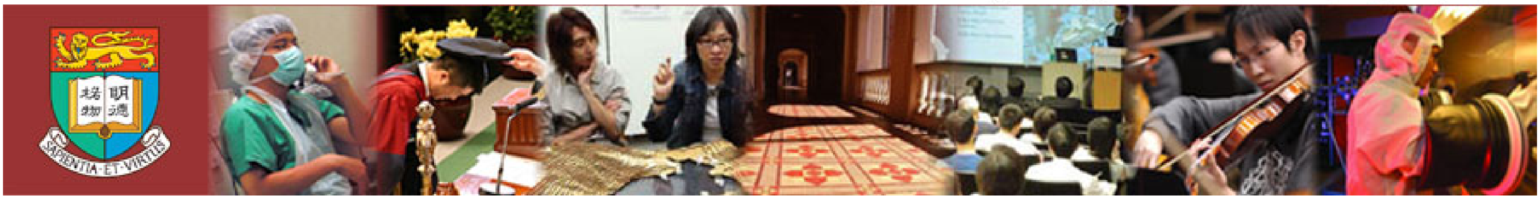


Title	Optimal layouts of midimew networks
Author(s)	Lau, FCM; Chen, G
Citation	Ieee Transactions On Parallel And Distributed Systems, 1996, v. 7 n. 9, p. 954-961
Issued Date	1996
URL	http://hdl.handle.net/10722/43634
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Optimal Layouts of Midimew Networks

Francis C.M. Lau, *Member, IEEE* and Guihai Chen

Abstract—Midimew networks [4] are mesh-connected networks derived from a subset of degree-4 circulant graphs. They have minimum diameter and average distance among all degree-4 circulant graphs, and are better than some of the most common topologies for parallel computers in terms of various cost measures. Among the many midimew networks, the rectangular ones appear to be most suitable for practical implementation. Unfortunately, with the normal way of laying out these networks on a 2D plane, long cross wires that grow with the size of the network exist. In this paper, we propose ways to lay out rectangular midimew networks in a 2D grid so that the length of the longest wire is at most a small constant. We prove that these constants are optimal under the assumption that rows and columns are moved as a whole during the layout process.

Index Terms—Dilation, graph embedding, interconnection networks, mesh-connected computers, midimew networks, parallel processing, VLSI.

1 Introduction

In designing interconnection networks, a constant low degree is generally more preferable than a high degree because of performance advantages, scalability, and various implementation considerations [1], [9], [17], [20]. Within the family of low-degree networks, degree-4 networks such as 2D meshes and 2D tori are among the most popular choices for processor interconnection in today's parallel computers [14], [16], [20]. One practical problem with these networks, however, is that their diameter and average distance tend to be large as the number of nodes increases, and so there has been a continuous effort in finding degree-4 or low-degree networks that have smaller diameters and average distances [2], [4], [6], [7], [11], [12]. One recent proposal is the *midimew* network (Minimum Distance Mesh with Warp-around links) [4]. This family of networks is isomorphic to a subset of circulant graphs [5] whose diameter and average distance are minimum among all degree-4 circulant graphs. The midimew network (or simply midimew) is worth studying because of its resemblance to the 2D torus (see Fig. 3a). In fact, by reconnecting the horizontal wrap-around links in a 2D torus, we have a midimew network with improved diameter. Specifically, the diameter is improved by factor of $\sqrt{2}$.[1] Similar improvement is also made to the average distance. Using diameter × degree as a measure, the midimew also outperforms the 3D mesh, and compares favorably with the 3D torus and the hypercube. Furthermore, because of its closeness in structure to the 2D torus, the midimew shares some of the positive characteristics of the torus, such as regularity, vertex-symmetry, fault-tolerance, and the existence of simple routing functions.

1. The midimew has a diameter of $\sqrt{\frac{N}{2}}$ or $\sqrt{N}-1$, and the torus has a diameter of $\sqrt{N}$, where N is the number of nodes.

• *The authors are with the Department of Computer Science, The University of Hong Kong, Hong Kong. E-mail: {fcmlau, gchen}@cs.hku.hk.*

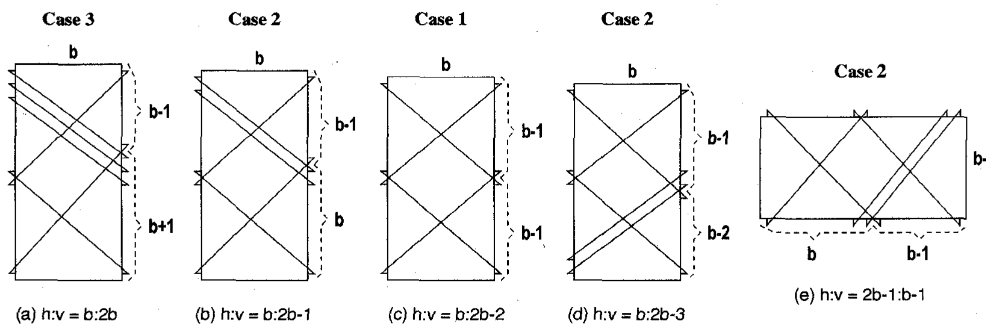
Manuscript received June 1, 1995; revised Dec. 1995. A preliminary version of this paper appeared in Proceedings of CONPAR 94–VAPP VI Joint International Conference, pp. 773-784, Linz, Austria, Sept. 1994.
For information on obtaining reprints of this article, please send e-mail to: transpds@computer.org, and reference IEEECS Log Number D95217.

