On the VLSI Area and Bisection Width of Star Graphs and Hierarchical Cubic Networks

Chi-Hsiang Yeh
Dept. of Electrical & Computer Engineering
Queen's University
Kingston, Ontario, K7L 3N6, Canada

Behrooz Parhami
Dept. of Electrical & Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA

Abstract

We solve an open question posed by Akers and Krishnamurthy in 1986 [1, 3] concerning VLSI layout of star graphs. We show that the area of the optimal layout of an N-node star graph, hierarchical cubic network (HCN), or hierarchical folded-hypercube network (HFN) is $N^2/16 \pm o(N^2)$ *under the Thompson model, or under the extended grid model where a node occupies a rectangle of sides that may range from* $n-1$ *to* $o(\sqrt{N})$ *for the n-star,* $\log_2 N + 1$ *to* $o(\sqrt{N})$ *for the HCN, and* $\log_2 N + 2$ *to* $o(\sqrt{N})$ *for the HFN. An n-dimensional star graph thus requires less area than any possible layout of a similar-size hypercube, but more than that of the much smaller n-cube. We also derive multilayer layout for star graphs that has area* $\frac{N^2}{8\lfloor L^2/2 \rfloor} \pm o(\frac{N^2}{L^2})$, *where a node occupies a rectangle of sides ranging from* $\lceil \frac{n-1}{4} \rceil$ *to* $o(\sqrt{N}/L)$ *and the number L of wiring layers satisfies* $2 \leq L = o(\sqrt{N}/n)$. *Finally we show that the bisection width of an N-node star graph is* $N/4 \pm o(N)$ *and the bisection width of an HCN or HFN is exactly* $N/4$.

1. Introduction

Star graphs [2] are a class of Cayley graphs that possess various desirable properties, such as vertex and edge symmetry, strongly hierarchical structure, maximal fault tolerance, as well as diameter, average distance, and node degree that are smaller than those of similar-size hypercubes. Various important issues, such as the development of algorithms and embeddings, have been intensely investigated for star graphs [6, 5, 7, 14, 21, 25], while VLSI layout (another aspect with important implication to the cost-performance of interconnection networks [10, 11, 12, 19, 22, 26, 27, 29]), has not been fully resolved for star graphs. In particular, the following open question was posed by Akers and Krishnamurthy [3, p. 565]: "Can the n-star graphs be laid out at least as efficiently as the comparable n-cubes?"

This question was partially answered by Sýkora and Vrt'o [22], who provided a layout for an N-node star graph that has an area of $4.5N^2$, which is within a constant factor from the $0.\bar{4}N^2$ area of a similar-size hypercube [27, 28]. However, its leading constant is larger than that of the hypercube layout area by a factor of 10.125. Moreover, the lower bound given in [22] for the area of the star graph is smaller than that of a similar-size hypercube [27, 28] by a factor of $348.\bar{4}$. Therefore, although the proofs in [22] show that the layout area of a star graph is within a constant factor from that of a similar-size hypercube, and the layout area of an n-star is asymptotically larger than that of an n-cube, the following questions remained unanswered:

(1) Can a star graph be laid out in a smaller area than a similar-size hypercube?

(2) Can an n-star be laid out at least as efficiently as an n-cube for networks of practical size?

In this paper, we completely resolve the open question posed in [1, 2, 3]. We accomplish this by providing an optimal layout for the star graph that has an area of $N^2/16 + o(N^2)$ under the Thompson model, a result that is $7.\bar{1}$ times smaller than the layout area of a similar-size hypercube [27, 28]. Moreover, we derive a tight lower bound on the layout area of star graphs and show that our obtained area is within a factor of $1 + o(1)$ from this lower bound. Thus, the area of a star graph is smaller than that of any possible layout of a similar-size hypercube, but cannot be smaller than that by a factor larger than $7.\bar{1} + o(1)$. Therefore, even for networks of practical size, the area of an $n!$-node n-star cannot be smaller than or comparable to that of a 2^n-node n-cube. We extend the VLSI layout model, and show that the area of the optimal layout of a star graph is also $N^2/16 \pm o(N^2)$ under the extended grid model for nodes that occupy a rectangle of sides ranging from $n-1$ to $o(\sqrt{N})$. We further show that the area of the optimal *X*-*Y* layout of a star graph is $\frac{N^2}{8\lfloor L^2/2 \rfloor} \pm o(\frac{N^2}{L^2})$ under the multilayer grid model, if a node occupies a cuboid with sectional area at least $2n-2$ for even depth h or $2n - 2 + \frac{2n-2}{h-1}$ for odd h and rectangle sides $o(\sqrt{N}/L)$, and the number L of wiring layers satisfies $2 \leq L = o(\sqrt{N}/n)$.

An N-node hierarchical cubic network (HCN) [15] (or hierarchical folded-hypercube network (HFN) [13]) is a 2-level network consisting of $\sqrt{N}$ $\frac{\log_2 N}{2}$-dimensional hypercube (or folded-hypercube) clusters, each having a link connecting it to each of the other clusters. If we view each

0-7695-0990-8/01/$10.00 (C) 2001 IEEE

cluster of an HCN or HFN as a supernode, then an HCN or HFN becomes a $\sqrt{N}$-node complete graph. Since complete graphs are the densest graphs among all networks, it was previously considered difficult and/or expensive to lay out networks derived from complete graphs. Based on the 2-D layout of complete graphs, we show that HCNs and HFNs can be laid out in $N^2/16 + o(N^2)$ area under the Thompson model or under the extended grid model where a node occupies a rectangle of sides that may range from $\log_2 N + 1$ to $o(\sqrt{N})$ for HCNs and from $\log_2 N + 2$ to $o(\sqrt{N})$ for HFNs. This result is smaller than the area of hypercube layouts [28] by a factor of $7.\bar{1}$, thus establishing that HCNs and HFNs can also be laid out more efficiently than hypercubes of the same size. We then show that these layouts are optimal within a factor of $1 + o(1)$ from the corresponding bisection-based lower bounds.

