DENSITY AND RELIABILITY OF INTERCONNECTION TOPOLOGIES FOR MULTICOMPUTERS

by

WILL EDWARD LELAND

Computer Sciences Technical Report #478

July 1982

DENSITY AND RELIABILITY OF INTERCONNECTION TOPOLOGIES FOR MULTICOMPUTERS

by

WILL EDWARD LELAND

A thesis submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN - MADISON

MAY 1982

© Copyright by Will Edward Leland 1982

All rights reserved

Abstract

In a multicomputer consisting of processors that communicate over point-to-point full-duplex communication lines, the distance between two processors is the number of message forwarding steps required for them to communicate. The diameter of a network is the longest distance between two processors, and the degree is the largest number of lines from any one processor. We examine the problem of minimizing the diameter k of a network as a function of the number N of processors, for a fixed degree d.

We define a new general method, the double-exchange topology, that produces families of interconnection topologies that are denser than any families previously proposed. In particular, we have discovered families of networks with d = 3, k = 1.47 lg N; d = 4, k = 0.947 lg N; and d = 5, k = .75 lg N. The best previously-known results were d = 3, k = 2 lg N; d = 4, k = lg N; and d = 5, k = .77 lg N.

These new families have a regular structure that admits efficient routing algorithms. They have a close relationship with previously-proposed (and less dense) families, allowing us to exploit known results on algorithm-mapping and VLSI layout.

Finally, we explore the issue of reliability. An intricate proof shows that our degree-3 families are 3-connected (the best possible connectivity for degree-3 networks) -- that is, at least three node failures are required to partition the network. We then propose two new measures of network reliability, called local reroutability and average random failset size, that are particularly appropriate for evaluating multicomputer topologies. Both by the standard of connectivity and by our proposed measures, our double-exchange families of interconnection topologies are highly reliable as well as exceptionally dense.

Acknowledgements

Professor Marvin Solomon has been a remarkable advisor: patient, encouraging, and always willing to take the time to think seriously about my research, to read carefully my interminable revisions, and to improve both beyond measure. The thoughtful comments and helpful conversations of Professors Raphael Finkel, S. Diane Smith, and Jim Goodman did much to clarify my writing and my reasoning; I am indebted to them and to Professor Charles Kime for their efforts with my thesis.

This thesis would never have happened without Professor Leonard Uhr, whose enthusiasm introduced me to the question of density, and whose interest and results encouraged me to pursue it. The expertise of Professors Sam Bent and E. F. Moore saved me much work, while the generosity of Sheryl Pomraning made the job of producing the finished thesis far less miserable than I had expected. I am grateful to my parents, my brother, and the Department of Computer Sciences for their undeserved patience and support. Bill Cox and Bill Wickart have been good friends and colleagues for years; their friendship, and the friendship of the Cross Trails Square Dance Club, the Friends of the Arboretum, the Arachne research group, and many others made a stressful time almost pleasant.

I cannot offer adequate words to thank my wife, Mary, for her hours with my thesis; I can offer my love and a promise to reciprocate for her thesis.

This research was supported in part by the Defense Advanced Projects Research Agency under Navy contract N00014-81-C-2151.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1. Introduction

The rapid development of microprocessors and VLSI makes the concept of a multicomputer increasingly attractive for achieving high effective processing rates. A multicomputer consists of many cooperating independent asynchronous processors; in a multicomputer, the processors must have reliable high-bandwidth communication facilities. Among the various possible interconnection designs, including busses and switching networks, point-to-point dedicated interconnections are very well suited to the VLSI requirements of restricted off-chip bandwidth and high logic-to-pin ratios. In particular, this research studies interconnection topologies for multicomputers in which each processor communicates with other processors by sending and forwarding messages over point-to-point full-duplex communication lines.

The designer of such multicomputers needs to discover interconnection topologies for the number of processors and of I/O ports per processor available, and to compare their effectiveness. Ad hoc design is adequate for small multicomputers, but has serious drawbacks when we consider scores or hundreds of processors. For each

new set of constraints, we must generate a topology; once we have found one, we must solve the problems of addressing and routing on it (even in the face of possible failures), find algorithms that exploit its particular interconnections, analyze its reliability, and compare it with other candidate topologies.

The designer is rescued from these difficulties by the formal definition of families of possible interconnections. The problems of addressing, routing, algorithm design, and reliability can then be solved for all members of a family at once, and entire families compared. We have discovered families of interconnections that are denser than any previously proposed. These interconnection topologies are also highly regular and resistant to isolated failures.

We investigate the question of fault tolerance in some detail, proposing new measures of reliability and developing mathematical tools for studying these measures.

1.2. Overview

Graph theory provides a convenient formal model for these interconnection topologies. The vertices (nodes, points) of a graph represent the processors and the edges (lines, arcs) represent the communication lines. The same formal model is used for switching networks (by interpreting some vertices as switches and others as processors or memories), and for computer communications networks (by interpreting the vertices as geographically remote hosts). To emphasize the distinction between our interpretation and other multiprocessor models that employ graph theory, we call our abstract models multicomputer graphs.

The different interpretations imply different criteria for evaluating graphs. Unlike switching networks, multicomputer graphs can ignore switch-setting algorithms and considerations of blocking; however, since the processors have limited total I/O bandwidth, our graphs must have very small degree (that is, very few edges per vertex). We concentrate on degrees 3, 4, and 5, with some consideration of general families that extend to higher degrees. Unlike computer communications networks, multicomputer graphs do not need to minimize the total number of edges or vertices involved, and cannot afford large ad hoc routing tables. We therefore

require graphs with address assignments that permit simple routing algorithms. This requirement is met by studying uniform infinite families of graphs, in which all members can use the same routing algorithm regardless of N, the number of processors.

A dense graph is one whose diameter (the greatest distance between any two vertices) is low for its degree and number of vertices; there is a limit to how small the diameter of any graph may be, given N and its degree. A dense topology is important not only for reduced communication delay and congestion, but also for enhanced reliability. Chapter 2 compares known dense multicomputer graph families using the cost measures of diameter and diameter times degree, as well as considerations of routing and interpolability (the ability to produce designs for any given number of processors). While reviewing previously-proposed multicomputer graph families in terms of these criteria, it presents a general construction -- the single-exchange family -- that includes most of the densest families proposed, and provides linear interpolation and a uniform routing algorithm for them. Finally, it introduces a new construction -- the double-exchange family -- that produces the densest known graph families at degrees 3, 4, and 5, as well as having simple routing and interpolation. Under

the diameter-degree product criterion, the degree-5 double-exchange family is, in fact, the densest family known at any degree.

Chapter 3 concentrates on a particular variety of double-exchange graph, exploring the problems of layout and of mapping algorithms to processors. A simple homomorphism from the well-studied shuffle-exchange family onto this double-exchange allows a computationally uniform emulation of any shuffle-exchange algorithm, and, by exploiting known results for the shuffle-exchange, provides both an asymptotic lower bound for the area required to lay out the new family and a good layout for it.

Chapter 4 returns to comparing all multicomputer graph families, using measures of reliability. Connectivity, the minimum number of failures required to disconnect a graph, is a measure of reliability often used for computer networks. After summarizing the known connectivities for previous multicomputer graph families, we show that three dense degree-3 families -- the well-known shuffle-exchange and two of our much denser double-exchange families -- can be made 3-connected by minor changes that preserve their degree, density, routing, and interpolability. No higher connectivity is possible for degree-3 graphs. We then propose two meas-

ures of reliability that are suitable for the distributed nature of the multicomputer: local reroutability and Average Random Failset size.

An infinite family of graphs is locally reroutable if there is some fixed number r, independent of N, such that any given isolated failure requires that at most r vertices modify their routing algorithms to detour messages around the fault. We measure the cost of this local rerouting by the minimal r that suffices for all members of a family, and by the increase in the length of the detoured paths.

The Average Random Failset (ARF) size of a graph is the expected number of random failures required to partition it; unlike most probabilistic measures of network reliability, it is independent of the fault probabilities of the individual elements. After determining the costs of local rerouting in the multicomputer graph families, Chapter 4 compares their ARF sizes by simulation and derives a lower bound on ARF size based on local reroutability.

The multicomputer designer can therefore use our research both for reliability indices that are suitable for comparing multicomputer topologies, and for specific families of topologies that have high density, natural address assignments, simple routing algorithms, uniform

rerouting methods for bypassing failed processors or communications lines, and high reliability.

Chapter 2

Dense Multicomputer Graphs

This chapter reviews families of graphs previously proposed for multicomputer interconnection, compares them according to several standard cost criteria, and develops a new construction that produces several cheaper families. To emphasize the distinction between our model, in which all vertices represent independent processors communicating via messages over full-duplex links (represented by edges), and other multiprocessor models that employ graph theory (such as permutation networks, in which most of the vertices represent switches in a network that provides interprocessor communication), we will call our abstract versions multicomputer graphs. The reader should be aware that there are terminological conflicts between several closely related research areas; for an example that will arise later in this chapter, the various permutation networks called "shuffle-exchange networks" [1] are not exactly the same graphs as the "shuffle-exchange graphs" in the multicomputer graph literature [2,3,4].

2.1. Definitions

Lg denotes the base two logarithm. $\lfloor x \rfloor$ denotes the largest integer not greater than x; $\lceil x \rceil$ denotes the smallest integer not less than x; and i div j denotes $\lfloor i/j \rfloor$. We use b^n to denote the set of b-ary strings of length n. Whenever we use integers drawn from the set $\{0,1,\ldots,c-1\}$ for any c, we employ the convention that our arithmetic is modulo c; for example, given $x \in b^n$, all arithmetic involving digits of x is modulo b, while arithmetic involving subscripts of its digits is modulo n. If x is a string of digits, x^* denotes a string formed by repeating x some number of times; the number of repetitions will be clear from context.

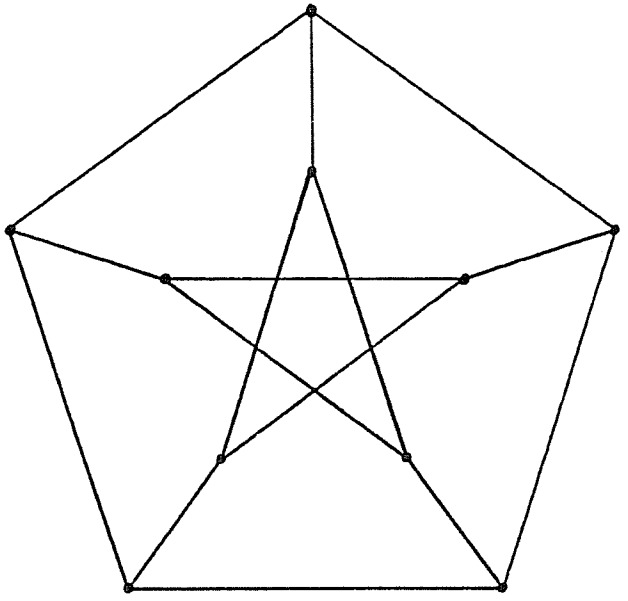
A graph consists of a nonempty set V of vertices and a set E of edges, or pairs of adjacent vertices. N will denote the number of vertices, and will sometimes be referred to as the size of a graph. All graphs will be simple, connected, and undirected. A simple graph has no self-loops or multiple edges; a connected graph has a path between any two vertices; an undirected graph has all edges undirected: if vertex v is adjacent to vertex w, then w is adjacent to v. The distance between two vertices is the length of a shortest path between them. The diameter, k, of a graph is the maximum distance between any two of its vertices. The degree of a

vertex is the number of edges incident on it; the degree of a graph, denoted d, is the maximum of the degrees of its vertices. A graph of degree three is called trivalent. If all vertices have the same degree, the graph is regular. (See Harary [5] for graph terminology not defined here.)

Two families of graphs that will be used heavily in later constructions have standard names in graph theory. C_N denotes a cycle (or ring) of N vertices, each connected to two neighbors. K_N denotes a complete graph with N vertices, each connected to all the others to give a diameter of 1. Another graph that appears often is the Petersen graph (Figure 2.1) which has degree 3, diameter 2, and 10 vertices.

Two quantitative cost criteria will be used in this chapter to compare the graph families as multicomputer graphs: diameter and diameter · degree. The latter criterion will be called the product cost of a graph. Graphs that have low diameter and product costs are called dense, and are extremely important for practical interconnection topologies: a network's diameter provides a lower bound on the message steps required by broadcast or echo algorithms [6] and an upper bound to the routing distance between any two vertices. Diameter also directly affects traffic congestion and several

Figure 2.1



The Petersen Graph

measures of network reliability [7,8,9].

The diameter that can be achieved for any given number of vertices, however, depends strongly on the degree of the graph. A theoretical limit is given by the Moore bound [10]. For degree 2, the best possible graphs are cycles, C_N, with N odd and diameter $k = \lfloor N/2 \rfloor$. For degree $d > 2$, $N \leq (d(d-1)^k - 2) / (d - 2)$, so $k \geq \lg(N) / \lg(d-1) + O(1)$. This bound is derived by enumerating the vertices in the tree created by taking any vertex for the root and treating its d neighbors as the roots of (d-1)-ary subtrees of height k-1. Clearly, this tree includes every vertex that could lie within distance k of its root. The Moore bound is provably unreachable for all diameters > 2, and, at diameter 2, for all degrees except 2, 3, 7, and possibly 57 [11,12]. This bound has not been refined, except for diameter 2, where Erdos [13] proved that, except for degrees 2, 3, 7, and possibly 57, all graphs have $N < d^2$.

The product cost, $k \cdot d$, is often used to compare families with different degrees, to account for the actual cost of allowing additional edges per vertex (or I/O ports per processor). In practical terms, higher degree not only requires more communication lines for a given number of processors but also, for single-chip microprocessors, decreases the bandwidth available for

each I/O port [14]. To compare infinite families of graphs, we will consider the diameter and product cost as functions of N. In particular, this thesis concentrates on families of small degree, whose diameters grow logarithmically with N. Recent examples of nonlogarithmic families, in which the diameter is N^r for some fixed $r < 1$, include the chordal ring networks [15], the twisted torus [16], and the snowflake [17]. For practical multicomputers, a family of multicomputer graphs will need to have small degree and, ideally, constant degree (so that the interface design does not depend on the number of processors in the network).

Two other important, but less easily quantified, cost criteria are the ease of interpolating graph families and of routing messages within given graphs. The graph families presented here will usually be defined either for all values of N that are a multiple of some constant c, or for all values of N that are a power of some c. We call the former linearly interpolable, the latter exponentially interpolable, and c the interpolation factor. Families that are defined for $N = mM$, where M is multiple of some constant c and m is proportional to lg M, are called semi-linearly interpolable. In other words, the sequence of sizes is arithmetic for linearly interpolable families, geometric for exponen-

tially interpolable families, and has the general form i log i for semi-linearly interpolable families. To reduce design constraints and simplify expanding existing systems, linear interpolability is preferable to exponential, and small factors to large ones. In part because of the importance of routing, most proposed multicomputer graph families have reasonable routing algorithms, so we will only mention routing occasionally (for example, when proposing new families). Further criteria, relating to reliability, will be discussed in chapter 4.

2.2. Graph Product Families

Several proposed families of graphs can be defined by the Cartesian product of graphs. Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, with diameters k_1 and k_2, degrees d_1 and d_2, their Cartesian product, $G_1 \times G_2$, is a graph with vertices $V_1 \times V_2$ and edges such that (v_1,v_2) is adjacent to (w_1,w_2) if and only if one of these two conditions holds: $v_1 = w_1$ and $(v_2,w_2) \in E_2$, or $v_2 = w_2$ and $(v_1,w_1) \in E_1$. There are N_1N_2 vertices in the resulting graph, which has diameter $k = k_1 + k_2$ and degree $d = d_1 + d_2$. If we take the Cartesian product of K_2 with itself n times, we have the binary hypercube [18] with $N = 2^n$, $k = n$, and $d = n$. The diameter and degree of the hypercube are both $\lg(N)$, forcing the degree to increase with N.

This product cost, $\lg^2(N)$, can be improved by using different initial graphs in the construction. Given any graph G, we can build an infinite family G_i by defining G_1 to be G and G_i to be $G \times G_{i-1}$. The resulting family has $N_i = N^i$, $k_i = ki$, and $d_i = di$, so that the product cost for G_i is $kdi^2 = (kd/\lg^2(N)) \cdot \lg^2(N_i)$. The vertices of G_i can be assigned addresses that are strings of i base-N digits, allowing a simple routing algorithm: Change each source digit in turn to the destination value by holding all other coordinates fixed, and route

according to the initial graph G to alter the current digit. In the worst case, k steps are required to modify each of the i digits. Since $N_i = N^i$, the family is exponentially interpolable with factor N, so initial graphs with small values of N may be preferable. For $N \leq 10$, the lowest cost is attained using the Petersen graph, which gives $k \cdot d = 0.54 \lg^2(N_i)$; for $N \leq 100$, $0.37 \lg^2(N_i)$ can be achieved using a graph with degree 4, diameter 4, and 95 vertices discovered by Bermond [19]. By using sufficiently large initial graphs, the later constructions in this chapter show that product graphs can attain an arbitrarily low coefficient of $\lg^2(N)$; however, this procedure succeeds only because the later families have O(lg N) costs.

2.2.1. The complete hypercube

The problems of exponential interpolability can be alleviated by allowing several different graphs in the product [20]. For two popular families built in this fashion, we can determine the best possible costs by choosing appropriate component graphs. One form of generalized hypercube [21] uses products of complete graphs. If the component complete graphs all have the same number of vertices, N_1, the vertices are given N_1-ary addresses; if the component graphs have various

sizes, say $N_1 \ldots N_r$, the vertices are given addresses in a mixed radix. In any case, for each vertex there are edges to all vertices whose addresses differ in exactly one digit. This complete hypercube then has, in the general case, $N = N_1 \cdot \ldots \cdot N_r$, $k = r$, and $d = \Sigma(N_i-1)$. The mixed radix form provides greater freedom in N, but does not produce denser graphs: For fixed r, elementary calculus shows that the least degree is attained when all the N_i are equal. So for given N, $k \cdot d = r \cdot d \geq r^2 \cdot (N^{1/r}-1)$. Changing the variable to $F = N^{1/r}$ shows that $k \cdot d \geq \log_F^2(N)\ (F-1) = (F-1)/\lg^2(F)\ \lg^2(N)$. The minimum possible product cost for any N is thus achieved by the same F, the one for which $(F-1)/\lg^2(F)$ is minimal. As F must be > 1, this function has a unique minimum value of approximately 0.741886, when F has a value of approximately 4.922. While this lower bound is unattainable in practice, as all factors must be integers, a product cost of $.741929 \lg^2(N)$ is attained for Cartesian products of K_5, leaving $N = 5^r$, $k = \log_5(N)$ and $d = 4k$.