The midimew network is of degree 4 and is most naturally laid out on a 2D plane. The normal way of laying them out on a plane, however, would lead to long "cross wires" whose length grows with the size of the network (see Fig. 3a, for example). In this paper we show how they can be laid out in a different manner so that the longest wires are confined to a length which is a constant and is independent of the size of the network. The importance of keeping wires short is well-recognized in the design of parallel machines and VLSI systems [3], [8], [10], [13], [15], [17], [21]. Formally, this is a problem of embedding (laying out) a guest graph of a midimew network onto a host graph of a 2D grid with minimization of the dilation or the wire length in this case. The constant wire lengths of the longest wire that we achieve through our layout procedures can be proved to be optimal under the assumption of "synchronous mapping" in which an entire row or column is moved at a time during the layout process, as opposed to moving individual nodes. The only directly-related work the authors are aware of is that of laying out doubly twisted torus networks on a plane [18], in which the procedure tries to reduce the amount of wire crossings but does not push for optimizing the wire lengths.

Section 2 presents the five types of rectangular midimews for a given number of nodes, which are the targets for our layout procedures. Section 3 presents the fundamental operations that are subsequently employed in the layout procedures. The actual layout procedures for the various cases and their optimality are presented in Section 4. In addition to minimizing wire lengths, the procedures should also try to minimize the area, as is done in [3], and the cross-section density. In Section 5, we prove that our layouts do not increase the area originally required and the cross-section density. The appendix reviews the basic structure of midimews.

2 Rectangular Midimews

The midimew has a diameter of b or $b-1$, where $b=\left\lceil\sqrt{\frac{N}{2}}\right\rceil$ and N is the number of nodes. For every $b > 2$ ($N > 8$), there are $4b-2$ midimew networks, and among them, five are rectangular (see



Fig, 1. The five rectangular midimews (for any given b).

the appendix). In this paper, we limit our discussion to rectangular midimews because the rectangular shape is likely to be the natural choice for practical implementations. Also, the routing function for a rectangular midimew should be easier to derive than that for a nonrectangular midimew.

We isolate the five rectangular midimew networks for any given b and display them in Fig. 1. For Figs.1a–d, the vertical wrap-around links that join a top node and a bottom node have a deviation of $r = 0$ (i.e., rectangular midimews; refer to Fig. 14 of the appendix); we call these and all the links between adjacent nodes in the underlying mesh "mesh links/wires." These links are omitted in the figure in order to highlight the other kind of links—the slanted links which we referred to as "cross links/wires." Fig. 1e is equal exactly to Fig. 1b through a rotation of 90 degrees, and Fig. 1d, when flipped vertically, is equivalent to Fig. 1b (the proportion between h and v does not matter in our layout procedures). From Fig. 1a–1d, we can see there are two sets of cross wires, the ones going from the lower left nodes to the upper right nodes, and the ones going from the upper left nodes to the lower right nodes; and similarly for Fig. 1e. For the purpose of reducing the lengths of wires, these five midimews can be classified into three cases based on the difference in number between the two sets of cross wires:

1) Fig. 1c for which the two sets of wires are equal in number,
2) Figs. 1b, 1d, 1e with a difference of 1, and
3) Fig. 1a with a difference of 2.

In the sequel, we take each of these cases, rearrange the nodes on the grid so that the connectivity is preserved. The main measure of interest is the length of the longest wire in the final layout. In the original layouts of midimews, the longest wire length is proportional to the size of the network, which is undesirable for real implementation. Our aim is to transform these long wires into constant-length wires. For easy visualization, we draw all wires as (shortest) straight lines, and so wires that join nodes at different rows and columns are slanted (actually the hypotenuse of a triangle). For a strict 2D layout, some of these wires might have to be "wired around" the nodes—i.e., as a series of vertical and horizontal wire segments. The resulting new length, however, is at most $\sqrt{2}$ times that of the length of a direct, slanted straight wire, and hence the two length measures should be more or less equivalent as far as optimizing wire length is concerned.

3 Tools for the Layout Procedures

Synchronous mapping: In order to produce an optimal layout, we take the midimew through a number of transformation steps, each of which corresponds to moving one or more rows or columns from one position to another. Note that an entire row or column is moved, not individual nodes—this we refer to as *synchronous mapping*. On the one hand, synchronous mappings make the analysis tractable; on the other hand, the maximum wire length after the transformation, as will be shown later, is but a small constant in all cases, thus diminishing the need for asynchronous mapping.

Shuffle: Consider Fig. 2a which shows a ring of n nodes, where n is even. We refer to this particular layout of the ring as the *natural form* in which the direct neighbors of a node are placed physically next to the node, except for the two extreme nodes. We index the nodes in this layout $0, 1, 2, \cdots, n/2 - 1, n/2, \cdots, n - 1$. A (perfect) shuffle operation divides the ring in the middle and then shuffles the two portions together so that the nodes of the two portions interleave evenly. The result is as shown in Fig. 2a. For the case of odd n, the operation is similar, and the result is as shown in Fig. 2b. The shuffle operation can be defined precisely as follows:

$$y = \begin{cases} 2x & \text{if } x < \lceil \frac{n}{2} \rceil \\ 2x + 1 - n & \text{otherwise} \end{cases}$$

where x is the position of a node before the shuffle, and y is its position after the shuffle, assuming that the top position is 0.

One important effect of the shuffle operation is that if there is a wire that joins a node in the upper portion to a node in the lower portion, the wire could become very short in length. We note at this point that all the cross wires of a midimew are actually of this type.