The bisection width is an important topological parameter for interconnection networks and is crucial to their cost and performance [9, 18]. However, this parameter is quite difficult to obtain for many networks. Upper bounds on the bisection widths of HCNs without diameter links and HFNs are available in [16], but exact values or lower bounds on the bisection widths of these networks were not known. In this paper we explore the relationships between VLSI layout area, bisection width, and the time to perform several total exchange tasks, which lead to the derivation of bisection widths for several interconnection networks. In particular, we show that the bisection width of an N-node star graph is $N/4 \pm o(N)$ and the bisection width of an HCN or HFN is exactly $N/4$.

2. Upper bounds on VLSI layouts

In this section, we present optimal layouts for complete graphs, star graphs, HCNs, and HFNs.

2.1. The Thompson and the extended grid models

In this subsection we present the Thompson model [23, 24] and introduce a variant to be referred to as the extended grid model.

In the *Thompson model*, a network is viewed as a graph whose nodes correspond to processing elements and edges correspond to wires. The graph is then embedded in a 2-D grid, where wires have unit width and a node of degree d occupies a square of side d. The wires can run either horizontally or vertically along grid lines; they can cross each other but cannot overlap with each other (including "bending" at the same grid point, which is allowed in another VLSI model called the *knock-knee model*). The area of a layout is the area of the smallest upright rectangle that contains all the nodes and wires. When there are two layers of wires, it is guaranteed that we can lay out the network within the area.

The *extended grid model* is almost the same as the Thompson model, except that a node may occupy an area considerably larger than d^2. A reason for proposing this extension is that a node in a parallel computer may include a large memory bank and one or several (complicated) processors, in addition to a router/switch, which may require considerably more than d^2 unit of area, where a unit of length is taken as the physical width for a wire. The range of actual node sizes must be specified explicitly in the extended grid model, which is usually taken to be between the minimum size required to implement a node (e.g., a square of side d or $d/4$, for a degree-d node in some technologies) and the maximum size allowed without affecting the leading constants for area, volume, and maximum wire length.

2.2. Optimal layouts for complete graphs

In this subsection we derive the layout area for complete graphs, which will then be used to derive optimal layout areas for star graphs, HCNs, and HFNs.

Lemma 2.1 *An N-node complete graph can be laid out in $N^4/16 + o(N^4)$ area under the Thompson model, or under the extended grid model where a node occupies a rectangle of sides that may range from $\lceil \frac{N-1}{4} \rceil$ to $o(N^{1.5})$.*

Proof: We first present the collinear layout of an m-node complete graph K_m by placing the m nodes, labeled 1 through m, along a row. Let a link be *type-i* if it connects two nodes whose addresses differ by i. Then the $m(m-1)/2$ links in K_m can be classified into types $1, 2, 3, \ldots, m-1$, and there are $m - i$ type-i links. To lay out K_m, we place all the type-1 links in one track, all the type-2 links in two tracks, where links connecting nodes with odd addresses are put in one track and links connecting nodes with even addresses in the other, and then all the type-i links in $\min(i, m-i)$ tracks for $i = 3, 4, 5, \ldots, m-1$. More precisely, type-i links are placed in i tracks if $i \le m/2$, where links are put in the same track if the remainders of their node-addresses modulo i are the same, and each of the $m - i$ type-i links is placed in a different track if $i > m/2$, It can be seen that such an arrangement will not result in overlapped links within a track. More details concerning the collinear layout of complete graphs found in [27, 31].

The total number of tracks in the layout described above is equal to

$$\sum_{i=1}^{m-1} \min(i, m-i) = \sum_{i=1}^{\lfloor m/2 \rfloor} i + \sum_{i=\lfloor m/2 \rfloor + 1}^{m-1} (m-i)$$

$$= \sum_{i=1}^{\lfloor m/2 \rfloor} i + \sum_{i=1}^{\lceil m/2 \rceil - 1} i = \lfloor m^2/4 \rfloor.$$

Since the bisection width of K_m is equal to $m^2/4$ when m is even and $(m^2-1)/4$ when m is odd, this layout is strictly optimal in terms of the number of tracks for collinear layout of complete graphs. This upper bound is 25% smaller than the one given in [11]. The above number of tracks leads to an area of $m(m-1)\lfloor m^2/4 \rfloor \approx m^4/4$.

Although the preceding method leads to the smallest possible number of tracks for the collinear layout of a complete graph, a smaller area can be achieved using 2-D layouts. Based on the collinear layout, we first derive an area-efficient layout for directed complete graphs, where each pair of nodes are connected by two directed edges. Without loss of generality, we assume that $m = m_1 \times m_2$ for some pair of integers $m_1, m_2 = \Theta(\sqrt{m})$.

To obtain an area-optimal layout, we place the m nodes of the complete graph, which we can label as (i, j) for $i =$

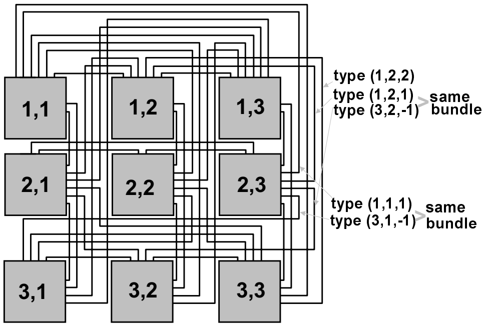


Figure 1. A 2-D layout for an undirected complete graph K_9.

$1, 2, \ldots, m_1$, $j = 1, 2, \ldots, m_2$, on an $m_1 \times m_2$ grid. Two neighboring rows are separated by $2m_1\lfloor m_2^2/4\rfloor$ tracks, while two neighboring columns are separated by $2m_2\lfloor m_1^2/4\rfloor$ tracks. We call a link connecting source node (i_1, j_1) to destination node (i_2, j_2) a *type*-$(i_1, j_1, j_2 - j_1)$ link. If $i_1 = i_2$ or $j_1 = j_2$, we can route the link as in the collinear layout. Otherwise, we first route it from the source node to the vicinity of the upper right corner of the *turning node* (i_1, j_2) along a horizontal track, and from there to the destination node (i_2, j_2) along a vertical track. Recall that we need $\min(k, m_2 - k)$ tracks for all the $m_2 - k$ type-k links in the collinear layout of an undirected K_{m_2}. Since m_1 links go from node (i_1, j_1) to the vicinity of node (i_1, j_2), which serves as the turning or destination node, and an equal number of links go in the reverse direction, we can replace each track in the collinear layout of a K_{m_2} with $2m_1$ tracks to accommodate the horizontal segments of the $2m_1$ directed links, leading to a total of $2m_1\lfloor m_2^2/4\rfloor$ tracks that have to be placed above each row of nodes.

