In Proceedings of the 1994 Symposium on the Theory of Computing
Montreal, Canada, May 23-25, 1994.

Scalable Expanders: Exploiting Hierarchical Random Wiring

Eric A. Brewer*
brewer@lcs.mit.edu

Frederic T. Chong†
ftchong@ai.mit.edu

F. Thomson Leighton‡
ftl@math.mit.edu

MIT Laboratory for Computer Science
MIT Artificial Intelligence Laboratory
MIT Department of Mathematics

Abstract

Recent work has shown many advantages to randomly wired expander-based networks. Unfortunately, the wiring complexity of such networks becomes physically problematic as they become large. This paper introduces a technique for scaling expanders that avoids this wiring complexity. Specifically, we make the following contributions:

1. *We introduce* hierarchical expanders, *which use a method of scaling small expanders to larger ones while maintaining practical physical construction. We present an example of such a scalable network, called the* metabutterfly, *which is scaled from the randomly wired multibutterfly.*
2. *We present a proof that we can scale any (α, β, M, N)-expander with $\alpha M \geq 1$ into an $(\alpha', \beta', kM, kN)$-expander with probability at least $1 - 2e^{-\alpha M}$, where $\alpha' = \frac{\alpha^2}{\beta^2 e^4 + 4\alpha}$ and $\beta' = \beta - 2$.*
3. *We present empirical evidence that the performance and fault tolerance of metabutterflies equals that of traditional randomly wired multibutterflies, despite the greatly simplified wiring of the metabutterfly.*

1 Introduction

This paper introduces *hierarchical expanders*, expanders that are scalable in practice. We construct these expanders with a transformation called a *k-extension*, which we prove can preserve expansion. We also introduce the *metabutterfly*, a novel hierarchical expander network constructed by k-extending a randomly wired multibutterfly. Finally, we present the results of detailed simulations of the metabutterfly that show that its performance and fault tolerance match those of the multibutterfly.

Although the theoretical results apply to expanders in general, this work was motivated by practical limitations on the scalability of randomly wired multibutterflies. In this section, we discuss the advantages of randomly wired multibutterflies and the difficulties encountered in the construction of large ones. We propose our solution to these problems, *hierarchical expanders*, in Section 2. In Section 3, we present a probabilistic analysis that proves that, with high probability, *k-extension* preserves expansion. Section 4 presents our empirical results, and Section 5 discusses some open issues.

1.1 Definitions

A *splitter network* is composed of multiple stages of routers, organized into splitters. The canonical butterfly is an example of a splitter network. It is helpful to view routing a message through a splitter network as a sorting function through equivalence classes of fewer and fewer routers. Specifically, for the i^{th} stage there are r^i equivalence classes, each with r^{s-i} routers, where r is the *radix* of the routers (the number of directions among which the router selects) and s is the number of stages in the network. Each equivalence class is connected to r equivalence classes in the next stage. An individual splitter consists of an equivalence class and its r associated equivalence classes in the next stage.

A bipartite graph with M inputs and N outputs is an (α, β, M, N)-*expander* if every set of $m \leq \alpha M$ inputs reaches at least βm outputs. For a radix-r splitter network to have expansion, each splitter must achieve expansion in each of the r directions. To achieve expansion, a splitter network must have routers with redundant connections in each of its r directions. We refer to this redundancy, d, as the *multiplicity*. The degree of any node in the splitter is then dr.

1.2 Multibutterflies

A *multibutterfly* is a splitter network with expansion. In particular, each M-input splitter of a multibutterfly is an $(\alpha, \beta, M, \frac{M}{r})$-expander in each of the r directions.

Bassalygo and Pinsker [BP74] first studied splitter networks with expansion. Recently, numerous results have

*Eric Brewer is supported in part by the National Science Foundation, grant CCR-8716884; by ARPA, contract N00014-91-J-1698; by an equipment grant from Digital Equipment Corporation; and by grants from AT&T and IBM.

†Fred Chong is supported in part by an Office of Naval Research Graduate Fellowship and ARPA contract N00014-91-J-1698

‡Tom Leighton is supported in part by Air Force Contract AFOSR F49620-92-J0125 and ARPA contracts N00014-91-J-1698 and N00014-92-J-1799.

been discovered that indicate that multibutterflies are ideally suited for message-routing applications. Among other things, multibutterflies can solve any one-to-one packet routing [Upf89], circuit-switching [ALM90], or non-blocking routing problem [ALM90] in optimal time, even if many of the routers in the network are faulty [LM89]. No other networks are known to be as powerful.

The reason behind the power of multibutterflies is that expansion roughly implies that βp outputs must be blocked or faulty for p inputs to be blocked, and thus it takes β^j faults to block one input j levels back. In contrast, *one* fault in a radix-2 butterfly blocks 2^j inputs j levels back. As a consequence, problems with faults and congestion that destroy the performance of traditional networks can be easily overcome in multibutterflies. (For a survey of the research on multibutterflies see [Pip93] [LM92].)

1.3 Wiring Complexity

Multibutterflies are generally constructed by randomly wiring redundant connections between the equivalence classes of each splitter. Although deterministic constructions are known [WZ93], none are known to produce expansion comparable to random wiring.

Unfortunately, random wiring and the known deterministic constructions of good expanders scale poorly in practice. For example, a 4096-endpoint machine with multiplicity $d = 2$ has 8192 wires in the first stage, almost all of which would be long cables with distinct logical endpoints. For comparison, a fat-tree [Lei85] might have a similar number of cables for the root node, but there are few logical endpoints, so huge groups of wires can be routed together. The groups connect to many boards, but the boards are located together and the connection of cables to boards is arbitrary and thus low labor. In the multibutterfly, the cables cannot be grouped and the connection of cables to boards is constrained. The other early stages also suffer from this problem.

At first glance, it appears that this wiring complexity is inherent to both expanders and random wiring. Indeed, given a splitter with M boards of input routers, M boards of output routers, and b routers per board, we can expect each board to be connected to about $\min(M, dbr)$ other boards when using random wiring. For typical values of M, d, b, and r, this means that we would need to connect *every* input board to *every* output board in a randomly wired splitter. Clearly, this becomes infeasible as M gets large and thus the randomly wired multibutterfly does not scale well in the practical setting where the network consists of boards of chips. A similar problem arises at the level of cabinets of boards for very large machines.