2.2.2. The mesh-connected hypercube

The other extreme in terms of component graphs are the mesh-connected hypercubes, in which each component graph of the product is a cycle (or K_2, as cycles have

at least 3 vertices). The formulae for degree and diameter are then more complex, but the best possible product cost is exactly the same as for the complete hypercubes. Each K_2 component contributes 1 to the degree and to the diameter, while increasing N by a factor of 2. Each cycle C_i contributes 2 to the degree and $\lfloor i/2 \rfloor$ to the diameter, and increases N by a factor of i. Any even cycle could replaced by an odd cycle, giving the product graph greater N for the same degree and diameter. Similarly, any two K_2 components could be replaced by a single C_5 component for an increase in N without affecting the degree or diameter. Thus, we can find the lower bound for cost as a function of N by using only the formulae appropriate for odd cycles: Given r cycles with $N_1 \ldots N_r$ vertices, respectively, the product diameter is $\geq (N_i-1)/2$ and the degree is 2r. As before, treating the N_i as continuous shows that the minimum cost is attained when all are equal, giving, once again, a lower bound for the product cost of $2r^2(N_1-1)/2 = r^2(N^{1/r}-1)$.

2.3. Tree Variations

The densities of the graph product families are surpassed by several families of graphs defined directly in terms of trees. For given degree $d > 2$, the simplest such family has a root vertex with d children, while its other internal vertices have d-1 children. If the root is assigned level 0 and the leaves level L, a tree has $d(d-1)^{L-1}$ leaf vertices (each with degree 1), and a total of $(d(d-1)^{L-2})/(d-2)$ vertices. Its diameter is 2L, so that $k = (2/\lg(d-1)) \lg(N) + O(1)$ and $k \cdot d = (2d/\lg(d-1)) \lg(N) + O(d)$. Routing is extremely simple, as is interpolation: any number N of vertices may be connected within diameter $2 \lceil \lg N/\lg(d-1) \rceil$ by adding a ragged lowest level. In terms of all the criteria used in this chapter, tree families are superior to the graph product families. (Of course, other criteria, such as the reliability issues considered in chapter 4, may make the tree less appealing.)

2.3.1. Multi-tree Structures

The low degree of a tree's leaf vertices can be exploited in several ways to improve density. Friedman [22] suggested joining d copies of the tree at the leaves, producing a regular graph with $N = 2d((d-1)^{L-1})/(d-2)$ and $k = 2L$, which reduces the lower-

order terms in the cost formulae, but does not improve the coefficient of lg(N). Arden and Lee proposed the family of Multi-Tree-Structures [23], in which some number, m, of component trees are interconnected at the top by a cycle among the roots, forcing each root vertex to have only d-2 children, and at the bottom by a cycle among the leaves of all the trees. When d is greater than 3, they also require that each leaf have enough additional connections to other leaves to make the graph regular, but these additional edges are not further specified. Although their exploration of possible Multi-Tree-Structures for degree 3 did uncover several graphs with small diameter for various small values of N, the best general bound they could establish was that, for degree 3, there are Multi-Tree-Structures with $N = 2^{(k+3)/2} + 2^{(k+3)/4}$. Thus, they further improved the low-order terms in the relationship between cost and N, but did not improve the coefficient of lg(N).

2.3.2. The hypertree family

The most successful tree-based multicomputer graphs, in terms of the cost criteria considered in this chapter, are the Hypertree families of Goodman and Sequin [24]. All the hypertree families are built from complete binary trees with L levels (that is, with the

leaves at level L), with additional hypercube-like edges being added to the vertices to lower the diameter. Hypertree m adds m edges to each vertex, carefully chosen so that the diameter is $\lfloor L(m+2)/(m+1) \rfloor = \lfloor (\lg(N+1)-1)(m+2)/(m+1) \rfloor$. The root has degree 2, the other internal vertices have degree 3+m, and the leaves have degree 1+m. The vertices in level i are assigned i-bit binary addresses; the additional edges connect vertices that differ in particular bits, depending on m and the level. More explicitly, given $x = x_0\ x_1 \cdots x_{n-1} \in 2^n$, define Smallest(x) to be the least i such that $x_{n-i} = 1$. Define Complement(x,i) to be x with bit i complemented. Then, in particular, Hypertree 1 has degree four and connects vertex v in level i with Complement(v, $\lceil i/2^{\text{Smallest}(i)} \rceil$). Similar, but more complex, definitions can be given when more edges are added.

For a hypertree with parameter m, $k = (m+2)/(m+1)\ \lg(N) + O(1) = (d-1)/(d-2)\ \lg(N) + O(1)$, and $k \cdot d = d(d-1)/(d-2)\ \lg(N) + O(d)$. The least product cost is then achieved by Hypertree 1, where d = 4 and $k \cdot d = 6\ \lg(N) + O(1)$. This cost is a great improvement over the costs of the graph product families, but is not as good as many other families. Hypertrees also suffer from somewhat more complex routing algorithms,

but, like the other tree families, are linearly interpolable with factor 1: They can be constructed for any value of N by adding a ragged lower edge.

2.4. Single-Exchange Graphs

The results of the graph product families and of the tree variations have been equaled or surpassed by several related families, such as shuffle-exchange graphs, de Bruijn networks, cube-connected-cycles, and the lens. All these families are specific instances of a general class that we call single-exchange graphs. The single-exchange construction is of some interest because it unifies so many popular multicomputer graph families and provides a common cost bound for them. Its inclusion here, however, is justified more by its relationship with the novel double-exchange construction developed below, and by the improved costs that can be achieved by exploiting the construct's generality. The definition provides linearly interpolable versions of several families that were originally proposed in an exponentially interpolable form.

2.4.1. Definitions

To provide a more intuitive understanding of the following definitions, we present the specific case of the standard binary shuffle-exchange multicomputer graph in parallel with the general definitions. The general definitions assume we have been given an arbitrary base $b > 1$ and some number $N > 1$ that is a multiple of b. In

our specific example, b is 2, and $N = 2^n$ for some $n > 0$.

For any $0 \leq x < N$, define $\rho(x)$ to be (xb) modulo N + xb div N. Xb div N plays the role of the high-order digit in a b-ary representation of x. In our example, ρ has the effect of rotating the high-order bit of the binary representation of x to the low-order position, and so becomes the familiar shuffle [25,26] operation. ρ maps the integers [0,N-1] into [0,N-1] and, if N is a multiple of b, is a bijection on [0,N-1] with inverse $\rho^{-1}(x) = (x \text{ modulo } b) \cdot (N/b) + (x \text{ div } b)$.

For any $0 \leq x < N$ and any $0 \leq i < b$, $SE_i(x) = (x \text{ div } b) \cdot b + (((x \text{ modulo } b)+i) \text{ modulo } b)$. This single-exchange operation can be viewed as replacing the low-order digit, x_{n-1}, in the b-ary representation of x with the digit $(x_{n-1}+i)$ modulo b. In particular, $SE_0(x) = x$. For our specific example, $SE_1(x)$ complements the least significant bit of x. Because N is a multiple of b, $SE_i(x)$ is a bijection on [0,N-1] for any i, with $SE_{-i}(SE_i(x)) = x$.

Using these definitions, we say an undirected graph is a single-exchange graph with S stages, base b, stage-size M, and cost C if its vertices can be assigned unique addresses of the form {s,x}, where $0 \leq s < S$ and $0 \leq x < M$, such that, $\forall$ vertices {s,x} and $\forall\ 0 \leq i < b$, there is a path of length at most C from {s,x} to {s+1 modulo S, $SE_i(\rho(x))$}. (If we view paths in a graph as operations on addresses, C is the cost of rotating in a new digit.) A vertex with address {s,x} is said to belong to stage s. Following our general arithmetic conventions, the stage portion of an address is always interpreted modulo S, the string portion is modulo M, and the subscript of any SE operation is modulo b. In the shuffle-exchange example, S is 1, b is 2, M is N, and C is 2.

2.4.2. The diameter of a single-exchange graph

Two more definitions then allow a simple expression for an upper bound on the diameter of any single-exchange graph G with cost C. Let $n = \lceil \log_b M \rceil$, and define Excess(s,n) to be the maximum over all i such that $0 \leq i < s$ of $\min\{(ai+bn) \text{ modulo } s \mid 0 \leq i < s,\ a = \pm 1,\ b = \pm 1\}$. Intuitively, Excess(s,n) is the maximum number of stages that must be traversed in order to reach a stage exactly n stages away from a desired destination. We can now prove the following bound on the diameter of G:

Theorem 1. A single-exchange graph G with cost C has diameter k bounded by $C \cdot (n+\text{Excess}(S,n))$.

Of course, this bound is interesting only because we can construct infinite families of single-exchange

graphs with small, bounded, values of C; any connected graph could trivially be considered a single-exchange graph with b = M = N, S = 1, and C = k. For the shuffle-exchange, Excess(1,n) is zero so the diameter bound is 2 lg N. Incidentally, this theorem also shows that G is connected.

The proof will follow from four lemmas. Lemma 1.1 shows that, for any j, a vertex {s,x} is within distance C of {s+1,xb+j}; Lemmas 1.2 and 1.3 use this result to show any vertex in stage s is within distance nC of any vertex in stages s+n or s-n; and Lemma 1.4 completes the proof by noting that any vertex is within distance C · Excess(S,n) of some vertex exactly n stages away from any desired destination.

Lemma 1.1. For any vertex {s,x} and any j, there is a path of length bounded by C from {s,x} to {s+1,xb+j}.

In terms of our example, this lemma states that a zero or one bit can be shifted in from the right in two or fewer steps.

Proof. Choosing i in the definition of the SE operation such that $SE_i(\rho(x))$ = (xb+j) modulo M will yield the desired result. Let i = j-(xb) div M. Then

$\rho(x)$ = (xb) div M + (xb) modulo M

and

$SE_i(\rho(x))$ = (xb) modulo M +
((xb) div M + j - (xb) div M) modulo b
= (xb+j) modulo M.

□

Lemma 1.2. Given any vertex {s,x}, there is a path to any vertex {s+n,y} of length nC, for any 0 ≤ y < M.

For our example, there is only one stage, and this lemma says we may reach any address within 2n steps by shifting in the destination address bits from the right.

Proof. Induction on n applications of Lemma 1.1 shows there is a path to any vertex {s+n,y} of length bounded by nC, where $y = x \cdot b^n + i_1 \cdot b^{n-1} + i_2 \cdot b^{n-2} \ldots + i_n$, modulo M, for some coefficients $i_1 \ldots i_n$ such that $0 \le i_j < b \ \forall\ j$. However, the term $\Sigma i_j b^{n-j}$ is merely an n-digit string of b-ary digits, so it can have any value between 0 and M-1, inclusive, allowing y to have any value in that range as well.

□

Lemma 1.3. Given any vertex {s,x}, there is a path to any vertex {s-n,y} of length nC, for any 0 ≤ y < M.

Proof. G is an undirected graph, so we can use Lemma 1.2 starting from {s-n,y}.

□

Lemma 1.4. Given any vertex {s,x} and any stage $0 \leq m < S$, there is a path to some vertex belonging either to stage m+n or to stage m-n of length bounded by C · Excess(n,S).

S is 1 for our specific example, Excess(n,1) is zero, and this lemma merely reflects the fact that no sequence of ρ operations is needed to change the stage.

Proof. By Lemma 1.1, there is a vertex in stage s+t within distance tC of {s,x}. In particular, there is a vertex in stage m+n within distance C·min(s-(m+n) modulo S, (m+n)-s modulo S) of {s,x}. Similarly, there is a vertex in stage m-n within distance C*min(s-(m-n) modulo S, (m-n)-s modulo S) of {s,x}. Excess(S,n) is defined as the maximum over all $0 \leq i < S$ of min((i-n) modulo S, (i+n) modulo S, (n-i) modulo S, (-i-n) modulo S). By considering the case i = s-m, we see that Excess(S,n) is at least the minimum of the distances from {s,x} to vertices in stages m+n and m-n.

∎

Proof of Theorem 1.

Lemma 1.4 guarantees a path of length bounded by C·Excess(S,n) from any source vertex to some vertex exactly n stages away from the destination; from here, Lemmas 1.2 and 1.3 guarantee a path to the destination

of length bounded by Cn, completing the proof of Theorem 1.

∎

In several constructions below, we will use the following corollary.

Corollary 1.5. If there is some constant $C_+ < C$ such that each vertex in stage s is within distance C_+ of some vertex in stage s+1, then k is bounded by Cn + C_+·Excess(S,n)

Proof. A straightforward modification of the proof of Lemma 1.4.

∎

2.4.3. Routing in single-exchange families

Many single-exchange graph constructions prove the validity of their cost parameter C by providing explicit paths of length ≤ C from any vertex {s,x} to the vertices $\{s+1, SE_i(\rho(x))\}$ ∀ i. In this case, the diameter proof just given also provides a routing algorithm that is guaranteed to find a path of distance ≤ C*(n+Excess(S,n)) between any two vertices.

Algorithm.

Input: A pair of vertices u and v, with addresses $\{s,x\}$ and $\{t,y\}$.

Output: A path from u to v of length $\leq C \cdot (n+\text{Excess}(S,n))$, where $n = \lfloor \log_b M \rfloor$.

Method: Compute $d^+ = \min(s-(t+n) \text{ modulo } S, (t+n)-s \text{ modulo } S)$ and $d^- = \min(s-(t-n) \text{ modulo } S, (t-n)-s \text{ modulo } S)$.

If $d^+ \leq d^-$,

If $(t+n-s) \text{ modulo } S \leq (s-(t+n)) \text{ modulo } S$,

define $z_0 = \rho^{d^+}(x)$, and construct a path P_0 from u to $\{t+n,z_0\}$ by applying the known routing for $\{s+1,SE_0(\rho(x))\}$ d^+ times to u.

Otherwise,

define $z_0 = \rho^{-d^+}(x)$, and construct a path P_0 from u to $\{t+n,z_0\}$ by applying the known routing for $\{s-1,\rho^{-1}(SE_0(x))\}$ d^+ times to u.

Compute $y-z_0b^n$ modulo M, and represent it as a b-ary string $i_0\ i_1 \ldots i_{n-1}$. For j = 1 through n, define $z_j = bz_{j-1}+i_{j-1}$ modulo M. z_n is then equal to $z_0b^n + \Sigma i_j b^{n-j}$ modulo $M = z_0b^n+(y-z_0b^n)$ modulo M = y. For $1 \leq j \leq n$, construct a path P_j from z_{j-1} to z_j of length $\leq C$ using the known paths for $SE_h(\rho(z_{j-1}))$, where $h = (i_{j-1} - bz_{j-1} \text{ div } M) \text{ modulo}$

b. (See Lemma 1.1.)

If $d^+ > d^-$,

If $(t-n-s) \text{ modulo } S \leq (s-(t-n)) \text{ modulo } S$,

define $z = \rho^{d^-}(x)$, and construct a path P_0 from u to $\{t-n,z\}$ by applying the known routing for $\{s+1,SE_0(\rho(x))\}$ d^- times to u.

Otherwise,

define $z = \rho^{-d^-}(x)$, and construct a path P_0 from u to $\{t-n,z\}$ by applying the known routing for $\{s-1,\rho^{-1}(SE_0(x))\}$ d^- times to u.

Compute $z-yb^n$ modulo M, and represent it as a b-ary string $i_0\ i_1 \ldots i_{n-1}$. Set $z_0 = y$ and, for j = 1 through n, define $z_j = bz_{j-1}+i_{j-1}$ modulo M. z_n therefore $= yb^n + \Sigma i_j b^{n-j}$ modulo $M = z$. For $1 \leq j \leq n$, construct a path P_j from z_{n-j+1} to z_{n-j} of length $\leq C$ by tracing backwards along the known (undirected) paths for $SE_h(\rho(z_{n-j}))$, where $h = (i_{n-j} - bz_{n-j} \text{ div } M) \text{ modulo } b$.

The desired path of length $\leq C \cdot (n+\text{Excess}(S,n))$ is then $P_0\ P_1 \ldots P_n$.

2.4.4. Specific single-exchange families

2.4.4.1. Cube-connected-cycles

Given integers $m \geq n > 0$, a cube-connected-cycles graph [27] with $m2^n$ vertices has vertex addresses of the

form $\{c,x\}$, where $0 \leq c < m$ and $x \in 2^n$. Each vertex $\{c,x\}$ has edges to $\{c+1 \text{ modulo } m,x\}$, $\{c-1 \text{ modulo } m,x\}$ and, if $c < n$, $\{c,\text{Complement}(x,c)\}$. As before, Complement(x,i) is x with bit i complemented. Intuitively, a cube-connected-cycles is an n-dimensional binary hypercube, with each vertex replaced by a cycle of n or more vertices. A vertex $\{c,x\}$ has at most three edges, so a cube-connected-cycles is always trivalent. (Preparata and Vuillemin originally defined the cube-connected-cycles only for m a power of two.)

Despite the use of Complement, a cube-connected-cycle with $m = n$ is, in fact, a single-exchange graph. Renumber each vertex $\{c,x\}$ to have address $\{c,\rho^c(x)\}$. The resulting graph has edges from $\{c,x\}$ to $\{c+1 \text{ modulo } m, \rho^{c+1}(x)\}$, $\{c-1 \text{ modulo } m, \rho^{c-1}(x)\}$, and $\{c,SE_1(x)\}$. The associated costs are $C = 2$ and $C_+ = 1$; $\text{Excess}(n,n) = \lfloor n/2 \rfloor$, so, by Corollary 1.5, the diameter is bounded by $\lfloor 5n/2 \rfloor$. Ad hoc arguments show that the diameter is actually 1 for $n = 1$, 4 for $n = 2$, 6 for $n = 3$, and $\lfloor 5n/2 \rfloor - 2$ for $n > 3$. Thus, $k = 5/2 \lg(N) + O(\lg \lg(N))$, and $k \cdot d = 15/2 \lg(N) + O(\lg \lg(N))$.