We next show that the vertical segments of all the links to the immediate right of a column can be placed in $2m_2\lfloor m_1^2/4\rfloor$ tracks. We present a possible arrangement as follows. We place all the type-(x, y, z) links within the *bundle* (j_1, k), if $y = j_1$ and $z = \pm k$ for some positive integers j_1 and k, for all integers $x = 1, 2, 3, \ldots, m_1$. In other words, links are put within the same bundle if their source nodes belong to the same column j_1 and the difference between the source and destination row numbers is the same (i.e., equal to k or $-k$). There are $m_2(m_1 - 1)$ bundles between two columns. Bundle (j_1, k) can be laid out using $2\min(k, m_1 - k)$ successive tracks, which is similar to the layout for two groups of type-k links in the collinear layout of a K_{m_1}. More precisely, a link of type (i_1, j_1, k) is placed in the first half of the bundle to which it belongs if $\lfloor (i_1 - 1)/k\rfloor$ is even, and placed in the second half otherwise. Within the same half of the bundle, a link is placed in the l^{th} track if $i_1 \bmod k = l$. Note that we place the vertical segments of links of types $(x, j_1, -k)$, (x, j_1, k), $(x+2k, j_1, -k)$, $(x+2k, j_1, k)$, and $(x+4k, j_1, -k)$, $\ldots$ along a vertical track when $k \le m_1/2$ to avoid overlapping. By arranging links according to the above rules, the vertical segments of the $2m_2(m_1 - k)$ links of type (x, y, k) and type $(x, y, -k)$, $x = 1, 2, 3, \ldots, m_1$, $y = 1, 2, 3, \ldots, m_2$, can be placed in $2m_2\min(k, m_1 - k)$ tracks. As a result, the number of vertical tracks required is equal to $2m_2$ times the number $\lfloor m_1^2/4\rfloor$ of tracks required for the collinear layout of a K_{m_1}, for a total of $2m_2\lfloor m_1^2/4\rfloor$ links.

Since a node occupies a square of side $m_1m_2 - 1$, the area required for the 2-D layout of the directed K_m is given by

$$m_1(2m_1\lfloor m_2^2/4\rfloor + m_1m_2 - 1) \times m_2(2m_2\lfloor m_1^2/4\rfloor + m_1m_2 - 1)$$

$$= m^4/4 + O(m^{3.5}).$$

For an undirected K_m, where each pair of nodes are connected by a single edge, the required area can be reduced to $m^4/16 + O(m^{3.5})$ by properly removing half of the tracks in both the horizontal and vertical directions. A possible method is to remove the links within the second half of each of the bundles (and their horizontal segments) as well as half of the links whose sources and destinations have the same row or column numbers. Figure 1 shows the resultant 2-D layout for an undirected K_9. Note that there were originally 12 tracks between 2 neighboring rows or columns in the layout of a directed K_9, but after the removal of the second halves of bundles, there are only 6 vertical tracks left between neighboring columns, and only 10, 2, and 6 horizontal tracks left above the 1st, 2nd, and 3rd rows, respectively. □

As will be shown in Section 3, this layout area is larger than the lower bounds by a factor of $1 + o(1)$ and is thus quite close to being strictly optimal.

2.3. Optimal layouts for star graphs

An n-dimensional star graph, n-star, is a symmetric graph that has $N = n!$ nodes of degree $n - 1$ [2, 3]. Each node in an n-star is assigned a label, which is a distinct permutation of the set of n symbols $\{1, 2, 3, \ldots, n\}$. Two nodes are connected by a dimension-i link, $2 \le i \le n$, if and only if the label of one node can be obtained from the other by interchanging the first symbol and the i^{th} symbol.

Lemma 2.2 *An n-dimensional star graph, with $N = n!$ nodes, can be laid out in $N^2/16 + o(N^2)$ area under the Thompson model, or under the extended grid model where a node occupies a rectangle of sides that may range from $\lceil \frac{n-1}{4}\rceil$ to $o(\sqrt{N})$.*

Proof: An n-star contains n disjoint $(n-1)$-stars as subgraphs, each pair of which are connected by $(n-2)!$ links. We can view an $(n-1)$-star as a supernode, and then the n-star becomes an n-supernode complete graph with multiple edges. Top-level views for a 6-star and its 5-star supernodes and the complete structures of a 4-star and a 3-star are illustrated in Fig. 2.

To lay out an n-star, we first place nodes belonging to the same $(n-1)$-star subgraph within a block. We refer to these blocks as *level-n blocks*, and the segments of dimension-n links outside these level-n blocks as *level-n inter-block links*. We can then lay out these level-n inter-block links using the preceding layout for a complete graph with multiple edges. We will eventually connect each of the level-n inter-block

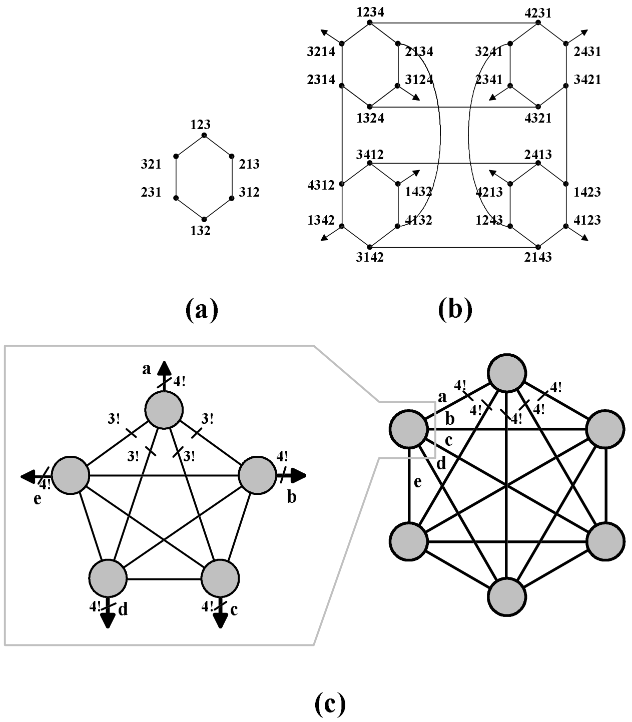


Figure 2. Structure of a (a) 3-star, (b) 4-star, and (c) top view of a 6-star and one of its 5-star subgraphs. Each pair of the 5-star subgraphs within the 6-star is connected by 4! dimension-6 links.

