Theory Comput. Systems **34**, 491–518 (2001)
DOI: 10.1007/s00224-001-1026-2

Theory of Computing Systems

On the Bisection Width and Expansion of Butterfly Networks*

C. F. Bornstein,[1,2] A. Litman,[3] B. M. Maggs,[4] R. K. Sitaraman,[2,5]
and T. Yatzkar[3]

[1]Instituto de Matemática, Universidade Federal do Rio de Janeiro,
COPPE/Sistemas,
Rio de Janeiro, Brazil
cfb@cos.ufrj.br

[2]Akamai Technologies, 500 Technology Square,
Cambridge, MA 02139, USA
ramesh@akamai.com

[3]Computer Science Department, Technion,
Technion City, Haifa 32000, Israel
{litman,ytali}@cs.technion.ac.il

[4]School of Computer Science, Carnegie Mellon University,
Pittsburgh, PA 15213, USA
bmm@cs.cmu.edu

[5]Department of Computer Science, University of Massachusetts,
Amherst, MA 01003, USA

Abstract. This paper proves tight bounds on the bisection width and expansion of butterfly networks with and without wraparound. We show that the bisection width of an n-input butterfly network is $2(\sqrt{2}-1)n + o(n) \approx 0.82n$ without wraparound, and n with wraparound. The former result is surprising, since it contradicts the prior "folklore" belief that the bisection width is n. We also show that every set of k nodes has at least $(k/(2\log k))(1-o(1))$ neighbors in a butterfly without wraparound, and at least $(k/\log k)(1-o(1))$ neighbors in a butterfly with wraparound, if k is $o(\sqrt{n})$ and $o(n)$, respectively.

* Claudson Bornstein is supported in part by CNPq under Grant 68.0031/99-5. Ami Litman is supported in part by BSF Grant No. 94-266. Bruce Maggs is supported in part by the National Science Foundation (NSF) under Grant CCR-9900304. Ramesh Sitaraman is supported in part by NSF CAREER Award CCR-9703017. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the NSF or the U.S. government.

1. Introduction

This paper analyzes the bisection width and expansion of a network called a *butterfly*. This network has been studied extensively and it, or one of its variants, has served as the routing network in several parallel computers and ATM switches. Surprisingly, however, the precise values of the butterfly's bisection width and expansion were not previously known. This paper proves upper and lower bounds on these parameters that are tight up to low-order additive terms.

1.1. *The Butterfly and Cube-Connected Cycles Networks*

Throughout this paper we use the following terminology to describe butterfly networks. The $(\log n)$-dimensional butterfly B_n has $N = n(\log n + 1)$ nodes arranged in $\log n + 1$ levels of n nodes each. (All logarithms in this paper are base 2.) Each node has a distinct label $\langle w, i\rangle$ where i is the level of the node $(0 \le i \le \log n)$ and w is a $(\log n)$-bit binary number drawn from $\{0, 1\}^{\log n}$ that denotes the *column* of the node. All nodes of the form $\langle w, i\rangle$, $0 \le i \le \log n$, are said to belong to column w. Similarly, the ith level L_i consists of all of the nodes $\langle w, i\rangle$, where w ranges over all $(\log n)$-bit binary numbers. For the purposes of this paper, the edges in the network are undirected. Two nodes $\langle w, i\rangle$ and $\langle w', i'\rangle$ are linked by an undirected edge if $i' = i + 1$ and either w and w' are identical or w and w' differ only in the bit in position i'. (The bit positions are numbered 1 through $\log n$, the most significant bit being numbered 1.) The nodes on level 0 are called the *input nodes* or just *inputs* of the network, and the nodes on level $\log n$ are called the *output nodes* or just *outputs*. The 32-node butterfly network B_8 ($N = 32$, $n = 8$, $\log n = 3$) is shown in Figure 1.

Sometimes the level 0 and $\log n$ nodes in each column are assumed to be the same node. In this case the butterfly is said to *wrap around* or to have *wraparound*. We use W_n to denote the $(\log n)$-dimensional butterfly with wraparound. This network has $n \log n$ nodes.

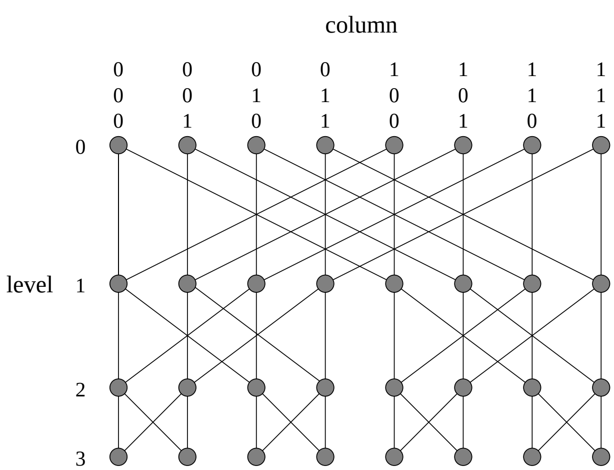

Fig. 1. The 32-node butterfly network B_8.

A number of properties of butterfly networks were known prior to our work. For example, it is not difficult to show that the diameter of B_n is $2 \log n$, and the diameter of W_n is $\lfloor (3 \log n)/2 \rfloor$, where the *diameter* of a network is the maximum, over all pairs of nodes u and v, of the length, in edges, of the shortest path between u and v. Also, the VLSI layout area of B_n is $(1 \pm o(1))n^2$ [3] and the layout area of W_n is $\Theta(n^2)$. (The leading constant in the layout area of W_n is not known). Furthermore, the three-dimensional layout volumes of B_n and W_n are $\Theta(n^{3/2})$ [16].

A network closely related to the butterfly is the *cube-connected cycles* [24]. A $\log n$-dimensional cube-connected cycles network CCC_n consists of n cycles, each containing $\log n$ nodes. Each cycle has a distinct $(\log n)$-bit label, and within a cycle each node is labeled with its position, a number between 1 and $\log n$, inclusive. Taken together, these labels give each node a distinct label $\langle w, i \rangle$, where w is the label of its cycle, and i is its position in the cycle. Two nodes in different cycles are connected by an edge if and only if they share the same position i within their respective cycles, and their cycle labels differ only in the bit in position i. That is, two nodes $\langle w, i \rangle$ and $\langle w', i \rangle$ are connected if w and w' differ in bit position i.

1.2. *Bisection Width*

The *bisection width* of an N-node network $G = (V, E)$ is defined as follows. A *cut* $(S, \bar{S})$ of G is a partition of its nodes into two sets S and $\bar{S}$, where $\bar{S} = V - S$. While this definition of a cut is given in terms of the nodes of G rather than its edges, it is often helpful to think of the cut $(S, \bar{S})$ as the set of all (undirected) edges with one endpoint in S and the other in $\bar{S}$. We call these the *cut edges* and say that they *cross* the cut. The *capacity* of a cut, $C(S, \bar{S})$, is the number of cut edges. We also say that the removal of the cut edges partitions G into S and $\bar{S}$, meaning that in the network that remains after these edges are removed from G, no edge connects a node in S to a node in $\bar{S}$. It is important to note, however, that two distinct cuts may have the same set of cut edges, so the two notions of a cut are not always equivalent. (As an example, in a network with $k > 1$ connected components, 2^k distinct cuts share the empty set as their set of cut edges.) Hence some care is required when viewing a cut as a set of edges and vice versa. A *bisection* of a network is a cut $(S, \bar{S})$ such that $|S| \leq \lceil N/2 \rceil$ and $|\bar{S}| \leq \lceil N/2 \rceil$. The *bisection width* $BW(G)$ is the minimum, over all bisections $(S, \bar{S})$, of $C(S, \bar{S})$. In other words, the bisection width is the minimum number of edges that must be removed from the network in order to partition its nodes into two equal-sized sets (to within one node).

The bisection width of a network is an important indicator of its power as a communications network. As an example, suppose that an N-node network G is used to route messages between the processors in a general-purpose parallel computer, with one processor attached to each node. If each processor sends a message to another processor chosen uniformly at random, then the expected number of messages that cross the bisection, in each direction, is $N/4$. Assuming that each edge of the network can transmit one message (in each direction) in one time step, the time required by the network to route the messages is at least $N/(4BW(G))$. Hence, the smaller the bisection width, the longer it will take to route the messages. Along these lines, in [7], [13], and [14], a network's *bandwidth* (also called *capacity*) is defined in terms of its ability to route messages with random destinations. We omit the precise definition of bandwidth here. In [13] the exact bandwidth of the n-input butterfly is shown to be $2n$, while in [7] tight bounds on the bandwidths of various other networks are proved. Using an argument similar to the one above, both of these papers prove that the bandwidth of a network cannot exceed four times its bisection width, although the definition of the bisection width of the butterfly given in [13] differs slightly from the definition given here in two respects. First, in [13] every edge connecting a node on level i with a node on level $i+1$ (for any $i \geq 0$) is *directed* from level i to level $i+1$, whereas here all edges are undirected. Second, in [13] the bisection width is defined as the minimum, over all cuts $(S, \bar{S})$ such that S contains at least $n/2$ inputs and $\bar{S}$ contains at least $n/2$ outputs, of the number of directed edges from S to $\bar{S}$. Thus, the upper bound on bandwidth in terms of bisection width combined with the exact bandwidth of the butterfly yields a lower bound of $n/2$ on the bisection width, i.e., the number of directed edges needed to separate $n/2$ inputs from $n/2$ outputs. Furthermore, the cut $\{S, \bar{S}\}$ in which S is the set of nodes whose column numbers begin with 0 achieves this bound. This result in similar in spirit to our Lemma 3.1.

In addition to the routing example, there are a large number of problems for which it is possible to prove some lower bound, I, on the number of messages that must cross any bisection of a parallel machine in order to solve the problem [28]. In each case, $I/(BW(G))$ is a lower bound on the time, T, to solve the problem.

The bisection width of a network also gives a lower bound on the VLSI layout area, A, of a network G. In particular, Thompson proved that $A \geq (BW(G))^2$ [28]. Combining this inequality with the inequality $T^2 \geq (I/BW(G))^2$ for any particular problem yields the so-called "AT^2" bound $AT^2 \geq \Omega(I^2)$. (See [28].)

1.3. *Expansion*

The *expansion* of a network G is defined as follows. The *edge expansion* of a set of nodes, S, is $C(S, \bar{S})$, i.e., the number of edges in the cut that separates S from the rest of the network. We define the *edge-expansion function* $EE(G, k)$ of the network to be

$$EE(G, k) = \min_{S:\ |S|=k} C(S, \bar{S})$$

for $1 \leq k \leq N$. In other words, the edge expansion function specifies, for each k, the minimum number of edges that must be removed to isolate a set of k nodes from the rest of the network.

The set of neighbors $\mathcal{N}(S)$ of a set S are the nodes in $\bar{S}$ that are adjacent to nodes in S, i.e.,

$$\mathcal{N}(S) = \{v \in \bar{S} \mid \exists u \in S, (u, v) \in E\}.$$

The *node expansion* of a set S is $|\mathcal{N}(S)|$. We say that a network G has *node-expansion function* $NE(G, k)$ if, for $1 \leq k \leq N$,

$$NE(G, k) = \min_{S:\ |S|=k} |\mathcal{N}(S)|.$$

In other words, the node expansion function specifies, for each k, the smallest number of neighbors possessed by any set of k nodes.

The expansion of a network G is an indicator of the speed at which information can be disseminated in G. In particular, if each node in a set of k nodes holds a small piece of information, they can increase the number of nodes holding the information to $k + NE(G, k)$ in a single step. Several load-balancing algorithms exploiting this property are reported in [8]. The expansion function can also be used to compare the computational powers of different networks. In particular, a difference in the expansion functions of a guest network and a host network has been used to prove lower bounds on the inefficiency of any emulation of the guest by the host [12], [25]. Finally, we observe that the only N-node bounded-degree networks known to be capable of routing and sorting deterministically in $O(\log N)$ time are those that incorporate some form of expansion (i.e., expansion functions of the form $NE(G, k) \geq (1 + \varepsilon)k$, for some fixed $\varepsilon > 0$) into their structures [1], [2], [17], [19], [29].