In what follows, we show how to (randomly) construct a special kind of expander for which there is no explosion in cabling cost. In particular, we show how to build a multibutterfly for which each board is connected to only dr other boards, no matter how large M and b become, thereby achieving full scalability. In effect, there are a few fat cables connected to each board instead of many thin cables. At the same time, the resulting network will still have all the same nice routing properties as a randomly wired multibutterfly. Hence, we gain scalability without sacrificing performance or fault tolerance.

In fact, once we have constructed a network with dr cables per board and k wires per cable, we then have the option to decrease the number of wires in each cable by multiplexing the logical connections among fewer physical wires. In effect, we can then have a few thin cables connecting to each board instead of a few fat cables, thereby decreasing the number of wires. This flexibility allows us to further reduce cabling and wiring cost; in particular, we can adjust the thickness of the cable based on the average load, which is significantly less than the peak load for a large group of wires. As long as the cable can handle the average load well, most traffic remains unaffected by the use of fewer wires.

In turn, if we are pin-limited on the board-level (e.g., say each board has only drb pins), decreasing the physical size of each cable, thereby cutting b, would allow us to increase dr without altering the pin count. Increasing d gives greater expansion, which results in better routing performance, and increasing r allows for fewer levels in the network, which results in less routing delay and lower hardware cost. Such design options could prove to be very valuable and are not available with traditional multibutterflies, because they provide no physical locality for wires.

2 Hierarchical Expanders

The wiring complexity of large expanders can be dramatically decreased by constructing them hierarchically. A *hierarchical expander* is an expander constructed from the application of a sequence of *random k-extensions* to an expander.

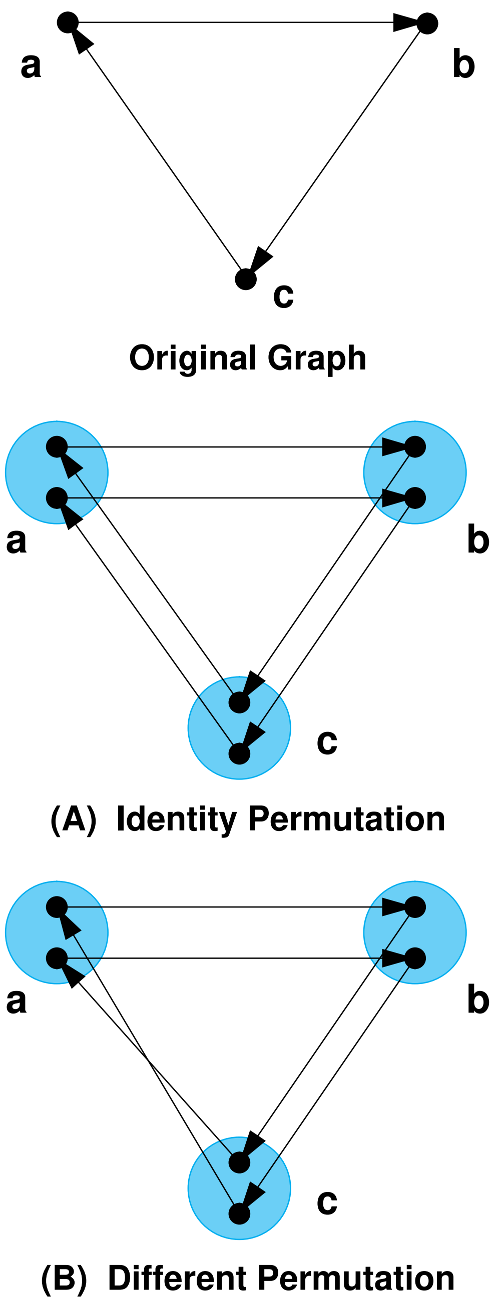
Given a directed graph $G \equiv (\mathcal{V}, \mathcal{E})$, an integer $k \geq 1$, and a set of permutations of [1,k], $\Pi = \{\pi_e | e \in \mathcal{E}\}$, we define the *k-extension of G induced by* Π to be the graph $G' \equiv (\mathcal{V}', \mathcal{E}')$ where:

$$\begin{aligned}\mathcal{V}' &\equiv \{\langle v, i\rangle \mid v \in \mathcal{V},\ i \in [1,k]\} \text{ and} \\ \mathcal{E}' &\equiv \{(\langle u, i\rangle, \langle v, j\rangle) \mid \\ &\qquad (u,v) \in \mathcal{E} \text{ and } \pi_{(u,v)}(i) = j\}\end{aligned}$$

Note that $|\mathcal{V}'| = k|\mathcal{V}|$ and $|\mathcal{E}'| = k|\mathcal{E}|$.

For example, two 2-extensions of a three-cycle are shown in Figure 1. Note that 2-extension (A) results in two disconnected copies of the original graph. In general, if all $\pi_e \in \Pi$ are the identity permutation, then the k-extension consists of k disjoint copies of the original graph.

Each edge in the original graph corresponds to k edges in the extended graph. These groups of k edges are called *channels*. The group of k nodes that correspond to one node in the original graph form one *metanode*; metanodes are shown in gray in Figure 1. The metanode/channel structure of G' is isomorphic to the vertex/edge structure of G.



$$\pi_{(a,b)} = \pi_{(b,c)} = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$$

$$\pi_{(c,a)} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

Figure 1: 2-extensions of a 3-cycle

We define a *random k-extension of a graph G* to be a k-extension induced by some Π such that each $\pi_e \in \Pi$ is an independently and uniformly chosen random permutation of $[1,k]$. Equivalently, a random k-extension of a graph G can be obtained by selecting randomly and uniformly over all of the $(k!)^{|\mathcal{E}|}$ possible k-extensions of G. In Section 3 we prove that random k-extensions preserve expansion, with very high probability, for any k.

2.1 Metabutterflies

A *metabutterfly* is a splitter network that is constructed from a multibutterfly through random k-extensions. Each splitter of the metabutterfly is a random k-extension of the corresponding splitter of the multibutterfly, with the possible exception of the last few stages. The value of k may differ for each splitter.

