute the entire path beforehand, encode and embed it in the message, and the routing will simply follow that embedded information from start to finish. A better approach would be self-routing in

[4]Let the length of the tail region be $l$. In general, for $n$ which is not trivially small, there is a $\frac{1}{2}$ chance for $l$ to be even, and a $\frac{1}{2}$ chance for $l$ to be odd. For even $l$, the length of the zigzag path is $2l$; for odd $l$, there is a $\frac{1}{2}$ chance that the path length is $2l+1$, and $\frac{1}{2}$ chance that it is $2l-1$. So, on average, the path length is $2l$.

which the message need not carry any routing information except the destination address. The following routing algorithm, to be executed by every node, implements self-routing.

Define for any pair of source or intermediate node $s = (y_s, x_s)$ and destination node $d = (y_d, x_d)$ the following routing directions.

$$\vec{x} = \begin{cases} +x & \text{if } x_d > x_s \\ -x & \text{if } x_d < x_s \end{cases}$$

$$\vec{y} = \begin{cases} +y & \text{if } y_d > y_s \text{ and } s \text{ is even} \\ -y & \text{if } y_d < y_s \text{ and } s \text{ is odd} \end{cases}$$

Define a *minimal link* as a link that by traversing it, the distance to $d$ is reduced by one.

**Algorithm 1** *(One-to-one routing) Let this node be $t$, and the message just received $m$.*

*If $t = d$, ACCEPT($m$);*
*elseif there is one minimal link, traverse that link (SEND($\vec{x}$) or SEND($\vec{y}$));*
*elseif there are two minimal links, SEND($\vec{y}$);*
*else traverse either one of the two horizontal (non-minimal) links*
*(SEND($+x$) or SEND($-x$)).*

Note that in line 3 above, the vertical link ($\vec{y}$) is preferred; if instead the horizontal link is taken, it is possible that the next link in line is a cut-away edge which could have been avoided if the vertical link was taken.

The above self-routing algorithm routes the message $m$ along a path which is equal to that proposed in Section 3. Hence, the algorithm is optimal.

Broadcast is an important network operation needed by many applications. Supposing that a node $t = (y_t, x_t)$ receives a broadcast message originating from a source node $s = (y_s, x_s)$, the following is the algorithm for $t$, where $\triangle y$ and $\triangle x$ are between $t$ and $s$, and $\vec{x}'$ means the direction opposite to $\vec{x}$.

**Algorithm 2** *((one-to-all) Broadcast)*

*if even $s$ and $y_t > y_s$ $y_s = y_s + 1$;*
*if odd $s$ and $y_t < y_s$ $y_s = y_s - 1$;*
**for the cross-lines:**
*if $\triangle y = \triangle x$*
*{SEND($\vec{x}$);*
*if (odd $\triangle x$ and $x_t > x_s$) or (even $\triangle x$ and $x_t < x_s$) SEND($\vec{x}'$)};*
*if $\triangle x = \triangle y + 1$*
*{SEND($\vec{x}$); SEND($\vec{y}$)};*
**for the regions $H$:**
*if $\triangle x > \triangle y + 1$*
*SEND($\vec{x}$);*

**for the regions $V$:**

*if* $\triangle x < \triangle y$

  $\{$*if (odd* $t$ *and* $y_t < y_s$*) or (even* $t$ *and* $y_t > y_s$*) SEND(*$\vec{y}$*)*

    *elseif even* $\triangle x$ *SEND(*$+x$*)*

    *elseif odd* $\triangle x$ *SEND(*$-x$*);*

  *if* $x_s < y_s$ *and* $x_s$ *is odd and* $x_t = 1$ *SEND(*$-x$*);*

  *if* $x_s < m - y_s - 1$ *and* $x_s$ *is odd and* $x_t = 1$ *SEND(*$-x$*);*

  *if* $n - x_s - 1 < y_s$ *and* $n - x_s - 1$ *is even and* $x_t = n - 2$ *SEND(*$+x$*);*

  *if* $n - x_s - 1 < n - y_s - 1$ *and* $n - x_s - 1$ *is even and* $x_t = n - 2$ *SEND(*$+x$*)*$\}$*;*

The cross-lines and regions $H$ and $V$ are as shown in the example in Fig. 6. The thick lines represent the spanning tree as generated by the above broadcast algorithm. The same tree resulted from $s$ being at the position as marked in the figure or the position that is immediately above it. The two stairs-like cross-lines partition the mesh into four regions, two denoted by $V$ and two by $H$, in which the nodes satisfy $\triangle x < \triangle y$ and $\triangle x > \triangle y + 1$, respectively. The nodes on the cross-lines are those that satisfy either $\triangle y = \triangle x$ or $\triangle x = \triangle y + 1$. In order to correctly compute $\triangle y$ and $\triangle x$, the coordinate $y_s$ of the source node $s$ needs to be adjusted, which is done by the first two lines of the algorithm.