1.4. *Lower Bounds Based on Embeddings*

Lower bounds on the bisection width and expansion of an N-node network H can often be proved by *embedding* the complete graph $G = K_N$ into H. In general, an embedding of a *guest* network G into a *host* network H is a mapping of nodes of G to nodes of H and edges of G to paths in H. The *load* l of an embedding is the maximum number of nodes of G mapped to any one node of H. The *congestion* c of the embedding is the maximum number of paths (corresponding to edges in G) that cross any one edge of H. The *dilation* d of an embedding is the length of the longest path. In proving lower bounds on bisection width and expansion, the chosen embedding typically has load 1, and routes the same number of paths, c, across each edge of H.

Given an embedding of K_N into H with load 1 and congestion c, a lower bound on $BW(H)$ is computed as follows. Let $(A, \bar{A})$ be a bisection of H with capacity $C(A, \bar{A}) = BW(H)$. Then removing the edges from K_N whose paths cross $(A, \bar{A})$ yields a bisection of K_N with capacity at most $c \cdot BW(H)$. Since $BW(K_N) = N^2/4$, we have $c \cdot BW(H) \geq N^2/4$, and hence $BW(H) \geq N^2/4c$. This approach readily yields $\Omega(n)$ lower bounds on the bisection widths of B_n and W_n, but without tight leading constants.

The same technique can be used to prove lower bounds on the edge expansion of a network. Suppose that K_N is embedded in H with load 1 and congestion c. Let A be a set of k nodes in H such that $C(A, \bar{A}) = EE(H, k)$. For each of the $EE(K_N, k)$ edges leading out of the corresponding set in K_N, a path must be routed out of A in H. Thus, we must have $c \cdot C(A, \bar{A}) \geq EE(K_N, k)$. Since the edge expansion of K_N is $EE(K_N, k) = k(N - k)$, we have $EE(H, k) = C(A, \bar{A}) \geq k(N - k)/c$. For $k \leq N/2$, we have $EE(H, k) \geq kN/2c$.

Prior to our work some bounds on $BW(B_n)$ were known, and $BW(W_n)$ had been analyzed exactly. It is not difficult to show that $BW(B_n) \leq n$ and $BW(W_n) \leq n$: partition the columns into those whose numbers start with a 0 and those whose numbers start with a 1. Similarly, $BW(CCC_n) \leq n/2$. For the cube-connected cycles network, Manabe et al. [20] proved the converse, namely $BW(CCC_n) \geq n/2$. (This paper appears only in Japanese, however!) The same approach can be used to show that $BW(W_n) \geq n$. Hence $BW(CCC_n) = n/2$ and $BW(W_n) = n$.

It is more difficult to prove an exact bound on the bisection width of the butterfly without wraparound, B_n, because it does not possess the same degree of symmetry as CCC_n and W_n. For example, the nodes on level 0 of B_n have two neighbors while those on level 1 have four, whereas in W_n every node has four neighbors. Prior to our work,

$BW(B_n)$ was known to be at least $n/2$. This lower bound is proved by embedding the graph $2K_N$ into B_n, where $2K_N$ is a variant of the complete graph in which any two nodes are connected by *two* parallel edges. There is an embedding of $2K_N = 2K_{n(\log n+1)}$ into B_n with load 1 and congestion $n(\log n+1)^2$. Since $BW(2K_{n(\log n+1)}) = (n(\log n+1))^2/2$, $BW(B_n) \geq n/2$.

These embeddings also imply that the edge expansion functions of B_n and W_n satisfy $EE(B_n, k) = \Omega(k/\log n)$ and $EE(W_n, k) = \Omega(k/\log n)$, for $k \leq N/2$.

1.5. *Related Networks*

Another network closely related to the butterfly is the Beneš network. A $(\log n)$-dimensional Beneš network consists of two back-to-back $(\log n)$-dimensional butterflies B_n and B'_n, where the ith node on level $\log n$ of B_n is identified with the ith node on level $\log n$ of B'_n. The nodes on level 0 of B_n are called the input nodes of the Beneš network, and the nodes on level 0 of B'_n are called the output nodes. Typically each input node is viewed as having two input ports (i.e., connections for edges), and each output node is viewed as having two output ports. The Beneš network is called *rearrangeable* because it is possible to route edge-disjoint paths between its $2n$ input ports and $2n$ output ports in any permutation [5], [6], [30].

In addition to the cube-connected cycles and Beneš networks, the butterfly has been shown to be closely related to the hypercube and other bounded-degree variants of the hypercube, including the shuffle-exchange network and the de Bruijn network. For example, it is not difficult to prove that an N-node butterfly network can be embedded in an N-node hypercube with constant load, congestion, and dilation. In fact, Greenberg et al. [10] proved that, for some sizes of N, the butterfly network is a subgraph of the hypercube. Also, Schwabe [12], [26] showed that an N-node butterfly network can emulate T steps of any computation of an N-node shuffle-exchange network (or de Bruijn network) in $O(T)$ steps, and vice versa.

More information about the structural and algorithmic properties of butterflies can be found in the book by Leighton [15]. Some of the parallel computers that use butterfly networks or its variants are described in [4], [9], [21]–[23]. Many network emulations are described in [12] and [18].

1.6. *Our Results*

We begin in Section 2 by proving that the bisection width of the n-input butterfly network without wraparound, B_n, is $2(\sqrt{2}-1)n+o(n)$. We show how to construct such a bisection and prove that no bisection is smaller. This result is surprising, because it contradicts the prior folklore belief that the bisection width is n. Next, in Section 3 we present an original proof that the bisection width of the butterfly with wraparound, W_n, is n. Although this result was proved previously by Manabe et al. [20], we include our proof because there is no English-language proof of this result in the literature. In Section 4 we prove upper and lower bounds on the edge- and node-expansion functions of W_n and B_n. For example, we show that every set of k nodes in B_n has at least $(k/(2\log k))(1 - o(1))$ neighbors, for $k = o(\sqrt{n})$. Several similar results were previously known. For example, Snir [27] proved tight bounds on the edge-expansion function for another variant of the butterfly network, which he calls Ω_n, that can be derived from $B_{n/2}$ by providing each input node in $B_{n/2}$ with a pair of input ports and each output node with a pair of output ports. These ports are technically not edges, as they do not connect pairs of nodes in $B_{n/2}$, but for the purposes of calculating the edge expansion function, these ports are counted as edges. In particular, $EE(\Omega_n, k)$ is defined as

$$EE(\Omega_n, k) = \min_{S:\ |S|=k} C(S, \bar{S}) + 2|L_0 \cap S| + 2|L_{(\log n)-1} \cap S|.$$

Let S be any set of nodes in Ω_n, and let $C = C(S, \bar{S}) + 2|L_0 \cap S| + 2|L_{(\log n)-1} \cap S|$ and $k = |S|$. Snir showed that $C \log C \geq 4k$. This translates to the lower bound $EE(\Omega_n, k) \geq (4 - o(1))k/\log k$, which is similar to the bound $EE(W_n, k) \geq (4 - o(1))k/\log k$ that we prove in Section 4. Note that Snir's bound holds for all k whereas ours holds only for $k = o(n)$. This is the result of counting the input and output ports in the edge expansion function for Ω_n; notice that $EE(\Omega_n, |\Omega_n|) = 4n$, whereas $EE(W_n, |W_n|) = 0$. Hong and Kung [11] prove a bound for yet another variant of B_n. Their FFT_n graph can be derived from B_n by adding a single input port to each input node and a single output port to each output node. They prove that if, for a set S of k nodes, there is a (not necessarily disjoint) set D of nodes such that every path from an input port to S passes through a node in D, then $k \leq 2|D| \log|D|$. This bound roughly corresponds to the lower bound $NE(B_n, k) \geq (\frac{1}{2} - o(1))k/\log k$ that we prove in Section 4.

2. The Bisection Width of the Butterfly

In this section we show that the bisection width of the butterfly, $BW(B_n)$, satisfies $2(\sqrt{2}-1)n < BW(B_n) \leq 2(\sqrt{2} - 1)n + o(n)$.

We reach this result as follows. We begin by introducing a highly symmetric network, the *mesh of stars*, and an embedding of the butterfly into this network. We use the embedding and the (as of yet unknown) bisection width of the mesh of stars to establish tight lower and upper bounds on $BW(B_n)$. We conclude by computing the bisection width of the mesh of stars.

What follows is a list of properties of the butterfly that we use in our constructions; most of these properties are well known and are given with no proof. Note that n, the number of inputs in a butterfly, is always a power of 2.

Lemma 2.1. *There is an automorphism of B_n (i.e., an embedding of B_n into B_n with load 1, congestion 1, and dilation 1) that maps each level L_i onto $L_{\log n - i}$.*

Lemma 2.2. *Let v and v' be two nodes on the same level of B_n. Then there is a level-preserving automorphism π of B_n such that $\pi(v) = v'$. Moreover, let $\{v, u\}$ and $\{v', u'\}$ be two edges of B_n such that v and v' are on the same level and u and u' are on the same level. Then there is a level-preserving automorphism π of B_n such that $\pi(v) = v'$ and $\pi(u) = u'$.*

Let p be a path through the butterfly. We call p *monotonic* if p visits any level at most once.

Lemma 2.3. *Let v and u be nodes of B_n, $v \in L_0$ and $u \in L_{\log n}$. Then there is exactly one monotonic path linking v and u.*

For $0 \le i \le j \le \log n$, let $B_n[i, j]$ denote the subgraph of B_n induced by levels $L_i, L_{i+1}, \ldots, L_j$.

Lemma 2.4. *Let $0 \le i \le j \le \log n$. Then $B_n[i, j]$ has $n/2^{j-i}$ connected components; each component is isomorphic to $B_{2^{j-i}}$; and the* kth *level of each component is a subset of the nodes on the* $(i+k)$th *level of B_n.*

Lemma 2.5. *Let $n > 1$. Then there is a partition of L_0, the first level of B_n, into two disjoint sets, I and O, each of cardinality $n/2$ such that if we assign two distinct "input ports" to each node of I and two distinct "output ports" to each node of O, then the resulting network is rearrangeable. That is, for any bijection of the input ports to the output ports there is a set of n edge-disjoint paths that link each input port with its image output port.*

Proof. The Beneš network is rearrangeable and there is an embedding of a $((\log n) - 1)$-dimensional Beneš network into B_n with load 1, congestion 1, and dilation 3. This embedding maps the I and O nodes of the Beneš network onto L_0. □

We say that a subset of nodes U is *compact* in a network G if for any given cut of G we can move all of U to one side of the cut without increasing its capacity. Formally, let $G = \langle V, E \rangle$ be a network and $U \subseteq V$. Then U is compact in G if for any cut $g = (A, \bar{A})$ of G there is a cut $g' = (A', \overline{A'})$ (possibly $g = g'$) such that

(1) either $U \subseteq A'$ or $U \subseteq \overline{A'}$,
(2) $A \cap (V - U) = A' \cap (V - U)$, and
(3) $C(g') \le C(g)$.