Preparata's original definition appears to be exponentially interpolable; however, the diameter bound and routing characteristics are preserved when our general single-exchange definitions are used, producing a semi-linearly interpolable family with factor 2. Cube-connected-cycles with $m > n$ can also be mapped into a version of the single-exchange construction, by using the same renumbering for $c < n$ and leaving the addresses of the vertices with $c \geq n$ unchanged. Such graphs are less dense, so this variation of the multistage single-exchange construction has not been pursued.

2.4.4.2. The lens

The lens family was originally proposed for a multicomputer model with busses instead of point-to-point communications [28]. However, as Finkel and Solomon point out, a completed lens with b busses per processor and b processors per bus can also be viewed as a regular graph of degree 2b. Given parameters b and $n > 1$, a lens in this interpretation can be defined that has nb^n vertices, degree 2b, and diameter $= \lfloor 3n/2 \rfloor$ as follows: Given $x = x_0\ x_1 \ldots x_{n-1} \in b^n$, define $\text{NewBit}(x,m,i) = x_0\ x_1 \ldots x_{m-1}\ i\ x_{m+1} \ldots x_{n-1}$, that is, x with the m-th bit replaced with i. Assign the vertices unique addresses of the form $\{s,x\}$, where $0 \leq s < n$, and $x \in b^n$. Vertex $\{s,x\}$ is then connected to vertex $\{s+1 \text{ modulo } n, \text{NewBit}(x,s,i)\}$ $\forall$ $0 \leq i < b$, and to vertices $\{s-1 \text{ modulo } n, \text{NewBit}(x,s-1,i)\}$ $\forall$ $0 \leq i < b$. This con-

struction has been deliberately written to show its similarity with the cube-connected-cycles; the same automorphism taking $\{s,x\}$ to $\{s,\rho^s(x)\}$ produces a multistage single-exchange graph with S = n, base b, d = 2b, and C = 1. By Theorem 1, the diameter is bounded by n + Excess(n,n) = $\lfloor 3n/2 \rfloor$; ad hoc arguments again show that this bound is exact. Thus, k = 3/(2 lg(d/2)) lg(N) + O(1) and k · d = 3d/(2 lg(d/2)) lg(N) + O(d).

When cast into the form of our single-exchange definitions, this family is semi-linearly interpolable with interpolation factor b, thereby answering the question of partial lenses proposed by Finkel and Solomon [28]. The same family of graphs, in the exponentially interpolable version, has been proposed for multicomputers by Farhi, under the name C_S graphs [29]. By designating most stages switches instead of processors, the base-2 lens can be reinterpreted as the popular permutation network variously called [30,31] the "multistage shuffle-exchange" [32,26], the S-W banyan with S = F = 2 [33], the omega [1], and the indirect binary n-cube [34].

2.4.4.3. The shuffle-exchange graph

The family of multicomputer graphs called shuffle-exchange graphs [2,3,4] has already appeared in sections

2.4.1 and 2.4.2 as a specific example of the single-exchange definition. The family is defined for N = 2^n by assigning unique addresses in 2^n to the vertices, and connecting vertex v to $\rho(v)$, $\rho^{-1}(v)$, and $SE_1(v)$. The resulting graphs are clearly trivalent single-exchange graphs with S = 1 and C = 2. Theorem 1 shows the diameter bounded by 2n = 2 lg(N); ad hoc arguments show the actual diameter is 2 lg(N) - 1, for a product cost of 6 lg(N) - 3.

2.4.4.4. The de Bruijn networks

For all even degrees above 3, the densest previously-known infinite families are the undirected versions of de Bruijn networks [35], where k = 1/lg(d/2) lg(N) and k · d = d/lg(d/2) lg(N). Although de Bruijn networks must have even degree, their asymptotic costs were until recently not only less than the costs of any previously published family of the same degree d, but also less than those of any family with degree d + 1. The double-exchange graphs introduced later in this chapter now excel for degrees 4 and 5, while a variation of de Bruijn graphs proposed by Farhi [29], called C'_S graphs, now excels at odd degrees of 7 or higher. For other recent variations and rediscoveries of de Bruijn networks, see Stone [36], Golunkov [37], Lam [38], Imase

and Itoh [39], and Pradhan [40].

Given a base b > 1 and some number N of vertices, where N is a multiple of b, a de Bruijn network with degree 2b is constructed by assigning each vertex a b-ary address and connecting vertex x with vertices $SE_i(\rho(x))$ $\forall$ $0 \leq i < b$. De Bruijn's original construction was a directed graph with N a power of the base b; when the construction is used for undirected graphs, some b^2 vertices (independent of N) have degree less than 2b, due to the specification of self-loops and multiple edges involving the vertices $(az)^*$ or $(az)^*a$ for arbitrary b-ary digits a and z. The edges of the original directed de Bruijn network provide exactly the same single-step processor-to-processor interconnections as the permutation network sometimes called the single-stage shuffle-exchange [26,36].

The undirected de Bruijn network is a single-exchange graph with S = 1 and cost parameter C = 1. Theorem 1 shows that the diameter is bounded by $\lceil \log_b N \rceil$; the observation that changing one b-ary digit requires at least 1 edge proves that the diameter is at least $\lceil \log_b N \rceil$. The diameter k is therefore $1/\lg(b)\ \lg(N) = 1/\lg(d/2)\ \lg(N)$, and $k \cdot d$ is $d/\lg(d/2)\ \lg(N)$. Interestingly, the ratio of either cost measure for the de Bruijn to the optimum possible is $\lg(d-1)/\lg(d/2)$,

which approaches 1 as d approaches infinity. Until the development of our double-exchange graphs, the degree-6 de Bruijn had the least product cost (3.785 lg N) of any family known.

The corresponding multistage single-exchange graph corresponds closely to the permutation network called the multistage shuffle-exchange. A de Bruijn with degree 2b, S stages, and stage-size M has, by Theorem 1, diameter bounded by n + Excess(S,n), where $n = \lceil \log_b M \rceil$. When S = n, the family includes the lens; when M is a power of b, Farhi [29] calls these C_S graphs, and proves that the diameter is in fact equal to n + Excess(S,n).

2.4.4.5. The C'_S graph

For odd degree at least 5, Farhi proposes the following variation, the C'_S graph, which gives $k = 2/\lg(\lfloor d/2 \rfloor \cdot \lceil d/2 \rceil)\ \lg(N) + O(1)$. Let $b = \lfloor d/2 \rfloor \cdot \lceil d/2 \rceil$, and let b_1 and b_2 denote $\lfloor d/2 \rfloor$ and $\lceil d/2 \rceil$ respectively. The graph consists of $Sb_1^n b_2^n$ vertices with addresses of the from {s,x,y}, where s is the stage, x is a string of n b_1-ary digits, and y is a string of n b_2-ary digits. When s is even, {s,x,y} has edges to $\{s+1, SE_i(\rho(x)), y\}$ and $\{s-1, x, \rho^{-1}(SE_j(y))\}$, where $0 \leq i < b_1$ and $0 \leq j < b_2$; when s is odd, {s,x,y} has edges to $\{s+1, x, SE_j(\rho(y))\}$ and $\{s-1, \rho^{-1}(SE_i(x)), y\}$. Any vertex can

reach any other in a stage 2n stages away by alternately moving in the x and y digits of the destination; other vertices are reached by first routing to any vertex whose stage is 2n away from the destination.

At degree 5, the C'_S family has diameter $k = (2/\lg 6) \lg N \doteq 0.774 \lg N$ and $k \cdot d \doteq 3.869 \lg N$. Until the development of our double-exchange graphs, this family had the least diameter of any degree-5 family known. Unlike the lens, a single-stage version of the C_S graphs is not possible: S must be even. Farhi's original definition requires M to be a power of b, but our general ρ operation (instead of the original rotate) makes the families linearly interpolable with factor 2b -- M can be any multiple of b, while S is at least 2.

2.4.5. Operations that construct single-exchange families

We can define several natural operations on graphs that build infinite families of single-exchange graph whose costs and routing algorithms are determined directly from the initial graph. The simplest operation on a given graph G with N_G vertices, degree d_G and diameter k_G is to build a single-exchange graph with S stages and any multiple, say m, of N_G vertices per stage. Arbitrarily assign unique addresses from the set

$\{0,1,\ldots,N_G-1\}$ to the vertices of G. Let the base $b = N_G$, the stage-size $M = mb$, and $n = \lfloor \log_b M \rfloor$. The vertices in the single-exchange graph are the formal pairs $\{s,x\}$, where $0 \le s < S$ and $0 \le x < M$. Connect vertex $\{s,x\}$ to vertices $\{s+1,\rho(x)\}$, $\{s-1,\rho^{-1}(x)\}$, and $\{\{s,y\} \mid$ x modulo b is adjacent to y modulo b in G$\}$. The resulting single-exchange graphs have degree $d = 2+d_G$ and mbS vertices. The value of x modulo b (intuitively, the least significant digit of x) can be changed by using the edges of G; a ρ requires a single edge, so $C = 1+k_G$, $C_+ = 1$, and, by Corollary 1.5, $k \le (1+k_G)n +$ Excess(S,n). This construction immediately creates the shuffle-exchange family by taking $G = K_2$ and $S = 1$, and the cube-connected-cycles by taking $G = K_2$, a stage-size of 2^n, and n stages. It also produces a degree 5 family with $k \le 3/\lg(10) \lg(N) \doteq .903 \lg(N)$, by using the Petersen graph for G; the only degree-5 families known with lower diameter are the C'_S family and our double-exchange family.

A slight generalization includes all de Bruijn networks and all lenses. Given G, choose an integer $e > 0$ and construct a single-exchange family with degree $= d+2e$, base $b = eN_G$, S stages, and stage-size mb for any m by defining the vertices as pairs $\{s,x\}$ as before. Define CopyNum(x) = (x modulo b) div N_G and Element(x) =

(x modulo b) modulo N_G. Each digit of x may be thought of as encoding e copies of G; in effect, CopyNum(x) selects one copy of G from the least significant digit of x, and Element(x) determines a particular vertex in G. Connect vertex {s,x} with vertices {{s+1,y} | Element(y) = Element(ρ(x))} {{s-1,y} | Element(y) = Element(ρ^{-1}(x))} and {{s,y} | CopyNum(y) = CopyNum(x) and Element(y) is adjacent to Element(x) in G}. The cost parameters C and C_+ of this single-exchange family are again $1+k_G$ and 1, respectively, so by Corollary 1.5, the diameter is bounded by $(1+k_G)n$ + Excess(S,n). By picking G = K_1, and S = 1, we create the degree 2e de Bruijn networks; choosing S = n produces the corresponding lens family. The diameter bound provided by Corollary 1.5 is tight for the cube-connected-cycles, the shuffle-exchange, the de Bruijn, and the lens; as mentioned above, ad hoc arguments show that the actual diameters are at most 2 less than our general bound for single-exchange families.

The graph operations just defined, together with Theorem 1 and its corollaries, allow any ad hoc inexpensive graph to be used to generate a linearly interpolable infinite family of inexpensive graphs. The constructions provide a simple routing algorithm for each family generated, for which one only needs to know how

to route in the original graph G: The destination address is shifted in, digit by digit, using the routing in G to implement the SE_i operations and our general ρ operation for the rotations. However, despite the recent progress in constructing ad hoc dense graphs for small diameters [41,42,29,43,44,45,46,19], these single-exchange graph operations cannot yet surpass the de Bruijn, $C_S^{\prime}$, and double-exchange families.

2.5. Double-Exchange Graphs

Surprisingly, a slight change in the single-exchange construction produces quite different -- and often superior -- graphs. We will first define this construction for single-stage double-exchange graphs in which N is a power of the base, $N = b^n$. The requirement that N is a power of the base not only simplifies the definitions, but also allows us to prove a lower diameter bound. Our construction uses the same rotate operation, ρ, but introduces a new "exchange-like" operation that we call the double-exchange.

2.5.1. Definitions

Given $x = x_0\ x_1 \ldots x_{n-1} \in b^n$, the digit sum of x, Sum(x), is Σx_i. (As usual, all arithmetic involving digits of x is modulo b, while arithmetic involving subscripts of its digits is modulo n.) We define $\rho(x) = x_1\ x_2 \ldots x_{n-1}\ x_0$ and $\rho^{-1}(x) = x_{n-1}\ x_0\ x_1 \ldots x_{n-2}$. If $n \geq 2$, then for all $0 \leq i < b$, the double-exchanges of x are $DE_i(x) = x_0\ x_1 \ldots x_{n-3}\ x_{n-2}-i\ x_{n-1}+i$. Given an integer $0 \leq i < b$, $x \oplus i$ will denote $x_0+i\ x_1+i \ldots x_{n-1}+i$. (So $DE_0(x) = x = x \oplus 0$.)

Let X be a nonempty set of vertices in a connected graph G. X is a double-exchange set $X_{b,n}$ with costs C_i if there exists a labeling of the vertices of X with distinct addresses in b^n such that:

a) If x and y are both in X, Sum(x) = Sum(y).

b) $\forall\ x \in X$ and $\forall\ 0 \leq i < b$, $DE_i(\rho(x)) \in X$;

c) There are constants C_i such that $\forall\ x \in X$ and $\forall\ 0 \leq i < b$, the distance from x to $DE_i(\rho(x))$ is $\leq C_i$.

If the double-exchange set X includes all vertices in G, G is a double-exchange graph, $G_{b,n}$ with costs C_i. We will show below that any double-exchange set $X_{b,n}$ has b^{n-1} vertices.

Given a vertex v in a double-exchange set, consider a path from v that begins with a ρ and consists of alternating ρ and DE operations: $\rho\ DE_{j_0}\ \rho\ DE_{j_1} \ldots \rho\ DE_{j_k}$. Each j_i satisfies $0 \leq j_i < b$, so the j's form a b-ary string $j_0 j_1 \ldots j_k$. Conversely, any b-ary string x can represent a sequence of DE subscripts. We write $\hat{x}$ for the corresponding path and $v \circ x$ for the vertex reached from v by following $\hat{x}$.

2.5.2. The diameter of a double-exchange set

In this section, we will prove the following:

Theorem 2. A double-exchange set $X_{b,n}$ has b^{n-1} vertices and diameter bounded by $\lfloor \frac{n}{b} \Sigma C_i \rfloor$.

Let u be a vertex in a double-exchange set $X_{b,n}$. Let v be a string in b^n. Lemmas 2.1 and 2.2 will show

that addresses with equal digit sums are connected by paths of the form $(\rho\ DE)^*$, so that X has b^{n-1} members; Lemma 2.4 will use the paths of Lemma 2.2 to prove the diameter bound.

Lemma 2.1. Given v and $x \in b^n$ such that $v = u \circ x$, $\forall\ 0 \le i < n$, $v_i = u_i + x_i - x_{i+1}$.

Proof. This lemma follows from a tedious but straightforward induction.

∎

Lemma 2.2. If Sum(u) = Sum(v), there is a path from u to v.

Proof. We will show there exists a string $x \in b^n$ such that $v = u \circ x$; property c will then guarantee that the path X exists. For fixed u and v, Lemma 2.1 provides n linear equations in the n unknowns x_i:

$$x_{i+1} = u_i - v_i + x_i,\ 0 \le i < n$$

However, these equations are not independent. Solving in terms of x_0 yields

$$x_i = \sum_{j<i}(u_j - v_j) + x_0, \quad 1 \le i < n$$

and $$x_0 = \sum_{j<n}(u_j - v_j) + x_0 = \text{Sum}(u) - \text{Sum}(v) + x_0$$

Therefore, Sum(u) must equal Sum(v). When this requirement is met, setting $x_0 = 0$ determines the remaining x_i such that $v = u \circ x$. For later reference, we note that x_0 may be assigned any value, yielding b distinct solutions of the form $x \oplus j$, $0 \le j < b$.

∎

Corollary 2.3. A double-exchange set $X_{b,n}$ has b^{n-1} members.

Proof. A double-exchange set is defined to be nonempty. Given $u \in X_{b,n}$, Lemma 2.2 shows there is a path from u to an address v using subpaths of the form $DE_i(\rho(x))$ whenever Sum(u) = Sum(v). Property b then implies that any such v is in X. There are b^{n-1} distinct strings v such that Sum(v) = Sum(u), since the first n-1 digits may be chosen freely, and the remaining digit chosen to produce the correct digit sum.

∎

In particular, Corollary 2.3 shows that a double-exchange graph has $N = b^{n-1}$.

Lemma 2.4. If Sum(u) = Sum(v), then there exists a path from u to v of length less than or equal to $\lfloor \frac{n}{b} \Sigma c_i \rfloor$.

Proof. Let $x \in b^n$ be the string defined in Lemma 2.2, where $\forall\ 0 \le i < b$, $v = u \circ (x \oplus i)$. In order to calculate the minimum distance from u to v along these paths, we

must find the minimum sum of the C_j's over the b alternative paths. For $0 \leq i < b$, let n_i be the number of j's such that $x_j = i$. The contribution of the C_i's to the length of $\mathfrak{X} \oplus i$ is $\Sigma C_j n_{j+i}$.

Suppose the minimum of this sum is greater than $\frac{n}{b} \Sigma C_j$. Then $\forall$ i, $\Sigma C_j n_{j+i} > \frac{n}{b} \Sigma C_j$. So $\Sigma(\Sigma C_j n_{j+i}) > n \Sigma C_j$. Since all the subscript arithmetic is modulo b, each C_j occurs in a product exactly once with each n_i. We therefore can interchange the summations to show that $\Sigma C_j(\Sigma n_i) > n \Sigma C_j$. But $\Sigma n_i = n$, because each digit in x is counted exactly once, yielding the contradiction that $\Sigma C_j\ n > n \Sigma C_j$. Therefore there is some i for which the length of $\mathfrak{X} \oplus i$ is $\leq \frac{n}{b} \Sigma C_j$. All distances in a graph are integral, so Distance(u,v) $\leq \lfloor \frac{n}{b} \Sigma C_i \rfloor$.

M

Proof of Theorem 2 Since all vertices in a double-exchange set have the same digit sum, Lemma 2.4 establishes the diameter bound of Theorem 2.

M

Some later constructions will use the following corollary.

Corollary 2.5. If a graph G contains a double-exchange set $X_{b,n}$ such that any vertex in G is at most distance r from some member of the double-exchange set X, then the diameter of G is less than or equal to 2r + the diameter of X.

Proof. Immediate.

M

Many of the later constructions in fact have a slightly lower diameter bound, although not enough lower to alter the asymptotic costs.