For the $\vec{y}$ and $\vec{x}$ above to make sense, the node $t$ has an imaginary destination $d$: for $t$ on a cross-line, $d$ is the node at the end of the cross-line; for $t$ in a $V$ or $H$ region, $d$ is a node on the boundary of the mesh that is within the region.

Some special nodes such as the nodes $a$, $b$ and $c$ in Fig. 6 need to be specially treated. The last four lines of the algorithm are used to detect whether $t$ is adjacent to such a node.

A broadcast message will be terminated at a boundary node where it cannot be sent any further; that is, the SEND function detects the following condition for the next node $(y_{next}, x_{next})$

$$inside = 0 \leq y_{next} \leq m - 1 \text{ and } 0 \leq x_{next} \leq n - 1.$$

If *inside* is true, the SEND function sends the message to the next node; otherwise it terminates the message.

The above broadcast algorithm is optimal as the path from $s$ to any other node is of the form generated by the *One-to-one* algorithm. It is message-optimal in that no node receives a broadcast message more than once.

## 5 Distances in the WT

Just as in the case of the OT which, by virtue of the wrap-around lines, has a better diameter and average distance than the OM, the WT has a better diameter and average distance than the WM. Recall that the WT is defined only for even number of rows, $m$.

For the source or intermediate node $s = (y_s, x_s)$ and the destination node $d = (y_d, x_d)$,