Lemma 2.6. *U is compact in G if U is compact in the subgraph of G induced by $U \cup \mathcal{N}(U)$.*

Proof. Follows from the definition of compact. □

Lemma 2.7. *Let U be a compact set of nodes in a network G. Then every connected component induced in G by U is also compact.*

Proof. Let $U_1, U_2, \ldots, U_n$ be the connected components of U in G. Assume by contradiction that U_1 is not compact in G, and let $g = (A, \bar{A})$ be a cut of G that partitions U_1 so that moving all of the vertices of U_1 into either A or $\bar{A}$ increases the capacity of the cut. Since $\mathcal{N}(U_1)$ does not contain any nodes in U, the partition of the remaining connected components of U by the cut g does not affect U_1's contribution toward $C(g)$. In particular, if $U - U_1$ is moved entirely into either A or $\bar{A}$, we obtain a new cut whose capacity (which might be larger than that of g) is made larger by moving U_1 entirely into either side of the cut, and thus U is not compact, a contradiction. □

Lemma 2.8. *Let $G = B_n$ and $U = \bigcup_{i=1}^{\log n} L_i$. Then U is compact in G.*

Proof. Let $g = (A, \bar{A})$ be a cut of B_n. Define the cut $g' = (A', \overline{A'})$ by $A' = A \cup U$. Clearly, g' satisfies requirements (1) and (2) of the definition of compact. We conclude by showing that $C(g') \leq C(g)$. Assume, without loss of generality, that $|\bar{A} \cap L_0| \leq |A \cap L_0|$. Let (I, O) be the partition of L_0 given by Lemma 2.5. Then

$$\begin{aligned} 2|\bar{A} \cap I| &= (n/2 - |A \cap I|) + (|\bar{A} \cap L_0| - |\bar{A} \cap O|) \\ &= (n/2 - |\bar{A} \cap O|) + (|\bar{A} \cap L_0| - |A \cap I|) \\ &\leq (n/2 - |\bar{A} \cap O|) + (|A \cap L_0| - |A \cap I|) \\ &= 2|A \cap O|. \end{aligned}$$

Hence $|\bar{A} \cap I| \leq |A \cap O|$. Also, by symmetry, $|\bar{A} \cap O| \leq |A \cap I|$. Pick a bijection π of I onto O such that $\pi(\bar{A} \cap I) \subseteq A \cap O$ and $\pi^{-1}(\bar{A} \cap O) \subseteq A \cap I$. By Lemma 2.5, there is a set of $2|\bar{A} \cap L_0|$ edge-disjoint paths realizing π such that each path has one end in $\bar{A} \cap L_0$ and the other end in $A \cap L_0$. Each of these paths must contain at least one edge that crosses the cut $g = (A, \bar{A})$. Hence, $C(g) \geq 2|\bar{A} \cap L_0|$. By our construction, $C(g') = 2|\bar{A} \cap L_0|$. Hence, $C(g') \leq C(g)$. □

Lemma 2.9. *Each connected component of $B_n[i, \log n]$ is compact in B_n, $1 \leq i \leq \log n$.*

Proof. Let B' be a connected component of $B_n[i, \log n]$, and let B'' be the other connected component of $B_n[i, \log n]$ such that $\mathcal{N}(B'') = \mathcal{N}(B')$. Both B' and B'' are isomorphic to $B_{n/2^i}$. Let G be the network induced in B_n by $B' \cup B'' \cup \mathcal{N}(B')$. G is isomorphic to $B_{n/2^{i-1}}$. Thus, by Lemma 2.8, $B' \cup B''$ is compact in G. Therefore, by Lemma 2.6, $B' \cup B''$ is compact in B_n. Furthermore, by Lemma 2.7, B' is separately compact in B_n. □

Lemma 2.10. *Let $0 \leq i \leq \log n$, $0 \leq j$, and $k = n2^j$, where i and j are integral and n is a power of 2. Then there is an embedding π of B_k into B_n such that*:

(1) *The dilation of the embedding is* 1.
(2) *The congestion of any edge is exactly* 2^j.
(3) *π maps $B_k[0, i-1]$ onto $B_n[0, i-1]$ with uniform load of 2^j.*
(4) *π maps $B_k[i+1+j, \log n + j]$ onto $B_n[i+1, \log n]$ with uniform load of 2^j.*
(5) *For each $l \in [i, i+j]$, π maps exactly 2^j nodes of the lth level of B_k onto each node of the ith level of B_n, so that the load on each node on the ith level of B_n is $(j+1)2^j$.*

Proof. We describe the embedding but do not give detailed proofs of the five properties listed in the statement of the lemma. For any $w \in \{0, 1\}^k$, the nodes of column w of B_k are all mapped to the column $w' \in \{0, 1\}^n$ of B_n for which the first i bit positions of w' match the first i bit positions of w, and the last $\log n - i$ bit positions of w' match the last $\log n - i$ bit positions of w. Within the column w of B_k, for any $l \in [0, i-1]$, the node with label $\langle w, l\rangle$ is mapped to the node with label $\langle w', l\rangle$ in B_n. For any $l \in [i, i+j]$, $\langle w, l\rangle$ is mapped to node $\langle w', i\rangle$. Finally, for $l \in [i+j+1, (\log n)+j]$, $\langle w, l\rangle$ is mapped to $\langle w', l-j\rangle$. □

2.1. *Reducing the Butterfly to a Mesh of Stars*

The $j \times k$ *mesh of stars*, denoted $MOS_{j,k}$, is the network obtained from the complete bipartite graph $K_{j,k}$ by replacing each edge with a path of length 2. This network has three levels that we refer to as M_1 (with j nodes), M_2 (with jk nodes), and M_3 (with k nodes).

Let $G = (V, E)$ be a network, let $g = (A, \bar{A})$ be a cut of G, and let $U \subseteq V$. We say that *g bisects U* if $|A \cap U| \le |\bar{A} \cap U| \le |A \cap U| + 1$. The *$U$-bisection width of G* is defined by

$$BW(G, U) = \min\{C(g)\colon\ g \text{ is a cut of } G \text{ that bisects } U\}.$$

In this section we show that

$$\frac{2BW(MOS_{n,n}, M_2)}{n^2} \le \frac{BW(B_n)}{n} \le \frac{2BW(MOS_{f(n),f(n)}, M_2)}{f(n)^2} + o(1)$$

for some function f such that $\lim_{n\to\infty} f(n) = \infty$. Later we compute the bisection width of the mesh of stars, which gives us lower and upper bounds on $BW(B_n)$.

We establish both bounds on $BW(B_n)$ via the following embedding of butterflies into meshes of stars.

Lemma 2.11. *Let $j, k > 1$, and suppose jk divides n. Then there is an embedding π of B_n into $MOS_{j,k}$ such that*:

(1) *The dilation of the embedding is* 1.
(2) *The congestion of any edge is exactly* $2n/jk$.
(3) *π maps the first $\log k$ levels of B_n onto M_1 with uniform load*, $(n/j)\log k$.
(4) *π maps the last $\log j$ levels of B_n onto M_3 with uniform load*, $(n/k)\log j$.
(5) *π maps the other nodes of B_n onto M_2 with uniform load*, $(n/jk)(\log(n/jk)+1)$. *Moreover, if $jk = n$, then $\pi^{-1}(\{v\})$ is compact for any node v of $MOS_{j,k}$, and the load of any node of M_2 is* 1.

Proof. We begin by introducing the following auxiliary graph G, which has three levels. The nodes on the first level are the connected components of $B_n[0, \log n - \log j]$, the nodes on the second level are the connected components of $B_n[\log k, \log n - \log j]$, and the nodes on the third level are the connected components of $B_n[\log k, \log n]$. Suppose that x is a node on the second level. Then the connected component x of B_n is contained in one of the connected components, w, on the first level, and also in one of the connected components, y, on the third level. Let G have (undirected) edges from x to both w and y.

We now show that G is isomorphic to $MOS_{j,k}$. By Lemma 2.4, the first level has j nodes, the second level jk nodes, and the third level k nodes. As we have seen, each node on the second level of G has exactly one edge leading to each of the other two levels. By Lemma 2.4, each node on the first level is a connected component of $B_n[0, \log n - \log j]$, which is isomorphic to $B_{n/j}$. This connected component contains a number of connected components belonging to $B_n[k, \log n - \log j]$. Applying Lemma 2.4 again, the set of connected components of $B_n[k, \log n - \log j]$ contained in one node on the first level is isomorphic to $B_{n/j}[k, \log n - \log j]$, and hence there are precisely k of these components. For each of these k components, there is a corresponding node on the second level of G, so the degree of each node on the first level of G is k. By a similar argument, the degree of each node on the third level of G is j. What remains is to show that for each node on the first level, there is a monotonic path of length 2 to each node on the third level. Consider a pair of nodes w and y on the first and third levels of G. By Lemma 2.3, any node on the first level of B_n is linked to each node on the last level of B_n by a single monotonic path. Hence, the path from any node on the first level of w in B_n to any node on the last level of y in B_n must pass through some connected component x of $B_n[\log k, \log n - \log j]$ that is contained in both w and y, which implies that there is a path from w to x to y in G.

We define the embedding π of B_n into G as follows. Let v be a node of B_n. If v is in $B_n[\log k, \log n - \log j]$, then v is mapped to the corresponding node of M_2. Otherwise, v belongs to either one connected component in $B_n[0, \log n - \log j]$, or one connected component in $B_n[\log k, \log n]$, but not both. In this case v is mapped to the corresponding node in M_1 or M_3.

Since the nodes of B_n mapped by π to M_1 are drawn from $B_n[0, \log k - 1]$, and the nodes mapped to M_3 are drawn from $B_n[\log n - \log j + 1, \log n]$, and jk divides n, no node of B_n that is mapped to M_1 is a neighbor in B_n of a node mapped to M_3. Hence, the dilation of π is 1, which satisifies requirement (1).

By symmetry (Lemma 2.2), the congestion of all M_1 to M_2 edges of G is equal. Since $2n$ edges of B_n are mapped across jk edges of G, the congestion of each edge is $2n/jk$. The same holds for the other level of edges of G; this establishes (2). (A more explicit way to see this is that, by Lemma 2.4, a connected component in $B_n[0, \log n - \log j]$ contains k connected components of $B_n[\log k, \log n - \log j]$, each of which is mapped to a different node of M_2. Since each component of $B_n[\log k, \log n - \log j]$ has n/jk input nodes on level $\log k$, and there are two edges from level $\log k - 1$ to each of these inputs, the congestion of each edge from M_1 to M_2 is $2n/jk$.)

By Lemma 2.4, π satisfies (3)–(5).

Assume now that $n = jk$. Then for $v \in M_2$, $\pi^{-1}(v)$ is a single node by our construction and hence is compact. For $v \in M_3$, $\pi^{-1}(v)$ is compact by Lemma 2.9. By Lemma 2.1, the same holds for $v \in M_1$. □

First, we establish the lower bound on the bisection width of B_n in terms of $BW(MOS_{n,n}, M_2)$. The proof makes use of the following lemma.

Lemma 2.12. *Let* $n > 1$. *Then*:

(1) *There is an* i *such that* $0 \le i \le \log n$ *and* $BW(B_n, L_i) \le BW(B_n)$.
(2) $BW(B_{n^2}, L_{\log n})/n^2 \le BW(B_n)/n$.

Proof. To establish (1), let $g = (A, \bar{A})$ be a bisection of B_n such that $C(g) = BW(B_n)$. Assume, without loss of generality, that $|A \cap L_0| \le n/2$. Then there is an i such that $|A \cap L_i| \le n/2 \le |A \cap L_{i+1}|$. Let $g' = (A', \overline{A'})$ be a cut of B_n (that does not necessarily bisect B_n) such that $C(g') \le C(g)$, $|A' \cap L_i| \le n/2 \le |A' \cap L_{i+1}|$, and $|A' \cap L_{i+1}| - |A' \cap L_i|$ is as small as possible. We establish (1) by showing that g' bisects either L_i or L_{i+1}. Assume otherwise. Then $|A' \cap L_i| < n/2 < |A' \cap L_{i+1}|$. Since the edges connecting nodes on level i with nodes on level $i+1$ in a butterfly can be partitioned into node- and edge-disjoint 4-cycles (which resemble butterflies when drawn, hence the name "butterfly"), there must be a 4-cycle, v–u–v'–u'–v, with $v, v' \in L_i$ and $u, u' \in L_{i+1}$ such that $|A' \cap \{v, v'\}| < |A' \cap \{u, u'\}|$. Hence, either $|A' \cap \{v, v'\}| = 0$ or $|A' \cap \{u, u'\}| = 2$. In both cases we can modify g' by moving a single node from A' to $\overline{A'}$ or vice versa to reduce $|A' \cap L_{i+1}| - |A' \cap L_i|$ without increasing the capacity of the cut, which yields a contradiction.