Property 1. *Given a ring of n nodes in natural form. After a shuffle operation, the wires have a length of* 2 *except the two wires joining the two topmost nodes to the two bottommost nodes in the transformed ring.*

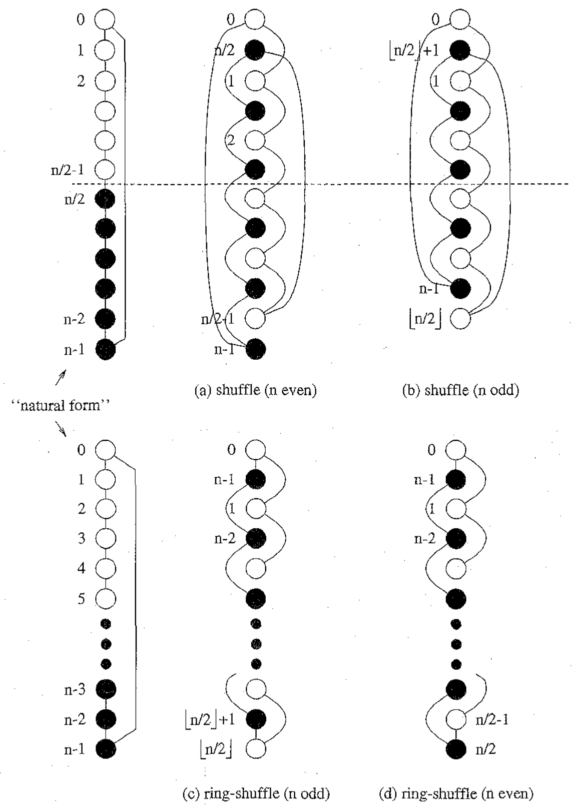


Fig. 2. Shuffle operations.

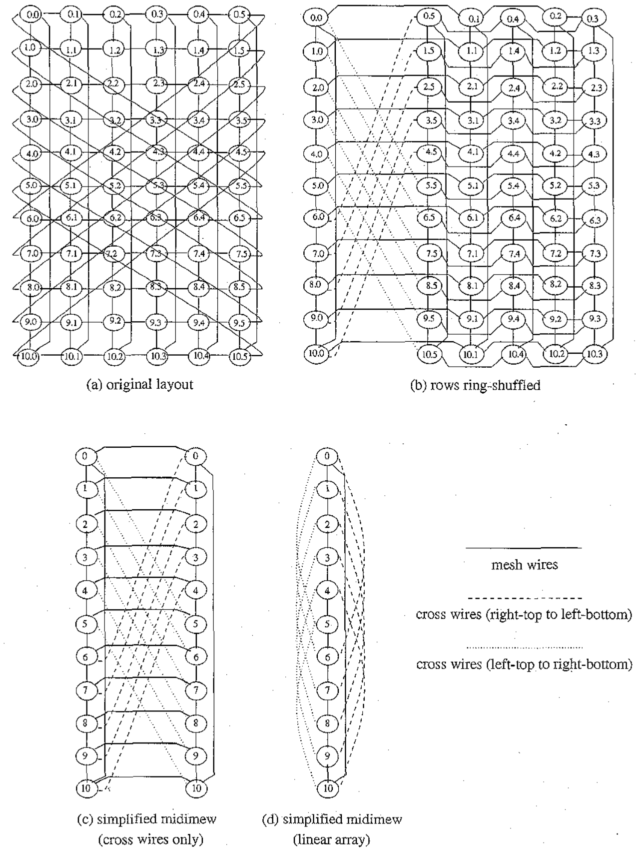


Fig. 3. A midimew example (Case 2, $b = 6$).

Ring-shuffle: Given a ring of n nodes in natural form, the result of ring-shuffle is as shown in Fig. 2c for odd n and Fig. 2d for even n. This operation is very similar to the shuffle operation except that the lower portion nodes are flipped vertically before they are shuffled into the upper portion nodes. The equation for ring-shuffle is as follows.

$$y = \begin{cases} 2x & \text{if } x < \left\lceil \frac{n}{2} \right\rceil \\ 2n - 2x - 1 & \text{otherwise} \end{cases}$$

PROPERTY 2. *The wire length after a ring-shuffle is at most 2. The two bottommost nodes ($n - 2$ and $n - 1$) are moved to the top and so the top four nodes after the operation are $0, n - 1, 1, n - 2$, in this order.*

4 Constant-Wire-Length Layouts of Midimews

We consider all five of the rectangular midimews for any value of b, which are separated into three cases as distinguished by the difference in number between the two sets of cross wires (Fig. 1). The problem can be transformed into one with a much simpler structure, as illustrated in Fig. 3. We note that in Fig. 3a, the cross wires are the hypotenuse of a right-angled triangle; therefore, by shortening the right-angled (vertical and/or horizontal) edges of the triangle, these wires will also be shortened; to shorten the horizontal edge, we apply a ring-shuffle to every row; the result is shown in Fig. 3b.

Given that we perform only synchronous mapping, we can represent Fig. 3b by Fig. 3c (and abbreviate the node indices accordingly) since the latter preserves both kinds of long wires (mesh wires and cross wires). Finally, we collapse the two columns, resulting in the linear array in Fig. 3d, in which the cross wires are very close approximations of the actual slanted wires. In the following, we use the simplified structure consisting of a linear array of nodes. Note that we use solid lines to represent mesh wires and dashed or dotted lines to represent cross wires in all subsequent midimew figures.

4.1 Case 1

This is the case in which the two sets of wires are equal in number—corresponding to Fig. 1c. For each b, $v = 2b - 2$.

THEOREM 1. *The lower bound for the wire length of laying out the nodes corresponding to a Case 1 midimew ($b > 4$) in a linear array is 4.*

PROOF. We try to keep the wire length to be ≤ 3 when placing the nodes in the linear array. Without loss of generality, assume node 0 is placed at one end of the linear array (Fig. 4a). Then, based on the structure of the Case 1 midimew (Fig. 1c), we have the picture of the neighborhood of node 0 as shown in Fig. 1b. Since node 0 is at one end of the linear array, its three direct neighbors, $(1, b - 1, 2b - 3)$, must be placed in the next three slots, in order to keep the wires within the length of 3. Then, there are four distinct neighbors $(2, b - 2, b, 2b - 4)$ of these three nodes which need to be placed. Clearly, no

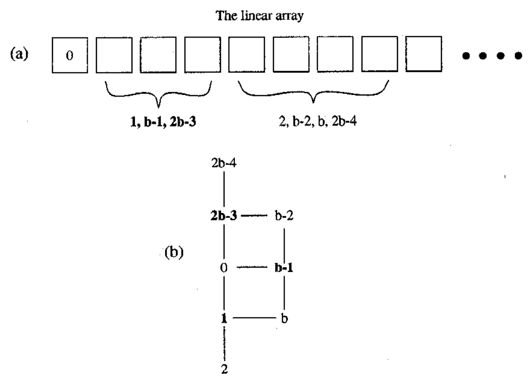


Fig. 4. Lower bound for Case 1.

matter how close to these three nodes they are placed, at least one of the wires would be ≥ 4. The result is valid provided that $b - 2 > 2$ (or $2b - 4 > b$)—i.e., $b > 4$.□

The layout procedure is as follows.