Corollary 2.6. If a double-exchange set $X_{b,n}$ has the property that there is a constant $C_R > 0$ such that $\forall$ vertices x and $\forall$ $0 \leq i < b$, the distance from x to DE(x,i) is bounded by $C_i - C_R$, then the diameter of X is less than or equal to $\lfloor \frac{n}{b} \Sigma C_i \rfloor - C_R$.

Proof. We can omit the initial ρ in any path $\mathfrak{X}$, and repeat the proof of Theorem 2. The counterpart of Lemma 2.1 is $\forall$ $0 \leq i < n$, $v_i = u_{i-1} + x_i - x_{i+1}$. The counterpart of Lemma 2.2 then yields the n equations

$$x_{i+1} = u_{i-1} - v_i + x_i,\ 0 \leq i < n$$

These equations are also dependent, and the rest of the proof continues unaltered, producing a final cost that is lower by the use of DE instead of ρ DE at the start of the path.

M

2.5.3. Routing in double-exchange families

The proof of Theorem 2 implies a routing algorithm for double-exchange sets that is analogous to the routing algorithm for single-exchange graphs. Because we have restricted our attention to addresses in b^n, however, the algorithm is much simpler.

Algorithm.

Input: A pair of vertices u and v in a double-exchange set $X_{b,n}$.

Output: A path from u to v of length $\leq \lfloor \frac{n}{b} \Sigma C_i \rfloor$.

Method: For $y \in b^n$, define

$$Cost(y) = \Sigma C_{y_i}.$$

Define $x_0 = 0$ and for $i = 0,\ldots,n-2$

$$x_{i+1} = u_i - v_i + x_i$$

For $0 \leq j < b$, compute $Cost(x \oplus j)$. Let j_{min} be the value of j such that $Cost(x \oplus j)$ is minimal. This calculation requires no more than O(bn) steps; for any particular family, b is fixed. The desired path is then $\bar{x} \oplus j_{min}$.

2.5.4. Specific double-exchange families

The double-exchange construction will allow us to surpass any previously published asymptotic density for degrees 3, 4, and 5. For any b and n, we can construct a double-exchange graph $G_{b,n}$ by providing an edge from each vertex for each of the operations ρ, ρ^{-1}, and DE_i for all $0 < i < b$. Since $d = 1 + b$, Corollary 2.6 ensures that $N = (d-1)^n$ and the diameter is $< (2d-3)/((d-1) \lg (d-1)) \lg(N)$. Fixing $b = 2$ yields a family of degree 3 graphs with $k < 3/2 \lg(N)$, compared to 2 lg N for the best previous trivalent families. For higher degrees, this basic construction does not surpass the de Bruijn density. At degree 4, $k \leq 5/(3 \lg 3) \lg(N) \approx 1.052 \lg(N)$, in contrast to $k = \lg(N)$ for the de Bruijn network. The comparison worsens as d increases: For large d, $k \approx 2/\lg d \lg(N)$, compared to $1/\lg \lfloor d/2 \rfloor \lg(N)$ for de Bruijn networks. However, there are construction techniques that produce cheaper infinite families by using paths instead of single edges to represent the operations.

2.5.4.1. Degree 3

The trivalent graphs just constructed with $b = 2$ have $N = 2^{n-1}$ vertices and a diameter bounded by $\lfloor 3n/2 \rfloor - 1$; *ad hoc* arguments show that the diameter is at

least $\lfloor 3n/2 \rfloor - 2$. In view of the simplicity of their construction and the fact that all previously published infinite trivalent families have diameter at least 2 lg(N), we will study them in more detail in later chapters. The construction defines two graphs for a given n -- one whose addresses have an even number of ones, and one whose addresses have an odd number. When n is odd, each address in one graph has its complement in the other, and the two graphs are isomorphic. When n is even, the graphs are not isomorphic, with the even parity graph having two vertices of degree 1 while the odd parity graph has none. If we always select the even parity graph for each n, the resulting even double-exchange family has a close relationship with the shuffle-exchanges, which is explored in the next chapter.

Our cheapest trivalent variation gives $k/\lg(N) < 7/(3 \lg 3) \approx 1.472$. For it, we use ternary addresses and two sets of vertices, $V \subseteq 3^n$ and $W \subseteq 3^{n-1}$. A vertex $v \in V$ has two neighbors in V, $\rho(v)$ and $\rho^{-1}(v)$, and one neighbor in W: $v_0\ v_1 \ldots v_{n-3}\ v_{n-2}+v_{n-1}$. A vertex $w \in W$ has three neighbors, all in V: v, $DE_1(v)$, and $DE_2(v)$, where $v = w_0\ w_1 \ldots w_{n-3}\ w_{n-2}\ 0$. V is a double-exchange set with $C_0 = 1$ and $C_1 = C_2 = 3$, so it has 3^{n-1} members, all within distance $\frac{7}{3} n - 1$. There are 3^{n-2} elements of

W adjacent to these v's, any two of which lie within distance $1 + \frac{7}{3}n$ of each other, by Corollary 2.5, giving the final graph a diameter bounded $1 + \frac{7}{3}n$ for $4 \cdot 3^{n-2}$ vertices. Its costs are then $k < 1.472 \lg(N)$ and $k \cdot d < 4.417 \lg(N)$.

2.5.4.2. Degree 4

The densest infinite family previously known with degree 4 is the binary de Bruijn network, which has $k = \lg(N)$. We can construct two double-exchange families that have lower cost. By using base 7, graphs can be constructed with $k \leq .967 \lg(N)$: connect vertex v with $\rho(v)$, $\rho^{-1}(v)$, $DE_1(v)$, and $DE_6(v)$. Then $C_0 = 1$, $C_1 = C_6 = 2$, $C_2 = C_5 = 3$, and $C_3 = C_4 = 4$. By Theorem 2, $N = 7^{n-1}$, while by Corollary 2.6 (with $C_R = 1$), $k < \lfloor 19\ n/7 \rfloor \leq 19/(7 \lg 7) \lg(N) \approx .967 \lg(N)$.

An even better infinite family of degree-4 double-exchange graphs is produced using base 5 strings and a double-exchange of plus or minus 1. More precisely, we define a graph in which a vertex whose address is the base 5 string v has neighbors $\rho(v)$, $\rho^{-1}(v)$, $DE_1(v)$, and $DE_4(v)$. This construction makes $C_0 = 1$ and $C_1 = C_4 = 2$. $DE_2 = DE_1\ DE_1$ and $DE_3 = DE_4\ DE_4$, so $C_2 = C_3 = 3$. Corollary 2.3 shows that $N = 5^{n-1}$, and Corollary 2.6, with $C_R = 1$, requires that $k < \lfloor 11\ n/5 \rfloor$, so $k < 11/(5 \lg 5)$

$\lg(N) = .947 \lg(N)$.

2.5.4.3. Degree 5

There are many infinite double-exchange families with degree 5 and $k < \lg(N)$. The best has $k \leq 3/4 \lg(N)$, and is constructed by letting $b = 4$ and connecting vertex v to $\rho(v)$, $DE_1(\rho(v))$, and $DE_2(v)$. Since DE_2 is its own inverse, each vertex has five or fewer neighbors. Clearly, $C_0 = C_1 = 1$ and $C_2 = 2$, while $C_3 = 2$, since $\rho\ DE_3$ can be implemented by $\rho\ DE_1\ DE_2$. Lemmas 2.2 and 2.4 then allow 4^{n-1} vertices to lie within diameter $n + \lfloor n/2 \rfloor$, giving $k \leq 3/(2 \lg 4) \lg(N) = 3/4 \lg(N)$. The resulting $k \cdot d$ of $15/4 \lg(N) = 3.75 \lg(N)$ is the lowest product cost for any published infinite family at any degree.

2.5.5. Variations of the double-exchange

We considered several variations of the double-exchange definition in the hope of improving the diameter bounds still further. Repeating the arguments of lemmas 2.1 and 2.2 shows that increasing the number of digits affected by the exchange to make "triple" or "quadruple" exchange families does not improve the diameter. On the chance that using noncontiguous digits in the exchange would give better diameters, we wrote a

program that explored all possible patterns of exchange within the last 6 bits of the even double-exchange, for n between 6 and 11 inclusive (so N was generally between 32 and 2048, depending on the number of connected components produced). None of these variations had greater density than the double-exchange; the ones that equaled it turned out to be logically equivalent.

There are, however, two related constructions that can produce denser graphs, although without improving the asymptotic value of $k/\lg(N)$. These are the twisted double-exchange and the multistage double-exchange. The diameter bounds for both constructions rely on Theorem 2.

2.5.5.1. The twisted double-exchange

For $x = x_0\ x_1\ \ldots\ x_{n-1} \in b^n$, define $\text{TwistedRotate}(x) = x_1\ x_2\ \ldots\ x_{n-1}\ x_0+1$. A twisted double-exchange set $X_{b,n}$ with costs C_i is defined in exact analogy with a double-exchange set, except that TwistedRotate is used in place of ρ. Let C_{min} be the least C_i.

Corollary 2.7. A twisted double-exchange set $X_{b,n}$ has b^n vertices and diameter less than or equal to $bC_{min} + \lfloor \frac{n}{b} \Sigma C_i \rfloor$.

Proof. Because each TwistedRotate increments the digit-sum, Sum(x), by one, a straightforward analogy of the proof of the double-exchange theorem allows any vertex u to reach any address v ∈ b^n such that Sum(v) - Sum(u) = n modulo b, in distance bounded by $\lfloor \frac{n}{b} \Sigma C_i \rfloor$. Therefore, any vertex u in X can reach any address v in b^n by an initial sequence of at most b TwistedRotates (a path of length at most bC_{min}), followed by a path of length $\lfloor \frac{n}{b} \Sigma C_i \rfloor$.

■

2.5.5.2. The Moebius graph

For small values of the C_i's, this construction then allows somewhat denser graphs than the untwisted version. In particular, the construction for the degree 3 even-parity double-exchange family given above is improved by using a twisted rotate, producing a family of trivalent graphs with $N = 2^n$ and $k \leq \lfloor 3n/2 \rfloor$. We call the resulting graphs Moebius graphs [47], because of their twist, and study them in more detail in chapter 4.

2.5.5.3. The multistage double-exchange

Multistage versions of double-exchange sets can be defined in exact analogy with the multistage single-exchange graph definition. Given a number of stages S,

a base b, and a parameter n, we assign addresses of the form {s,x} to vertices in the set, where $0 \leq s < S$ and $x \in b^n$. StageRotate({s,x}) is thus {s+1,ρ(x)}, and $StageDE_i(\{s,x\})$ is $\{s,DE_i(x)\}$. A multistage double-exchange set with costs C_i is then defined by replacing the operations in the double-exchange definition with the corresponding staged versions. Again, let C_{min} be the least C_i.

Corollary 2.8. A multistage double-exchange set with base b, parameter n, S stages, and costs C_i has b^{n-1} vertices and diameter less than or equal to $\lfloor \frac{n}{b} \Sigma C_i \rfloor$ + C_{min} Excess(S,n).

Proof. Using the proof method of Theorem 2, we show that a vertex {s,x} can route to any vertex {s+n,y} or {s-n,y} in distance less than or equal to $\lfloor \frac{n}{b} \Sigma C_i \rfloor$. Then, as in Theorem 1, we show that we can route to some vertex n stages away from any given destination within distance C_{min} Excess(S,n).

■

Corollaries 2.7 and 2.8 improve only the low order terms in the cost functions; however, such a reduction can make a large difference in the number of vertices that a graph of given degree and diameter can have. At degree 3, for example, the most vertices for a given di-

ameter under the various choices are:

diameter:	3	4	5	6	7	8	9	10
shuffle-exchange:	4	--	8	--	16	--	32	--
double-exchange:	--	8	16	--	32	64	--	128
twisted DE:	8	16	--	32	64	--	128	256
multistage DE:	10	14	24	72	88	208	256	448

2.5.6. Interpolating double-exchange families

The double-exchange constructions presented so far produce families that are highly dense but exponentially interpolable (for a fixed number of stages). If linear interpolability is more critical than diameter, the equivalent linearly interpolable constructions can be defined in analogy with the single-exchange construction. The generalized rotate operation is the same for both constructions, but the definition of the double-exchange is more complex than the single-exchange: Given a base $b > 1$ and some number $N > 1$ that is a multiple of b^2, define $\rho(x)$ to be (xb) modulo N + xb div N, and define $DE_i(x)$ to be

$$(x \text{ div } b^2) \cdot b^2 +$$
$$(((x \text{ modulo } b^2) \text{ div } b - i) \text{ modulo } b) \cdot b +$$
$$((x \text{ modulo } b) + i) \text{ modulo } b.$$

Intuitively, the first term in the DE expression sets

the two low-order digits of x to 0, the second term subtracts i from the next-to-last digit, and the third term adds i to the last digit. N is a multiple of b^2, so $DE_i(x)$ is a bijection on [0,N-1] for any i, with $DE_{-i}(DE_i(x)) = x$. Unfortunately, we have not found a tight diameter bound for such constructions. Letting C be the maximum of the costs C_i, we can prove the following:

Corollary 2.9. An interpolated double-exchange set X has diameter less than or equal to $C \lceil \log_b N \rceil$.

Proof. From any vertex u, there is a path of length at most C to (ub+i) modulo N; as in Lemma 1.2, $\lceil \log_b N \rceil$ iterations reach any v.

∎

For the degrees we have explored, the linearly interpolated double-exchange graphs are generally denser than the corresponding single-exchange graphs, and twisted graphs are denser than the untwisted versions. Considering the 250 trivalent graphs with N a multiple of four between 4 and 1000, for example, the mean value of k/lg N is 1.55 for the shuffle-exchange, 1.49 for the even double-exchange, and 1.37 for the Moebius. The definition can be extended to include multiple stages for further improvements in density; for comparison, we re-

peat the example from the last section with the addition of linear interpolation:

diameter:	3	4	5	6	7	8	9	10
shuffle-exchange:	4	-	8	12	20	68	84	156
double-exchange:	-	8	16	24	36	68	120	188
twisted DE:	8	16	--	36	64	108	180	276
multistage DE:	10	14	24	72	88	208	300	484
multistage twist DE:	16	24	32	80	108	192	260	600

The values for the multistage versions represent the best results found among graphs with not more than 17 stages and not more than 256 vertices per stage. Although it may not represent the densest possible multistage Moebius for diameter 10, the value of 600 vertices represents the densest trivalent graph currently known at that diameter [46].

2.6. Summary

The degrees and costs of the many families of graphs proposed for multicomputer interconnection models are summarized in Table 2.1. The densest have fixed degree and diameters logarithmic in N, the number of processors. Our double-exchange families have notably high density as well as simple routing algorithms. In particular, for degrees three, four, and five, our double-exchange families have the lowest diameters; our best degree five family also has the lowest product cost of any family yet reported.

Table 2.1

Degrees and Costs of Graph Families

	Diameter	Product Cost
Degree 3		
tree	2 lgN	6 lgN
cube-connected cycles	2.5 lgN	7.5 lgN
shuffle-exchange	2 lgN	6 lgN
double-exchange	1.5 lgN	4.5 lgN
ternary double-exchange	1.472 lgN	4.417 lgN
Degree 4		
tree	1.262 lgN	5.047 lgN
Hypertree 1	1.5 lgN	6 lgN
binary lens	1.5 lgN	6 lgN
binary De Bruijn	1 lgN	4 lgN
double-exchange	0.947 lgN	3.790 lgN
Degree 5		
tree	1 lgN	5 lgN
Hypertree 2	1.333 lgN	6.667 lgN
$C_S^{'}$	0.774 lgN	3.869 lgN
double-exchange	0.75 lgN	3.75 lgN
Unbounded Degree		
binary hypercube	1 lgN	1 lg^2N
mesh-connected hypercube	variable	0.742 lg^2N
complete hypercube	variable	0.742 lg^1N

Chapter 3

Algorithms and Layout

In the previous chapter, we defined the shuffle-exchange and even double-exchange families of trivalent graphs. These families have very similar definitions; the even double-exchange family uses the double-exchange, and the shuffle-exchange family uses the single-exchange. Nevertheless, the even double-exchange has much better diameter: 1.5 lg N instead of 2 lg N. This chapter briefly explores the consequences of a homomorphism between the two families that allows the even double-exchange family to exploit known results on algorithms [2,48,26,49] and layouts [3,4] for shuffle-exchange graphs.

Throughout this chapter, all strings are in 2^n; continuing the conventions of Chapter 2, all arithmetic involving bits is modulo 2 and all arithmetic involving subscripts is modulo n. The bitwise complement of v is denoted by ∇ or $\tilde{}v$.

3.1. Properties of the Homomorphism

A graph homomorphism is a mapping that preserves adjacency. For a given n, define a function h from all binary strings of length n to the even-parity binary strings of length n by $h(v_0v_1\ldots v_{n-1}) \rightarrow w_0w_1\ldots w_{n-1}$, where $w_i = v_i \oplus v_{i+1}$. Intuitively, this function identifies each vertex v of a shuffle-exchange graph with its complementary vertex $\bar{v}$.

Theorem 1. The mapping h is a homomorphism from a shuffle-exchange graph with 2^n vertices onto an even double-exchange graph with 2^{n-1} vertices.

Proof. H is clearly onto, mapping two strings v to each even-parity string w: v_0 may be chosen arbitrarily, and for $i > 0$, $v_i = v_0 + \Sigma\{w_j \mid 0 \le j < i\}$. We call the string where v_0 is zero, w^0, and the other w^1.

The mapping h preserves adjacency. If $u = \rho(v)$ or $\rho^{-1}(v)$, the fact that the subscripts are calculated modulo n immediately ensures that $h(u) = \rho(h(v))$ or $\rho^{-1}(h(v))$, respectively. If $u = SE(v) = v_0v_1\ldots v_{n-2}\bar{v}_{n-1}$, $h(u) = v_0 \oplus v_1 \; v_1 \oplus v_2 \; \ldots \; v_{n-3} \oplus v_{n-2} \; v_{n-2} \oplus \bar{v}_{n-1} \; \bar{v}_{n-1} \oplus v_0 = v_0 \oplus v_1 \; v_1 \oplus v_2 \; \ldots \; v_{n-3} \oplus v_{n-2} \; \sim(v_{n-2} \oplus v_{n-1}) \; \sim(v_{n-1} \oplus v_0) = DE(h(v))$.

∎

Corollary 1.1 $h(u) = h(v)$ if and only if $u = v$ or $u = \bar{v}$.

Proof The "if" follows from the definition of h; the "only if" is established by a straightforward induction on the bits of u and v.