links to a certain node within the block, in order to lay out the corresponding dimension-n link completely. We continue to lay out the $(n-1)$-star within each of the level-n blocks We place nodes belonging to the same $(n-2)$-star subgraph within a level-$(n-1)$ block, and lay out level-$(n-1)$ inter-block links (i.e., the segments of dimension-$(n-1)$ links outside level-$(n-1)$ blocks) using a complete graph layout with multiple edges. We will eventually connect each of the links incident to the level-$(n-1)$ block (including dimension-n links and dimension-$(n-1)$ links) to a certain node within the level-$(n-1)$ block. This process can be done recursively until the number of nodes within a block to be laid out is small. Then we use any viable method to lay out these small substars.

Let $n_1 = \lceil \sqrt{n} \rceil$ and $n_2 = \lceil n/n_1 \rceil$. To lay out all the level-n inter-block links, $N^2/16 + o(N^2)$ area will be required based on the layout of a $K_{n_1 n_2}$ with $(n-2)!$ edges between each pair of nodes, calculated as follows. Recall that the grid layout for a $K_{n_1 n_2}$ with 2 edges between each pair of nodes requires $n^4/4 + o(n^4)$ area. As a result, a K_n with $(n-2)!$ edges between each pair of nodes can be laid out in $(n^2(n-2)!)^2/16 + o(n^2(n-2)!)^2 = N^2/16 + o(N^2)$ area, which can be easily done be expanding each side-$(2n-2)$ node into a side-$(n-1)!$ node and replicating each link into $(n-2)!/2$ links.

To lay out a level-i block, $i = n-1, n-2, n-3, \ldots, l$, we first connect the wires outside the block to appropriate level-$(i-1)$ blocks within it, and then lay out the level-i inter-block links of the i-star within the level-i block using the layout for $K_{i_1 i_2}$, where $i_1 = \lceil \sqrt{i} \rceil$ and $i_2 = \lceil i/i_1 \rceil$. Note that $i_1 i_2 - i$ out of the $i_1 i_2$ level-i blocks can be removed since there are only i $(i-1)$-stars within a block. The number l can be any integer smaller than n and greater than 3. In what follows, we assume $l = O(1)$. The area for a level-l block is $O(n^2)$ since there are only $O(1)$ nodes in it and $O(n)$ links incident to it.

The segments of dimension-n links inside a level-n block and all the dimension-i links, $i = n-1, n-2, ..., l$, inside the block require a square of side $O(N/n)$ for wiring them. Since the network nodes are arranged as a 2-D grid, the side of a level-n block will be $o(N)$, which does not increase the leading constant of the total area, as long as a node occupies a rectangle of side $o(\sqrt{N})$. As a result, the layout area for an n-star is $N^2/16 + o(N^2)$, which is mainly occupied by dimension-n links. □

This layout area is smaller than the one given in [22] by a factor of 72 and is optimal within a factor of $1 + o(1)$, as will be seen from the lower bound derived in Section 3. Similar results can be obtained for pancake graphs and bubble-sort graphs [3].

2.4. Multilayer layout for star graphs

In the multilayer 2-D grid model [31, 30], a network is viewed as a graph whose nodes correspond to processing elements and edges correspond to communication links. The nodes and edges of the graph are then embedded in a 3-D grid, where edges have unit width, can run along grid lines, but cannot cross or overlap with each other (i.e., the paths for embedding these edges must be edge- and node-disjoint). The nodes of the graph are embedded in the 2-D grid of the first layer (i.e., $z = 1$). As in the extended grid model, the range of actual node sizes must be specified explicitly in this model. As an example, a node that occupies a square of side $w = o(\sqrt{N/\log N})$ and depth $h = 2$ will be mapped to $w^2 h = o(N/\log N)$ neighboring grid points at the first two layers. The sectional area for the cuboid occupied by a node is $wh = o(\sqrt{N/\log N})$.

The area A of a layout is defined as the area of the smallest upright rectangle along the x-y directions that contains all the nodes and links. The volume of a layout is equal to the number L of layers times its area A. A network with area A' under the Thompson model or the extended grid model can always be laid out with area $A \leq A'$ under the multilayer 2-D grid model with $L = 2$ layers, so the former can be viewed as a special case of the latter. Note, however, that if we derive layouts directly for the two-layer 2-D grid model, we may be able to obtain layouts with area smaller than the best possible layout derived for the Thompson model or the extended grid model. The cost of a layout under the multilayer 2-D grid model is a function of A and L as well as other parameters.

Recall that there are $(n-2)!$ dimension-n links between any pair of level-n blocks. We partition these links into $L/2$ groups, each having at most $\lceil 2(n-2)!/L \rceil$ links. To lay out these links using L layers, we simply assign each group to a particular pair of layers and then simply lay out these dimension-n links based on the layout of a $K_{n_1 n_2}$ with $\lceil 2(n-2)!/L \rceil$ edges between each pair of nodes (for the segments outside level-n blocks). Similarly, dimension-$(n-1)$ links that are confined within level-n blocks can be laid out by first partitioning them in $L/2$ groups and then laying out each of the groups using a pair of neighboring wiring layers based on the layout of an $(n-1)$-node complete graph with $\lceil 2(n-3)!/L \rceil$ edges between each pair of nodes (for the segments outside level-$(n-1)$ blocks); dimension-i links that are confined within level-$(i-1)$ blocks, $i = n-2, n-3, ..., l+1, l = O(1)$, can also be laid out based on the layout of an i-node complete graph with $\lceil 2(i-2)!/L \rceil$ edges between each pair of nodes (for the segments outside level-i blocks). The segments of dimension-i links inside level-i blocks, $i = n, n-1, n-2, ..., l$, are laid out using the same pair of wiring layers that are used to lay out the segments outside level-i blocks, and only require negligible area as in the layout under the Thompson model.