PROCEDURE 1 (CASE 1).

1) Shuffle: After the shuffle operation, the order of the nodes becomes $0, v/2, 1, v/2 + 1, \cdots$.
2) Node pairing: Group each pair of adjacent nodes together to form a "compound node"—that is, $(0, v/2)$, $(1, v/2 + 1), \ldots□, (v/2 - 1, v - 1)$; the compound nodes form a ring of $v/2$ nodes in natural form.
3) Ring-shuffle: applied to the ring of compound nodes.

THEOREM 2. *The maximum wire length due to Procedure* 1 *for Case* 1 *midimew is* 4 *which is optimal.*

PROOF. The cross wires are shortened to a length of 1 by the shuffle operation; they remain unchanged till the end because of node pairing. By Property 1, the shuffle operation changes the length of the mesh wires to 2 except for the wires joining the four extreme nodes. These four extreme nodes become the two extreme nodes in the ring of compound nodes (call this the "compound ring"). After applying a ring-shuffle to the compound ring, by Property 2, all the wires in this compound ring become of length 2 or 1, and the two extreme compound nodes are placed next to each other. Expanding (from compound nodes back to the original nodes), we have a set of mesh wires that are of length 4 or 2, except those joining the four extreme nodes, which are of length at most 3 as these nodes are placed together. Comparing with the lower bound result (Theorem 1), the transformation is optimal. □

An example for $b = 5$ and $v = 2b - 2 = 8$ is shown in Fig. 5. Fig. 5f shows the complete look of the midimew after the above transformation.

Note that since the two sets of cross wires are equal in number, the above procedure is applicable to and the theorem valid for both odd and even b.

4.2 Case 2

This case corresponds to Figs. 1b, 1d, 1e of which the difference in number between the two sets of cross wires is 1. For each b, $v = 2b - 1$ or $2b - 3$. Fig. 7 shows an example with $b = 5$ and $v = 2b - 1 = 9$.

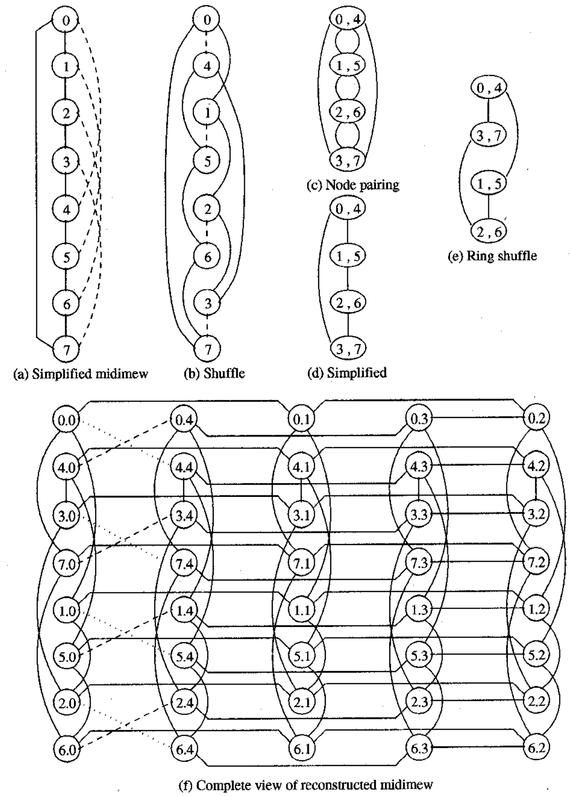


Figure 5. Case 1 ($b = 5$).

THEOREM 3. *The lower bound for the wire length of laying out the nodes corresponding to a Case* 2 *midimew* ($b > 4$) *in a linear array is* 4.

PROOF. Similarly to the lower bound proof for Case 1, we place node 0 at one end of the array. By noting that node 0 has four neighbors, and that these four nodes have four other distinct neighbors (Fig. 6), we have a lower bound of 4. This result is valid for $b - 2 > 2$ (or $2b - 3 > b + 1$)—i.e., $b > 4$. □

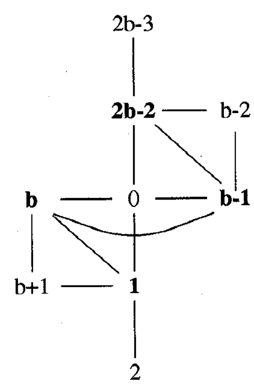


Fig. 6. Structure of node 0's neighborhood in Case 2.

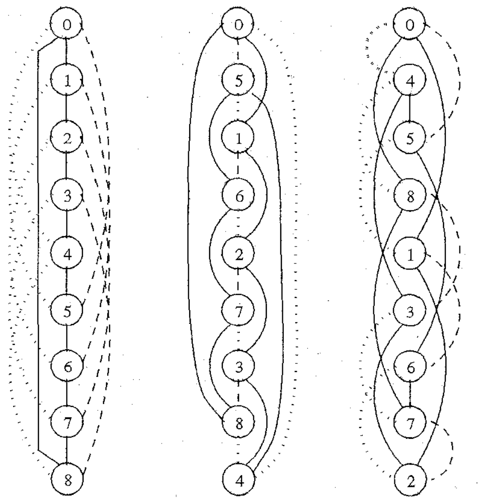
The layout procedure is as follows:

PROCEDURE 2 (CASE 2).