For the late stages of an M-input, radix-r multibutterfly, the splitters may be expanders only because $\alpha M_i < 1$, where $M_i \equiv \frac{M}{r^i}$ is the input size of an i^{th}-stage splitter. In other words, the late stages are not really providing expansion, since only sets of size zero get expansion. In this case, we construct the metabutterfly splitter, which has $M_i k$ inputs, out of an $M_i k$-input multibutterfly splitter. This avoids hierarchical wiring, but does not affect the practical scalability because M_i is small. Furthermore, the replaced splitters are typically complete bipartite graphs, in which case hierarchical wiring does not reduce the wiring complexity. Alternatively, in practice it may simpler and sufficient to k extend these end stages, even though the resulting splitters may not provably have expansion. The simulations presented in Section 4 use this simplification.

If all of the stages actually expand sets of size at least one, then we replace the final output metanodes, which each have k nodes, with k-input multibutterflies. This ensures that the network resolves destinations to the correct node rather than just the correct metanode.

For example, a 1024-node metabutterfly can be implemented with a 64-extended 16-node multibutterfly plus 16 64-node multibutterflies for the output metanodes. The total counts of nodes, wires, and stages are each the same as for a 1024-node multibutterfly; the only difference is the wiring pattern. Figure 2 shows a radix-2 64-input metabutterfly that is an 8-extended 8-input multibutterfly.

Unlike the multibutterfly, in which the first-stage wiring is unconstrained, the connections are constrained into a two-level hierarchy. The top level of the hierarchy is the channel wiring, which reduces the number of inter-metanode connections from roughly $M\frac{M}{r}$ to at most Md, where d is the multiplicity. The wires within the channels form the second level of the hierarchy and do not affect the number of inter-metanode connections.

For example, for a 4096-processor machine with metanodes of size 64, the number of logical endpoints has been reduced from 4096 to $\frac{4096}{64} = 64$. With $d = 2$, this takes us from 8192 individual wires per stage to 128 groups of 64

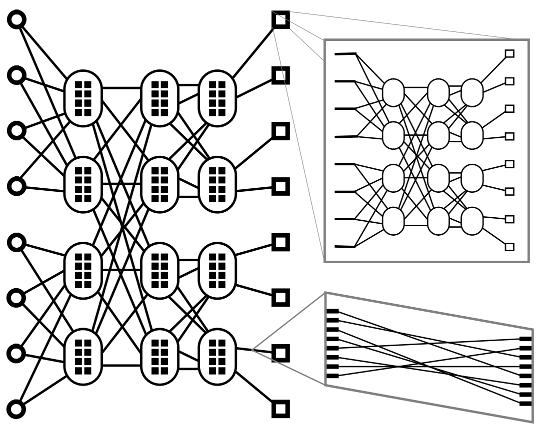


Figure 2: A radix-2, multiplicity-2, 64-endpoint metabutterfly with metanodes of size 8. Each circle on the left contains 8 inputs, and each oval metanode contains 8 routers. Each router (solid square) is a 4×4 switch. Each output metanode (hollow square), shown expanded at the top right, is an 8-input multibutterfly; each channel, shown expanded at the bottom right, contains 8 wires. Typically, the metanodes correspond to boards.

wires, with each group routed as a unit. The wires within a group can be connected to the endpoint routers arbitrarily, since any (random) one-to-one mapping is sufficient.

This example has a two-level hierarchy, but deeper hierarchies are possible and actually make sense for very large networks. For example, if a two-level hierarchy requires that k be very large, then it may not be possible to group k nodes onto one board. A three-level hierarchy provides k^2 times as many nodes, which allows a much smaller k for the same total number of inputs. For example, a 64K-node machine might be constructed as a (64, 16)-extended 64-input multibutterfly ($64 \times 16 \times 64 = 64$K). Each board would contain 16 nodes, 64 boards would be assembled as one cabinet, and the 64 cabinets would be connected as a 64-input multibutterfly (with very thick inter-cabinet cables). The top level of the hierarchy simplifies the inter-cabinet wiring, and the second level allows the large inter-cabinet cables to be constructed as groups of 64 inter-board cables. Connecting the inter-board cables to the boards is trivial, since the assignment is random. Finally, note that the boards within a cabinet are not connected; they are located together only for wiring convenience. Likewise, the routers on a board are completely independent.

With large cables, the option of multiplexing becomes both more cost effective and more likely to deliver good performance. For example, optical fiber would provide enough bandwidth to replace very large cables. Note that a packet routing scheme across such fiber would provide little degradation in performance until the load reached bandwidth limits. The number and capacity of fibers can be designed to accomodate expected load. The larger the original cable being multiplexed, the more likely the average load will be significantly lower than the peak load. We can exploit such differences to build more cost-effective networks.

The relationship between metabutterflies and multibutterflies is quite interesting. The set of all metabutterfly wirings is a strict subset of the set of all randomly wired multibutterfly wirings. However, it does not follow automatically that a metabutterfly has expansion with high probability! Since the metabutterfly allows only a subset of the random wirings, the percentage of bad wirings may no longer be vanishingly small. A primary result of this paper is that metabutterflies are in fact multibutterflies; that is, all the splitters of the metabutterfly have expansion.

We also present simulation results in Section 4 that show that the performance and fault tolerance of the metabutterfly is statistically indistinguishable from that of the multibutterfly. This is somewhat surprising since the metabutterfly constrains the randomness of the wiring in order to ensure that the network remains scalable in practice. Thus, the metabutterfly provides the size, performance and fault tolerance of a large multibutterfly, but with the wiring complexity of a small one.

3 Theoretical Results

The k-extension of a graph inherits many of the properties of the underlying graph. For example, if the underlying graph is d-regular, then so is the k-extension. In this section, we prove a somewhat more difficult and important fact, namely, that if G is an expander, then a random k-extension of G is also an expander with very high probability.

It is useful to establish some intuition about the expansion of a k-extension given that the original graph is an (α, β, M, N)-expander. First, if a node in the original graph has d neighbors, then each of the k nodes in the corresponding metanode have d neighbors, all of which are distinct, for a total of dk nodes. Extending this notion, if a set, $\mathcal{S}$, of size $m \leq \alpha M$ nodes in the original graph expands to set $\mathcal{T}$ of size βm, then the corresponding set of metanodes, which contain km nodes, expands to βkm nodes covering βm metanodes. This gives us an expansion factor of β for any k.