This homomorphism means that any distributed algorithm running on a shuffle-exchange with N processors can be emulated by an even double-exchange with N/2 processors, with each processor in the even double-exchange emulating exactly two processors in the shuffle-exchange. (In Fishburn's terminology [50], the emulation is computationally uniform.) The even double-exchange has 3/4 the diameter of the shuffle-exchange, so that algorithms that depend on broadcasting or echoing [6] should run in less than twice the time with half as many processors. The lower diameter and average distance can also reduce the average traffic load per processor, reducing queuing delays and congestion. The next theorem shows that most edges in the even double-exchange correspond to exactly two edges in the shuffle-exchange. This result suggests that emulation should not worsen traffic congestion, and also allows us to establish bounds on the VLSI layout area required by even double-exchanges.

Theorem 2. Each edge in the even double-exchange is the image of exactly two edges in the shuffle-exchange, except for the edge $(1^*0,1^*01)$ when n is odd and the edge $((10)^*,(01)^*)$ when n is a multiple of four. These two edges are each the image of four edges in the shuffle-exchange.

Proof. Each edge (u,v) in the even double-exchange is the image of at least two edges in the shuffle-exchange. If $v = \rho(u)$, the edge is an image of the edges $(u^0,\rho(u^0))$ and $(u^1,\rho(u^1))$, and similarly for ρ^{-1}; if $v = DE(u)$, the edge is an image of the edges $(u^0,SE(u^0))$ and $(u^1,SE(u^1))$. If h maps any other edge in the shuffle-exchange graph to (u,v), u^0 must obey one of these three conditions:

a) $\rho(u^0) = \sim(SE(u^0))$,

b) $\rho^{-1}(u^0) = \sim(SE(u^0))$,

c) $\rho(u^0) = \sim(\rho^{-1}(u^0))$.

Condition a is only satisfied when n is odd and u^0 has the form $(01)^*0$; condition b is only satisfied when n is odd and u^0 has the form $(01)^*1$. In these cases, either u is 1^*0 and v is 1^*01, or vice versa. Condition c is only satisfied when n is a multiple of four and u^0 has the form $(0011)^*$ or $(0101)^*$. In both cases, either u is $(10)^*$ and v is $(01)^*$, or vice versa.

M

3.2. VLSI Layout of the Even Double-Exchange

If advances in VLSI and wafer-scale integration permit entire networks of simple prccessors to be on one chip or wafer, the additional criterion of layout area may become important. Our model for the area required for the VLSI layout of a graph is due to Thompson [51,52]. In this model, processors and lines are laid out on a uniform rectilinear grid, with processors only being placed on grid intersections and lines only being allowed to follow the grid. (In other words, a processor must have integral coordinates and all points in a line must have at least one integral coordinate.) A line may not cross a processor, and is terminated by a processor at each end. Two lines may not occupy the same path in the grid, and may cross only at a grid intersection. The area of a layout is then the area of the smallest rectangle that contains all processors and lines. Among trivalent graphs, both the cube-connected-cycles and the shuffle-exchange require $Theta(N^2/lg^2N)$ area [52,27,4]. We will use these results and the homomorphism h to show that the even double-exchange also requires $Omega(N^2/lg^2N)$ area, and can be laid out in $O(N^2/lg^{3/2}N)$ area.

Definitions. A bisection of a graph $G = \{V,E\}$ is a subset E_B of E and a partition of V into two sets V_1 and V_2 such that $|V_1|$ is within one of $|V_2|$ and E_B contains each edge connecting a vertex in V_1 with a vertex in V_2. The width of a bisection is $|E_B|$; the minimum bisection width of a graph is the least width of any bisection of the graph.

Theorem 3. If there is a bisection of an even double-exchange graph of width m, then there is a bisection of the corresponding shuffle-exchange graph of width either 2m or 2m+2.

Proof. An even double-exchange graph always has an even number of vertices, so any bisection must have $|V_1| = |V_2|$. If E_B has m edges, then by Theorem 2, $h^{-1}(E_B)$ has either 2m or 2m+2 edges; $h^{-1}(V_1)$ and $h^{-1}(V_2)$ partition the vertices of the shuffle-exchange, and, by Theorem 1, have equal cardinality. Furthermore, $h^{-1}(E_B)$ must include any edge of the shuffle-exchange between a vertex v in $h^{-1}(V_1)$ and a vertex w in $h^{-1}(V_2)$, since the edge (h(v),h(w)) connects a member of V_1 with a member of V_2 in the even double-exchange. Thus, the inverse images of E_B, V_1, and V_2 provide a bisection of the corresponding shuffle-exchange, $h^{-1}(G)$.

M

Corollary 3.1. The minimum bisection width of an even double-exchange graph is at least Omega(N/lg N).

Proof. By Theorem 3, if the even double-exchange had a smaller bisection width, then so would the shuffle-exchange. Thompson, however, proved that the minimum bisection width of the shuffle-exchange is Omega(N/lg N).

M

Corollary 3.2. Any VLSI layout of the even double-exchange family requires area $\text{Omega}(N^2/\lg^2 N)$.

Proof. Thompson proved that the VLSI layout of a graph requires at least the square of its minimum bisection width.

M

Any set of m edges can be added to a layout by adding no more than m vertical and m horizontal tracks, increasing the area by $O(m^2)$. Together with Corollary 3.2, this fact simplifies the description of proposed layouts for the even double-exchange family, by allowing any particular set of O(N/lg N) edges or vertices to be ignored without invalidating the proof of the final area requirements.

Theorem 4. The even double-exchange family has a VLSI layout with $O(N^2/\lg^{3/2}N)$ area.

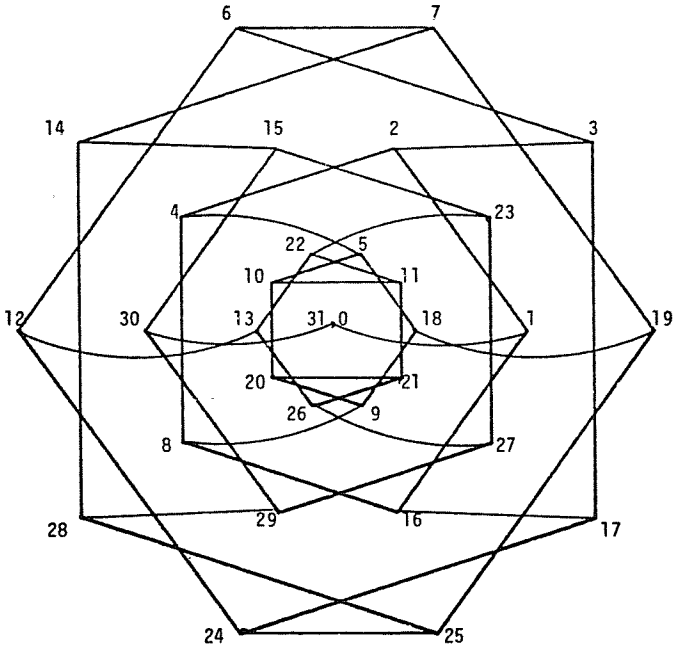
Proof. A necklace is a cycle of addresses created by successive ρ operations; each ρ edge and each vertex in a shuffle-exchange or even double-exchange graph lies in exactly one necklace, with at most $n = \lg N$ members. The length of any necklace must divide n, so at most $N^{1/2} \lg N$ vertices belong to necklaces with fewer than n members; this result in turn implies that there are $O(N/\lg N)$ necklaces. For the remainder of the proof, we assume all necklaces have n members, as $N^{1/2} \lg N$ is $O(N/\lg N)$.

Hoey and Leiserson [3] defined a layout of the shuffle-exchange based on a mapping g of the n-bit binary strings into the complex plane:

$$g(x_0x_1\cdots x_{n-1}) = \sum_{0\le j<n} x_j z^{n-1-j},$$

where $z = e^{2\pi i/n}$, the principal primitive complex n-th root of 1. Hoey and Leiserson prove that $g(\rho(x))$ in the complex plane is $g(x)$ rotated counterclockwise around the origin by $2\pi/n$ radians, so all vertices in a necklace are the same distance from the origin, and that $g(SE(x))$ is $g(x)$ plus or minus 1, so all SE edges connect vertices at the same distance from the real axis. The shuffle-exchange graph for $n = 5$, as it appears

Figure 3.1



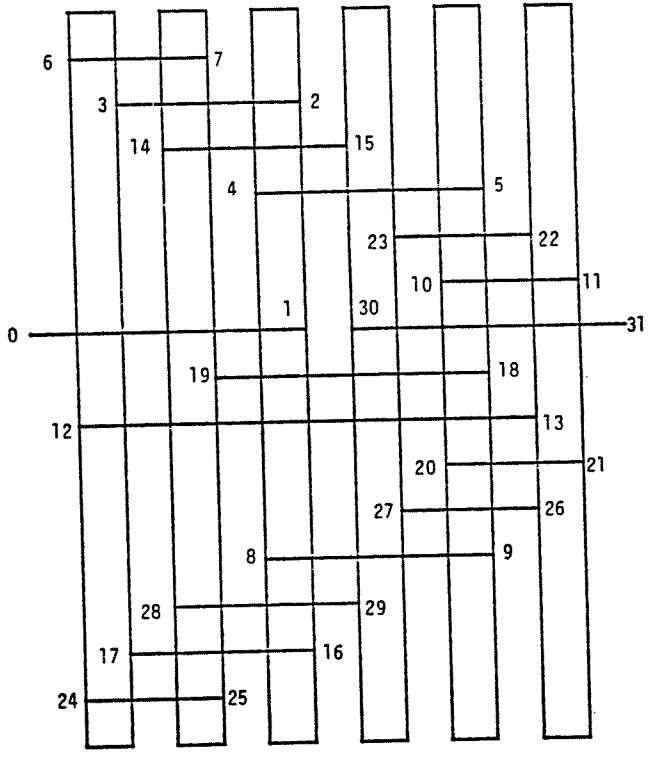
Mapping of the Shuffle-Exchange to the Complex Plane, for n = 5

under the mapping g, is illustrated in Figure 3.1 (from [3]).

To produce a layout for the shuffle-exchange, each necklace is given two adjacent vertical tracks for its vertices: the righthand track holds the vertices from the right half of the necklace's representation under g, and the left track holds the left half. All ρ edges are thus routed using a total of $O(N/\lg N)$ vertical tracks, together with two horizontal tracks to connect the two halves of each necklace. Because the vertices are placed on the vertical tracks according to their distance from the x-axis under g, every SE edge lies in a single horizontal track. Kleitman, Leighton, Lepley, and Miller [4] show that when the necklaces are ordered along the x-axis according to their distance in the complex plane from the origin, only $O(N/\lg^{1/2}N)$ horizontal tracks are needed to hold all SE edges, so the Hoey and Leiserson layout has $O(N^2/\lg^{3/2}N)$ area. Figure 3.2 shows the resulting layout for $n = 5$.

We may now use the homomorphism h to define an $O(N^2/\lg^{3/2}N)$ area layout for the even double-exchange family. There are at most $O(N/\lg N)$ addresses that g maps to the origin, so we can ignore all x such that $g(h^{-1}(x)) = \{0\}$. For all other x, we define Model(x) to be $y \in h^{-1}(x)$ such that $Im(g(y)) > 0$ or $Im(g(y)) = 0$ and

Figure 3.2



Layout of the Shuffle-Exchange, for n = 5

$Re(g(y)) > 0$. Because $g(x) = -g(\bar{x})$, Model(x) is single-valued. We now place each vertex x of the even double-exchange according to the position of Model(x) in the shuffle-exchange layout.

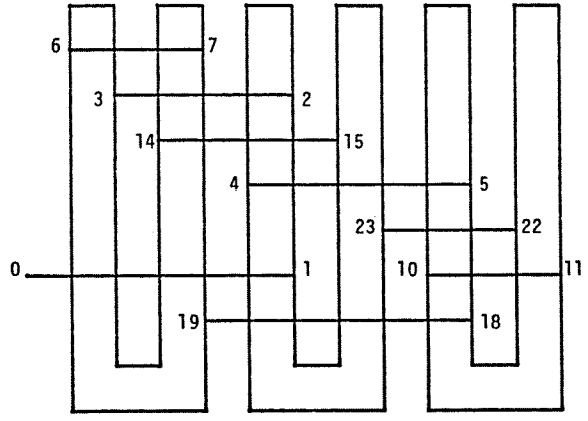
If Model(x) and Model(ρ(x)) lie in the same necklace of the shuffle-exchange, we connect x and ρ(x) with the same layout routing. In the complex plane representation of the shuffle-exchange, each necklace has only two ρ edges connecting vertices below the real line with vertices on or above it, so at most two ρ edges in each even double-exchange necklace will not be handled by using the corresponding shuffle-exchange layout routing. These O(N/lg N) edges can be ignored, as usual. Similarly, we can connect x with DE(x) by using the layout routing for the edge between Model(x) and Model(DE(x)); the only possible exceptions arise when SE(Model(x)) lies on the real axis and has a negative imaginary component. Each necklace in the shuffle-exchange has at most two vertices whose images under g are on the real axis, so, once again, there are at most O(N/lg N) DE edges in the even double-exchange that this layout does not handle. Thus, all but O(N/lg N) edges in the even double-exchange can be routed within area $O(N^2/\lg^{3/2}N)$ using the Model mapping and the shuffle-exchange routings. Figure 3.3 shows the even double-exchange layout

for n = 5 corresponding to the shuffle-exchange layout of Figure 3.2.

M

We conjecture that Kleitman's $O(N^2/\lg^2 N)$ layout of the shuffle-exchange may similarly be used to provide an optimal layout of the even double-exchange; in any case, ad hoc layouts for practical values of N can easily improve on these general constructions for the shuffle-exchange as well as the even double-exchange.

Figure 3.3



Layout of the Even Double-Exchange, for n = 5

3.3. Summary

The diameter of the even double-exchange is qualitatively superior to the shuffle-exchange and all other trivalent families previously proposed, and yet this family can be efficiently laid out and effectively used to emulate algorithms proposed for the shuffle-exchange. In the next chapter, we show that the even double-exchange is at least as reliable as the shuffle-exchange.

Chapter 4
Reliability

A multicomputer graph might be highly successful in terms of the cost measures considered in Chapter 2 yet still be impractical. Any pragmatic comparison of multicomputer graph families must measure their reliability: How seriously do failures affect them? Of the three major effects of failures -- disconnection, increased diameter, and increased traffic congestion -- this chapter concentrates on the most serious, disconnection. We are deliberately excluding the issue of fault diagnosis, so the measures presented here assume that failed elements are identified by some underlying protocol.

Knowing that a graph remains connected despite some number of failures, however, is not an adequate measure of reliability for a distributed computer. Even if the graph is still connected, we need to know if messages can be routed using local knowledge, or if their routing requires global knowledge of the remaining network. Global knowledge is a particularly onerous requirement for distributed computers because extensive exchanges of routing information increase the congestion on a network already suffering from failures. Local rerouting

methods for handling failures allow us, in turn, to establish lower bounds for a general reliability measure, the Average Random Failset size: If failures are randomly and uniformly distributed without replacement, how many can be tolerated on the average before the graph becomes disconnected?

4.1. Connectivity

In graph-theoretic analyses, a common measure of reliability [53,7,54,9] is connectivity: the least number of failures that can disconnect the network. For edge-connectivity, the failures are of edges; for vertex-connectivity (or simply 'connectivity'), the failures are of vertices. Vertex-connectivity is sometimes also called survivability [55,56] or invulnerability [57,54]. By elementary graph theory, the vertex-connectivity is bounded above by the edge-connectivity, which in turn is bounded above by the smallest degree of any vertex. Connectivity provides a worst-case bound for failures; most of the proposed families have their vertex-connectivity equal to their minimum degree, and therefore equal to their edge-connectivity.

4.1.1. Previous multicomputer graphs

4.1.1.1. Graph product families

The graph product families generally have high connectivity, in compensation for their poor performance in respect to degree and cost. Given two graphs G_1 and G_2 with vertex-connectivities c_1 and c_2 respectively, the connectivity of their Cartesian product $G_1 \times G_2$ is greater than or equal to $\min(c_1+c_2, c_1N_2, c_2N_1)$. When the

operand graphs' connectivities are as high as possible, with $c_1 = d_1$ and $c_2 = d_2$, the product's connectivity is bounded above by $c_1 + c_2$ (the degree of the product). This result shows that the connectivity and the edge-connectivity of the binary, the complete, and the mesh-connected hypercubes are exactly their degree.

4.1.1.2. Tree variations

Trees are, of course, 1-connected, while Friedman's joined trees of degree d are d-connected. Trivalent Multi-Tree-Structures are 3-connected; for higher degrees, Multi-Tree-Structure graphs may have a connectivity as low as three, depending on the details of their construction. Hypertrees always have a root vertex of degree 2, and so are no better than 2-connected at any degree. Goodman and Sequin prove that Hypertree 1 has an edge-connectivity of 2, by showing how to route around a single failed edge. More generally, all hypertrees are proven 2-connected by noting that there is always a route around any single failed vertex: If the vertex is the root, its two children are always connected by a hypercube-like edge; if the vertex is a leaf, all other vertices are always connected by the underlying binary tree; otherwise, each of the two children of the failed vertex always has a hypercube-like edge to a

vertex that was not a child of the failed vertex, from which all vertices outside the failed vertex's subtree can be reached (including the vertex connected to the other child, so its subtree is still accessible.)

4.1.1.3. Single-exchange graphs

Connectivity varies widely among single-exchange families. At degree three, the cube-connected-cycles have connectivity 3, while the shuffle-exchange graphs always have two vertices of degree 1, and so are only 1-connected. At degree four, the binary de Bruijn networks have two vertices of degree 2, so cannot be more than 2-connected; the rerouting method defined in a later section proves that they are, in fact, 2-connected. Saluja and Reddy [58] have modified the binary de Bruijn to create a 4-connected family with, however, each graph having two vertices of degree 5. In general, de Bruijn networks with base b and degree d = 2b are (d-2)-connected. The lens with any degree d = 2b is d-connected, provided n is at least 3, as is the multistage de Bruijn.

4.1.2. Connectivity of the Moebius graph

The Moebius graphs can be modified to make them 3-connected, while preserving their degree, diameter, interpolation, and routing properties. No graph with degree 3 can have higher connectivity, so, by this measure of reliability, these modifications make the Moebius family as reliable as possible without increasing the degree. Throughout this section, ρ denotes the twisted rotate operation, $\rho(v_0v_1\cdots v_{n-1}) = v_1v_2\cdots v_{n-1}\bar{v}_0$.