To prove (2), select an i that satisfies $BW(B_n, L_i) \le BW(B_n)$ (by (1) we know that such an i exists), and let $g = (A, \bar{A})$ be a cut of B_n that bisects L_i and for which $C(g) = BW(B_n, L_i)$. Apply Lemma 2.10 with $j = \log n$ and $k = n^2$. Let π be the embedding of B_{n^2} into B_n given by this lemma. Define a cut of B_{n^2} by $g' = (\pi^{-1}(A), \pi^{-1}(\bar{A}))$. Since the congestion of each edge of B_n is exactly n, $C(g) \cdot n = C(g')$, and hence $C(g')/n^2 = C(g)/n$. Since g bisects the ith level of B_n, by property (5) of Lemma 2.10, g' bisects every level of $B_{n^2}[i, i + \log n]$. Furthermore, since $\log n \in [i, i + \log n]$, g' bisects the $(\log n)$th level of B_{n^2}. Hence, $BW(B_{n^2}, L_{\log n})/n^2 \le C(g')/n^2 = C(g)/n = BW(B_n, L_i) \le BW(B_n)/n$. □

Lemma 2.13. $2BW(MOS_{n,n}, M_2)/n^2 \le BW(B_n)/n$.

Proof. This inequality clearly holds for $n = 1$. Assume henceforth that $n > 1$. By Lemma 2.12, $BW(B_{n^2}, L_{\log n})/n^2 \le BW(B_n)/n$. Let $g = (A, \bar{A})$ be a cut of B_{n^2} that bisects $L_{\log n}$ such that $C(g) = BW(B_{n^2}, L_{\log n})$. Apply Lemma 2.11 on B_{n^2} with $j = k = n$, and let π be the embedding of B_{n^2} into $MOS_{n,n}$ provided by this lemma. Note that for $v \in M_2$, $\pi^{-1}(v)$ is a singleton in B_{n^2}.

The sets $\{\pi^{-1}(v)\colon v \in M_1 \cup M_3\}$ are pairwise disjoint, compact (by Lemma 2.9), and do not intersect $L_{\log n}$. Hence, we may assume, without loss of generality, that each of these sets is a subset of either A or $\bar{A}$. Define a cut $g' = (A', \overline{A'})$ of $MOS_{n,n}$ by $v \in A' \iff \pi^{-1}(v) \subset A$. Since $\pi^{-1}(v)$ is a singleton for any $v \in M_2$ and g bisects $L_{\log n}$, g' bisects M_2. Since the congestion of π is exactly 2, $2C(g') = C(g)$. Hence, $2BW(MOS_{n,n}, M_2)/n^2 \le 2C(g')/n^2 = C(g)/n^2 = BW(B_{n^2}, L_{\log n})/n^2 \le BW(B_n)/n$. □

We now establish the upper bound on the bisection width of B_n. Let $G = \langle V, E\rangle$ be a network, let $g = (A, \bar{A})$ be a cut of G, and let $U \subset V$. We say that U *is amenable with respect to* g if it is possible to shift nodes of U from A to $\bar{A}$ or vice versa so that any number of nodes in U (but not necessarily any subset of U) from 0 to $|U|$ can be placed on either side of the cut without increasing the capacity of the cut, i.e., U is amenable with respect to g in the network G if for every $0 \le k \le |U|$ there is a cut $g' = (A', \overline{A'})$ such that:

(1) $A' \cap (V - U) = A \cap (V - U)$,
(2) $|A' \cap U| = k$, and
(3) $C(g') \le C(g)$.

Lemma 2.14. *Let $g = (A, \bar{A})$ be a cut of G, let U be a set of nodes, and let $W = U \cup \mathcal{N}(U)$. Then U is amenable with respect to g in G iff U is amenable with respect to $g|_W \stackrel{\triangle}{=} (A \cap W, \bar{A} \cap W)$ in the subgraph of G induced by W.*

Proof. The lemma follows from the definition of "amenable." □

Lemma 2.15. *Let $n > 2$, let U be a connected component of $B_n[1, \log n - 1]$, and let $g = (A, \bar{A})$ be a cut of B_n such that $L_0 \cap \mathcal{N}(U) \subseteq A$ and $L_{\log n} \cap \mathcal{N}(U) \subseteq \bar{A}$. Then U is amenable with respect to g.*

Proof. By Lemma 2.4, the subgraph of B_n induced by U is isomorphic to $B_{n/4}$. By Lemma 2.14, we can restrict ourselves to the subgraph of B_n induced by $U \cup \mathcal{N}(U)$. Call this network G, its levels $L_0^G, \ldots, L_{\log n}^G$, and let g' be the cut g restricted to G.

There is a set P of $n/2$ monotonic edge-disjoint paths of G leading from L_0^G to $L_{\log n}^G$ and covering all the edges of G. Each path of P has one endpoint in A and the other in $\bar{A}$, hence, $n/2 \le C(g')$.

Consider a cut $g'' = (A'', \overline{A''})$ of G with the following property:

$$(*)\ \ L_0^G \subset A'',\ L_{\log n}^G \subset \overline{A''},\ \text{and}\ \forall i,\ 0 < i < \log n, L_i^G \bigcap A'' \neq \emptyset \Rightarrow L_{i-1}^G \subset A''.$$

In other words, there is some j, $0 < j < \log n$, such that levels 0 through $j - 1$ of G are contained entirely in A'' and levels $j + 1$ through $\log n$ are contained entirely in $\overline{A''}$, while some of the nodes on level j may belong to A and others to $\overline{A''}$. Clearly, there is a cut g'' satisfying both $(*)$ and $|A'' \cap U| = k$ for any $0 \le k \le |U|$. Now, if g'' satisfies $(*)$, then any path of P contributes exactly one to the capacity of g''; since there are no other edges, $C(g'') = n/2 \le C(g')$. □

The following lemma provides a tight upper bound on the bisection width of B_n in terms of the M_2-bisection width of $MOS_{j,j}$. The proof not only establishes the inequality, but also demonstrates how to find a bisection of B_n with capacity at most $2n \cdot (BW(MOS_{j,j}, M_2)/j^2 + 2/j)$, for any j such that $j^3 + 2j - 1 \le \log n$. In Section 2.2 we show that as j grows to infinity, $BW(MOS_{j,j}, M_2)/j^2$ converges to $\sqrt{2} - 1$. Thus, for large n, we can choose a large j satisfying $j^3 + 2j - 1 \le \log n$, and construct a bisection of B_n with capacity close to $2(\sqrt{(2)} - 1)n$.

Lemma 2.16. *$BW(B_n)/n \le 2BW(MOS_{j,j}, M_2)/j^2 + 4/j$ for any j such that j is a power of 2 and $\log n \ge j^3 + 2j - 1$.*

Proof. We begin by finding a bisection of M_2 in $MOS_{j,j}$ with minimum capacity. Let $j > 1$ be a power of 2 and let g^* be a cut of $MOS_{j,j}$ that bisects M_2 such that $C(g^*) = BW(MOS_{j,j}, M_2)$. Pick two nodes, $u, v \in M_2$, on different sides of g^*, $u \in A$ and $v \in \bar{A}$. Next, for reasons that will soon become clear, we move at most one neighbor of u and one neighbor of v to the other side of g^* to produce a cut $g = (A, \bar{A})$ in which each of u and v has one neighbor in A and the other in $\bar{A}$. By this construction, g bisects M_2 and $C(g) \le BW(MOS_{j,j}, M_2) + 2j$, since the degree of the neighbors of u and v is j.

Next, we embed B_n into $MOS_{j,j}$, and use the embedding combined with the cut g of $MOSj, j$ to induce a cut g' of B_n. Let $k = j$, $n > j^2$ and let π be the embedding of B_n into $MOS_{j,j}$ given by Lemma 2.11. Define a cut of B_n by $g' = (A', \overline{A'}) = (\pi^{-1}(A), \pi^{-1}(\bar{A}))$. Since the congestion is $2n/j^2$, we have $C(g') = 2nC(g)/j^2$. Hence, $C(g')/n = 2C(g)/j^2$. Because g bisects M_2, g' bisects the nodes of B_n that are mapped to M_2, i.e., the nodes in $B_n[j, \log n - j]$. The remaining nodes in B_n, those in $B_n[0, j-1]$ and $B_n[\log n - j + 1]$, might be mapped to either side of g' in an arbitrary fashion. Hence, the cut g' does not necessarily bisect B_n.

In order to transform g' into a bisection of B_n, we move some nodes from A' to $\overline{A'}$, or vice versa. Let $N = n(1 + \log n)$ be the number of nodes of B_n. Since only the nodes in $B_n[0, j-1]$ are mapped to M_1, and only the nodes of $B_n[\log n - j + 1, \log n]$ are mapped to M_3, the (absolute) difference between $|A'|$ and $|\overline{A'}|$ can be at most $N(2j/(1+\log n))$. Hence, for $j = o(\log n)$, the imbalance is $o(N)$. By Lemma 2.11(5), for each $w \in M_2$, $\pi^{-1}(w)$ is a connected component of $B_n[j, \log n - j]$, and hence is isomorphic to B_{n/j^2}. Thus, $|\pi^{-1}(u)| = |\pi^{-1}(v)| = |B_{n/j^2}| = n/j^2(1 + \log(n/j^2)) = N(1 - 2j/(1 + \log n))/j^2$. Since each of u and v has one neighbor in A and the other in $\bar{A}$, by Lemma 2.15, both $\pi^{-1}(u)$ and $\pi^{-1}(v)$ are amenable with respect to g'. By our construction, one is a subset of A' and the other of $\overline{A'}$. Therefore, provided that there are enough nodes in $\pi^{-1}(u)$ and $\pi^{-1}(v)$, we can correct the imbalance without increasing the capacity of the cut by either moving nodes from $\pi^{-1}(u)$ to $\overline{A'}$ or by moving nodes from $\pi^{-1}(v)$ to A'. Moving one node (in the right direction) decreases the imbalance by 2. Hence, there will be enough nodes to move provided that $2 \cdot N(1-2j/(1+\log n))/j^2 \ge N(2j/(1+\log n))$. This inequality holds when $j^3 + 2j - 1 \le \log n$.

Hence, for $j^3 + 2j - 1 \le \log n$, $BW(B_n) \le C(g')$ and $BW(B_n)/n \le C(g')/n = 2C(g)/j^2 \le 2BW(MOS_{j,j}, M_2)/j^2 + 4/j$. □

2.2. *The Bisection Width of the Mesh of Stars*

In this subsection we show

$$\sqrt{2} - 1 < \frac{BW(MOS_{j,j}, M_2)}{j^2} \le \sqrt{2} - 1 + o(1).$$

As the following lemma shows, the real function $f(x, y) \triangleq x + y - \min(1, 2xy)$, defined on the closed domain $D = \{\langle x, y\rangle\colon 0 \le x, y \le 1 \text{ and } 1 \le x + y\}$, is related to $BW(MOS_{j,j}, M_2)$.