But it does not follow that the k-extension is an (α, β, Mk, Nk)-expander. In particular, if the k-extension were such an expander, then *any* set of at most $\alpha M k$ nodes must achieve β expansion. The argument given above requires that the $\alpha M k$ selected nodes cover at most αM metanodes and be spread evenly among metanodes (although we can avoid the latter restriction). If the set covers more metanodes, then the expansion of the underlying graph does not apply, since $|\mathcal{S}| > \alpha M$. However, for the k-extension the $\alpha M k$ restriction *does* apply. Thus, the difficult part of showing that the k-extension is an expander is handling the case in which the selected set, with size at most $\alpha M k$, covers more than αM metanodes.

$$\begin{bmatrix} s_1 & s_{\lfloor \alpha M \rfloor+1} & \cdots & s_{(C-1)\lfloor \alpha M \rfloor+1} \\ s_2 & s_{\lfloor \alpha M \rfloor+2} & \cdots & s_{(C-1)\lfloor \alpha M \rfloor+2} \\ \vdots & \vdots & \vdots & \vdots \\ s_{\lfloor \alpha M \rfloor} & s_{2\lfloor \alpha M \rfloor} & \cdots & s_{C\lfloor \alpha M \rfloor} \end{bmatrix}$$

Figure 3: The structure of $\mathbf{F}$.

It is easy to show that the k-extension is an $(\alpha' \equiv \frac{\alpha}{k}, \beta, Mk, Nk)$-expander. By limiting α' to $\frac{\alpha}{k}$ we know that the selected set has size at most $\frac{\alpha M k}{k} = \alpha M$ and thus can cover at most αM metanodes, which avoids the difficult case. Naturally, we would like the expansion to be independent of k.

If we keep α' independent of k, however, then not all k-extensions are expanders, since some of the extensions are not even connected, as shown in Figure 1(A). In particular, if we choose k large enough so that the size of one copy is less than $\alpha' Mk$, then the copy must expand. However, the copy is disconnected from the rest of the graph and thus can not expand. Thus, not all k-extensions of an expander are expanders.

Fortunately, the following result shows that the vast majority of k-extensions of an expander are also expanders, for any k. We will later use this fact to prove that given a multibutterfly with sufficient expansion, then, with very high probability, each splitter of a metabutterfly will have expansion, since it is a random k-extension of the corresponding multibutterfly splitter.

Theorem 1 *If $G \equiv (\mathcal{U} \cup \mathcal{V},\ \mathcal{E})$ is an (α, β, M, N)-expander from $\mathcal{U}$ to $\mathcal{V}$, with $\alpha M \geq 1$, then for any $k \geq 1$, a random k-extension $G' \equiv (\mathcal{U}' \cup \mathcal{V}',\ \mathcal{E}')$ of G is an $(\alpha', \beta', kM, kN)$-expander from $\mathcal{U}'$ to $\mathcal{V}'$ with probability at least $1 - 2e^{-\alpha M}$, where $\alpha' = \frac{\alpha^2}{\beta^2 e^4 + 4\alpha}$ and $\beta' = \beta - 2$.*

Proof: In what follows, we refer to the edges of G as *channels* in the k-extension of G. In addition, the nodes of G correspond to *metanodes* in G'. Consequently, we use the sets $\mathcal{U}$ and $\mathcal{V}$ when referring to either the nodes of G or the metanodes of G'. We use the sets $\mathcal{U}'$ and $\mathcal{V}'$ when referring to the nodes of G'.

Let $\mathcal{S}$ be any subset of $\mathcal{U}'$ with at most $\alpha' kM$ nodes, and let $\mathcal{N}(\mathcal{S}) \subseteq \mathcal{V}'$ be the neighborhood of $\mathcal{S}$, which is the set to which $\mathcal{S}$ expands. In order to show that G' is an expander, we must show that $|\mathcal{N}(\mathcal{S})| \geq \beta' |\mathcal{S}|$ for all $\mathcal{S}$.

We define s_i to be the number of nodes of $\mathcal{S}$ contained in the i^{th} metanode of $\mathcal{U}$ and we order the metanodes so that $s_1 \geq s_2 \geq \cdots \geq s_M$. Next, we arrange the s_i's into a matrix, $\mathbf{F} \equiv \{f_{i,j}\}$, with $\lfloor \alpha M \rfloor \geq 1$ rows, so that the values appear in column-major order. That is, $f_{i,j} \geq f_{i',j'}$ if and only if $j < j'$ or $j = j'$ and $i \leq i'$. Since there are M metanodes and $\lfloor \alpha M \rfloor$ rows, there must be $C \equiv \left\lceil \frac{M}{\lfloor \alpha M \rfloor} \right\rceil$ columns. Figure 3 shows the structure of $\mathbf{F}$. Since M may be less than $C\lfloor \alpha M \rfloor$, we pad the bottom of the rightmost column with zeroes; that is, $s_i \equiv 0$ for all $i > M$. This matrix and the others used in this proof are strictly organizational tools: we exploit no properties of matrices other than their two-dimensional structure.

Note that $S \equiv |\mathcal{S}| = \sum_{i,j} f_{i,j}$. We partition the metanodes into C *groups* corresponding to the columns of $\mathbf{F}$; the j^{th} group consists of the metanodes corresponding to the values $f_{1,j}, f_{2,j}, \ldots, f_{\lfloor \alpha M \rfloor, j}$. Thus, the first group contains the $\lfloor \alpha M \rfloor$ metanodes that contain the most nodes in $\mathcal{S}$. For concreteness, we let $u_{i,j}$ denote the metanode that corresponds to $f_{i,j}$, for all $f_{i,j} > 0$. (The restriction on $f_{i,j}$ exists because we padded $\mathbf{F}$ with zeroes; there may not be a corresponding metanode if $f_{i,j} = 0$.)