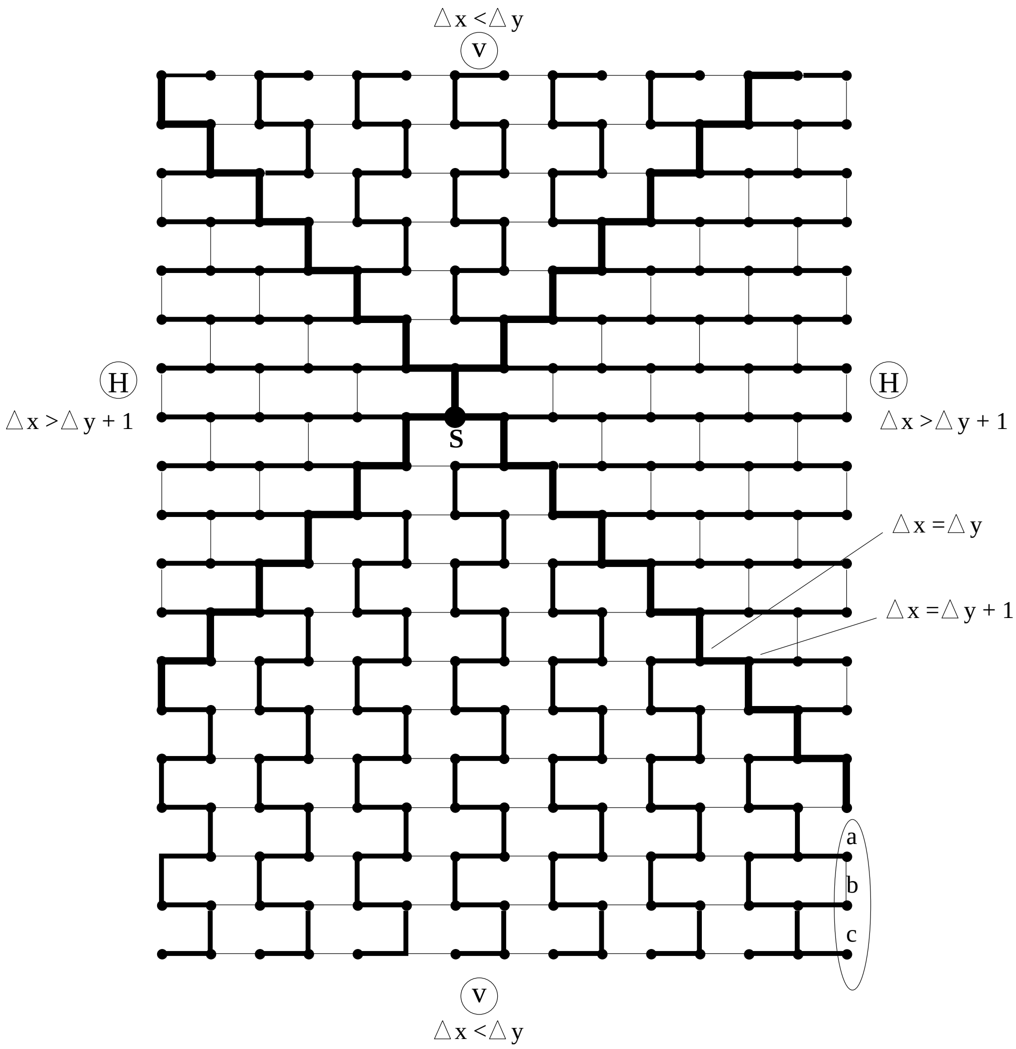



Figure 6: Broadcast in WM(19,15).

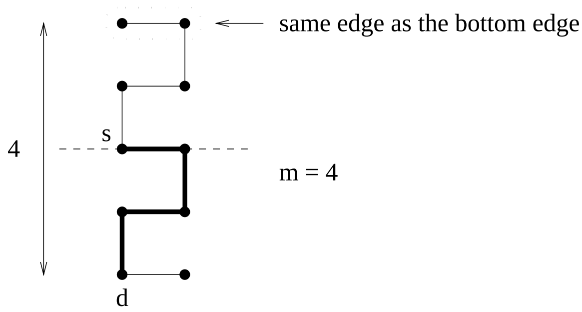



Figure 7: A vertical-cut path in a WT.

we define

$$\Diamond y = \text{MIN}\{\triangle y, m - \triangle y\} \qquad \Diamond x = \text{MIN}\{\triangle x, n - \triangle x\}.$$

$\vec{y}$ and $\vec{x}$ can then be defined to be based on $\Diamond y$ and $\Diamond x$, as follows.

$$\vec{x} = \text{the direction } (+x \text{ or } -x) \text{ as implied by } \Diamond x$$

$$\vec{y} = \begin{cases} +y \text{ if } \triangle y = m - \triangle y \text{ and } s \text{ is even} \\ -y \text{ if } \triangle y = m - \triangle y \text{ and } s \text{ is odd} \\ \text{the direction } (+y \text{ or } -y) \text{ as implied by } \Diamond y\text{, otherwise} \end{cases}$$

Note that $+x$ means $x = x + 1 \pmod{m}$; similarly for $-x$, $+y$, and $-y$.

**Lemma 4** *The length of an optimal path from any node $s = (y_s, x_s)$ to any other node $d = (y_d, x_d)$ in a WT is*

$$Distance(s, d) = \begin{cases} \Diamond x + \Diamond y & \text{if } \Diamond x \geq \Diamond y \\ 2\Diamond y \pm 1 & \text{if } \Diamond x < \Diamond y \end{cases}.$$

**Proof:** Apply the style of routing as proposed in Section 4 (i.e., a square followed by an optional tail) and follow the directions for $x$ and $y$ traversal as given above. □

**Theorem 4** *The diameter of WT($m, n$) is*

$$Diameter(WT(m, n)) = \begin{cases} \lfloor \frac{n}{2} \rfloor + \frac{m}{2} & \text{if } n \geq m \\ m & \text{if } n < m \end{cases}.$$

**Proof:** Similar to the proof of Theorem 5 except that in this case, the longest vertical-cut path is shorter (by about half) than that in the WM because of the wrap-around links. Since $m$ is even, the length of of such a path is $m$ (see the example, for $m = 4$, in Fig. 7). □

Clearly, the square WT has the same diameter as its OT counterpart.