Lemma 2.17. *Let j be an even integer ($j > 0$), and let $\langle x, y\rangle \in D$ such that xj and yj are integers. Let B be the set of cuts $(A, \bar{A})$ of $MOS_{j,j}$ that bisect M_2 and satisfy $|A \cap M_1| = xj$ and $|A \cap M_3| = yj$. Then*

$$\min\{C(g)\colon g \in B\} = f(x, y)j^2.$$

Proof. Let $g = (A, \bar{A})$ be a cut in B such that $C(g)$ is as small as possible. The network $MOS_{j,j}$ has $x(1-y)j^2$ monotonic paths (of length 2) leading from $A \cap M_1$ to $\bar{A} \cap M_3$, and $(1-x)yj^2$ monotonic paths leading from $\bar{A} \cap M_1$ to $A \cap M_3$. Each of these paths contributes one to the capacity of g. In addition, there are xyj^2 monotonic paths from $A \cap M_1$ to $A \cap M_3$. Since $C(g)$ is as small as possible, as many of the middles nodes on these paths as possible are in A. Assume without loss of generality that $(1-x)(1-y) < xy$ (otherwise we can reverse the roles of A and $\bar{A}$, and examine cut $g' = (\bar{A}, A)$ instead, noting that $C(g') = C(g)$). Since g bisects M_2, if $\frac{1}{2} < xy$, then the middle nodes of exactly $(xy - \frac{1}{2})j^2$ of these paths are in $\bar{A}$. (Here we need j even so that $\frac{1}{2}j^2$ is integral.) Each of these paths contributes two to the capacity of g. Otherwise, if $xy \leq \frac{1}{2}$, then, because g is minimal, the middles node of all of these paths are in A, and the paths contribute nothing to the capacity of g. Since $(1-x)(1-y) < xy$, there are at most $j^2/2$ paths from $\bar{A} \cap M_1$ to $\bar{A} \cap M_3$, and hence the middle nodes on these paths are all in $\bar{A}$ and the paths contribute nothing to the capacity of g.

In summary, $C(g)/j^2 = x(1-y) + (1-x)y + 2\max(xy - \frac{1}{2}, 0) = x + y - 2xy + \max(2xy - 1, 0) = x + y + \max(-1, -2xy) = x + y - \min(1, 2xy) = f(x, y)$. □

Lemma 2.18. *The function* $f = x + y - \min(1, 2xy)$ *is continuous in the domain* $D = \{\langle x, y\rangle \colon 0 \leq x, y \leq 1$ *and* $1 \leq x + y\}$, *and* $f\left(\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}\right) = \sqrt{2} - 1$ *is a* (*global*) *minimum of* f.

Proof. Clearly, f is continuous. We have $f(x, y) = x + y - \min(1, 2xy) = x + y - \min(1, [(x+y)^2 - (x-y)^2]/2)$. For any fixed value of $x+y$, $f(x, y)$ is minimized when $x = y$. Hence, f has a minimum on the line $x = y$. Consider the univariate function $f(x, x) = 2x - \min(1, 2x^2)$ for $\langle x, x\rangle \in D$. In this domain $x \geq \frac{1}{2}$. This function is monotonic decreasing for $x < \sqrt{\frac{1}{2}}$ and monotonic increasing for $\sqrt{\frac{1}{2}} < x$. Hence, it reaches a minimum at $x = \sqrt{\frac{1}{2}}$. □

Lemma 2.19. *As a function of variable* j, *which is positive, even, and integral*,

$$\sqrt{2} - 1 < BW(MOS_{j,j}, M_2)/j^2 \leq \sqrt{2} - 1 + o(1).$$

Proof. Let $g = (A, \bar{A})$ be a cut of $MOS_{j,j}$ that bisects M_2 such that $C(g)$ is as small as possible. Assume, without loss of generality, that $j \leq |A \cap (M_1 \cup M_3)|$. (Otherwise, since $|M_1| + |M_3| = 2j$, swap A and $\bar{A}$.) Let $x = |A \cap M_1|/j$ and $y = |A \cap M_3|/j$. Clearly, $\langle x, y\rangle \in D$.

The first inequality, $\sqrt{2} - 1 < BW(MOS_{j,j}, M_2)/j^2$, follows from Lemmas 2.17 and 2.18 and the fact that $\sqrt{2} - 1$ is irrational while $BW(MOS_{j,j}, M_2)/j^2$ is not.

The second inequality, $BW(MOS_{j,j}, M_2)/j^2 \leq \sqrt{2} - 1 + o(1)$, follows from the same two lemmas and the fact that, as j goes to infinity, for a minimum cut, $g = (A, \bar{A})$, $x = |A \cap M_1|/j$ becomes arbitrarily close to $\sqrt{\frac{1}{2}}$, as does $y = |A \cap M_3|/j$, so that $f(x, y)$ converges to the minimum value $f\left(\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}\right) = \sqrt{2} - 1$. □

Lemmas 2.13, 2.16, and 2.19 imply:

Theorem 2.20. *For $n > 1$,*

$$2(\sqrt{2}-1)n < BW(B_n) \leq 2(\sqrt{2}-1)n + o(n).$$

3. Bisection Width of W_n

In this section we prove that the bisection width of W_n is n. The proof makes use of the following lemma.

Lemma 3.1. *Any cut of the butterfly network B_n that bisects its inputs* (*or outputs*), *or the set of inputs and outputs of the network taken together, has capacity at least n.*

Proof. We prove this by embedding the complete bipartite graph $K_{n,n}$ in the butterfly network B_n. Call the sides of $K_{n,n}$ *left* and *right*, respectively. The embedding maps the nodes on the left side of $K_{n,n}$ to the input nodes of B_n and the nodes on the right side to the output nodes of B_n in a one-to-one fashion. Each edge of $K_{n,n}$ is mapped to the unique shortest path that connects the corresponding input and output nodes of the butterfly, that is, the usual route used to connect those two nodes in the butterfly network. The embedding has load 1, congestion $n/2$, and dilation $\log n$.

Now take a cut $(S, \bar{S})$ of B_n that bisects its inputs. The removal of the edges of $K_{n,n}$ that are routed through any of the edges in the cut $(S, \bar{S})$ results in a cut of $K_{n,n}$ that has capacity at most $C(S, \bar{S}) \cdot n/2$. Next, take a minimum cut of $K_{n,n}$ that bisects the left side of $K_{n,n}$. Let b and $n - b$ be the number of right nodes on each side of the cut. The capacity of this cut is $(b + n - b) \cdot n/2$, i.e., $n^2/2$. Since this is a minimum cut, any other cut of $K_{n,n}$ bisecting its left side, and, in particular, the one induced in $K_{n,n}$ by $(S, \bar{S})$ must have capacity at least $n^2/2$. Thus, $C(S, \bar{S}) \geq n$. A cut bisecting the outputs of the butterfly network is analogous.

Now consider a minimum cut that does not necessarily bisect either the left or right sides of $K_{n,n}$, but does bisect $K_{n,n}$ itself. Let b be the number of nodes from the left side in one of the partitions induced by the cut. Then this partition has $n - b$ nodes from the right side of $K_{n,n}$, while the other partition has $n - b$ nodes from the left side and b nodes from the right. The capacity of the cut is $b^2 + (n - b)^2$. However, $b^2 + (n - b)^2$ has its minimum value when $b = n/2$, and thus the cut has capacity at least $n^2/2$. If we map this cut onto the butterfly network, according to our embedding, we obtain a cut $(S, \bar{S})$ of the set of input and output nodes of the butterfly network. Again, we have $C(S, \bar{S}) \cdot n/2 \geq n^2/2$, thus proving the lemma. □

Lemma 3.2. *The bisection width of the butterfly network with wraparound is $BW(W_n) = n$.*

Proof. It is straighforward to prove that $BW(W_n) \leq n$ by exhibiting a bisection with capacity n. For example, consider the cut $(S, \bar{S})$ where S is the set of nodes in the first $n/2$ columns of the butterfly network.

To show that $BW(W_n) \geq n$, we start with a cut $g = (S, \bar{S})$ that bisects W_n and is as small as possible, and show how to translate g into a cut of equal capacity that bisects the inputs of B_n. Lemma 3.1 can then be used to provide a lower bound on the capacity of g. Because g bisects W_n, either there exists a level i with exactly $n/2$ nodes in S, or there is a level i with more than $n/2$ nodes in S such that on level $i + 1 (\bmod \log n)$ there are more than $n/2$ nodes in $\bar{S}$. By the symmetry of the butterfly with wraparound, we can renumber the levels of W_n so that, without loss of generality, $i = 0$. The cut $(S, \bar{S})$ is translated to a cut of B_n by transmuting W_n into B_n in a standard fashion: each node v on level 0 of W_n is replaced by a pair of nodes. One of these new nodes remains on level 0 in the same column as v and inherits the edges connecting v to its two neighbors on level 1. The other new node remains in the same column of v but becomes part of a new level $\log n$, and inherits the edges connecting v to its neighbors on level $\log n - 1$. The resulting network is isomorphic to B_n. Moreover, the edges in the cut $(S, \bar{S})$ of W_n now also form a cut of B_n. As long as the majority of the nodes in level 0 are in S, there must be some node s on level 0 that is in S and that has a neighbor on level 1 that belongs to $\bar{S}$ (since any k nodes on level 0 have at least k neighbors on level 1). Moving s from S to $\bar{S}$ does not increase the capacity of the cut. We can repeat this process until level 0 is bisected by $(S, \bar{S})$. By Lemma 3.1 this cut must have capacity at least n. Therefore $C(S, \bar{S}) \geq n$, and hence $BW(W_n) \geq n$. □

Lemma 3.3. *The bisection width of the* $(\log n)$*-dimensional cube-connected cycles network is* $BW(CCC_n) = n/2$.

Proof. A bisection that cuts one of the cube dimensions has size $n/2$, and thus $BW(CCC_n) \leq n/2$. To prove a matching lower bound, we embed W_n in CCC_n as follows. Suppose that $\langle w, i\rangle$ and $\langle w', i'\rangle$ are neighbors in W_n, where $i' = i+1 (\bmod \log n)$. If $w = w'$, then map the edge to the corresponding edge in CCC_n. Otherwise, map the edge to a path of length two in CCC_n that passes through $\langle w, i'\rangle$, which is connected to both $\langle w, i\rangle$ and $\langle w', i'\rangle$. This embedding has congestion 2, and thus $BW(CCC_n) \geq n/2$. □

4. Expansion of W_n and B_n

In this section we derive upper and lower bounds on the edge and node expansion of W_n and B_n.

4.1. *Expansion of* W_n

In this section we determine the edge-expansion function $EE(W_n, k)$ and the node-expansion function $NE(W_n, k)$ of W_n. First, we show that, for $k = o(n)$ (which implies

$k = o(N)$), $(4 - o(1))k/\log k \leq EE(W_n, k) \leq (4 + o(1))k/\log k$, which we also write more succinctly as $EE(W_n, k) = (4 \pm o(1))k/\log k$. The lower bound cannot be extended to hold for all values of k up to $N/2$ because, for $k = N/2$, the value of the expansion function cannot exceed the bisection width of W_n, which we showed in Section 3 to be $BW(W_n) = n \leq (1+o(1))N/\log N$. Hence, $EE(W_n, N/2) \leq BW(W_n) \leq (2+o(1))(N/2)/\log(N/2)$, which is smaller than the lower bound that holds for $k = o(n)$ by a factor of about 2. For larger values of k, however, we can use the technique of embedding K_N into W_n, which gives a bound of $EE(W_n, k) = \Omega(k/\log n)$ for $k \leq N/2$. For $k = n^{\varepsilon}$, for any fixed $\varepsilon > 0$, this lower bound is $\Omega(k/\log k)$. Hence, for all $k \leq N/2$, $EE(W_n, k) = \Theta(k/\log k)$. Our bounds for $NE(W_n, k)$ are not as tight. We show that $(1 - o(1))k/\log k \leq NE(W_n, k) \leq (3 + o(1))k/\log k$, for $k = o(n)$. For larger values of k we can again use the technique of embedding K_N into W_n, which yields $NE(W_n, k) = \Theta(k/\log k)$ for $k \leq N/2$.