For each group of metanodes (with the possible exception of the last, which may contain less than $\lfloor \alpha M \rfloor$ metanodes), we identify a particular set of $\beta \lfloor \alpha M \rfloor$ channels, such that each channel connects a metanode in the group to one of a set of $\beta \lfloor \alpha M \rfloor$ metanodes in $\mathcal{V}$. We can always find such a set of channels since any set of size $m \leq \alpha M$ in $\mathcal{U}$ expands to a set βm in $\mathcal{V}$. The channels and metanodes that we select must satisfy certain additional properties, however. In particular, if we weight each channel with the value of $f_{i,j}$ for the connected metanode in $\mathcal{U}$, then we require that the weight of the heaviest βl channels in the j^{th} group each be at least $f_{l,j}$, for all l and j. We can show that such a collection of channels and metanodes in $\mathcal{V}$ can always be found by induction on i.

The base case of $i = 1$ is trivial. Since, without loss of generality, $f_{1,j} > 0$ we can just use the channels linking $u_{1,j}$ to β of its neighbors in $\mathcal{V}$. Once we have found a set of channels satisfying the property for $l-1$, we can augment it to a set that satisfies the property for l as follows. If $f_{l,j} = 0$ then we are done immediately. Otherwise, we examine the neighbors of $\mathcal{U}^* \equiv \{u_{1,j}, u_{2,j}, \ldots, u_{l,j}\}$ in $\mathcal{V}$. By the expansion properties of G, there are at least βl neighbors of this set in $\mathcal{V}$ and each channel linking $\mathcal{U}^*$ to $\mathcal{N}(\mathcal{U}^*)$ has weight at least $f_{l,j}$. Since, by induction, we have already found $\beta(l-1)$ nodes each with weight at least $f_{l-1,j}$, we can augment the set by choosing any β previously unchosen metanodes from $\mathcal{N}(\mathcal{U}^*)$. The additional metanodes each have weight at least $f_{l,j}$ and we are done.

We next construct an $N \times C$ matrix of weights $\mathbf{H}^* \equiv \{h^*_{i,j}\}$ by setting $h^*_{i,j}$ to be the weight of the channel that connects the i^{th} metanode of $\mathcal{V}$ to the j^{th} group of metanodes from $\mathcal{U}$ just described. If there is no such connection, then we set $h^*_{i,j} \equiv 0$. By the preceding analysis, we know that the βl largest entries in the j^{th} column each have size at least $f_{l,j}$ for all l and j.

This means that we can define another $N \times C$ matrix $\mathbf{H} \equiv \{h_{i,j}\}$ so that $0 \leq h_{i,j} \leq h^*_{i,j}$ and so that there are precisely β copies of each $f_{i,j}$ in the j^{th} column of $\mathbf{H}$. Essentially, we take the β largest items in the column, which each have size at least $f_{1,j}$, and replace them with $f_{1,j}$. Similarly, we take the next β largest items and replace them with $f_{2,j}$, and continue until we have β copies of each $f_{i,j}$. This gives us the following two properties:

$$\sum_{i=1}^{N} h_{i,j} = \beta \sum_{i=1}^{\lfloor \alpha M \rfloor} f_{i,j} \qquad \text{for all } j \text{, and}$$

$$\beta S = \sum_{i,j} h_{i,j}. \tag{1}$$

The matrix **H** plays a crucial role in describing how many nodes in $\mathcal{V}'$ are likely to be a neighbor of $\mathcal{S}$. In particular, from the definition of $h_{i,j}$ we know that the i^{th} metanode of $\mathcal{V}$ is connected to a metanode in $\mathcal{U}$ that contains at least $h_{i,j}$ items from $\mathcal{S}$. Since every channel is wired in a one-to-one fashion, this means that the i^{th} metanode in $\mathcal{V}$ contains $h_{i,j}$ neighbors of $\mathcal{S}$. As an immediate consequence of this fact, we can deduce that:

$$|\mathcal{N}(\mathcal{S})| \geq \sum_{i=1}^{N} h_{i,j} \qquad \text{for all } j.$$

Although the preceding fact is helpful, it is not sufficient, since we must show that (with high probability) $|\mathcal{N}(\mathcal{S})|$ is close to $\beta S = \sum_{i,j} h_{i,j}$. To obtain the stronger bound, we rely on the fact that the i^{th} metanode of $\mathcal{V}$ does indeed contain at least $\sum_j h_{i,j}$ neighbors of $\mathcal{S}$, when neighbors are counted according to multiplicity. Since each channel is wired with a random permutation, we use probabilistic methods to show that, with high probability, most of these $\sum_j h_{i,j}$ neighbors are distinct (at least on average over the whole graph).

The analysis depends crucially on the following simple facts about row and column sums in **F** and **H**. We define $a_{min} \equiv f_{\lfloor \alpha M \rfloor,1}$ to be the smallest item in the first column of **F**. We also define:

$$\begin{aligned} b_{min} &\equiv f_{\lfloor \alpha M \rfloor,2} + f_{\lfloor \alpha M \rfloor,3} + \cdots + f_{\lfloor \alpha M \rfloor,C} \\ b_{max} &\equiv f_{1,2} + f_{1,3} + \cdots + f_{1,C} \end{aligned}$$

to be the smallest and largest row sums in **F** when the first column is excluded from the sum.

The first key fact is that:

$$b_{max} \leq a_{min} + b_{min} \tag{2}$$

To see this, we expand it over two lines:

$$\begin{array}{lllll} & f_{1,2} & + \cdots & + f_{1,C} & \\ \leq & f_{\lfloor \alpha M \rfloor,1} & + \cdots & + f_{\lfloor \alpha M \rfloor,C-1} & + f_{\lfloor \alpha M \rfloor,C} \end{array}$$

The key is that column-major order ensures that for each term on top, the term below it is at least as large. In addition, since S is the sum of the elements in **F**, we know that:

$$S \geq \lfloor \alpha M \rfloor (a_{min} + b_{min}) \geq \lfloor \alpha M \rfloor b_{max} \tag{3}$$

Since $S \leq \alpha' M k$, we can therefore conclude that:

$$\lfloor \alpha M \rfloor b_{max} \leq \alpha' M k$$

and, after defining $\alpha' \equiv \rho \alpha$, that:

$$b_{max} \leq \frac{\rho \alpha M k}{\lfloor \alpha M \rfloor} \leq \rho' k \tag{4}$$

where $\rho' \equiv \rho \frac{\alpha M}{\lfloor \alpha M \rfloor} \approx \rho$. (The value of ρ will be determined later.)