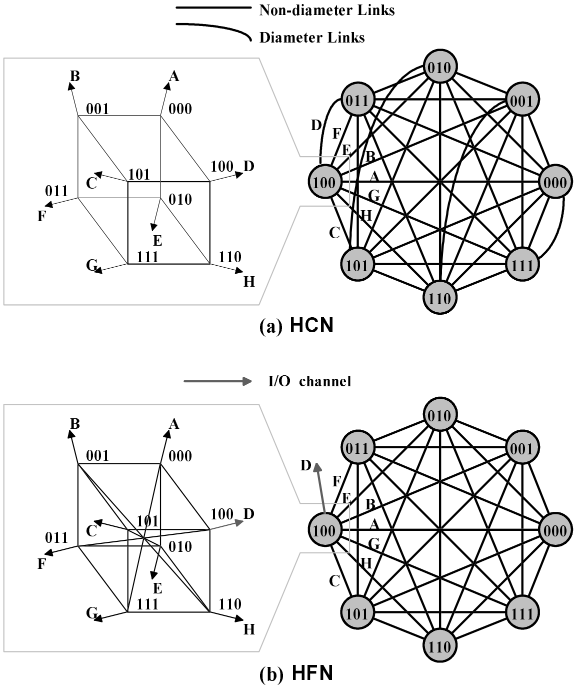
Similar to the star layout under the Thompson model, the dimension-n links dominate the layout area when L is not very large. The preceding layout method can be easily generalized to odd L using the technique in [31, 30], leading to the following lemma.

Lemma 2.3 *An n-star can be laid out in* $\frac{N^2}{4L^2} + o\left(\frac{N^2}{L^2}\right)$ *area for even L or in* $\frac{N^2}{4(L^2-1)} + o\left(\frac{N^2}{L^2}\right)$ *area for odd L under the multilayer 2-D grid model, where* $N = n!$, *a node occupies a rectangle of sides ranging from* $\lceil \frac{n-1}{4} \rceil$ *to* $o\left(\frac{\sqrt{N}}{L}\right)$ *with any depth* $h \le L$, *and the number L of wiring layers satisfies* $2 \le L = o\left(\frac{\sqrt{N}}{n}\right)$.

The preceding technique for laying out star graphs can be applied to various other networks as long as the network can be partitioned into clusters with multiple links connecting a pair of neighboring clusters. More details will be reported in the near future.

2.5. Optimal layouts for HCNs and HFNs

A hierarchical cubic network (HCN) [15] is a 2-level network consisting of $\sqrt{N}$ $(\log_2 N/2)$-dimensional hypercube clusters, each having an inter-cluster link connecting it to each of the other hypercube clusters. In addition to these links, the i^{th} hypercube cluster has a *diameter link* connecting it to the $(\sqrt{N} - i + 1)$-th hypercube, for $i = 1, 2, ..., \sqrt{N}$. The top view for a 64-node HCN is seen in Fig. 3a. A hierarchical folded-hypercube network (HFN) [13] is a 2-level network consisting of $\sqrt{N}$ $(\log_2 N/2)$-dimensional folded hypercubes, each having an *inter-cluster link* connecting it to each of the other folded hypercubes. The top view for a 64-node HFN is seen in Fig. 3b. Since clusters in an HCN or HFN are connected as a complete graph, they can be laid out using a method similar to that used for star graphs.



(a) HCN

(b) HFN

Figure 3. The 64-node (a) HCN and (b) HFN. The hypercube or folded-hypercube clusters are inter-connected as a complete graph.

Lemma 2.4 *An N-node HCN or HFN can be laid out using* $N^2/16 + o(N^2)$ *area under the Thompson model, or under the extended grid model where a node occupies a rectangle of sides that may range from* $\lceil \frac{\log_2 N + 1}{4} \rceil$ *to* $o(\sqrt{N})$ *for the HCN or from* $\lceil \frac{\log_2 N + 2}{4} \rceil$ *to* $o(\sqrt{N})$ *for the HFN.*

Proof: Let $m_1 = \lceil N^{1/4} \rceil$ and $m_2 = \lceil \sqrt{N}/m_1 \rceil$. To lay out an HFN, we first place nodes belonging to the same folded-hypercube cluster within a block of side $\sqrt{N} - 1$, and lay out the inter-cluster links based on the previous layout for an undirected $K_{m_1 m_2}$. Note that the last $m_1 m_2 - \sqrt{N}$ blocks and their associated links are not needed and can be removed. We will then lay out the folded hypercube within each of the blocks and connect each of the links incident to the block to a certain node within the block. These blocks will subsequently be expanded to an area larger than $(\sqrt{N} - 1)^2$ in order to accommodate the routing of the *intra-cluster links* within a folded-hypercube cluster.

We can lay out an HCN almost identically to the previous method for HFNs. The only difference is that an HCN has $\sqrt{N}/2$ additional diameter links, and the subgraph within a block is a hypercube.

We need area $(\sqrt{N})^4/16 + o(\sqrt{N})^4 = N^2/16 + o(N^2)$ to route the inter-cluster links of an HCN or an HFN. Since a $\sqrt{N}$-node hypercube or a folded hypercube can be laid out in a square of side $O(\sqrt{N})$ [27, 28], the expanded area will not affect the leading constant for the layout area. The result then follows, given that the diameter links for the HCN only imply $O(N\sqrt{N})$ additional area. □

As will be shown in Section 3, the above layouts for HCNs and HFNs are optimal within a factor of $1 + o(1)$.

3. Tight bounds on layout areas

In this section we introduce several lower-bound techniques and show that all the VLSI layouts derived in the previous section are optimal within a factor of $1+o(1)$.

3.1. Lower bounds based on BATT

A basic communication task that arises often in applications is the total exchange (TE) task [8, 17] (also called all-to-all personalized communication), where each node has to send a different (personalized) packet to every other node of the network. In this subsection we propose to use the *best achievable TE throughput (BATT)* as a convenient and powerful tool for deriving lower bounds on VLSI layout areas of interconnection networks.