Definitions. The nodes of a rooted tree can be arranged in levels in the following manner. The root of the tree is in level 0. A node whose parent is in level i belongs to level $i + 1$. The nodes of a d-dimensional sub-butterfly of W_n or B_n can be arranged in $d + 1$ levels of 2^d nodes each. We refer to the nodes in level 0 (resp. level d) of the sub-butterfly as the inputs (resp. outputs) of the sub-butterfly. Note that the inputs and outputs of a sub-butterfly may or may not be inputs or outputs of W_n or B_n.

The *down-tree* T_u is an n-leaf complete binary tree rooted at node u of W_n, where the children of a node on level i are located on level $i + 1 (\mathrm{mod} \log n)$. Let node u be in level i of W_n. Tree T_u is a subgraph of W_n such that the jth level of T_u consists of nodes in level $i + j (\mathrm{mod} \log n)$ of W_n. Note that the leaves of T_u also belong to level i of W_n.

The *up-tree* T'_u is an n-leaf complete binary tree rooted at node u of W_n, where the children of a node on level i are located on level $i - 1 (\mathrm{mod} \log n)$. Let node u be in level i of W_n. Tree T'_u is a subgraph of W_n such that the jth level of T'_u consists of nodes in level $i - j (\mathrm{mod} \log n)$ of W_n. Note that the leaves of T'_u also belong to level i of W_n.

4.1.1. *Edge Expansion of W_n.* In this section we determine $EE(W_n, k)$ to within lower-order terms by proving upper and lower bounds.

Lemma 4.1. *The edge-expansion function $EE(W_n, k)$ is at most $(4+o(1))k/\log k$, for $1 \leq k \leq N$.*

Proof. Let A be a sub-butterfly of W_n with k nodes. Each level of the sub-butterfly A has $(1 + o(1))k/\log k$ nodes. Each input and output node of sub-butterfly A has two incident edges that belong to cut $(A, \bar{A})$. Thus, the total number of edges in cut $(A, \bar{A})$ is $(4 + o(1))k/\log k$. Therefore, $EE(W_n, k) \leq (4 + o(1))k/\log k$. □

Lemma 4.2. *The edge-expansion function $EE(W_n, k)$ is at least $(4-o(1))k/\log k$, for $k = o(n)$.*

Proof. Let A be *any* set of $k = o(n)$ nodes of W_n. To prove the lemma, we use a credit

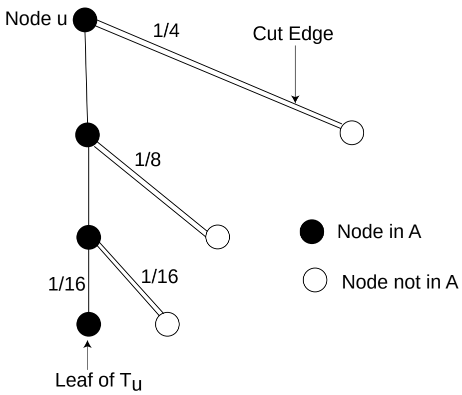

Fig. 2. Node u passes $\frac{1}{2}$ unit of credit down tree T_u. A fractional value next to an edge indicates the number of units of credit retained by the edge.

distribution scheme to show that $C(A, \bar{A})$ is at least $(4 - o(1))k/\log k$. Each node $u \in A$ distributes 1 unit of credit to edges in cut $(A, \bar{A})$ using the following procedure. Let T_u be the down-tree rooted at node u. Furthermore, let the edges of T_u be directed from root to leaf. Node u passes $\frac{1}{2}$ unit of credit down the tree T_u using an iterative procedure. Figure 2 shows a small example in which there is a path from a node u to a leaf of T_u consisting entirely of nodes in A, in which all of the siblings of the nodes below u on the path are in $\bar{A}$. First, the two outgoing edges of u in tree T_u receive $\frac{1}{4}$ unit of credit each. Iteratively, each tree edge (v, w) does one of the following:

- If tree edge (v, w) is an edge in cut $(A, \bar{A})$ or if w is a leaf of T_u, edge (v, w) *retains* all the credit it received.
- Otherwise, edge (v, w) retains none of the credit it received and passes half the credit it received to each of the two outgoing tree edges of w.

In a similar fashion, node u distributes $\frac{1}{2}$ unit of credit via the up-tree T'_u rooted at u.

We bound the total units of credit retained by the edges in cut $(A, \bar{A})$ as follows. Each node $u \in A$ distributes 1 unit of credit, of which some portion is retained by edges in cut $(A, \bar{A})$ and the rest is retained by edges (v, w) such that $w \in A$ is a leaf of T_u or T'_u. If an edge (v, w) not in cut $(A, \bar{A})$ retains credit from node u, then there is a path of length $\log n$ from node u to w such that every node in the path belongs to A. Note that a node w may appear as a leaf in both T_u and T'_u. Since there are at most k nodes, $w \in A$, there are at most $2k$ edges not in cut $(A, \bar{A})$ that retain credit from node u, and each such edge retains $1/2^{\log n+1} = 1/(2n)$ units of credit from u. Thus the number of units of credit from node $u \in A$ that is retained by edges in cut$(A, \bar{A})$ is at least $1 - 2k/2n = 1 - k/n$. Since there are k nodes in A, the total units of credit retained by the edges in cut $(A, \bar{A})$ is at least

$$k\left(1 - \frac{k}{n}\right) = (1 - o(1))k, \tag{1}$$

since $k = o(n)$.

Next, we show that each edge in cut $(A, \bar{A})$ retains a total of at most $(\lfloor \log k \rfloor + 1)/4$ units of credit. Let (v, w) be a cut edge such that $v \in A$ and $w \in \bar{A}$. Without loss of generality, let nodes v and w be in levels i and $i + 1 (\mathrm{mod} \log n)$ of W_n, respectively. Let T'_v be the up-tree rooted at node v, and let T_w denote the down-tree rooted at w. The edge (v, w) only retains units of credit that are passed down from T'_v and not up from T_w because w does not have any units of credit to distribute initially, and any units of credit that are passed up through T_w are retained before they reach (v, w). Thus, the cut edge (v, w) retains the most credit when *all* k of the nodes in A are placed in the first $\lfloor \log k \rfloor + 1$ levels of tree T'_v as close to v as possible, i.e., when all nodes in levels 0 through $\lfloor \log k \rfloor - 1$ and some nodes in level $\lfloor \log k \rfloor$ of tree T'_v are in A. Each node $u \in A$ at level $j \geq 0$ of tree T'_v begins with $\frac{1}{2}$ unit of credit to pass down, and $1/2^{j+2}$ units of this credit reach the cut edge (v, w), while the rest of the original $\frac{1}{2}$ unit of credit passed down by u exits the tree T'_v. Since there are 2^j nodes in level j of tree T'_v, the total units of credit retained by cut edge (v, w) is at most

$$\sum_{j=0}^{\lfloor \log k \rfloor} \left(2^j \cdot \frac{1}{2^{j+2}} \right) = \frac{\lfloor \log k \rfloor + 1}{4}. \tag{2}$$

It follows from (1) and (2) that the number of edges in cut $(A, \bar{A})$ is at least

$$C(A, \bar{A}) \geq (1 - o(1))k \cdot \frac{4}{\lfloor \log k \rfloor + 1} \geq (4 - o(1)) \frac{k}{\log k}. \qquad \square$$

Theorem 4.3. *For $k \leq N/2$, the edge-expansion function $EE(W_n, k)$ is $\Theta(k/\log k)$.*

Proof. The upper bound follows from Lemma 4.1. For $k = o(n)$, the lower bound follows from Lemma 4.2.

For $n^{\varepsilon} < k \leq N/2$, for any fixed $\varepsilon > 0$, the bound is proved by embedding K_N into W_n. For example, the following not-too-elegant embedding, π, will do. Suppose that each node w of K_N is given a distinct label $l(w) \in [1, N]$, and that the nodes of K_N are mapped to the nodes of W_n in an arbitrary one-to-one fashion. Let u and v be two nodes in K_N for which $l(u) < l(v)$. Then the path for the edge between u and v in K_N is routed in W_n from $\pi^{-1}(u)$ to $\pi^{-1}(v)$ as follows. First, the path travels monotonically up the column in which $\pi^{-1}(u)$ resides, in order of decreasing level numbers, until it reaches level 0. Then the path travels monotonically along a path of length $\log n$ in order of increasing level number to the node on level $\log n (\mathrm{mod} \log n)$ (i.e., level 0) in the column containing $\pi^{-1}(v)$, following a path of length $\log n$ even if $\pi^{-1}(u)$ and $\pi^{-1}(v)$ are in the same column. Finally, the path travels monotonically from level $\log n (\mathrm{mod} \log n)$ to $\pi^{-1}(v)$ in order of decreasing level number. The dilation of the embedding is $3 \log n - 2$. Note that the paths specified by π are not necessarily simple, and despite the symmetry in both K_N and W_n, the paths are not symmetric, and neither the congestion nor the dilation is uniform. However, for the purposes of proving an asymptotic bound, these properties are not essential.

To analyze the congestion, c, of π it is easiest to bound the congestion due to the three parts of the paths separately. The number of paths originating in a column is less than $N \log n$, as is the number of paths terminating in a column. Hence, the congestion due to the first and last parts of the paths is at most $2N \log n$. To compute the congestion of an edge spanning levels i and $i+1 (\bmod \log n)$ of W_n due to the middle parts of the paths, notice that the edge can be reached via a monotonic path in order of increasing level number from at most 2^i different inputs, and that at most $n/2^{i+1}$ outputs can be reached from the edge. The middle parts of at most $N \log n$ different paths originate at each of these 2^i inputs, but of these at most $N \log n/2^{i+1}$ have destinations in the same columns as the $n/2^{i+1}$ outputs that can be reached from the edge. Hence, the congestion on the edge is at most $(2^i N \log n)/2^{i+1} = (N \log n)/2$. Thus the total congestion c is at most $O(N \log n)$.

As explained in the Section 1.4, $EE(W_n, k) \geq kN/2c$. Since $c = O(N \log n)$ and $\log n = \Theta(\log k)$ (since $n^\varepsilon < k \leq \frac{1}{2} n \log n$, for some $\varepsilon > 0$, by assumption), we have $EE(W_n, k) \geq \Omega(k/\log k)$. This embedding can be adapted to prove the same $\Omega(k/\log k)$ lower bound on $EE(B_n, k)$, and, since both W_n and B_n are bounded-degree networks, for $NE(W_n, k)$ and $NE(B_n, k)$. □

4.1.2. *Node Expansion of* W_n. In this section we determine $NE(W_n, k)$ to within constant factors by proving upper and lower bounds.

Lemma 4.4. *The node-expansion function* $NE(W_n, k)$ *is at most* $(3 + o(1))k/\log k$, *when* $1 \leq k \leq N$.

Proof. Let A consist of two nonintersecting sub-butterflies B' and B'' of W_n such that B' and B'' have $k/2$ nodes each, and such that both B' and B'' are contained in a sub-butterfly B one dimension larger. Each level of the sub-butterflies B' and B'' has $(\frac{1}{2} + o(1))k/\log k$ nodes, and each level of B has $(1+o(1))k/\log k$ nodes. The neighbors in $\bar{A}$ of the input nodes of B' and B'' are the $(1 + o(1))k/\log k$ inputs of B. Each output node of B' and B'' has two nodes in $\bar{A}$ as neighbors. Thus, the output nodes of B' and B'' have $(2 + o(1))k/\log k$ neighbors. Therefore, the total number of nodes in $\mathcal{N}(A)$ is $(3 + o(1))k/\log k$. Therefore, $NE(W_n, k) \leq (3 + o(1))k/\log k$. □

Lemma 4.5. *The node-expansion function* $NE(W_n, k)$ *is at least* $(1 - o(1))k/\log k$, *when* $k = o(n)$.