Next, define $a_i \equiv h_{i,1}$ to be the first item in the i^{th} row of **H**, and

$$b_i \equiv h_{i,2} + h_{i,3} + \cdots + h_{i,C} \tag{5}$$

to be the sum of the remaining elements in the i^{th} row of **H**. By the manner in which **H** was constructed, it should be clear that $b_i \leq b_{max}$ for all i and that:

$$\sum_{i=1}^{N} \frac{b_i(a_i + b_i)}{k - b_i} \leq \frac{b_{max}}{k - b_{max}} \sum_{i=1}^{N} (a_i + b_i),$$

and by applying Equations 1 and 4, we get:

$$\sum_{i=1}^{N} \frac{b_i(a_i + b_i)}{k - b_i} \leq \frac{b_{max}}{(1 - \rho')k} \beta S. \tag{6}$$

We are now ready for the probabilistic analysis. Consider the i^{th} metanode v_i in $\mathcal{V}$. By definition, v_i is incident to a metanode from the first group of metanodes in $\mathcal{U}$, which contains at least $a_i \equiv h_{i,1}$ nodes in $\mathcal{S}$. Since each channel is wired one-to-one, this means that v_i contains at least a_i nodes in $\mathcal{N}(\mathcal{S})$. In addition, v_i is incident to a metanode in the j^{th} group that contains at least $h_{i,j}$ nodes from $\mathcal{S}$ for each $j \geq 2$. Unfortunately, this does not mean that v_i contains at least:

$$a_i + \sum_{j=2}^{C} h_{i,j} = a_i + b_i$$

nodes in $\mathcal{N}(\mathcal{S})$ since there may be overlap among the neighbors of each group. However, since each channel is wired randomly and independently, we will be able to show that the amount of overlap is small with very high probability (at least on average over all v_i).

In the probabilistic analysis that follows, we will only account for a_i distinct neighbors from the first group and b_i distinct neighbors from the other groups. That is, we will assume, without loss of generality, that v_i is connected to metanodes in $\mathcal{U}$ that contain a_i, $h_{i,2}$, $h_{i,3}, \ldots,$ $h_{i,C}$ nodes of $\mathcal{S}$. In fact, v_i may have more metanode neighbors in $\mathcal{U}$ and each may contain more nodes of $\mathcal{S}$, but we will undercount by ignoring this potential for additional neighbors in $\mathcal{N}(\mathcal{S})$.

In addition, we think of each channel as being randomly wired in sequence, starting with channels connecting to $u_{1,1}, u_{2,1} \ldots$, and continuing in column-major order through the metanodes, and starting with the wires that are connected to nodes in $\mathcal{S}$ within each metanode. Then, regardless of the existing connections, the probability that the wire currently being connected (from a node in $\mathcal{S}$) connects to a node already in $\mathcal{N}(\mathcal{S})$ (because of previous connections among those that we are counting) is at most:

$$\begin{cases} 0 & \text{for the } a_i \text{ connections being made from the first group} \\ \dfrac{a_i+b_i}{k-b_i} & \text{for the } b_i \text{ connections being made for later groups} \end{cases}$$

This is because there is no chance for overlap for the first channel, and because for subsequent channels, there are still at least $k - b_i$ choices for nodes, at most $a_i + b_i$ of which can lead to previously selected nodes.

We can now use a Chernoff bound (see Lemma 1.7 of [Lei92]) to show that the probability that there are σT overlaps over all metanodes is at most:

$$e^{-\sigma T \ln \frac{\sigma}{e}}$$

for any $\sigma > 1$, where by Equation 6:

$$T = \sum_i \frac{b_i(a_i+b_i)}{k-b_i} \le \frac{b_{max}}{(1-\rho')k}\beta S \qquad (7)$$

is an upper bound on the expected number of overlaps. When we consider this probability over all possible choices for $\mathcal{S}$ of size S, we find that with probability at most:

$$P_S \equiv \binom{Mk}{S} e^{-\sigma T \ln \frac{\sigma}{e}}$$

there exists some $\mathcal{S}$ of size S with at least σT overlaps. Thus, with probability $1 - P_S$ there is *no* set of size S that has σT overlaps.

In order to make P_S small, we must make σ and/or T be large. On the other hand, we do not want σT to be too large since:

$$\mathcal{N}(\mathcal{S}) \ge \sum_i (a_i + b_i) - \sigma T = \beta S - \sigma T$$

needs to be at least $\beta' S$. Hence, we define $\sigma \equiv \frac{(\beta-\beta')S}{T}$ to ensure that we achieve the required expansion. It now remains to select values for $\rho \equiv \frac{\alpha'}{\alpha}$ and β' that ensure that $P_S \le e^{-S}$ and that $\sigma > 1$.

We start this process by observing that:

$$P_S \le \left(\frac{Mke}{S}\right)^S e^{-\sigma T \ln \frac{\sigma}{e}}$$

$$= (e)^{S \ln \frac{Mke}{S} - (\beta-\beta')S \ln \frac{(\beta-\beta')S}{Te}}.$$

This quantity is at most e^{-S} provided that:

$$\ln \frac{Mke}{S} - (\beta-\beta') \ln \frac{(\beta-\beta')S}{Te} \le -1$$

which is satisfied when:

$$\frac{Mke^2}{S} \le \left[\frac{(\beta-\beta')S}{Te}\right]^{\beta-\beta'}. \qquad (8)$$

From Equation 7 we find that:

$$\left[\frac{(\beta-\beta')S}{Te}\right]^{\beta-\beta'} \ge \left[\frac{(\beta-\beta')(1-\rho')k}{b_{max}\beta e}\right]^{\beta-\beta'}$$

and we find from Equation 3 that:

$$\frac{Mke^2}{S} \le \frac{Mke^2}{\lfloor \alpha M \rfloor b_{max}} \le \frac{\alpha M}{\alpha \lfloor \alpha M \rfloor} \frac{ke^2}{b_{max}}$$

$$\le \frac{2ke^2}{\alpha b_{max}}$$

Hence Equation 8 is true provided that:

$$\frac{2ke^2}{\alpha b_{max}} \le \left[\frac{(\beta-\beta')(1-\rho')k}{b_{max}\beta e}\right]^{\beta-\beta'}$$

which is satisfied when:

$$\frac{2\beta e^3}{\alpha(\beta-\beta')(1-\rho')} \le \left[\frac{(\beta-\beta')(1-\rho')k}{b_{max}\beta e}\right]^{\beta-\beta'-1}.$$