In [23, 24], Thompson has shown that the layout area of a graph is at least B^2 under the Thompson model, where a node occupies a square of side d, B is the bisection width of the graph, and d is the degree of the node. The same lower bound can be obtained for the extended grid model.

Theorem 3.1 *The layout area of a network is at least B^2 under the extended grid model if a node occupies a rectangle of sides at least d, where B is the bisection width of the graph.*

Theorem 3.2 *Assume that $f(N)$ TE tasks can be executed in $f(N)T_{TE}$ communication steps in an N-node interconnection network for some integer function $f(N)$, under the all-port communication model. Then the layout area of the network is at least*

$$\frac{\lfloor N/2\rfloor^2 \times \lceil N/2\rceil^2}{T_{TE}^2} \approx \frac{N^4}{16T_{TE}^2}$$

under the Thompson model, or the extended grid model if a node occupies a rectangle of sides at least d.

Most interconnection networks of interest can be laid out efficiently using *multilayer X-Y layouts*, a class of layouts under the multilayer grid models, where odd-numbered layers implement the segments of links that are in the horizontal (X) direction and even-numbered layers implement the segments of links that are in the vertical (Y) direction. A layout under the Thompson model can always be transformed into a two-layer X-Y layout [24]. We can derive the following lower bound for X-Y layouts under the multilayer grid models.

Theorem 3.3 *The area for any X-Y layout of a network is at least*

$$\begin{cases} \frac{4B^2}{L^2} & \textit{for even L, or} \\ \frac{4B^2}{L^2-1} & \textit{for odd L} \end{cases}$$

under the multilayer grid models, with $L \geq 2$ wiring layers, if a node occupies a cuboid with sectional area at least $2d$ for even depth h or $2d+\frac{2d}{h-1}$ for odd h, where B is the bisection width of the graph.

Theorem 3.4 *Assume that $f(N)$ TE tasks can be executed in $f(N)T_{TE}$ communication steps in an N-node interconnection network for some integer function $f(N)$, under the all-port communication model. Then the area of any X-Y layout for the network with $L \geq 2$ wiring layers is at least*

$$\begin{cases} \frac{4\lfloor N/2\rfloor^2\times\lceil N/2\rceil^2}{L^2T_{TE}^2} \approx \frac{N^4}{4L^2T_{TE}^2} & \textit{for even L, or} \\ \frac{4\lfloor N/2\rfloor^2\times\lceil N/2\rceil^2}{(L^2-1)T_{TE}^2} \approx \frac{N^4}{4(L^2-1)T_{TE}^2} & \textit{for odd L} \end{cases}$$

under the multilayer grid model if a node occupies a cuboid with sectional area at least $2d$ for even depth h or $2d+\frac{2d}{h-1}$ for odd h.

Note that a node in Theorems 3.1, 3.2, 3.3, and 3.4 is allowed to be very large (e.g., as large as a square with side $\sqrt{N}/\log_2 N$), without reducing the lower bound. Also, in performing the TE tasks, we assume that links are bidirectional and that nodes can have infinitely large routing tables and buffer space and perform infinitely many computation steps if required. (Links are still assumed to have unit width in the layout.)

Similar to BAR^2 technique, we can also use the *best achievable unicast throughput (BAUT)* to derive area lower bounds, which yields the highest possible injection rate for unicast routing with uniformly distributed sources and destinations after balancing the traffic over all network nodes and links. The proposed BATT and BAUT techniques are both following the idea of the lower-bound technique proposed by Thompson that derives area lower bounds based on the time complexity for performing fast Fourier transform (FFT) [23]. However, BATT and BAUT are different in that more than one instance of the communication tasks are executed at the same time to simplify the derivation of the best achievable throughput or even to improve the throughput to a degree that cannot be achieved by using a single (possibly pipelined) instance. Also, we found that the TE tasks and unicast routing are more convenient and/or achieve better bounds compared to FFT and other communication tasks such as sorting; these communication tasks have not been previously used for this purpose or in this form. The BATT and BAUT techniques are closely related to the method that derives area lower bounds by embedding multiple complete graphs, in contrast to embedding a single complete graph as in most previous papers.

3.2. Minimal layout areas of the complete graph, star graph, HCN, and HFN

The minimal layout of a network is the layout(s) of the network that requires the smallest possible area. Except when a strictly optimal layout has been obtained, we specify the range for the possible minimal layout area of a network by giving the best-known lower bound on any possible layouts of the network and the the best-known area for the network (i.e., either the area of a layout with explicit construction or the area of a layout whose existence has been proven). This range is useful in indicating the cost-performance of the network. Our goal is to find optimal layouts for interconnection networks of interest, where an optimal layout is a layout of the network whose area has the same leading constant as its minimal layout. A layout whose area has the same order as its minimal layout is referred to as an *asymptotically optimal layout*. Based on the lower-bound techniques proposed in Subsection 3.1, we will show that all the VLSI layouts presented in Section 2 are optimal.

Theorem 3.5 *The area of the minimal layout of an N-node complete graph is $N^4/16 + o(N^4)$ under the Thompson model, or under the extended grid model if a node occupies a rectangle of sides that may range from $N-1$ to $o(N^{1.5})$. The number of tracks required for the minimal collinear layout of an N-node complete graph is $\lfloor N^2/4 \rfloor$.*

In [14], it is shown that TE can be performed on an N-node star graph in $2N + o(N)$ time under the all port communication model. This is the fastest algorithm for the TE task reported in the literature thus far. By simply using this TE execution time and Theorem 3.2, we can easily obtain an area lower bound that is 12.25 times better than the one given in [22].

From Theorem 3.2, however, we know that the throughput for executing TE tasks, rather than the completion time for a single TE task, is what counts for the lower bound on VLSI layout area. This greatly simplifies the problem and allow us to further improve the lower bound on the star-graph area by another factor of 4.

Lemma 3.6 *[27] $(n-1)$ TE tasks can be executed in $nN + o(nN)$ communication time in an n-star under the all-port communication model.*

By combining this result with Theorem 3.2, we can now obtain the minimal area for star graphs.