The minimum degree of vertices in the unmodified Moebius graphs is two, setting an upper bound on their connectivity. However, very few vertices in a given Moebius graph have so few edges.

Lemma 1.1. Given a Moebius graph with $N = 2^n$ vertices ($n > 1$), if n is even, all vertices have degree three; if n is odd, all vertices have degree 3 except the vertex with address $\lfloor N/3 \rfloor$ and the vertex with the complementary address, which have degree 2 and are connected to one another.

Proof. There are three operations that define edges in a Moebius: DE, ρ, and ρ^{-1}. Matching corresponding bits between x and DE(x) immediately yields the contradiction that $x_0 = \bar{x}_0$, so that DE(x) cannot equal x; similar arguments show that $\rho(x)$ and $\rho^{-1}(x)$ also cannot equal x, and that DE(x) is never equal to $\rho(x)$ or $\rho^{-1}(x)$. These results prove that every vertex in a Moebius graph has degree at least two. Suppose that $\rho^{-1}(x) = \rho(x)$.

Matching bits as before gives the equations $x_1 = \bar{x}_{n-1}$, $x_0 = \bar{x}_{n-2}$, and $\forall\ i < n-2$, $x_i = x_{i+2}$. If n is even, these equations are inconsistent; if n is odd, there are two solutions: $x = \lfloor N/3 \rfloor = (01)^*0$ and $x = (10)^*1$. These two addresses are connected to each other by ρ; all other vertices cannot have $\rho^{-1} = \rho$, and so are trivalent.

M

Because so few vertices in a Moebius graph have degree less than 3, it is easy to modify the Moebius construction to produce regular trivalent graphs that are 3-connected: If n is odd, delete the two connected vertices of degree two and create an edge connecting the two vertices that were adjacent to them. We call this operation eliding the degree-2 vertices. The Moebius routing algorithm still works for the elided Moebius, and the diameter bound is unharmed, since any path through the elided vertices is shortened. In particular, the two vertices that once neighbored on the elided vertices do not need to make any change in their message forwarding: A message that would have gone through one of the elided vertices must get there on a DE edge, and, if sent out on the prescribed DE edge reaches exactly the vertex that lies two steps farther along its calculated path.

Theorem 1. The elided Moebius graph is 3-connected.

For $n < 5$, exhaustive testing shows that the graph is 3-connected; for $n \geq 5$, we prove Theorem 1 by defining some sets of distinguished vertices such that despite any two vertex failures, there is a path from any nondistinguished vertex to some surviving distinguished vertex (Lemma 1.2), and a path between any two surviving distinguished vertices (Lemma 1.2).

Definitions. The proof employs three sets of distinguished vertices, Zero, One, and Altern:

$$\begin{aligned}
\text{Zero} &= \{0^n\} \\
\text{One} &= \{1^n\} \\
\text{Altern} &= \{(01)^{n/2}, (10)^{n/2}\} \text{ for } n \text{ even} \\
&\quad \{(01)^{(n-3)/2}001, (01)^{(n-1)/2}1, \\
&\qquad (10)^{(n-1)/2}0, (10)^{(n-3)/2}110\} \text{ for } n \text{ odd}
\end{aligned}$$

The names Zero, One, and Altern are used both for the sets and for their members. An address not in Zero, One, or Altern is called nondistinguished.

Let $x = x_0x_1 \ldots x_{n-1}$, and define InitialZero(x) to be the number of leading zeros in x. InitialZero(x) is thus the least i such that $x_i = 1$, or n if x is Zero. Similarly, InitialOne(x) denotes the number of leading ones in x, and InitialAltern(x) denotes the length of the longest initial string of alternating bits. The following elementary properties of these initial pattern

lengths are used in the proofs below:

a) InitialZero(x) > 0 if and only if InitialOne(x) = 0;

b) InitialAltern(x) > 1 if and only if InitialZero(x) $\leq$ 1 and InitialOne(x) $\leq$ 1;

c) If x and y differ in their values of InitialZero, or of InitialOne, or of InitialAltern, then $x \neq y$.

We occasionally use the analogous properties for the number of trailing zeros, ones, or alternating bits in an address x, denoted FinalZero(x), FinalOne(x), and FinalAltern(x) respectively.

Lemma 1.2. For any nondistinguished vertex x there are vertex-disjoint paths from x to Zero, One, and some vertex A $\in$ Altern.

Proof. Lemmas 1.2.1, 1.2.2, and 1.2.3 will establish that, from any nondistinguished x, there are paths $P_0(x)$, $P_1(x)$, and $P_A(x)$, to Zero, One, and some A in Altern, respectively, which monotonically increase in their vertices' values of InitialZero, InitialOne, and InitialAltern, respectively. Although these paths eventually acquire sufficiently long initial bit patterns to guarantee the disjointness of their remaining vertices, they do not suffice to prove Lemma 1.2; for example, if x = 111101, then both $P_0(x)$ and $P_1(x)$ include the vertex

111110.

Lemma 1.2.4 will circumvent this problem (and complete the proof) by specifying initial paths for each x, denoted $InnerPath_0(x)$, $InnerPath_1(x)$, and $InnerPath_A(x)$, to vertices $StartP_0(x)$, $StartP_1(x)$, and $StartP_A(x)$ whose values of InitialZero, InitialOne, and InitialAltern, respectively, are large enough to keep the outer paths $P_0(StartP_0(x))$, $P_1(StartP_1(x))$, and $P_A(StartP_A(x))$ disjoint from one another and from the inner paths. The inner paths will also be constrained to be vertex-disjoint; the outer paths will be disjoint and will contain no elided vertices because they will have, respectively, InitialZero, InitialOne, and InitialAltern greater than one.

Lemma 1.2.1. Given any nondistinguished vertex x, there is a path, $P_0(x)$, to Zero whose vertices increase monotonically in their InitialZero values.

Proof. There are two cases to consider, depending on the value of InitialZero(x). If InitialZero(x) is less than n-2, we can route to an address y such that InitialZero(y) = InitialZero(x) + 1: If x ends with a one, let y be $\rho^{-1}(x)$; otherwise, let y be $\rho^{-1}(DE(x))$. Since InitialZero(x) < n-2 the DE does not change InitialZero(x), while the ρ^{-1} increases InitialZero(x)

by exactly 1. Applying this construction a finite number of times must reach an address whose InitialZero value is at least n-2, along a path whose InitialZero values increase monotonically.

If InitialZero(x) is at least n-2, we can demonstrate a path from x to Zero whose vertices increase monotonically in their value of InitialZero. All bits to the left of x_{n-2} must be 0, so there are four cases to consider. If the rightmost two bits are both zero, x is Zero. If the rightmost two bits are both one, x's DE edge connects directly to Zero. If the rightmost two bits are 10, a DE changes them to 01, increasing the InitialZero value by one and reducing to the fourth case: If the rightmost two bits are 01, x's ρ^{-1} edge connects directly to Zero.

M

We note that if InitialZero(x) is greater than one, $P_0(x)$ can contain no elided vertices because InitialAltern will be 1 for the entire path.

Lemma 1.2.2. Given any nondistinguished vertex x, there is a path, $P_1(x)$, to One whose vertices increase monotonically in their InitialOne values.

Proof. The proof exactly parallels that of Lemma 1.2.1.

M

If InitialOne(x) is greater than one, $P_1(x)$ can contain no elided vertices.

Lemma 1.2.3. Given any nondistinguished vertex x, there is a path, $P_A(x)$, to some A ∈ Altern whose vertices increase monotonically in their InitialAltern values.

Proof. For InitialAltern(x) < n-2 or n even, the proof exactly parallels that of Lemma 1.2.1. When n is odd, the only nondistinguished vertices with InitialAltern ≥ n-2 are $(01)^*000$ and $(10)^*111$, from each of which a DE edge increases the value of InitialAltern by one and connects to a member of Altern.

M

Because InitialAltern increases by no more than one at each step, $P_A(x)$ could only reach an elided vertex by first passing through a vertex with InitialAltern = n-1; when n is odd, both such vertices are members of Altern, so $P_A(x)$ ends with them.

Lemma 1.2.4. Given any nondistinguished vertex x, there are vertices $StartP_0(x)$, $StartP_1(x)$, and $StartP_A(x)$, and paths $InnerPath_0(x)$, $InnerPath_1(x)$, and $InnerPath_A(x)$ reaching $StartP_0(x)$, $StartP_1(x)$, and $StartP_A(x)$ respectively, which satisfy three properties:

a) the addresses in the proposed paths do not represent elided vertices;

b) the inner paths are vertex-disjoint;

c) $InnerPath_0(x)$ does not intersect $P_1(StartP_1(x))$ or $P_A(StartP_A(x))$; $InnerPath_1(x)$ does not intersect $P_0(StartP_0(x))$ or $P_A(StartP_A(x))$; and $InnerPath_A(x)$ does not intersect $P_0(StartP_0(x))$ or $P_1(StartP_1(x))$.

Proof. Unfortunately, there are numerous cases, distinguished by the initial and final bit patterns of x, that must be proven separately. Only addresses beginning with 0 will be discussed; those beginning with 1 are handled by complementing all addresses. The detailed verification of the properties is very similar for each case, and so is presented for only one; correct inner paths for all cases are listed in Table 4.1.

Case InitialZero(x) > 1 and x ends in 10

$InnerPath_0 = DE$

$InnerPath_1 = \rho^{-1}\ DE\ \rho^{-1}$

$InnerPath_A = \rho\ DE\ \rho^{-2}$

The vertex x must have the form $00x_2...x_{n-3}10$, with at least one bit between the 00 and the 10. $InnerPath_0$ contains only $StartP_0 = 00x_2...01$. Neither it nor the vertices in $P_0(StartP_0)$ are elided because they have

$InitialZero \geq InitialZero(x) \geq 2$. $InnerPath_1$ contains $100x_2...1$, $100x_2...0$, and $StartP_1 = 1100x_2....$ These addresses have $InitialAltern \leq 2$, and so cannot have been elided; they have $InitialOne \geq 1$, and so are not in $InnerPath_0$ or $P_0(StartP_0)$. The vertices in $P_1(StartP_1)$ have $InitialOne \geq 2$ and therefore do not include any members of $InnerPath_0$.

$InnerPath_A$ contains $0x_2...101$, $0x_2...110$, $10x_2...11$, and $StartP_A = 010x_2...1$. Since x is not elided, $\rho(x)$ cannot be; the other vertices are not elided because they have $FinalAltern \leq 2$ when n is odd. The addresses in $InnerPath_A$ are distinct from those of $InnerPath_0$ and $P_0(StartP_0)$ because they have $InitialZero < InitialZero(x)$, and are distinct from those of $P_1(StartP_1)$ because they have $InitialOne < 2$. Only $10x_2...11$ could possibly be in $InnerPath_1$, but it begins with $10^{InitialZero(x)-1}1$, while each address in $InnerPath_1$ starts with either $10^{InitialZero(x)}1$ or 11. Finally, each address in $P_A(StartP_A)$ has $InitialAltern \geq 3$, and so cannot belong to $InnerPath_0$ or $InnerPath_1$.

Analogous arguments establish the three necessary conditions for the inner paths for all the other cases, completing the proofs of lemmas 1.2.4 and 1.1.

■

Table 4.1

InitialAltern(x) = n-1

If n is even, x must be $(01)^{(n-2)/2}00$.

$InnerPath_0 = DE\ \rho^{-2}$

$InnerPath_1 = \rho^2\ DE\ \rho^{-2}$

$InnerPath_A = \rho^{-1}$

If n is odd, x is distinguished.

InitialAltern(x) = n-2

If n is even

If $x = (01)^{(n-2)/2}01$,

$InnerPath_0 = \rho^2\ DE\ \rho^{-1}$

$InnerPath_1 = \rho^{-1}\ DE\ \rho^{-1}$

$InnerPath_A = DE$

If $x = (01)^{(n-2)/2}11$,

$InnerPath_0 = \rho^{-1}$

$InnerPath_1 = \rho\ DE\ \rho^{-1}$

$InnerPath_A = DE\ \rho^{-1}$

If n is odd

If $x = (01)^{(n-3)/2}000$,

$InnerPath_0 = DE\ \rho^{-1}$

$InnerPath_1 = \rho\ DE\ \rho^{-1}$

$InnerPath_A = \rho^{-1}$

If $x = (01)^{(n-3)/2}001$, it's distinguished.

InitialAltern(x) > 1 but < n-2

If x ends in 0,

$InnerPath_0 = DE\ \rho^{-1}$

$InnerPath_1 = \rho\ DE\ \rho^{-1}$

$InnerPath_A = \rho^{-1}$

If x ends in 1,

$InnerPath_0 = \rho^{-1}$

$InnerPath_1 = \rho\ DE\ \rho^{-1}$

Table 4.1 (continued)

InitialAltern(x) = 1 (so InitialZero(x) > 1)

If x ends in 00

If InitialZero(x) = n, x is distinguished

If InitialZero(x) < n-2

$InnerPath_0 = DE$

$InnerPath_1 = \rho^{-2}$

$InnerPath_A = \rho\ DE\ \rho^{-1}$

If x ends in 10

$InnerPath_0 = DE$

$InnerPath_1 = \rho^{-1}\ DE\ \rho^{-1}$

$InnerPath_A = \rho\ DE\ \rho^{-1}$

If x ends in 01

If InitialZero(x) = n-1

$InnerPath_0 = \rho^{-1}$

$InnerPath_1 = \rho^2\ DE\ \rho^{-2}$

$InnerPath_A = DE\ \rho^{-1}$

If InitialZero(x) < n-2

$InnerPath_0 = \rho^{-1}$

$InnerPath_1 = DE\ \rho^{-1}\ DE\ \rho^{-1}$

$InnerPath_A = \rho\ DE\ \rho^{-1}\ DE\ \rho^{-1}$

If x ends in 11

If InitialZero(x) = n-2

$InnerPath_0 = DE$

$InnerPath_1 = \rho\ DE\ \rho^{-2}$

$InnerPath_A = \rho^{-1}\ DE\ \rho^{-2}$

If InitialZero(x) < n-2

$InnerPath_0 = \rho^{-1}$

$InnerPath_1 = DE\ \rho^{-2}$

$InnerPath_A = \rho\ DE\ \rho^{-1}\ DE\ \rho^{-1}$

Lemma 1.3. Despite any two vertex failures, there is a path between any two surviving distinguished vertices.

Proof. The proof uses four sublemmas. We show that there are three vertex-disjoint paths between Zero and One (Lemma 1.3.1), three between any member of Altern and Zero (Lemma 1.3.2), and three between any member of Altern and One (Lemma 1.3.3). Finally, we show that despite any two vertex failures, there is a path between any two surviving members of Altern (Lemma 1.3.4).

Lemma 1.3.1. There are three vertex-disjoint paths between Zero and One that contain no elided vertices.

Proof. Starting from Zero, define paths P_L, P_R, and P_M:

$$P_L = \rho^{-(n-2)}\ DE$$
$$P_R = DE\ \rho^{n-2}$$
$$P_M = (\rho\ DE)^{n-1}\ \rho^{-1}$$

P_L introduces 1's from the left, P_R from the right, and P_M from the right, but with the rightmost two bits being either 01 or 10. Straightforward induction arguments exactly characterize all nonterminal vertices in these three paths: those in P_L have the form 11^*00^*, those in P_R have the form 00^*11^*, and those in P_M have the form of either 0^*1^*01 or 0^*1^*10. These characterizations prove that the sets of nonterminal vertices of P_L, P_R, and P_M are pairwise disjoint and, since n is greater

that 4, contain no elided vertices.

M

Lemma 1.3.2. There are three vertex-disjoint paths from any member of Altern to Zero.

Proof. The required paths are defined using two convenient shorter paths:

$$Rightshift(x) = \rho^2\ DE\ \rho\ DE\ \rho,$$
$$Leftshift(x) = DE\ \rho^{-1}\ DE\ \rho^{-3}.$$

Letting a denote either 0 or 1, then for any X, Rightshift(aāaāX) = Xaaaa and Leftshift(Xāaāa) = aaaaX. In the addresses along the paths defined by Leftshift(Xāaāa) and Rightshift(aāaāX), the substring X is shifted in position, but never changed. Leftshift(Xāaāa) also has the property that the number of initial a's increases monotonically: Xāaāa goes to Xāaaā, aXāaa, aXāāā, aaXāā, aaaXā, and aaaaX in turn.

For the remainder of this lemma, let q denote n div 4 and r denote n modulo 4. The particular paths between Altern and Zero depend on r, but share a common pattern: P_L shifts in zeros from the left, P_R shifts in zeroes from the right, and P_1 first shifts in ones until the address is almost entirely ones and then shifts in zeros.

Case r = 0: Altern has two members, $(0101)^q$ and $(1010)^q$.

Starting from $(0101)^q$,

$P_L = \rho^{-1}$ Leftshift^{q-1} DE ρ^{-1},

$P_R = \rho$ DE ρ Rightshift^{q-1},

$P_1 = $ DE ρ DE ρ^{-1} DE ρ^{-2} Leftshift^{q-1} $\rho^{-(n-2)}$ DE.

The initial ρ^{-1} of P_L transforms $(0101)^q$ to $0010(1010)^{q-1}$. From here, the value of InitialZero increases monotonically. The final DE converts the last address of the last Leftshift, $(0000)^{q-1}0010$, to $0^{n-1}1$, which the ρ^{-1} takes directly to Zero. None of these internal addresses can appear in P_R, because all addresses along that path have InitialAltern > 1: P_R first reaches $(1010)^{q-1}1011$, $(1010)^{q-1}1000$, and $(0101)^{q-1}0000$; the next q-2 Rightshifts produce only addresses starting with 01 or 10, ending with $0101(0000)^{q-1}$; the final Rightshift then produces $101(0000)^{q-1}1$, $01(0000)^{q-1}10$, $01(0000)^{q-1}01$, $1(0000)^{q-1}011$, 10^{n-1}, and Zero. Except for $0^{n-1}1$ in P_L, all addresses in P_L and P_R contain the pattern 10. This fact, together with their initial bit patterns, distinguishes them from most addresses in P_1.