By Equation 4, the latter inequality holds if:

$$\frac{2\beta e^3}{\alpha(\beta-\beta')(1-\rho')} \le \left[\frac{(\beta-\beta')(1-\rho')}{\rho'\beta e}\right]^{\beta-\beta'-1} \qquad (9)$$

and the dependence on k is finally gone. There are many ways to set ρ and β' so that Equation 9 is satisfied. In fact, we can make β' arbitrarily close to $\beta - 1$ simply by making ρ be a very small constant (assuming α and β are constant). For the theorem we set $\beta' \equiv \beta - 2$ and solve for ρ:

$$\frac{2\beta e^3}{\alpha(2)(1-\rho')} \le \left[\frac{2(1-\rho')}{\rho'\beta e}\right]^1$$

Simplifying:

$$\rho'\beta^2 e^4 \le 2\alpha(1-\rho')^2$$

We bound $(1-\rho')^2$ with $(1-2\rho')$ and simplify:

$$\rho'(\beta^2 e^4 + 4\alpha) \le 2\alpha$$

which gives us the desired value for α':

$$\rho' \le \frac{2\alpha}{\beta^2 e^4 + 4\alpha}$$

$$\alpha' \equiv \rho\alpha \le \frac{\rho'\alpha}{2} \le \frac{\alpha^2}{\beta^2 e^4 + 4\alpha} \qquad (10)$$

We must also show that $\sigma \equiv \frac{(\beta-\beta')S}{T} > 1$. From Equation 7 we know:

$$\frac{S}{T} \ge \frac{(1-\rho')k}{b_{max}}$$

and after substituting for β' and b_{max} we get:

$$\sigma \geq \frac{2S}{T} \geq \frac{2(1-\rho')}{\rho'} > 1$$

since $\rho' < \frac{1}{2}$.

Finally, we observe that the probability that we fail to achieve the desired expansion for a random k-extension of an expander is P_S summed over all possible sizes of $\mathcal{S}$. We can assume that $S \geq \alpha M$; otherwise, $\mathcal{S}$ can cover at most αM metanodes and there is no need for a probabilistic analysis. Thus, the probability that we fail to achieve the desired expansion is at most:

$$\sum_{s=\lceil \alpha M \rceil}^{Mk} e^{-s} < \int_{\alpha M - \frac{1}{2}}^{\infty} e^{-s} ds < 2e^{-\alpha M}$$

This completes the proof. □

It is perhaps worth noting the importance of Equation 2 in the previous analysis. It means that one of the following three statements is true: First, b_{min} and b_{max} are close, in which case each metanode in $\mathcal{V}$ has about the same number of chances for overlaps, which makes the Chernoff bound more favorable. Second, b_{max} is small, which decreases the potential for overlap. Third, a_{min} is large, which means that we get a reasonable amount of expansion for free.

Given Theorem 1 we can prove that metabutterflies are multibutterflies, that is, that each splitter has expansion. For most stages in the underlying multibutterfly, $\alpha M_i \geq 1$, and we can just k-extend the splitter. By Theorem 1 we know the k-extended splitter is an expander. If the i^{th} stage has $\alpha M_i < 1$, and thus only expands empty sets, we replace it and all later stages with an $M_i k$-input multibutterfly, which provides (α, β)-expansion for these stages. Thus, for any k, M, and α, and any $\beta > 3$, we can convert an M-input multibutterfly with (α, β)-expansion into an Mk-input metabutterfly in which each splitter has at least (α', β')-expansion, where α' and β' are those given in Theorem 1.

This gives us a metabutterfly with a two-level hierarchy. Deeper hierarchies are obtained by k-extending a metabutterfly splitter, so that each "wire" in the original graph is itself a channel and each "node" in the original graph is itself a metanode. Although the k-extensions can be applied recursively, it should be noted that α' shrinks rapidly with each k-extension. Fortunately, practical applications should never need more than a three-level hierarchy.

4 Empirical Results

In this section we present empirical evidence that the performance and fault tolerance of metabutterflies is identical to that of multibutterflies. We use the methodology of previous studies [CED92] [CK92] and investigate connectivity, partitioning, and performance with uniformly distributed router failures within each network.

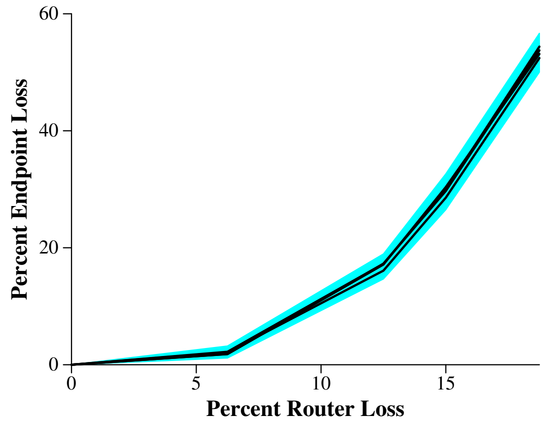


Figure 4: Network partitioning with the Leighton-Maggs Fault Propagation algorithm. The four curves represent partitioning for a 1024-endpoint multibutterfly and 1024-endpoint metabutterflies with metanodes of size 4, 16, and 32. The shaded area represents the aggregate confidence interval of the curves.

We first measure the connectivity, which is the probability that all input-output pairs remain connected for a given percentage of failed routers. We assume that a failed router fails completely; that is, all of its inputs are blocked. We compared connectivity for a 1024-endpoint multibutterfly and for 1024-endpoint metabutterflies with metanode sizes of 4, 16, and 32. The routers had a radix of 4 and a multiplicity of 2. We found no significant differences in the connectivity of all four networks. However, connectivity is not a good measure of fault tolerance because it makes no guarantees about the performance of the surviving input-output pairs. Under this metric, bottlenecks due to synchronization constraints have been shown to degrade application performance significantly.