Proof. To prove the lemma, we use a credit distribution scheme similar to that in the proof of Lemma 4.2. We show that $|\mathcal{N}(A)|$ is at least $(1 - o(1))k/\log k$, for *any* set A with $k = o(n)$ nodes of W_n. Each node $u \in A$ distributes 1 unit of credit to the nodes in $\mathcal{N}(A)$ using the following procedure. Let T_u be the down-tree rooted at node u. Node u passes $\frac{1}{2}$ unit of credit down the tree T_u using an iterative procedure. First, each of the two nodes adjacent to u in tree T_u receive $\frac{1}{4}$ unit of credit each. Iteratively, each node v does one of the following:

- If node v is in $\mathcal{N}(A)$ or if v is a leaf of T_u, node v *retains* all the credit it received.
- Otherwise, node v retains none of the credit it received and passes half the credit it received to each of the two nodes adjacent to v in the next level of T_u.

In a similar fashion, node u distributes $\frac{1}{2}$ unit of credit via the up-tree T'_u rooted at u to the nodes in $\mathcal{N}(A)$.

We bound the total units of credit retained by the nodes in $\mathcal{N}(A)$ as follows. Each node $u \in A$ distributes 1 unit of credit, of which some portion is retained by nodes in $\mathcal{N}(A)$ and the rest is retained by nodes v such that $v \in A$ and v is a leaf of T_u or T'_u. If a node v not in $\mathcal{N}(A)$ retains credit from node u, then there is a path of length $\log n$ from node u to v such that every node in the path belongs to A. There are at most k nodes not in $\mathcal{N}(A)$ that retain credit from node u, and each such node retains at most $2/2^{\log n+1} = 1/n$ units of credit from u (counting units of credit from both T_u and T'_u). Thus, the number of units of credit from node $u \in A$ that is retained by nodes in $\mathcal{N}(A)$ is at least $1 - k/n$. Since there are k nodes in A, the total units of credit retained by the nodes in $\mathcal{N}(A)$ is at least

$$k\left(1 - \frac{k}{n}\right) = (1 - o(1))k, \tag{3}$$

since $k = o(n)$.

Next, we show that each node in $\mathcal{N}(A)$ retains a total of at most $\lfloor \log k \rfloor$ units of credit. Let v be a node in $\mathcal{N}(A)$ and let T_v and T'_v be the down-tree and the up-tree rooted at v, respectively. Node v retains the most units of credit when *all* k of the nodes in A are placed in levels 1 to $\lfloor \log k \rfloor$ of the trees T_v and T'_v as close to v as possible. To obtain an upper bound on the number of units of credit retained by node v, assume that all the nodes in levels 1 to $\lfloor \log k \rfloor$ of T_v and T'_v belong to A. Since each node $u \in A$ at level $j \geq 1$ of tree T_v or T'_v contributes $1/2^{j+1}$ units of credit to node v, and since there are a total of 2^{j+1} nodes in level j of trees T_v and T'_v, the total units of credit retained by node $v \in \mathcal{N}(A)$ is at most

$$\sum_{j=1}^{\lfloor \log k \rfloor} \left(2^{j+1} \cdot \frac{1}{2^{j+1}}\right) = \lfloor \log k \rfloor. \tag{4}$$

It follows from (3) and (4) that the number of nodes in $\mathcal{N}(A)$ is at least

$$(1 - o(1))k \cdot \frac{1}{\lfloor \log k \rfloor} \geq (1 - o(1))\frac{k}{\log k}. \qquad \square$$

Theorem 4.6. *For $k \leq N/2$, the node-expansion function $NE(W_n, k)$ is $\Theta(k/\log k)$.*

Proof. The upper bound follows from Lemma 4.4. For $k = o(n)$, the lower bound follows from Lemma 4.5. For $n^{\varepsilon} < k \leq N/2$, for any fixed $\varepsilon > 0$, the lower bound is proved by embedding K_N into W_n. $\square$

4.2. *Expansion of B_n*

In this section we determine the edge-expansion function $EE(B_n, k)$ and the node-expansion function $NE(B_n, k)$ of B_n. We begin by showing that $EE(B_n, k) = (2 \pm o(1))k/\log k$ for $k = o(\sqrt{n})$. The lower bound cannot be extended to hold for all values of k up to $N/2$ because, for $k = N/2$, the value of the expansion function cannot exceed the bisection width of B_n, which we showed in Section 2 to be $BW(B_n) \leq$

$(2(\sqrt{2}-1)+o(1))n \le (2(\sqrt{2}-1)+o(1))N/\log N$. Hence, $EE(B_n, N/2) \le BW(B_n) \approx (1.66 + o(1))(N/2)/\log(N/2)$. For larger values of k, however, we can use the technique of embedding K_N into B_n, which gives a bound of $EE(B_n, k) = \Omega(k/\log n)$. For $k = n^{\varepsilon}$, for any fixed $\varepsilon > 0$, this lower bound is $\Omega(k/\log k)$. Hence, for all values of k, $EE(B_n, k) = \Theta(k/\log k)$. Our bounds for $NE(B_n, k)$ are not as tight. We show that $(\frac{1}{2} - o(1))k/\log k \le NE(B_n, k) \le (1 + o(1))k/\log k$, for $k = o(\sqrt{n})$. For larger values of k we can again use the technique of embedding K_N into B_n, which yields $NE(W_n, k) = \Theta(k/\log k)$ over the whole range of k.

Definitions. Let node u be in level i of B_n.

The *down-tree* T_u is an $(n/2^i)$-leaf complete binary tree rooted at node u of B_n. Tree T_u is a subgraph of B_n such that the jth level of T_u consists of nodes in level $i + j$ of B_n. Note that the leaves of T_u belong to level $\log n$ of B_n.

The *up-tree* T'_u is a 2^i-leaf complete binary tree rooted at node u of B_n. Let node u be in level i of B_n. Tree T'_u is a subgraph of B_n such that the jth level of T'_u consists of nodes in level $i - j$ of B_n. Note that the leaves of T'_u belong to level 0 of B_n.

4.2.1. *Edge Expansion of* B_n. In this section we determine $EE(B_n, k)$ to within lower-order terms by proving upper and lower bounds.

Lemma 4.7. *The edge-expansion function* $EE(B_n, k)$ *is at most* $(2 + o(1))k/\log k$, *when* $1 \le k \le N$.

Proof. Let A be a sub-butterfly of B_n with k nodes such that the nodes in level 0 of the sub-butterfly are in level 0 of B_n. Each level of the sub-butterfly A has $(1+o(1))k/\log k$ nodes. Each output node of sub-butterfly A has two incident edges that belong to cut $(A, \bar{A})$. Thus, the total number of edges in cut $(A, \bar{A})$ is $(2 + o(1))k/\log k$. Therefore, $EE(B_n, k) \le (2 + o(1))k/\log k$. □

Lemma 4.8. *The edge-expansion function* $EE(B_n, k)$ *is at least* $(2 - o(1))k/\log k$, *when* $k = o(\sqrt{n})$.

Proof. Let A be *any* subset of the nodes of B_n of cardinality k. To prove the lemma, we use a credit distribution scheme to show that $C(A, \bar{A})$ is at least $(2 - o(1))k/\log k$. Each node $u \in A$ distributes 1 unit of credit to the edges in cut $(A, \bar{A})$ using the following procedure. First, assume that node $u \in A$ is in level i of B_n such that $0 \le i < \lfloor(\log n + 1)/2\rfloor$. Let T_u be the down-tree rooted at node u. Furthermore, let the edges of T_u be directed from root to leaf. Node u passes 1 unit of credit down the tree T_u using an iterative procedure. First, the two outgoing edges of u in tree T_u receive $\frac{1}{2}$ unit of credit each. Iteratively, each tree edge (v, w) does one of the following:

- If tree edge (v, w) is in cut $(A, \bar{A})$ or if w is a leaf of T_u, edge (v, w) *retains* all the credit it received.
- Otherwise, edge (v, w) retains none of the credit it received, and passes half the credit it received to each of the two outgoing tree edges of w.

In a similar fashion, each node $u \in A$ in level i of B_n, such that $\lfloor(\log n + 1)/2\rfloor \le i \le \log n$, distributes 1 unit of credit via the up-tree T'_u rooted at u.

We bound the total number of units of credit retained by edges in cut $(A, \bar{A})$ as follows. Each node $u \in A$ distributes 1 unit of credit, of which some portion is retained by edges in cut $(A, \bar{A})$ and the rest is retained by edges (v, w) such that $w \in A$ is a leaf of T_u or T'_u. If an edge (v, w) not in cut $(A, \bar{A})$ retains credit from node u, there is a path of length at least $\lfloor(\log n + 1)/2\rfloor$ from node u to w such that every node in the path belongs to A. Therefore, there are at most k edges not in cut $(A, \bar{A})$ that retain credit from node u, and each such edge retains at most

$$1/2^{\lfloor(\log n+1)/2\rfloor} \le 1/\sqrt{n}$$

units of credit. Thus, the number of units of credit distributed by a node $u \in A$ that is retained by edges in cut$(A, \bar{A})$ is at least $1 - (k/\sqrt{n})$. Since there are k nodes in A, the total units of credit retained by the edges in cut $(A, \bar{A})$ is at least

$$k\left(1 - \frac{k}{\sqrt{n}}\right) = (1 - o(1))k, \tag{5}$$

since $k = o(\sqrt{n})$.

Next, we show that each edge in cut $(A, \bar{A})$ retains a total of at most $(\lfloor\log k\rfloor + 1)/2$ units of credit. Let (v, w) be a cut edge such that $v \in A$ and $w \in \bar{A}$. Without loss of generality, let nodes v and w be in levels i and $i + 1$ of B_n, respectively. Let T'_v be the up-tree rooted at node v. The cut edge (v, w) retains the most number of units of credit when *all* k of the nodes of A are placed in the first $\lfloor\log k\rfloor + 1$ levels of tree T'_v as close to v as possible, i.e., when all nodes of levels 0 through $\lfloor\log k\rfloor - 1$ and some nodes in level $\lfloor\log k\rfloor$ of tree T'_v are in A. Since each node $u \in A$ at level $j \ge 0$ of tree T'_v contributes $1/2^{j+1}$ units of credit to cut edge (v, w), and since there are 2^j nodes in level j of tree T'_v, the total units of credit retained by cut edge (v, w) is at most

$$\sum_{j=0}^{\lfloor\log k\rfloor}\left(2^j \cdot \frac{1}{2^{j+1}}\right) = \frac{\lfloor\log k\rfloor + 1}{2}. \tag{6}$$

It follows from (5) and (6) that the number of edges in cut $(A, \bar{A})$ is at least

$$C(A, \bar{A}) \ge (1 - o(1))k \cdot \frac{2}{\lfloor\log k\rfloor + 1} \ge (2 - o(1))\frac{k}{\log k}. \qquad \square$$

Theorem 4.9. *For $k \le N/2$, the edge-expansion function $EE(B_n, k)$ is $\Theta(k/\log k)$.*

Proof. The upper bound follows from Lemma 4.7. For $k = o(\sqrt{n})$, the lower bound follows from Lemma 4.8. For $n^\varepsilon < k \le N/2$, for any fixed $\varepsilon > 0$, the lower bound is proved by embedding K_N into B_n. $\square$

4.2.2. *Node Expansion of B_n.* In this section we determine $NE(B_n, k)$ to within constant factors by proving upper and lower bounds.