1) Shuffle: after this operation, the order of nodes is 0, b, 1, $b+1, \cdots, 2b-2, b-1$ (assuming the case of $v = 2b-1$).
2) Ring-shuffle

THEOREM 4. *The maximum wire length due to Procedure 2 for Case 2 midimew is 4 which is optimal.*

PROOF. It can be easily verified that the cross wires form a ring. By Property 1, the shuffle operation reduces the mesh wires to a length of 2 except for the ones connecting the four extreme nodes. The shuffle operation also rearranges the ring formed by the cross wires into natural form. This is because node 0 is connected to node b by a cross link, and b to 1 by another cross link, and so on. The ring-shuffle then reduces the cross wires to a length of at most 2, by Property 2. Note that each mesh wire after the shuffle operation overlaps with two cross wires, except the ones connecting the four extreme nodes, which overlap with more. Therefore, by the ring-shuffle, these wires are stretched to a length of at most 4. By Property 2, the four extreme nodes are placed together at the top, and hence the wires that connect them are of length at most 3. □



(a) Simplified midimew (b) Shuffle (c) Ring-shuffle

Fig. 7. Case 2 ($b = 5$).

4.3 Case 3

This is the difficult case corresponding to Fig. 1a. The same structure is shown with more details in Fig. 8a. The difference in number between the two sets of cross wires is 2. For each b, $v = 2b$. We first derive a lower bound for the wire length which is specific to this case.

THEOREM 5. *The lower bound for the wire length of laying out the nodes corresponding to a Case 3 midimew ($b > 4$) in a linear array is 5.*

PROOF. As in the proofs of Theorems 1 and 3, we try to place the nodes such that the wire length is ≤ 4. Without loss of generality, assume node 0 is placed at one end of the linear array. Then, from the structure of the Case 3 midimew as shown in Fig. 8a, we see that node 0 has four neighbors, 1, $b-1$, $b+1$, $2b-1$, which in turn have five distinct neighbors, 2, $b-2$, b, $b+2$, $2b-2$ (Fig. 9b). Node ∅ s four neighbors must be placed in the four slots next to node 0. Clearly, then, no matter how close to these four nodes the other five nodes are placed, at least one of the wires would be ≥ 5 (Fig. 9a). The result is valid provided that $b-2 > 2$ (or $2b-2 > b+2$)—i.e., $b > 4$. □

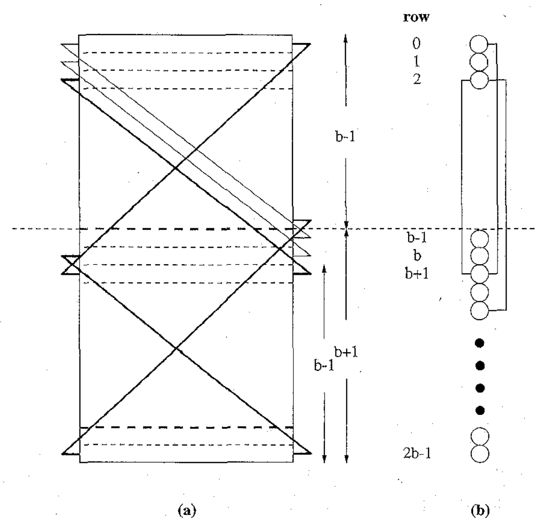


Fig. 8. Structure of Case 3.

Let's first consider a special property of Case 3 which is important for deriving the layout procedures and the proofs below. In Fig. 8a, we see that the cross wires from left-top to right-bottom have a displacement of $b-1$ while the cross wires from right-top to left-bottom have a displacement of $b+1$. If we trace the wires, starting from right-top, as in Fig. 8b in which the two-column structure has been simplified into a linear array of nodes, we find the following connected components.

- **If b is odd.** $b-1$ is even, then nodes 0, $b+1$, 2, $b+3$, $\cdots$, $2b-2$, $b-1$ form a ring via the cross wires, and nodes 1, $b+2$, 3, $b+4$, $\cdots$, $2b-1$, b form another ring; these two rings are referred to as the "first" and "second" ring, respectively.

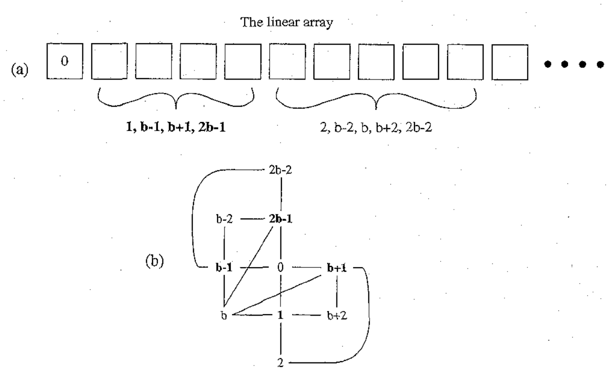


Fig. 9. Lower bound for Case 3.

- **If b is even.** $b - 1$ is odd, then all the nodes form a single ring: $0, b+1, 2, b+3, \cdots, b-2, 2b-1, b, 1, b+2, \cdots, b-1$.

Fig. 10 and Fig. 11 show two examples of Case 3 with an odd b and an even b, respectively, including the rings that are formed by the cross links.

The layout procedure for the case of odd b is as follows. In the description, we make references to the example in Fig. 10.

PROCEDURE 3 (CASE 3; ODD b).

1) Node pairing: Take the second ring as identified above for the odd-b case, shift it clockwise by one position—i.e., the order of nodes becomes $b, 1, b+2, \cdots$ (Fig. 10a2); then, pair the corresponding nodes of the two rings, resulting in a ring of compound nodes: $(0, b)$, $(b+1, 1)$, $(2, b+2), \cdots, (b-1, 2b-1)$ (Fig. 10b).
2) Ring-shuffle: applied to the compound ring (Fig. 10c).
3) Expanding: Each compound node is expanded into two original nodes being placed next to each other in the final layout; the order of the two nodes can be arbitrary (Fig. 10d).