To avoid such bottlenecks, we choose a partition, a subset of endpoints to use, with the Leighton-Maggs Fault Propagation algorithm [LM92]. This algorithm treats a router as faulty unless it has at least one unblocked output in each direction; faults propagate backward when there is insufficient bandwidth through a router. The resulting partitions have been shown to have high bandwidth between all pairs of endpoints. Figure 4 shows the percentage of endpoints that remain connected under this more conservative definition. The partitionings of all four networks are statistically indistinguishable.

We simulated performance on these networks under these partitioning situations. The routers simulated were based upon the RN1, a full-custom, high-speed VLSI crossbar that performs source-responsible, pipelined, circuit-switched routing [MDK91]. We used a synthetic, barrier-synchronized network load that models shared-memory applications studied in [CFKA90].

In over 500 trials simulated on the CM5 [TMC91], we found that performance had a 0.9997 correlation to partitioning. This confirms our expectation that these partitionings guarantee high bandwidth between surviving input-output pairs. It also means that the *performance* of the four networks is statistically indistinguishable, even with many faults.

5 Open Issues

An open question is whether a small set of permutations would be sufficient to provide expansion for a k-extension of an expander. Clearly, one permutation would not be enough. If $\mathcal{S}$ consisted of the first node in each of $\alpha M k$ metanodes in $\mathcal{U}$, then a single permutation would connect these $\alpha M k$ nodes to no more than $\frac{M}{r}$ nodes in $\mathcal{V}'$. This means that we lose a factor of k in expansion. If there were such a set, then the network could be wired with only a few types of cables (one for each permutation). However, in practice, cable connectors are still attached manually, which means that is it actually easier to make a cable with a random permutation than it is to make a standard cable (which requires the identity permutation).

It would also be interesting to show that a randomly cabled metabutterfly with multiplicity-2 can route any permutation in $\mathcal{O}(\log n)$ steps. Multiplicity-2 multibutterflies do not have expansion but still route well. By the results in this paper, multiplicity-4 metabutterflies have sufficient expansion to guarantee $\mathcal{O}(\log n)$-time packet routing, but multiplicity-2 metabutterflies may not. However, multiplicity-2 metabutterflies do route well empirically.

On the practical side, it remains to be shown that the multiplexing allowed by hierarchical wiring significantly reduces the number of wires required to achieve a particular level of throughput. Since the average load per cable should be well less than the peak load of a full-thickness cable, we expect to be able to reduce the number of wires per cable significantly without reducing the effective bandwidth of the network. The use of randomness helps us here as well, since we know that the actual load per cable will not differ from the average load per cable by very much; that is, randomness ensures relatively even load.

Finally, we are also looking at *dynamic random multiplexing*, in which the routers randomly select a wire within the cable. This provides several advantages. First, the wiring within the cable need not be random, an off-the-shelf cable works fine. Second, the effective multiplicity goes up, which increases both the fault tolerance and the performance. For example, if a cable connects ten routers to ten routers, each input router can reach each output router, as opposed to only two routers for random wiring with a multiplicity of two. When a output fails (or is busy), each input gets $\frac{9}{10}$ of the bandwidth, rather than eight getting their full share and two getting half their bandwidth. The full practical benefits of this technique remain to be investigated.

6 Conclusion

We have proven that, with high probability, random k-extensions preserve expansion. Random k-extensions allow us to build hierarchical expanders that are much more scalable than other expanders. An example of a hierarchical expander is the metabutterfly, which is based on the randomly wired multibutterfly. Results from detailed performance simulations indicate that the fault tolerance and performance of the metabutterfly match those of the multibutterfly, despite the metabutterfly's greatly simplified wiring and resulting scalability.

7 Acknowledgments

We would like to thank Bobby Blumofe, Tom Knight and Charles Leiserson for their comments on this work.

[ALM90] S. Arora, F. T. Leighton, and B. Maggs. On-line algorithms for path selection in a non-blocking network. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 149–158, May 1990.

[BP74] L. A. Bassalygo and M. S. Pinsker. Complexity of optimum nonblocking switching networks without reconnections. *Problems of Information Transmission*, 9:64–66, 1974.

[CED92] F. T. Chong, E. Egozy, and A. DeHon. Fault tolerance and performance of multipath multistage interconnection networks. In T. F. Knight Jr. and J. Savage, editors, *Advanced Research in VLSI and Parallel Systems 1992*, pages 227–242. MIT Press, March 1992.

[CFKA90] D. Chaiken, C. Fields, K. Kurihara, and A. Agarwal. Directory-based cache-coherence in large-scale multiprocessors. *IEEE Computer*, 23(6):41–58, June 1990.

[CK92] F. T. Chong and T. F. Knight, Jr. Design and performance of multipath MIN architectures. In *Symposium on Parallel Architectures and Algorithms*, pages 286–295, San Diego, California, June 1992. ACM.

[Lei85] C. E. Leiserson. Fat-trees: Universal networks for hardware efficient supercomputing. *IEEE Transactions on Computers*, C-34(10):892–901, October 1985.

[Lei92] F. T. Leighton. *Introduction to parallel algorithms and architectures*. Morgan Kaufmann, San Mateo, CA, 1992.

[LM89] F. T. Leighton and B. Maggs. Expanders might be practical: Fast algorithms for routing around faults on multibutterflies. In *IEEE 30th Annual Symposium on Foundations of Computer Science*, 1989.

[LM92] F. T. Leighton and B. Maggs. Fast algorithms for routing around faults in multibutterflies and randomly-wired splitter networks. *IEEE Transactions on Computers*, 41(5):1–10, May 1992.

[MDK91] H. Minsky, A. DeHon, and T. F. Knight Jr. RN1: Low-latency, dilated, crossbar router. In *Hot Chips Symposium III*, 1991.

[Pip93] N. Pippenger. Self-routing superconcentrators. In *25th Annual ACM Symposium on the Theory of Computing*, pages 355–361. ACM, May 1993.

[TMC91] Thinking Machines Corporation, Cambridge, MA. *CM5 Technical Summary*, October 1991.

[Upf89] E. Upfal. An $O(\log N)$ deterministic packet routing scheme. In *21st Annual ACM Symposium on Theory of Computing*, pages 241–250. ACM, May 1989.

[WZ93] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: explicit construction and applications. In *25th Annual ACM Symposium on the Theory of Computing*, pages 245–251. ACM, May 1993.