Theorem 3.7 *The area of the minimal layout of an n-dimensional star graph with $N = n!$ nodes is equal to $N^2/16 \pm o(N^2)$ under the Thompson model, or under the extended grid model if a node occupies a rectangle of sides that may range from $n-1$ to $o(\sqrt{N})$.*

The ratio of the upper bound over the lower bound for the layout area of a star graph under the Thompson model is significantly reduced from 3528 in [22] to $1 + o(1)$.

Theorem 3.8 *The area of the minimal X-Y layout of an N-node star graph is equal to*

$$\begin{cases} \frac{N^2}{4L^2} + o\left(\frac{N^2}{L^2}\right) & \text{for even } L\text{, or} \\ \frac{N^2}{4(L^2-1)} + o\left(\frac{N^2}{L^2}\right) & \text{for odd } L \end{cases}$$

under the multilayer 2-D grid model, if a node occupies a cuboid with sectional area at least $2d$ for even depth h or $2d + \frac{2d}{h-1}$ for odd h and rectangle sides $o(\sqrt{N})$, where the number L of wiring layers satisfies $2 \le L = o\left(\frac{\sqrt{N}}{n}\right)$.

These are the first bounds thus far for the area of multilayer star layouts, and are optimal within a factor of $1 + o(1)$.

Lemma 3.9 *[27] The throughput for executing TE tasks can be arbitrarily close to $1/N$ in an N-node HCN or HFN under the all-port communication model.*

By combining Lemma 3.9 with Theorem 3.2, we can prove that the layouts presented in Subsection 2.5 are also close to being strictly optimal.

Theorem 3.10 *The area of the minimal layout of an N-node HCN or HFN is equal to $N^2/16 + o(N^2)$ under the Thompson model, or under the extended grid model if a node occupies a rectangle of sides that may range from $\log_2 N + 1$ to $o(\sqrt{N})$ for the HCN and $\log_2 N + 2$ to $o(\sqrt{N})$ for the HFN.*

4. Bisection widths of the star graph and HCN

The bisection width of a network is the minimum number of links that have to be removed to partition the network into two (disjoint) halves. It is an important topological parameter for interconnection networks and is crucial to their cost and performance. However, this parameter is quite difficult to obtain for many networks. In what follows, we will show that tight bounds on the bisection width for star graphs, HCNs, and HFNs can be obtained from the upper bounds on their TE throughput and VLSI areas.

Theorem 4.1 *The bisection width of an N-node star graph is equal to $N/4 \pm o(N)$.*

Proof: If the bisection width is $N/4 + \varepsilon N$ for some positive constant ε, then any layout for a star graph would have area at least $(1/16 + 2\varepsilon)N^2$, which conflicts with Lemma 2.2. If the bisection width is $N/4 - \varepsilon N$ for some positive constant ε, then T_{TE} must be at least $N/(1-4\varepsilon)$, which conflicts with Lemma 3.6. So the bisection width of an N-node star graph must be $N/4 \pm o(N)$. Note that when n is odd, we cannot obtain the preceding upper bound by partitioning the n-star based on its $(n-1)$-star clusters (e.g., $\lceil n/2 \rceil$ $(n-1)$-substars on one side and $\lfloor n/2 \rfloor$ $(n-1)$-substars on the other), while partitioning it based on its $(n-2)$-star clusters will involve dimension-$(n-1)$ links. □

In the following theorem we derive the exact bisection width for HCNs and HFNs.

Theorem 4.2 *The bisection width of an N-node HCN or HFN is equal to $N/4$.*

Proof: We first derive a lower bound on the bisection width of an HCN or HFN based on the average throughput for performing a set of TE tasks in it. The total time for executing $f(N) = 10N$ TE tasks is at most $10N^2 + 2N$ (see Lemma 3.9 and [27]). Since

$$B \ge \frac{\lfloor N/2 \rfloor \times \lceil N/2 \rceil}{T_{TE}},$$

the bisection width B of an HFN must satisfy

$$B \ge \frac{N^2}{\frac{4(10N^2+2N)}{10N}} = \frac{N}{4} - 0.05 + \frac{0.04}{N+0.2} > \frac{N}{4} - 0.05.$$

From the definition of HCNs and HFNs, we know that the size N of an HCN or HFN is a multiple of 4 since the size of a hypercube or folded-hypercube cluster is an even integer $\sqrt{N}$. Since the bisection width B must be an integer, we have $B \ge \lceil N/4 - 0.05 \rceil = N/4$.

By separating the hypercube (or folded-hypercube) clusters in an HCN (or HFN, respectively) into two equal parts, each having $\sqrt{N}/2$ clusters, there will be exactly $N/4$ inter-cluster links connecting these two parts. This leads to an upper bound $B \leq N/4$ on the bisection width of an HFN. Thus, the bisection width of an N-node HFN is exactly $N/4$. If one of the two parts has clusters $0, 1, 2, ..., \sqrt{N}/4 - 1$ and clusters $3\sqrt{N}/4, 3\sqrt{N}/4 + 1, ..., \sqrt{N} - 1$, then all diameter links in an N-node HCN will be confined within the same side. Therefore, the bisection width of an HCN is also upper bounded by $N/4$, which is the exact bisection width. □

Bisection width is often used for obtaining lower bounds on the VLSI layout area for graphs and the lower bounds on various communication tasks [24]. We have essentially reversed this process, using upper bounds on the VLSI area and TE throughput, which are actually easier to find for these networks, to obtain exact values or tight bounds on their bisection widths.

5. Conclusion

We completely resolved an open question posed by Akers and Krishnamurthy [1, 3] by showing that a star graph does indeed have a more compact layout than a similar-size hypercube, while only by a constant factor of $7.\bar{1}$. We presented layouts for star graphs, complete graphs, HCNs, and HFNs that are optimal within a factor of $1 + o(1)$. We also derived the exact bisection widths of HCNs and HFNs, and tight bounds on the bisection width of star graphs. The techniques proposed in this paper can be applied to a wide variety of other interconnection networks [27].

References

[1] Akers, S.B. and B. Krishnamurthy, "A group theoretic model for symmetric interconnection networks," *Proc. Int'l Conf. Parallel Processing,* 1986, pp. 216-223.