The path P_1 begins with $(0101)^{q-1}0110$, $1(0101)^{q-1}101$, $1(0101)^{q-1}110$, and $11(0101)^{q-1}11$. InitialOne increases monotonically from here until, after

the Leftshifts, the path reaches One. All addresses in this portion of P_1 have InitialZero < 2, and so do not appear in P_L, while those with InitialOne > 1 must be distinct from all in P_R. The only addresses in this portion with InitialOne $\leq$ 1 are the first three, which clearly differ from the first two addresses in P_R, and are distinct from later addresses in P_R because the latter all contain a sequence of four or more zeros. After reaching One, P_1 employs ρ^{-1}s to produce addresses of the form 0^*1^*11. These addresses do not contain the pattern 10, and so are distinct from all addresses in P_L and P_R. The last ρ^{-1} in P_1 produces $0^{n-2}11$, from which the concluding DE immediately reaches Zero.

Thus all three proposed paths reach Zero without intersecting, and the first case in the proof of Lemma 1.3.2 is established. The paths for $(1010)^q$ and for the remaining cases are listed in Table 4.2; their proofs are so similar that they are omitted.

∎

Table 4.2

Case r = 0: Altern has two members

Starting from $(0101)^q$,

$P_L = \rho^{-1}\ \text{Leftshift}^{q-1}\ DE\ \rho^{-1}$

$P_R = \rho\ DE\ \rho\ \text{Rightshift}^{q-1}$

$P_1 = DE\ \rho\ DE\ \rho^{-1}\ DE\ \text{Leftshift}^{q-1}\ \rho^{-(n-2)}\ DE$

Starting from $(1010)^q$,

$P_L = \text{Leftshift}^{q-1}\ DE\ \rho^{-1}\ DE\ \rho^{-1}\ DE$

$P_R = \rho\ \text{Rightshift}^{q-1}\ \rho^2\ DE\ \rho$

$P_1 = \rho^{-1}\ \text{Leftshift}^{q-1}\ DE\ \rho^{n-1}$

Case r = 1: Altern has four members

Starting from $(0101)^q 1$

$P_L = \rho^{-2}\ \text{Leftshift}^{q-1}\ DE\ \rho^{-1}$

$P_R = \text{Rightshift}^q\ \rho$

$P_1 = DE\ \rho^{-1}\ \text{Rightshift}^q\ \rho^{-(n-3)}\ DE$

Starting from $(0101)^{q-1}01001$,

$P_L = \rho^{-4}\ \text{Leftshift}^{q-1}\ DE\ \rho^{-2}\ DE$

$P_R = \text{Rightshift}^q\ \rho\ DE\ \rho^{-1}$

$P_1 = DE\ \rho\ DE\ \text{Rightshift}^q\ \rho\ DE\ \rho^{n-1}$

Starting from $(1010)^q 0$,

$P_L = \rho^{-1}\ \text{Leftshift}^q\ \rho\ DE$

$P_R = \rho\ \text{Rightshift}^{q-1}\ \rho^2\ DE\ \rho^{-1}$

$P_1 = DE\ \text{Rightshift}^{q-1}\ \rho^2\ DE\ \rho^{n-1}$

Starting from $(1010)^{q-1}10110$,

$P_L = \rho^{-3}\ \text{Leftshift}^{q-1}\ (DE\ \rho^{-1})^2$

$P_R = \rho\ \text{Rightshift}^{q-1}\ \rho^2\ DE\ \rho\ DE$

Table 4.2 (continued)

$P_1 = DE\ \rho\ \text{Rightshift}^q\ \rho^{n-2}$

Case r = 2: Altern has two members

Starting from $(0101)^q 01$,

$P_L = \text{Leftshift}^q\ \rho^{-1}$

$P_R = \rho\ \text{Rightshift}^q\ \rho^2$

$P_1 = \rho^{-1}\ \text{Leftshift}^q\ \rho^{-n}\ DE$

Starting from $(1010)^q 10$,

$P_L = \text{Leftshift}^q\ DE\ \rho^{-1}$

$P_R = \rho\ \text{Rightshift}^q$

$P_1 = \rho^{-1}\ \text{Leftshift}^q\ \rho^{-(n-2)}\ DE$

Case r = 3: Altern has four members

Starting from $(0101)^q 001$,

$P_L = \rho^{-2}\ \text{Leftshift}^q\ DE\ \rho^{-3}$

$P_R = \text{Rightshift}^q\ \rho^2\ DE\ \rho$

$P_1 = DE\ \text{Rightshift}^q\ \rho^2\ DE\ \rho^{-(n-3)}\ DE$

Starting from $(0101)^q 011$,

$P_L = \rho^{-2}\ \text{Leftshift}^{q+1}$

$P_R = \text{Rightshift}^{q+1}\ DE$

$P_1 = DE\ \rho\ \text{Rightshift}^{q+1}\ \rho^{n-2}$

Starting from $(1010)^q 100$,

$P_L = \rho^{-1}\ \text{Leftshift}^q\ (DE\ \rho^{-1})^2$

$P_R = \rho\ \text{Rightshift}^q$

$P_1 = DE\ \text{Rightshift}^q\ \rho^{-(n-2)}\ DE$

Starting from $(1010)^q 110$,

$P_L = \rho^{-3}\ \text{Leftshift}^q\ \rho^{-1}$

$P_R = \rho\ \text{Rightshift}^q\ \rho$

$P_1 = DE\ \rho\ \text{Rightshift}^q\ \rho^{-(n-3)}\ DE$

Lemma 1.3.3. There are three vertex-disjoint paths between any member of Altern and One.

Proof. The complement of any member of Altern is also in Altern, so complementing all addresses used in the proof of Lemma 1.3.2 produces the required paths from any member of Altern to One.

∎

Lemma 1.3.4. Despite any two vertex failures, there is a path between any two distinct members, A and B, of Altern.

Proof. Lemmas 1.3.2 and 1.3.3 have established that there are three vertex-disjoint paths between A and Zero, between A and One, between B and Zero, and between B and One. These paths ensure that, provided at least one of Zero and One has not failed, no other two failures can disconnect A from B. Lemma 1.3.4 is proven, then, if there is at least one path between A and B that goes through neither Zero nor One. If n is even, A and B must be $(01)^*$ and $(10)^*$, and the requisite path is produced by applying ρ n times; each vertex on this path has either the form $(10)^*1(10)^*1$ or the form $(01)^*(10)^*$, which guarantees that none is Zero or One for $n > 2$.

If n is odd, Altern has four members; for brevity, we name them by their final two bits: $A_{00} = (10)^*0$, $A_{01} = (01)^*001$, $A_{10} = (10)^*110$, and $A_{11} = (01)^*1$. The elided Moebius has an edge directly connecting A_{01} with A_{10}. A_{11} connects to A_{00} through a single vertex: $\rho(A_{11}) = (10)^*111$, and $DE((10)^*111) = A_{00}$. Because n is greater than 3, the middle vertex on this path is neither Zero nor One. Finally, the path $DE\ \rho\ (DE\ \rho^2)^{(n-3)/2}$ connects A_{00} to A_{10} without going through Zero or One. Each vertex on this path has one of the three forms $(10)^*1(10)^*11$, $(01)^*(10)^*110$, or $(01)^*(10)^*1$. Because n is greater than 3, no vertex is Zero or One, and lemmas 1.3.4 and 1.3 are proven.

∎

Proof of Theorem 1.

We can now complete the proof of Theorem 1, that the elided Moebius is 3-connected, by combining Lemmas 1.2 and 1.3. Despite any two failed vertices, there is a path between any two vertices v and w, which runs from v to some distinguished vertex D_v, from D_v to some distinguished vertex D_w, and from D_w to w. (Of course, not all of these formal vertices need actually be distinct.)

∎

4.1.3. Connectivity of the shuffle-exchange

There are also several ways in which shuffle-exchange graphs can be modified to be 3-connected, without increasing their degree; the elided shuffle-exchange family presented here satisfies an additional reliability criterion, vertex reroutability, defined in the next section. Once self-loops and duplicate edges are eliminated, shuffle-exchange graphs have two vertices of degree 1, 0^* and 1^*, and, if n is even, two adjacent vertices of degree two, $(01)^*$ and $(10)^*$. To make the graphs 3-connected, we delete 0^* and 1^*, reducing the degree of 0^*1 and 1^* to two. We replace these vertices in turn by edges connecting 10^* to 0^*10 and 01^* to 1^*01. If n is even, we also elide $(01)^*$ and $(10)^*$, making an edge from $(01)^*00$ to $(10)^*11$. (Thus we reduce N by 4 when n is odd and by 6 when n is even.) For $n > 3$, these elisions reduce the diameter while preserving the degree and the routing algorithms; using the idea of distinguished vertices and monotonic paths again, a case-by-case argument proves that these graphs are 3-connected. Zero is now the set $\{0^*10\}$, and One is $\{1^*01\}$; if n is odd, Altern is $\{(01)^*0, (10)^*1\}$, while if n is even, Altern is $\{(01)^*00, (10)^*11\}$. As the proof so closely parallels the Moebius proof, its details are omitted.

4.1.4. Connectivity of the even double-exchange

The even double-exchange graphs discussed in chapter 3 are only 1-connected, because each contains the degree-1 address 0^*. The technique of deleting vertices of degree 1 and eliding vertices of degree 2 again produces a family that is both 3-connected and vertex reroutable. The elisions required, and the particular distinguished vertices used to prove 3-connectness, depend on n modulo 4.

When n modulo 4 is 0, we remove the two vertices of degree 1, 0^* and 1^*, and elide their neighbors, 0^*11 and 1^*00 (thereby connecting 0^*110 with 10^*1 and 1^*001 with 01^*0), as well as eliding the two degree 2 vertices, $(01)^*$ and $(10)^*$ (producing an edge from $(01)^*10$ to $(10)^*01$). As long as n is at least 8, this produces a 3-connected graph; we set Zero = 0^*110, One = 1^*001, and Altern = $(01)^*10$. When n modulo 4 is 2, the only the vertices with degree below three are 0^* and 1^*, so we remove them and elide their neighbors, again connecting 0^*110 with 10^*1 and 1^*001 with 01^*0. In this case, if n is at least 6, we can use Zero = 0^*110, One = 1^*001, and Altern = $\{(01)^*00, (10)^*11\}$. Finally, when n is odd and at least 5, we remove the single vertex of degree 1, 0^*, and elide its neighbor 0^*11, to connect 0^*110 with 10^*1. The only degree-2 vertices are 1^*0 and its neighbor

1^*01, which we elide to make an edge from 01^* to 1^*011. Here, Zero = 0^*110 and One = 1^*000; Altern is $(01)^*0$ when n modulo 4 is 1, and $(10)^*1$ otherwise.

As before, monotonic paths to Zero, One, and Altern are easily demonstrated, and a case-by-case argument then proves that each elided even double-exchange graph is 3-connected, for n at least 5. Furthermore, this elision does not increase the asymptotic space requirements of the layouts discussed in Chapter 3, because only a fixed number of vertices and edges are affected.

4.2. Local Rerouting

The 3-connectedness of the elided Moebius and shuffle-exchange families proves that they satisfy a popular measure of reliability for multicomputer graphs without, however, guaranteeing that even a single failure can be handled in an actual distributed system. The proofs employ very impractical message routes, several times longer than those of simple routing algorithms and with a severe bottleneck problem from the distinguished vertices. Instead of relying on distinguished vertices or universal knowledge of a failure, a realistic distributed routing algorithm should be able to reroute locally around an isolated failure. Our formal model for the cost of local rerouting turns out to be useful not only for providing a measure of the reliability of a distributed computer, but also for bounding the diameter costs of failures in multicomputer graphs and for determining an analytic lower bound on a general reliability measure, the Average Random Failset size.

4.2.1. Definitions

Given a graph G = {V,E} that is at least 2-connected, it is always possible to reroute around an isolated failure. The cost of doing such rerouting locally will be measured by the minimum number of vertices

4.2. Local Rerouting

The 3-connectedness of the elided Moebius and shuffle-exchange families proves that they satisfy a popular measure of reliability for multicomputer graphs without, however, guaranteeing that even a single failure can be handled in an actual distributed system. The proofs employ very impractical message routes, several times longer than those of simple routing algorithms and with a severe bottleneck problem from the distinguished vertices. Instead of relying on distinguished vertices or universal knowledge of a failure, a realistic distributed routing algorithm should be able to reroute locally around an isolated failure. Our formal model for the cost of local rerouting turns out to be useful not only for providing a measure of the reliability of a distributed computer, but also for bounding the diameter costs of failures in multicomputer graphs and for determining an analytic lower bound on a general reliability measure, the Average Random Failset size.

4.2.1. Definitions

Given a graph $G = \{V,E\}$ that is at least 2-connected, it is always possible to reroute around an isolated failure. The cost of doing such rerouting locally will be measured by the minimum number of vertices

that must be involved in routing around a failure, and by the increase in the length of a path caused by routing within such a minimum set. Formally, for an edge $e = (v,w) \in E$, RegionSize(e) is the number of vertices in the smallest cycle containing e. If all vertices within this cycle -- the rerouting region of e -- know that e has failed, no other vertex needs to change its routing algorithm: Any message being routed through e reaches, say, v normally, detours through the rerouting region to w, and then continues on its original route. Such a local rerouting increases the length of a path through e by at most RegionSize(e) - 2.

For a graph $G = \{V,E\}$, EdgeRegionSize(G) is the maximum over all edges e in E of RegionSize(e). An infinite family of graphs is edge reroutable if there is some fixed r such that EdgeRegionSize(G) $\leq$ r for every graph G in the family; EdgeRegionSize for the family is then the least such r.

The corresponding definition for the cost of locally rerouting around vertex-failures is slightly more complex. Given some vertex $v \in V$, RegionSize(v) is $1 + |V_R(v)|$, where $V_R(v)$ is a minimal set of vertices satisfying

a) $V_R(v)$ does not contain v;

b) given any pair of distinct neighbors of v, there is

a path between them using only vertices in $V_R(v)$. Any path through v may be detoured around v without the knowledge of vertices outside the rerouting region of v, $\{v\} \cup V_R(v)$. The RerouteCost(v) of locally rerouting around v is the maximum over all pairs of neighbors of v of the difference between the length of the shortest path in $V_R(v)$ connecting them and their true distance.

For a graph $G = \{V,E\}$, VertexRegionSize(G) is the maximum over all v in V of RegionSize(v), and RerouteCost(G) is the maximum over all v of RerouteCost(v). In analogy with edge reroutability, an infinite family of graphs is vertex reroutable if there is some fixed r such that VertexRegionSize(G) $\leq$ r for every graph G in the family; VertexRegionSize for the family is then the least such r, and RerouteCost for the family is the maximum of RerouteCost(G). A vertex reroutable family necessarily is edge reroutable and has a finite RerouteCost. Any vertex rerouting region must include all of the vertex's neighbors, so VertexRegionSize for a graph is always at least 1 + d.

To our knowledge, local reroutability is the only measure of its kind for comparing the suitability of infinite graph families for distributed computers. The importance of diameter has led to several extensions of connectivity to include diameter issues [59]. Hartman

and Rubin [60] have proposed the question of the diameter stability of graphs; Boesch [9] define the persistence of a graph as the smallest number of vertices whose removal increases the diameter. Perhaps the closest proposed criterion is given by Farley [61], who considers isolated failure immune graphs -- graphs that are not disconnected by any number of isolated (nonadjacent) failures. Unfortunately, the dense multicomputer graph families proposed are not immune to isolated failures; the families that are locally reroutable are, however, immune to failures separated by the diameter of the vertex rerouting region, which is typically small.

4.2.2. Reroutability of specific families

Although most members of the proposed multicomputer graph families are at least 2-connected, they vary considerably in their local rerouting costs -- both in the number of processors affected (EdgeRegionSize and VertexRegionSize) and in the increased path lengths forced by local rerouting (RerouteCost). Indeed, several of the proposed infinite families are not vertex or even edge reroutable, despite being highly connected.

4.2.2.1. The hypercube

Theorem 2. The binary, complete, and mesh-connected hypercube families are edge reroutable, with EdgeRegionSize $\leq$ 4, but are not vertex reroutable.

Proof. Each vertex of a graph in the graph product families can be assigned a (possibly mixed radix) address of n digits, where n is the number of graph operands, and the value of a particular digit denotes a vertex in the corresponding operand graph. An edge connects two addresses that differ in exactly one position, so each edge is in a 4-cycle whose edges respectively change a given digit, change any other digit, restore the given digit to its original value, and finally restore the other digit. A complete hypercube in which each operand graph has at least three vertices always has a 3-cycle for any edge (change a digit, change it to something else, restore it), and so has EdgeRegionSize = 3; other hypercubes have EdgeRegionSize = 4. Because the degrees of the members of each of these families increase without bound, the families are not vertex reroutable.

M

4.2.2.2. Tree variations

Theorem 3. The tree-based families of Friedman [22] are neither edge nor vertex reroutable.

Proof. In Friedman's construction, which joins d d-ary trees with L levels each by identifying their leaves, routing around a root or an edge adjacent to a root requires a path down through that tree, up to a root of another tree, and back again. Hence for a given graph, EdgeRegionSize = 4L, VertexRegionSize = 2dL-d+2, and RerouteCost = 4L-4.

M

Theorem 4. For each m, the family Hypertree m of Goodman and Sequin [62] is edge and vertex reroutable, with EdgeRegionSize 6, VertexRegionSize 12+3m, and RerouteCost 8.

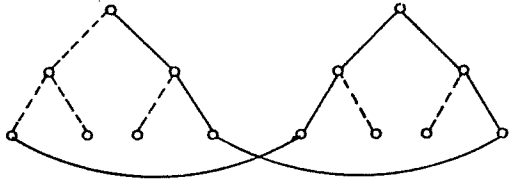
Proof. Goodman and Sequin present a general cycle for an edge in a Hypertree with four or more levels, which shows that the EdgeRegionSize is no more than 6. This cycle is shown in Figure 4.1a; it consists of two vertices joined by a hypercube-like edge, together with their parents and siblings. The distance up a tree from a level L to the nearest level whose hypercube-like edges complement the corresponding address bit is

L/(l+m), so their cycle is minimal: A cycle whose length is independent of L and which includes one hypercube-like edge from level L must use another hypercube-like edge at the same level. Similarly, to connect a vertex to its grandparent without going through its parent, any vertex rerouting region must include a cycle at least as long as the 10-vertex cycle shown in Figure 4.1b. The vertex rerouting region is produced by merging these cycles, as shown in Figure 4.1c.