THEOREM 6. *The maximum wire length due to Procedure 3 for a Case 3 midimew, where b is odd, is 5 which is optimal.*

PROOF. Consider nodes 0, 1, 2, where 0 and 2 are in the first and third position of the first ring, respectively, and 1 is in the first position of the second ring initially. By shifting of the second ring, node 1 is moved to the second position. Then, when the two rings are compounded, node 1 of the second ring appears in the second position (together with node $b + 1$ of the first ring) which is its "natural" position in between 0 and 2; the same applies to all other nodes which get placed into their natural positions in the compound ring—i.e., node i is right before node $i + 1 \bmod 2b$ and right after node $i - 1 \bmod 2b$, except for the nodes at the two ends. Therefore, if we reconnect the mesh wires, we have another ring which coincides exactly with the compound ring formed by cross wires. Both rings are in their natural form. The ring-shuffle then transforms them into a form in which all the wires are of length at most 2, by Property 2. Expanding, we have the length of these wires (cross wires and mesh wires) ≤ 5 because the longest wire that spans six nodes (expanded from three compound nodes) is at most 5 in length. □

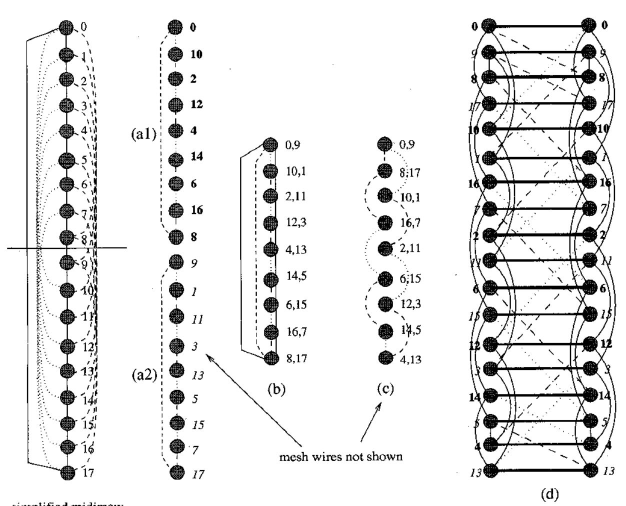


Fig. 10. Case 3—b odd ($b = 9$).

PROCEDURE 4 (CASE 3; EVEN b).

1) Node pairing: Divide the ring in the middle and compound the nodes of the upper portion and the nodes in the lower portion (Figs. 11a, b); the compound ring is in natural form, and the order of the nodes is $(0, b)$, $(b+1, 1)$, $(2, b+2)$, $\cdots$, $(2b-1, b-1)$.
2) Ring-shuffle: applied to the compound ring (Fig. 11c).
3) Expanding: Each compound node is expanded into two original nodes being placed next to each other in the final layout; the order of the two nodes can be arbitrary (Fig. 10d).

THEOREM 7. *The maximum wire length due to Procedure 4 for a Case 3 midimew, where b is even, is 5 which is optimal.*

PROOF. The proof is similar to the one for Theorem 6: The compound ring is in natural form and its nodes are in their natural positions; after the ring-shuffle, the links (mesh and cross links) are of length at most 2, and hence the maximum wire length in the expanded final layout is 5. □

5 AREA AND CROSS-SECTION DENSITY

In the following, we show that our layout procedures do not change the area originally required and the cross-section density.

With reference to Thompson's model [19], the natural way of laying out a midimew with aspect ratio $1b/2b$ (width/height) needs at least an area of $16b^2$. Take Fig. 3a for example:

- The vertical mesh wires and the vertical long wires contribute a width of $2 \times 1b = 2b$.
- The horizontal mesh wires contribute a height of $1 \times 2b = 2b$.

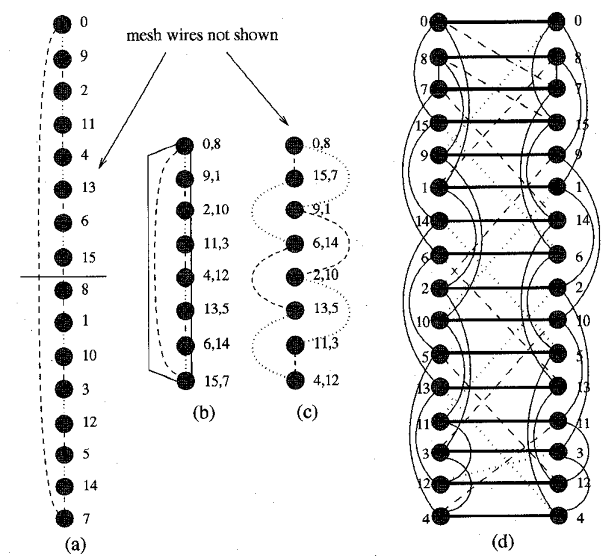


Fig. 11. Case 3—b even ($b = 8$).

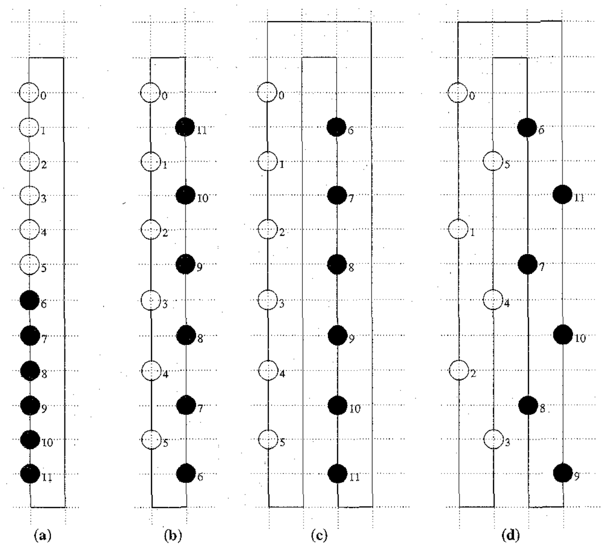


Fig. 12. (b) ring-shuffle; (c) shuffle; (d) shuffle followed by ring-shuffle.