[2] Akers, S.B., D. Harel, and B. Krishnamurthy, "The star graph: an attractive alternative to the n-cube," *Proc. Int'l Conf. Parallel Processing,* 1987, pp. 393-400.

[3] Akers, S.B. and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Trans. Comput.,* vol. 38, Apr. 1989, pp. 555-565.

[4] Akl, S.G. and K.A. Lyons, *Parallel Computational Geometry,* Prentice Hall, Englewood Cliffs, NJ, 1993.

[5] Al-Ayyoub, A.-E. and K. Day, "Matrix decomposition on the star graph," *IEEE Trans. Parallel and Distributed Systems,* Vol. 8, no. 8, Aug. 1997, pp. 803-812.

[6] Azevedo, M.M., "Star-based interconnection networks and fault-tolerant clock synchronization for large multicomputers," Ph.D. dissertation, Dept. Electrical & Comp. Eng., Univ. of California, Irvine, 1997.

[7] Bagherzadeh, N., M. Dowd, and S. Latifi, "A well-behaved enumeration of star graphs," *IEEE Trans. Parallel Distrib. Sys.,* vol. 6, no. 5, May 1995, pp. 531-535.

[8] Bertsekas, D.P. and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods,* Athena Scientific, 1997.

[9] Bornstein, C., A. Litman, B.M. Maggs, R.K. Sitaraman, and T. Yatzkar, On the bisection width and expansion of butterfly networks," *Proc. 12th International Parallel Processing Symposium & 9th Symposium on Parallel and Distributed Processing*, Apr. 1998, pp. 144-150.

[10] Calamoneri T. and R. Petreschi, "Optimal layout of trivalent Cayley interconnection networks," *Int'l J. Foundations of Computer Science*, Vol. 10, no. 3, pp. 277-287, 1999.

[11] Chen, C. and D.P. Agrawal, "dBCube: a new class of hierarchical multiprocessor interconnection networks with area efficient layout," *IEEE Trans. Parallel Distrib. Sys.,* vol. 4, no. 12, Dec. 1993, pp. 1332-1344.

[12] Even, S., S. Muthukrishnan, M.S. Paterson, and S. Cenk Sahinalp, "Layout of the Batcher bitonic sorter," *Proc. ACM Symp. Parallel Algorithms and Architectures,* 1998, pp. 172-181.

[13] Duh, D., G. Chen, and J. Fang, "Algorithms and properties of a new two-level network with folded hypercubes as basic modules," *IEEE Trans. Parallel Distrib. Sys.,* vol. 6, no. 7, Jul. 1995, pp. 714-723.

[14] Fragopoulou, P. and S.G. Akl, "Optimal communication algorithms on star graphs using spanning tree constructions," *J. Parallel Distrib. Computing.,* vol. 24, 1995, pp. 55-71.

[15] Ghose, K. and R. Desai, "Hierarchical cubic networks," *IEEE Trans. Parallel Distrib. Sys.,* vol. 6, no. 4, Apr. 1995, pp. 427-435.

[16] Hamdi, M., "Communication-efficient interconnection networks for parallel computations," Ph.D. dissertation, Dept. Electrical Eng., Univ. of Pittsburgh, PA, 1991.

[17] Johnsson, S.L. and C.-T. Ho, "Optimum broadcasting and personalized communication in hypercubes," *IEEE Trans. Computers,* vol. 38, no. 9, Sep. 1989, pp. 1249-1268.

[18] Kalpakis, K. and Y. Yesha, "On the bisection width of the transposition network," *Networks,* vol. 29, no. 1, Jan. 1997, pp. 69-76.

[19] Leiserson, C.E., "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Trans. Computers*, vol. C-34, no. 10, Oct. 1985, pp. 892-901.

[20] Parhami, B., *Introduction to Parallel Processing: Algorithms and Architectures*, Plenum Press, 1999.

[21] Saikia, D.K., R. Badrinath, and R.K. Sen, "Embedding torus on the star graph," *IEEE Trans. Parallel and Distributed Systems,* Vol. 9, no. 7, Jul. 1998, pp. 650-663

[22] Sýkora O. and I. Vrt'o, "On VLSI layouts of the star graph and related networks," *Integration, VLSI J.* 1994, pp. 83-93.

[23] Thompson, C.D., "Area-time complexity for VLSI," *Proc. ACM Symp. Theory of Computing*, 1979, pp. 81-88.

[24] Thompson, C.D., "A complexity theory for VLSI," Ph.D. dissertation, Dept. of Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA, 1980.

[25] Tseng, Y.-C., S.-H. Chang, and J.-P. Sheu, "Fault-tolerant ring embedding in a star graph with both link and node failures," *IEEE Trans. Parallel and Distributed Systems,* Vol. 8, no. 12, Dec. 1997, pp. 1185-1195.

[26] Wise, D.S., "Compact layouts of banyan/FFT networks," *VLSI Systems and Computations,* Computer Science Press, 1981, pp. 186-195.

[27] Yeh, C.-H., "Efficient low-degree interconnection networks for parallel processing: topologies, algorithms, VLSI layouts, and fault tolerance," Ph.D. dissertation, Dept. Electrical & Computer Engineering, Univ. of California, Santa Barbara, Mar. 1998.

[28] Yeh, C.-H., E.A. Varvarigos, and B. Parhami, "Efficient VLSI layouts of hypercubic networks," *Proc. Symp. Frontiers of Massively Parallel Computation*, Feb. 1999, pp. 98-105.

[29] Yeh, C.-H., B. Parhami, and E.A. Varvarigos, "The recursive grid layout scheme for VLSI layout of hierarchical networks," *Proc. Merged Int'l Parallel Processing Symp. & Symp. Parallel and Distributed Processing*, Apr. 1999, pp. 441-445.

[30] Yeh, C.-H., E.A. Varvarigos, and B. Parhami, "Multilayer VLSI layout for interconnection networks," *Proc. Int'l Conf. Parallel Processing,* 2000, pp. 33-40,

[31] Yeh, C.-H., B. Parhami, E.A. Varvarigos, and H. Lee, "VLSI layout and packaging of butterfly networks," *Proc. ACM Symp. Parallel Algorithms and Architectures,* 2000, pp. 196-205.