Lemma 4.10. *The node-expansion function $NE(B_n, k)$ is at most $(1 + o(1))k/\log k$, when $1 \leq k \leq N$.*

Proof. Let A consist of two nonintersecting sub-butterflies B' and B'' of B_n such that B' and B'' have $k/2$ nodes each, both B' and B'' are contained in B, which is one dimension larger, and the output nodes of B', B'', and B are on level $\log n$ of B_n. Each level of the sub-butterflies B' and B'' has $(\frac{1}{2} + o(1))k/\log k$ nodes, and each level of B has $(1 + o(1))k/\log k$ nodes. The neighbors in $\bar{A}$ of the input nodes of B' and B'' are the $(1 + o(1))k/\log k$ inputs of B. The output nodes of B' and B'' have no neighbors in $\bar{A}$. Therefore, the total number of nodes in $\mathcal{N}(A)$ is $(1 + o(1))k/\log k$. Therefore, $NE(B_n, k) \leq (1 + o(1))k/\log k$. □

Lemma 4.11. *The node-expansion function $NE(B_n, k)$ is at least $(\frac{1}{2} - o(1))k/\log k$, when $k = o(\sqrt{n})$.*

Proof. Let A be *any* subset of the nodes of B_n of cardinality k. To prove the lemma, we use a credit distribution scheme similar to that in the proof of Lemma 4.8. We show that $|\mathcal{N}(A)|$ is at least $(\frac{1}{2} - o(1))k/\log k$. Each node $u \in A$ distributes 1 unit of credit to the nodes in $\mathcal{N}(A)$ using the following procedure. First, assume that node $u \in A$ is in level i of B_n such that $0 \leq i < \lfloor(\log n + 1)/2\rfloor$. Let T_u be the down-tree rooted at node u. Node u passes 1 unit of credit down the tree T_u using an iterative procedure. First, the two nodes adjacent to u in tree T_u receive $\frac{1}{2}$ unit of credit each. Iteratively, each node v does one of the following:

- If node v is in $\mathcal{N}(A)$ or if v is a leaf of T_u, node v *retains* all the credit it received.
- Otherwise, node v retains none of the credit it received, and passes half the credit it received to each of the two nodes adjacent to v in the next level of tree T_u.

In a similar fashion, each node $u \in A$ in level i of B_n, such that $\lfloor(\log n + 1)/2\rfloor \leq i \leq \log n$, distributes 1 unit of credit via the up-tree T'_u rooted at u.

We bound the total number of units of credit retained by nodes in $\mathcal{N}(A)$ as follows. Each node $u \in A$ distributes 1 unit of credit, of which some portion is retained by nodes in $\mathcal{N}(A)$ and the rest is retained by nodes v such that $v \in A$ is a leaf of T_u or T'_u. If a node v not in $\mathcal{N}(A)$ retains credit from node u, there is a path of length at least $\lfloor(\log n + 1)/2\rfloor$ from node u to v such that every node in the path belongs to A. Therefore, there are at most k nodes that do not belong to $\mathcal{N}(A)$ that retain credit from node u, and each such node retains at most

$$1/2^{\lfloor(\log n+1)/2\rfloor} \leq 1/\sqrt{n}$$

units of credit. Thus, the number of units of credit distributed by a node $u \in A$ that is retained by nodes in $\mathcal{N}(A)$ is at least $1 - (k/\sqrt{n})$. Since there are k nodes in A, the total units of credit retained by the nodes in $\mathcal{N}(A)$ is at least

$$k\left(1 - \frac{k}{\sqrt{n}}\right) = (1 - o(1))k, \tag{7}$$

since $k = o(\sqrt{n})$.

Next, we show that each node $v \in \mathcal{N}(A)$ retains a total of at most $2\lfloor \log k \rfloor$ units of credit. Let T_v and T'_v be the down-tree and up-tree rooted at node v, respectively. The node v retains the most number of units of credit when *all* k of the nodes of A are placed in levels 1 to $\lfloor \log k \rfloor$ of the trees T_v and T'_v as close to v as possible. To obtain an upper bound on the number of units of credit retained by node v, assume that all the nodes in levels 1 to $\lfloor \log k \rfloor$ of T_v and T'_v belong to A. Since each node $u \in A$ at level $j \geq 1$ of tree T_v or T'_v contributes $1/2^j$ units of credit to node v, and since there are a total of 2^{j+1} nodes in level j of trees T_v and T'_v, the total units of credit retained by node $v \in \mathcal{N}(A)$ is at most

$$\sum_{j=1}^{\lfloor \log k \rfloor} \left(2^{j+1} \cdot \frac{1}{2^j}\right) = 2\lfloor \log k \rfloor. \tag{8}$$

It follows from (7) and (8) that the number of nodes in cut $\mathcal{N}(A)$ is at least

$$(1 - o(1))k \cdot \frac{1}{2\lfloor \log k \rfloor} \geq \left(\tfrac{1}{2} - o(1)\right)\frac{k}{\log k}. \qquad \square$$

Theorem 4.12. *For $k \leq N/2$, the node-expansion function $NE(B_n, k)$ is $\Theta(k/\log k)$.*

Proof. The upper bound follows from Lemma 4.10. For $k = o(\sqrt{n})$, the lower bound follows from Lemma 4.11. For $n^{\varepsilon} < k \leq N/2$, for any fixed $\varepsilon > 0$, the lower bound is proved by embedding K_N into B_n. $\square$

4.3. *Summary*

The lower bounds proved in this section are summarized below:

	$k = o(\sqrt{n})$	$k = o(n)$	$k \leq N/2$
$EE(W_n, k)$		$(4 - o(1))k/\log k$	$\Omega(k/\log k)$
$NE(W_n, k)$		$(1 - o(1))k/\log k$	$\Omega(k/\log k)$
$EE(B_n, k)$	$(2 - o(1))k/\log k$		$\Omega(k/\log k)$
$NE(B_n, k)$	$(\frac{1}{2} - o(1))k/\log k$		$\Omega(k/\log k)$

The upper bounds are shown below:

	$k \leq N$
$EE(W_n, k)$	$(4 + o(1))k/\log k$
$NE(W_n, k)$	$(3 + o(1))k/\log k$
$EE(B_n, k)$	$(2 + o(1))k/\log k$
$NE(B_n, k)$	$(1 + o(1))k/\log k$

Acknowledgments

The authors thank two anonymous referees for their many thoughtful comments and suggestions.

References

[1] M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in $c \log n$ parallel steps. *Combinatorica*, 3:1–19, 1983.

[2] S. Arora, F. T. Leighton, and B. M. Maggs. On-line algorithms for path selection in a non-blocking network. *SIAM Journal on Computing*, 25(3):600–625, June 1996.

[3] A. Avior, T. Calamoneri, S. Even, A. Litman, and A. L. Rosenberg. A tight layout of the butterfly network. *Theory of Computing Systems*, 31(4):475–488, July/August 1998.

[4] Butterfly^{TM} parallel processor overview. BBN Report No. 6148, Version 1, BBN Advanced Computers, Inc., Cambridge, MA, March 1986.

[5] B. Beizer. The analysis and synthesis of signal switching networks. In *Proceedings of the Symposium on Mathematical Theory of Automata*, pages 563–576, Brooklyn, NY, 1962.

[6] V. E. Beneš. Optimal rearrangeable multistage connecting networks. *Bell System Technical Journal*, 43:1641–1656, July 1964.

[7] P. de la Torre and C. P. Kruskal. Exact bandwidth of parallel processor-networks. In *Proceedings of the 3rd International Colloquium on Structural Information and Communcation Complexity* (*SIROCCO '96*), pages 195–209, June 1996.

[8] B. Ghosh, F. T. Leighton, B. M. Maggs, S. Muthukrishnan, C. G. Plaxton, R. Rajaraman, A. W. Richa, R. E. Tarjan, and D. Zuckerman. Tight analyses of two local load balancing algorithms. *SIAM Journal on Computing*, 29(1):29–64, February 1999.

[9] A. Gottlieb. An overview of the NYU Ultracomputer Project. In J. J. Dongarra, editor, *Experimental Parallel Computing Architectures*, pages 25–95. Elsevier, Amsterdam, 1987.

[10] D. S. Greenberg, L. S. Heath, and A. L. Rosenberg. Optimal embeddings of butterfly-like graphs in the hypercube. *Mathematical Systems Theory*, 23:61–77, 1990.

[11] J.-W. Hong and H. T. Kung. I/O complexity: the red–blue pebble game. In *Conference Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*, pages 326–333, May 1981.

[12] R. R. Koch, F. T. Leighton, B. M. Maggs, S. B. Rao, A. L. Rosenberg, and E. J. Schwabe. Work-preserving emulations of fixed-connection networks. *Journal of the ACM*, 44(1):104–147, January 1997.

[13] C. P. Kruskal and M. Snir. Optimal interconnection networks for parallel processors: the importance of being square. In Y. Yemeni, editor, *Current Advances in Distributed Computing and Communications*, pages 91–113. Computer Science Press, Rockville, MD, 1987.

[14] C. P. Kruskal and M. Snir. Cost-performance tradeoffs for communication networks. *Discrete Applied Mathematics*, 37–38:359–385, July 1992.

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

[16] F. T. Leighton, M. Lepley, and G. L. Miller. Layouts for the shuffle-exchange graph based on the complex plane diagram. *SIAM Journal of Algebraic and Discrete Methods*, 5(2):202–215, June 1984.

[17] F. T. Leighton and B. M. Maggs. Fast algorithms for routing around faults in multi-butterflies and randomly-wired splitter networks. *IEEE Transactions on Computers*, 41(5):578–587, May 1992.

[18] B. M. Maggs and E. J. Schwabe. Real-time emulations of bounded-degree networks. *Information Processing Letters*, 66(5):269–276, 1998.

[19] B. M. Maggs and B. Vöcking. Improved routing and sorting on multibutterflies. *Algorithmica*, 28(4):438–464, 2000.

[20] Y. Manabe, K. Hagihara, and N. Tokura. The minimum bisection widths of the cube-connected cycles graph and the cube graph. *Transactions of the IEICE* (*D*), J67-D(6):647–654, June 1984. In Japanese.

[21] T. Nakata, S. Matsushita, N. Tanabe, N. Kajihara, H. Onozuka, Y. Asano, and N. Koike. Parallel programming on Cenju: a multiprocessor system for modular circuit simulation. *NEC Research & Development*, 32(3):421–429, July 1991.

[22] W. Oed. Cray Y-MP C90: system features and early benchmark results. *Parallel Computing*, 18(8):947–954, August 1992.

[23] G. F. Pfister, W. C. Brantley, D. A. George, S. L. Harvey, W. J. Kleinfelder, K. P. McAuliffe, E. A. Melton, V. A. Norton, and J. Weiss. An introduction to the IBM research parallel processor prototype (RP3). In J. J. Dongarra, editor, *Experimental Parallel Computing Architectures*, pages 123–140. Elsevier, Amsterdam, 1987.

[24] F. P. Preparata and J. Vuillemin. The cube-connected cycles: a versatile network for parallel computation. *Communications of the ACM*, 24(5):300–309, May 1981.

[25] K. J. Rappoport. On the slowdown of efficient simulations of multibutterflies. In *Proceedings of the Eighth ACM Symposium on Parallel Algorithms and Architectures*, pages 176–182, June 1996.

[26] E. J. Schwabe. On the computational equivalence of hypercube-derived networks. In *Proceedings of the Second Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 388–397, July 1990.

[27] M. Snir. I/O limitations on multi-chip VLSI systems. In *Proceedings of the Nineteenth Annual Allerton Conference on Communication, Control, and Computing*, pages 224–233, September 1981.

[28] C. D. Thompson. A Complexity Theory for VLSI. Ph.D. thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1980.

[29] E. Upfal. An $O(\log N)$ deterministic packet routing scheme. *Journal of the ACM*, 39(1):55–70, January 1992.

[30] A. Waksman. A permutation network. *Journal of the ACM*, 15(1):159–163, January 1968.

Received September 30, 1997*, and in final form July* 30, 2001. *Online publication November* 23, 2001.