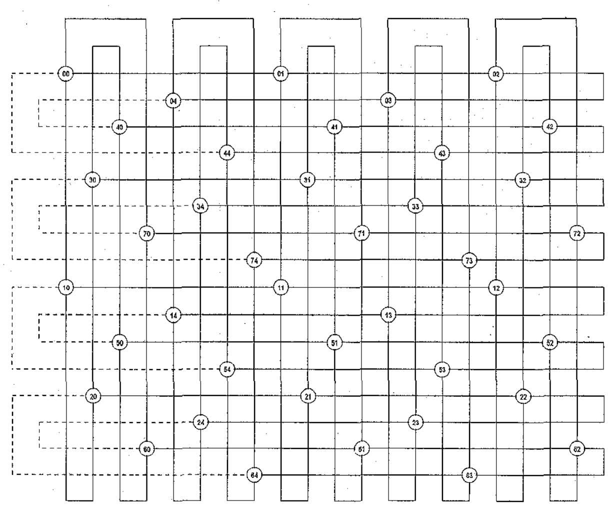


Fig. 13. Layout of Figure 5.

- Each of the cross links needs to go through at least one horizontal track and one vertical track in order to join the two nodes that are on opposite sides. Hence, each cross link contributes a width of 1 and a height of 1; and there are $2b$ such links.

We now show in the following that our layouts use exactly the same area of $16b^2$. We first map them to a Thompson grid. Fig. 12 shows how to lay out one column (a ring) with even number of nodes of a midimew on the grid. In Fig. 12d, we see that the width is4 which is two times the original width (Fig. 12a)—hence a width of $4b$ for the midimew when laid out on the grid.

The height remains unchanged, which is $2b$. But then referring to Fig. 3b (the same is true for all other cases), we observe that each horizontal ring joins every other node in a row, instead of adjacent nodes. Therefore, we need to create one extra horizontal track between every two adjacent horizontal tracks in Fig. 12d so that a node in a column can connect to the other node two columns away through a straight wire—this gives rise to a height of $4b$, and hence a total area of $16b^2$. A completely laid out example of the Case 1 ($b = 5$) midimew in Fig. 5 is shown in Fig. 13. For odd number of nodes in a column, the layout method is similar.

Regarding cross-section density which is the number of wires through a cross section, we note that for a midimew of aspect ratio $1b/2b$, the (worst-case) minimum density is $4b$. This is so because when bisecting the network horizontally or vertically, at least one vertical (or horizontal) cross wire per row (or column) is cut (see Fig. 3a, for example); together with the short mesh wires, the minimum density is therefore $2 \times 2b = 4b$. It turns out that the transformed networks when laid out on a Thompson grid have exactly the same minimum density. This comes naturally as a corollary of the area arguments above: each unit of area corresponds to one wire running vertically and one wire running horizontally; therefore a $4b \times 4b$ area gives a minimum cross section density of $4b$, regardless of which way the network is bisected.

6 Conclusion

We have shown how all five types (in three cases) of rectangular midimews for any given b can be laid out in a 2D grid with constant wire lengths and without increasing the area required or the cross-section density. The wire lengths for the three cases are 4, 4, and 5, respectively, which are optimal under the assumption of synchronous mapping. These constants are sufficiently small, therefore diminishing the need for the investigation of asynchronous mapping (i.e., moving individual nodes). Together with the other positive characteristics of midimew networks, these small constant wire lengths make midimew an attractive candidate for network interconnection in real situations. The layout procedures we have derived can be easily converted into a set of formulas for computing the final grid positions of the nodes of a given midimew, which should be useful to the system implementor. Moreover, the techniques developed in this paper should be applicable to other mesh-based structures as well—for example, the torus, for which ring-shuffle alone can produce an optimal layout (which is used in the Cray T3D) because of the simple structure of torus as compared with midimew.

Appendix

Basic Structure of Midimews

Given a number of nodes N, $N > 2$, the following dimension parameters determine the overall structure of a midimew network which is as shown in Fig. 14 in which a filled circle represents a node, and an empty circle an unused slot on the grid.

$$b = \left\lceil \frac{N}{2} \right\rceil$$

$$r = \left\lceil \frac{N}{b} \right\rceil b - N, \quad 0 \le r \le b$$

$$h = b + r = \left(b \left\lceil \frac{N}{b} + b \right\rceil \right) - N$$

$$v = \left\lceil \frac{N}{b} \right\rceil - r = N - (b-1) \left\lceil \frac{N}{b} \right\rceil$$

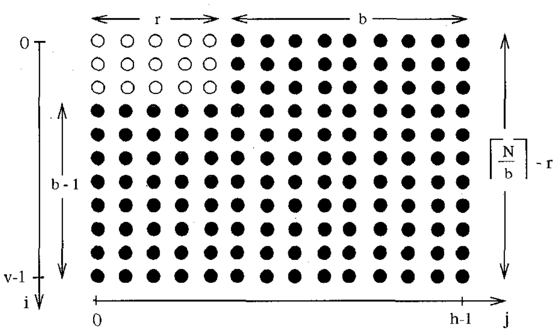


Fig. 14. The dimensions of midimew.

Fig. 15 shows how the structure of Fig. 14 comes about: Fig. 15a shows an approximately $b \times \frac{N}{b}$ mesh with r missing nodes; from Fig. 15a to Fig. 15b, the r empty slots are moved to the side, and rotated upward; then, the entire shaded block, together with the bottom row of filled circles, is tilted upward and placed alongside the main block. After this, the connections are made, which is according to the following two rules:

RULE 1. Vertically each bottom node $(v - 1, j)$ is connected to the top node of the column $(j + r) \bmod h$; that is, there is a deviation of r for vertical wrap-around links.