**Theorem 5** *The average distance of the square WT($n, n$) is $\frac{7n^4+2n^2}{12n^3-12n}$, or approximately $\frac{7}{12}n$.*

**Proof:** Consider the distribution function of the random variable $\diamond x$ (or $\diamond y$) from any fixed node. Since the WT is node-symmetric, the frequency of $\diamond x$ is

$$\begin{array}{rcccccc} \diamond x = & 0, & 1, & 2, & \ldots, & n/2-1, & n/2 \\ Frequency = & 1, & 2, & 2, & \ldots, & 2, & 1 \end{array}$$

We use $P(\diamond x = i)$ to denote the probability of the variable $\diamond x$ having the value $i$. Hence

$$P(\diamond x = i) = \begin{cases} 1/n & \text{if } i = 0 \\ 2/n & \text{if } 0 < i < n/2 \\ 1/n & \text{if } i = n/2 \end{cases}$$

$$P(\diamond x < i) = \sum_{i=0}^{i-1} P(\diamond x = i) = \frac{2i-1}{n}.$$

The average distance is

$$\begin{aligned} & P(\diamond x \geq \diamond y) \times (\diamond x + \diamond y) + P(\diamond x < \diamond y) \times 2\diamond y \\ = \; & \sum_{i=1}^{n/2} P(\diamond x = i) \sum_{j=0}^{i} P(\diamond y = j)(i+j) + \sum_{i=1}^{n/2} P(\diamond y = i) P(\diamond x < i) 2i \\ = \; & P(\diamond x = n/2) P(\diamond y = 0) n/2 + P(\diamond x = n/2) P(\diamond y = n/2) n \\ & + P(\diamond x = n/2) \sum_{j=1}^{n/2-1} P(\diamond y = j)(n/2 + j) \\ & + \sum_{i=1}^{n/2-1} P(\diamond x = i) \sum_{j=1}^{i} P(\diamond y = j)(i+j) \\ & + \sum_{i=1}^{n/2-1} P(\diamond x = i) P(\diamond y = 0) i + P(\diamond y = n/2) P(\diamond x < n/2) n \\ & + \sum_{i=1}^{n/2-1} P(\diamond y = i) P(\diamond x < i) 2i \\ = \; & \frac{1}{n}\frac{1}{n}\frac{n}{2} + \frac{1}{n}\frac{1}{n} n + \frac{1}{n} \sum_{j=1}^{n/2-1} \frac{2}{n}\left(\frac{n}{2} + j\right) + \sum_{i=1}^{n/2-1} \frac{2}{n} \sum_{j=1}^{i} \frac{2}{n}(i+j) \\ & + \sum_{i=1}^{n/2-1} \frac{2}{n}\frac{1}{n} i + \frac{1}{n}\frac{n-1}{n} n + \sum_{i=1}^{n/2-1} \frac{2}{n}\frac{2i-1}{n} 2i \\ = \; & \frac{7n^2+2}{12n}. \end{aligned}$$

Multiplying by the correcting factor $\frac{n^2}{n^2-1}$ which accounts for the fact that the source and the destination must differ, the final result is $\frac{7n^4+2n^2}{12n^3-12n}$, or approximately $\frac{7}{12}n$. $\square$

This represents a 16.7% increase in average distance from that of the corresponding OT.

We omit the detailed description of the routing algorithms for WT as the *One-to-one routing* algorithm and the *Broadcast* algorithm presented before for the WM can be easily modified for the WT.

Table 1: Performance comparison of square WM/WT and OM/OT

| Network | Degree | Diameter | Average Distance | Connectivity | Node Symmetry |
|---|---|---|---|---|---|
| M(n,n) | 4 | $2n-2$ | $\frac{2}{3}n$ | | $\times$ |
| T(n,n) | 4 | $n$ | $\frac{1}{2}n$ | 4 | $\surd$ |
| WM(n,n) | 3 | $2n-1$ | $\frac{4}{5}n$ | | $\times$ |
| WT(n,n) | 3 | $n$ | $\frac{7}{12}n$ | 3 | $\surd$ |

# 6 Comparison

Table 6 summarizes the comparison of the WM/WT with OM/OT in terms of various fundamental measures. We consider only the square networks as they should be the ones that are most practical. From the table we see the following.

- From OM/OT to WM/WT, the **diameter** is not affected although one degree is deleted.
- From WM to WT, just like from OM to OT, the **diameter** is reduced by half.
- As for **cost-effectiveness**, typically measured by *degree* $\times$ *diameter*, WM/WT is clearly superior to OM/OT. Even if the measure is *degree*$\times$ *average-distance*, WM/WT is still superior. In particular, WM has a $\frac{1}{5}$ or 20% increase in average distance when compared with OM, and WT a $\frac{1}{6}$ or 16.7% increase when compared with OT, but their degree has a 25% decrease from that of OM/OT.
- Considering the fact that the **channel width** of networks is often limited by node size (or node pin count) [1, 3], WM/WT has a wider channel width than OM/OT and hence a faster transmission speed along the wires.
- WT, like OT, can be laid out on a plane without **long wires** [10, 5].
- WT, like OT, has optimal **fault tolerance** since its connectivity is 3 which is equal to its degree. Although the number of the shortest paths between two nodes in WM/WT is much less than that in OM/OT, the optimal fault tolerance of WT is not affected. It is not difficult to see that there exist three node-disjoint paths between any two nodes in a WT.

The above points apply only to square WMs/WTs (*i.e.*, $n = m$). In general, the difference in distance performance between WM/WT and OM/OT is directly proportional to the ratio $\frac{m}{n}$.

# 7 Concluding Remarks

The motivation underlying the work presented here is the question of how much we can reduce the hardware costs of an ordinary mesh while keeping its performance. Our answer is the wall mesh in which every node uses one fewer link (a 25% reduction in hardware link costs) and the overall performance is only slightly affected.

The WM bears a direct resemblance to the honeycomb architecture [8] and the honeycomb mesh (HM) [11], which all use 6-vertex cells as the basic building blocks. In the honeycomb architecture, however, the cell (not vertex) is a processor. On the other hand, in the HM as well as the WM, each vertex is a processor. The main difference between the the HM and the WM is that the former uses an $(x, y, z)$-labeling scheme for the nodes, while the latter uses $(x, y)$-labeling. Which labeling scheme to use has much bearing on the design of the routing scheme. The three-coordinate scheme obviously would result in a routing scheme which is more complicated than that of a two-coordinate scheme. The three-coordinate scheme tends to favor an HM that is bounded by a hexagon. The WM is different, which is being conceived as a variant of the ordinary mesh—hence the WM is rectangular, its nodes are labeled as in the ordinary mesh, and its basic routing algorithm is as simple as that of the ordinary mesh.

Further research problems include:

- Higher dimensional WMs: How to extend the wall mesh to having three or more dimensions?
- Mesh with special links: Suppose the deleted links can be added back; how should these links be connected so that the resulting mesh can overcome some of the problems of WM or OM, such as the long diameter and the lack of connections for some specific applications?
- A variant of the Moore problem [4]: Given a known interconnection network of degree $k$ and diameter $d$, what is the minimum diameter of the network if we are allowed to add $p$ degrees to each node?
- Other degree-3 meshes: Are there better methods to delete one degree from each node of the ordinary mesh?
- Is the WT a Cayley graph [2]?

# References

[1] A. Agarwal, "Limits on Interconnection Network Performance", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 2, No. 4, October 1991, pp. 398–412.

[2] S.B. Akers and B. Krishnamurthy, "A Group-Theoretic Model for Symmetric Interconnection Networks", *IEEE Trans. on Computers*, Vol. 38, No. 4, April 1989, pp. 555–566.

[3] W.J. Dally, "Express Cube: Improving the Performance of k-ary n-cube Interconnection Network", *IEEE Trans. on Computers*, Vol. 40, No. 9, September 1991, pp. 1016–1023.

[4] M.J. Dinneen and P.R. Hafner, "New Results for the Degree/Diameter Problem", *Networks*, Vol. 24, 1994, pp. 359–367.

[5] F.C.M. Lau and G. Chen, "Optimal Layout of Midimew Network", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 7, No. 9, September 1996, pp. 954–961.

[6] F.T. Leighton, B. Maggs, and S. Rao, "Universal Packet Routing Algorithms", *Proc. of 29th Ann. Symp. on Foundations of Computer Science*, 1988, pp. 256–269.

[7] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercube*, Morgan Kaufmann Publishers, 1992.

[8] V. Milutinovic, V. Milutinovic, and B. Soucek, "The Honeycomb Architecture", *IEEE Computer*, V. 20, No. 4, 1987, pp. 81–83.

[9] I.D. Scherson and A.S. Youssef, *Interconnection Networks for High-Performance Parallel Computers*, IEEE Computer Society Press, 1994.

[10] C.H. Sequin, "Doubly Twisted Torus Networks for VLSI Processor Array", *Proc. of 8th Annual Symposium on Computer Architecture*, 1981, pp. 471–480.

[11] I. Stojmenovic, Honeycomb Networks: Topological Properties and Communication Algorithms, TR-95-01, Computer Science Department, University of Ottawa, January 1995.

[12] C. Wu and T. Feng, *Interconnection Networks for Parallel and Distributed Processing*, IEEE Computer Society Press, 1984.