RULE 2. Horizontally each leftmost node $(i, 0)$ is connected to the rightmost node of the row $(i + b - 1) \bmod v$; that is, there is a deviation of $b - 1$ for horizontal wrap-around links.

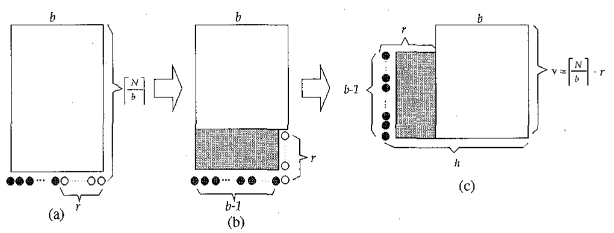


Fig. 15. Laying out the nodes.

When N is a multiple of b, i.e., $r = 0$, the midimew network is rectangular, such as the one in Fig. 3a for which $N = 66$, $b = 6$, $r = 0$, $h = 6$, and $v = 2b - 1 = 11$. For additional properties, please refer to [4].

ACKNOWLEDGMENTS

We thank the referees for their detailed comments, and Prof. Lionel Ni for his many useful suggestions.

REFERENCES

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

[2] B.W. Arden and K.W. Tang, "Representations and Routing for Cayley Graphs, "*IEEE Trans. Comm.*, vol. 39, no. 11, pp. 1,533–1,537, Nov. 1991.

[3] R. Beigel and C.P. Kruskal, "Processor Networks and Interconnection Networks without Long Wires," *Proc. First ACM Symp. Parallel Algorithms and Architectures*, pp. 42–51, 1989.

[4] R. Beivide, E. Herrada, J.L. Balcázar, and A. Arruabarrena, "Optimal Distance Networks of Low Degree for Parallel Computers," *IEEE Trans. Computers*, vol. 40, no. 10, pp. 1,109–1,124, Oct. 1991.

[5] F.T. Boesch and R. Tindell, "Circulants and Their Connectivities," *J. Graph Theory*, vol. 8, pp. 487–499, 1984.

[6] H. Braun and F.C. Stephan, "On Optimizing Diameter and Average Distance of Directed Interconnected Networks," *IEEE Trans. Computers*, vol. 42, no. 3, pp. 353–358, Mar. 1993.

[7] L. Campbell et al., "Small Diameter Symmetric Networks from Linear Groups," *IEEE Trans. Computers*, vol. 41, no. pp. 2, 218–220, Feb. 1992.

[8] W.J. Dally, "Wire Efficient VLSI Multiprocessor Communication Networks," *Proc. Stanford Conf. Advanced Research in VLSI*, pp. 391–415, 1987.

[9] W.J. Dally, "Performance Analysis of *k*-ary *n*-cube Interconnection Networks," *IEEE Trans. Computers*, vol. 39, no. 6, pp. 775–785, June 1990.

[10] Y. Feldman and E. Shapiro, "Spatial Machines: A More Realistic Approach to Parallel Computation," *CACM*, vol. 35, no. 10, pp. 61–73, Oct. 1992.

[11] G.A. Gessesse and S. Chalasani, "New Degree Four Networks: Properties and Performance," *Proc. Seventh Int'l Parallel Processing Symp.*, pp. 168–172, Apr. 1993.

[12] V. Hatz, "Interconnection Networks Based on Block Designs," *Proc. CONPAR 92–VAPP V*, pp. 37–42, Sept. 1992.

[13] W.D. Hillis, *The Connection Machine*. MIT Press, 1985.

[14] L.M. Ni and P.K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," *Computer*, vol. 26, pp. 62–76, Feb. 1993.

[15] M.S. Paterson, W.L. Ruzzo, and L. Snyder, "Bounds on Minimax Edge Length for Complete Binary Trees," *Proc. 13th ACM Symp. Theory of Computing*, pp. 293–299, 1981.

[16] G. Ramanathan and J. Oren, "Survey of Commercial Parallel Machines," *ACM Computer Architecture News*, vol. 21, no. 3, pp. 13–33, June 1993.

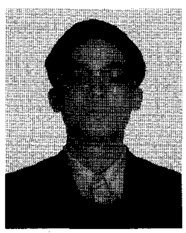
[17] A.G. Ranade and S.L. Johnsson, "The Communication Efficiency of Meshes, Boolean Cubes and Cube Connected Cycles for Wafer Scale Integration," *Proc. 1987 Int'l Conf. Parallel Processing*, pp. 479–482, 1987.

[18] C.H. Sequin, "Doubly Twisted Torus Networks for VLSI Processor Arrays," *Proc. Eighth Ann. Symp. Computer Architecture*, pp. 471–480, 1981.

[19] C.D. Thompson, "Area-Time Complexity for VLSI," *Proc. 11th Ann. Symp. Theory of Computing*, pp. 81–88, 1979.

[20] A. Varma and C.S. Raghavendra, *Interconnection Networks for Multiprocessors and Multicomputers: Theory and Practice.* IEEE C. S. Press, 1994.

[21] P.M.B. Vitányi, "Locality, Communication, and Interconnect Length in Multicomputers," *SIAM J. Computing*, vol. 17, no. 4, pp. 659–672, Aug. 1988.



Francis C.M. Lau (S'78-M'87) has a BSc degree from Acadia University, Canada, and an MMath and a PhD degree from the University of Waterloo, Canada. He joined the Department of Computer Science at the University of Hong Kong in 1987 where he is now an associate professor. His research interests are in parallel and distributed computing, object-oriented programming, and operating systems.

Guihai Chen received a BS degree in computer science from Nanjing University, China, in 1984, and an ME degree in computer engineering from Southeast University, China, in 1987. From 1987 to 1992, he was a lecturer at Hohai University, China. Currently, he is pursuing a PhD degree in computer science at the University of Hong Kong. His research interests include interconnection networks, VLSI layouts, high-performance computer architectures, parallel algorithms, and database management.