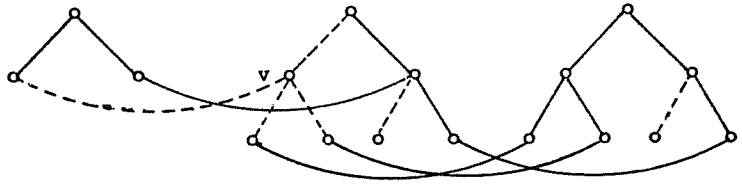
Figure 4.1

a) Edge rerouting region

b) Routing to the grandparent



c) Vertex rerouting region for v



Rerouting for the Hypertree

4.2.2.3. The cube-connected-cycles

Theorem 5. The cube-connected-cycles family (with n > 1) is edge and vertex reroutable, with EdgeRegionSize 8, VertexRegionSize 14, and RerouteCost 10.

Proof. For a cube-connected-cycles graph with n > 1, an edge is always part of an 8-cycle of the form ρ, SE, ρ^{-1}, SE, ρ, SE, ρ^{-1}, SE, where ρ is an untwisted rotate; letting X denote the substring $x_1x_2\ldots x_{n-2}$, the addresses in our single-exchange formulation of the family are $\{s,x_0Xx_{n-1}\}$, $\{s+1,Xx_{n-1}x_0\}$, $\{s+1,Xx_{n-1}\bar{x}_0\}$, $\{s,\bar{x}_0Xx_{n-1}\}$, $\{s,\bar{x}_0X\bar{x}_{n-1}\}$, $\{s+1,X\bar{x}_{n-1}\bar{x}_0\}$, $\{s+1,X\bar{x}_{n-1}x_0\}$, and $\{s,x_0X\bar{x}_{n-1}\}$. No two edges in this sequence are the same because no two addresses are identical: They disagree in either their stage components or corresponding bits of their binary string components.

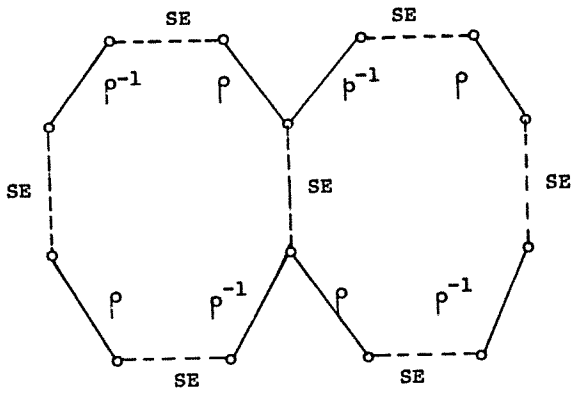
No shorter cycle, or collection of shorter cycles, will contain every SE edge. Any cycle must have one SE between a ρ and a ρ^{-1}, and, by parity, an even number of SE operations. A cycle of length less than n must have exactly as many ρ as ρ^{-1} edges, in order to return to the original stage; however, between any two successive SEs, it must have an unequal number of ρ and ρ^{-1} edges, to avoid loops in the path. The only two sequences (ignoring cyclic permutations) of seven or fewer operations

that satisfy these constraints for n greater than 6 are SE ρ SE ρ^{-1} and SE ρ^2 SE ρ^{-2}, neither of which defines a cycle. (For n less than 7, the constraints allow a few more sequences, none of which is a cycle.)

Combining two of these 8-cycles gives us the minimal vertex-rerouting set, with 14 vertices, as shown in Figure 4.2. All 14 addresses in this region are distinct, as before. Because the 8-cycles are minimal for rerouting around vertex v from $\rho^{-1}(v)$ or $\rho(v)$ to SE(v), the set is minimal and VertexRegionSize is 14. If this region is unique, then the cube-connected-cycles have a RerouteCost of 10. Uniqueness follows from the uniqueness of our 8-cycle for edge rerouting, which is established by enumerating all sequences of eight operations satisfying the cycle constraints. Aside from cyclic permutations, the only such sequences are our edge cycle, $(\text{SE}\ \rho)^2\ (\text{SE}\ \rho^{-1})^2$, SE ρ^3 SE ρ^{-3}, and a few additional sequences that only obey the constraints when n is less than 7. Of these, only our pattern is actually a cycle.

■

Figure 4.2



Vertex Rerouting for the Cube-Connected-Cycles

4.2.2.4. The shuffle-exchange

Theorem 6. The shuffle-exchange family is neither edge nor vertex reroutable.

Proof. Shuffle-exchange graphs are 1-connected.

∎

Theorem 7. The elided shuffle-exchange family is edge and vertex reroutable, with EdgeRegionSize 8, VertexRegionSize 14, and RerouteCost 10.

Proof. For $n > 4$, an edge is always part of an 8-cycle with the pattern of Theorem 5: ρ, SE, ρ^{-1}, SE, ρ, SE, ρ^{-1}, SE; again letting X denote $x_1x_2\ldots x_{n-2}$, the corresponding addresses are x_0Xx_{n-1}, $Xx_{n-1}x_0$, $Xx_{n-1}\bar{x}_0$, $\bar{x}_0Xx_{n-1}$, $\bar{x}_0X\bar{x}_{n-1}$, $X\bar{x}_{n-1}\bar{x}_0$, $X\bar{x}_{n-1}x_0$, and $x_0X\bar{x}_{n-1}$. Unlike Theorem 5, however, the addresses are not necessarily distinct and may represent elided vertices. Solving the possible equalities in the sequence shows that the only occasions where two nonadjacent addresses in the sequence can be equal are when the cycle contains two of the elided addresses, 0^* and 0^*1, or their complements. Fortunately, the new edges created by the elisions each form a subsequence of the canonical 8-cycle, and so belong to smaller cycles: the edges from 10^* to 0^*10 and from 01^* to 1^*01 each represents the sequence ρ, SE,

ρ^{-1}, SE, and ρ (placing them in 4-cycles that end with SE, ρ^{-1}, SE), while for even n, the edge from $(01)^*00$ to $(10)^*11$ represents SE, ρ, and SE, placing it in a 6-cycle. Conversely, any canonical 8-cycle that contains an elided address must have it inside a subsequence represented by one of the new edges. For all n greater than 4, there are edges which do not belong to cycles containing new edges, so their RegionSizes are actually 8. Minimality and uniqueness of the 8-cycle are proved by another enumeration of the possible short sequences; given these conditions, the vertex rerouting region argument follows the line of Theorem 5.

∎

4.2.2.5. The lens

Theorem 8. A lens family of base b and degree 2b is edge and vertex reroutable, with EdgeRegionSize 4, VertexRegionSize 6+2b, and RerouteCost 4.

Proof. In the lens, $\{s,x\}$ is connected to $\{s+1,SE_i(\rho(x))\}$ for all $0 \le i < b$, where ρ is the untwisted rotate. An edge always lies in a cycle of the form $\{s,x_0X\}$, $\{s+1,Xi+x_0\}$, $\{s,\bar{x}_0X\}$, and $\{s+1,Xi+\bar{x}_0\}$, where X is the substring $x_1...x_{n-1}$ and $\bar{x}_0$ is any digit not equal to x_0. These addresses are necessarily distinct, so the formal sequence does define an actual

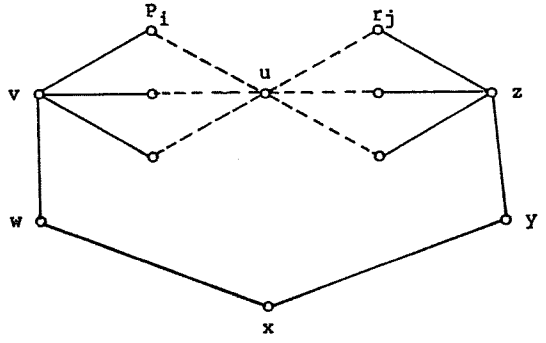
cycle of length four. For n greater than 1, the edge $(\{0,0^*\}, \{1,0^*1\})$ is never in a 3-cycle, so EdgeRegionSize is 4.

For n at least 6, a vertex $u = \{s,u_0u_1...u_{n-1}\}$ has a rerouting region with 6+2b vertices. Letting $\bar{u}_0$ and $\bar{u}_{n-1}$ represent fixed b-ary digits not equal to u_0 and u_{n-1} respectively, and U denote the substring $u_1u_2...u_{n-2}$, the region in general contains u, an outer path of 5 addresses ($v = \{s,u_0U\bar{u}_{n-1}\}$, $w = \{s+1,U\bar{u}_{n-1}\bar{u}_{n-1}\}$, $x = \{s,\bar{u}_0U\bar{u}_{n-1}\}$, $y = \{s-1,\bar{u}_0\bar{u}_0U\}$, and $z = \{s,\bar{u}_0Uu_{n-1})\}$, b addresses adjacent to both u and v ($p_i = \{s-1,iu_0U\}$ ∀ $0 \le i < b$), and b addresses adjacent to both u and z ($r_i = \{s+1,Uu_{n-1}i\}$ ∀ $0 \le i < b$). This set contains all neighbors of u, and, since no address in the outer path can equal u, it always provides a path from one neighbor to another without going through u. Figure 4.3 illustrates the region for a lens of degree 6.

This set is minimal, assuming that n is at least 6. First, we observe that the outer path is a shortest path between v and z. Any path between v and z must have either at least n edges or an even number of edges, because v and z are in the same stage. The current outer path has length 4, so any shorter path between them must have length 2; enumerating the neighbors of v and z

shows that they have none in common. (There are b^2 paths between v and z of length 4, all of which go through x; the particular values of w and y were chosen so that the corresponding outer path can also be used for the de Bruijn proof below.) By a similar argument, the unique shortest path from any p_i to any r_j goes through v and z. Finally, enumerating the possible neighbors of the p_i shows that a rerouting region for u that does not contain an address of the form v would have at least 2+5b vertices just to provide a path between any two of the p_i; similarly, a region without z could not have fewer than 2+5b vertices. So a minimal vertex set for rerouting around u must contain v and z; to provide a path from the set of p_i to the set of r_i, it must either include the outer path (connecting v with z) or a path of length min(6,n-2) from some p_i to some r_j. Hence a lens with base b and n at least 6 has VertexRegionSize = 6+2b and RerouteCost = 4. (For n less than 6, there is a shorter path between v and z that does not go through u: v connects to $\{s-1, \rho^{-1}(u)\}$, while z connects to $\{s+1, \rho(x)\}$. $\{s+1, \rho(x)\}$ and $\{s-1, \rho^{-1}(u)\}$ are neighbors of u, and are connected by n-2 ρ edges, defining a vertex rerouting region of 2b+n vertices.)

Figure 4.3



Vertex Rerouting for the Lens

4.2.2.6. The de Bruijn and C'_S graphs

Corollary 8.1. A multistage de Bruijn of base b, degree d = 2b, and S at least 3 is edge and vertex reroutable, with EdgeRegionSize 4, VertexRegionSize 6+d, and RerouteCost 4.

Proof. Because S is at least 3, all the formal addresses used in the proof of Theorem 8 are necessarily distinct, so the proof carries over directly.

M

Corollary 8.2. A C'_S graph of degree d = 2m+1, base b = m(m+1), and S at least 4 is edge and vertex reroutable, with EdgeRegionSize 4, VertexRegionSize 6+d, and RerouteCost 4.

Proof. The edge patterns of Theorem 8 can be used directly; the only change in the vertex rerouting pattern required is that there are either m left neighbors P_i and m+1 right neighbors, or vice versa, depending on whether the initial stage is odd or even.

M

Theorem 9. A de Bruijn family of base b and degree 2b is edge and vertex reroutable, with EdgeRegionSize 4, VertexRegionSize 6+2b, and RerouteCost 4.

Proof. The pattern of operations that worked for the lens works for the de Bruijn: every edge lies in a cycle of the form x_0X, $Xi+x_0$, $\bar{x}_0X$, and $Xi+\bar{x}_0$. Unlike the cycle in the lens, these four addresses are not necessarily distinct; however an actual sequence of four such addresses can have at most two identical, and those two must be adjacent in the formal sequence: x cannot equal $\bar{x}_0X$ and $Xi+x_0$ cannot equal $Xi+\bar{x}_0$; if $x = Xi+x_0$ then $x = a^n$ for some b-ary digit a, so $\bar{x}_0X = \bar{a}a^{n-1}$ while $Xi+\bar{x}_0 = a^{n-1}\bar{a}$; an analogous situation holds if $\bar{x}_0X = Xi+\bar{x}_0$; finally, $Xi+x_0$ cannot equal $\bar{x}_0X$, and x cannot equal $Xi+\bar{x}_0$. Hence the cycle of four formally distinct addresses must represent an actual cycle of either three or four distinct vertices. For $n > 2$, not all vertices in a de Bruijn are in 3-cycles so this edge rerouting region is minimal.

For $n > 5$, the minimal vertex rerouting region in a de Bruijn with base b has 6+2b formally distinct addresses, which are exactly analogous with those of the lens. The proofs that this vertex set provides paths between all neighbors of a vertex, and that no smaller set suffices for all addresses when n is greater than 5, are also analogous with those of Theorem 8, and are omitted.

M

Theorem 10. Saluja and Reddy's 4-connected modification of the binary de Bruijn family is edge reroutable with EdgeRegionSize 4, but not vertex reroutable.

Proof. For given n, let Altern be $\{(01)^*, (10)^*\}$ for n even, and $\{(01)^*0, (10)^*1\}$ for n odd. Saluja and Reddy [58] make the binary de Bruijn network 4-connected by connecting each of 0^* and 1^* to both members of Altern. These new edges form a 4-cycle, and all other edges lie in 3-cycles or 4-cycles by Theorem 9, so the new family is edge reroutable. Consider, however, the vertex rerouting region for 0^*. The region must contain a path from the vertices 0^*1 and 10^* to the vertices in Altern that does not go through 0^*. If this path uses any of the new edges, it must go through 1^*.

Let P be a shortest path from any vertex in $\{0^*1, 10^*\}$ to any vertex in Altern $\bigcup \{1^*\}$ that does not go through 0^*. P must contain only edges in the unmodified de Bruijn; each such edge increases the number of ones in an address by at most one; therefore, P must contain at least $\lfloor n/2 \rfloor - 1$ edges to go from one of its initial addresses to any member of its destination addresses. Therefore, the vertex rerouting region for 0^* must contain more than n/2 vertices. (More detailed arguments show that the actual VertexRegionSize for one of these modified graphs is n+2 for n greater than 3).

M

4.2.2.7. The elided Moebius

Theorem 11. The elided Moebius family is edge and vertex reroutable, with EdgeRegionSize 8, VertexRegionSize 14, and RerouteCost 10.

Proof. A sequence of address operations analogous to that for theorems 5 and 7 defines an 8-cycle for any particular edge in the Moebius graph: ρ, DE, ρ^{-1}, DE, ρ, DE, ρ^{-1}, DE; here ρ is the twisted rotate. Letting X represent the substring $x_1x_2\ldots x_{n-3}$, the addresses produced are $x = x_0Xx_{n-2}x_{n-1}$, $Xx_{n-2}x_{n-1}\bar{x}_0$, $Xx_{n-2}\bar{x}_{n-1}x_0$, $\bar{x}_0Xx_{n-2}\bar{x}_{n-1}$, $\bar{x}_0X\bar{x}_{n-2}x_{n-1}$, $X\bar{x}_{n-2}x_{n-1}x_0$, $X\bar{x}_{n-2}\bar{x}_{n-1}\bar{x}_0$, and $x_0X\bar{x}_{n-2}\bar{x}_{n-1}$. Unlike Theorem 7, all these addresses are distinct: Because the twisted rotate changes a string's parity, the addresses fall into two sets of four having opposite parity, $\{x, \bar{x}_0Xx_{n-2}\bar{x}_{n-1}, \bar{x}_0X\bar{x}_{n-2}x_{n-1}, x_0X\bar{x}_{n-2}\bar{x}_{n-1}\}$ and $\{Xx_{n-2}x_{n-1}\bar{x}_0, Xx_{n-2}\bar{x}_{n-1}x_0, X\bar{x}_{n-2}x_{n-1}x_0, X\bar{x}_{n-2}\bar{x}_{n-1}\bar{x}_0\}$. Within the first set, the addresses are distinguished by their first and last bits; within the second, they are distinguished by their last two bits.

If n is odd, addresses in this cycle may represent elided vertices. An unelided address, however, could only be transformed to an elided one by a DE operation, since the two elided addresses are transformed into each

other by the ρ and ρ^{-1} operations. Thus the new edge created by elision represents both the sequence DE ρ DE and the sequence DE ρ^{-1} DE; any formal cycle containing elided addresses then has a corresponding actual cycle in which one of these subsequences is replaced by a single edge. For n less than 6, there are shorter cycles for all edges, giving EdgeRegionSizes of 3, 4, 6, and 7, respectively, for elided Moebius graphs with n = 2, 3, 4, and 5. For n at least 6, enumeration of the possible sequences of operations again shows that no other cycle or set of cycles with length less than 9 contains all edges; the description and proof of the vertex rerouting region therefore follow Theorem 5.

M

4.2.2.8. The elided even double-exchange

Theorem 12. The elided even double-exchange family is edge and vertex reroutable, with EdgeRegionSize 8, VertexRegionSize 14, and RerouteCost 10.

Proof. The canonical 8-cycle, ρ, DE, ρ^{-1}, DE, ρ, DE, ρ^{-1}, DE again provides the edge rerouting region for n at least 6; here ρ is untwisted rotation. The addresses produced are $x = x_0Xx_{n-2}x_{n-1}$, $Xx_{n-2}x_{n-1}x_0$, $Xx_{n-2}\bar{x}_{n-1}\bar{x}_0$, $\bar{x}_0Xx_{n-2}\bar{x}_{n-1}$, $\bar{x}_0X\bar{x}_{n-2}x_{n-1}$, $X\bar{x}_{n-2}x_{n-1}\bar{x}_0$, $X\bar{x}_{n-2}\bar{x}_{n-1}x_0$, and $x_0X\bar{x}_{n-2}\bar{x}_{n-1}$. Unlike Theorem 11, the ρ operation does

not alter parity, and not all these addresses are necessarily distinct in the nonelided version of the family. However, the elisions remove precisely those cases where there are duplicated addresses: any address that returned to itself after a sequence of one or two operations would have degree less than 3, and be elided; the only remaining possible looping subsequences are DE ρ DE and DE ρ^{-1} DE, both of which imply that any starting address that returns to itself has a DE edge to 0^* or 1^*, and so is elided. Conversely, each of the new edges added by elision represents a subsequence in this cycle, and so belongs to a cycle of length less than 8.

There are no elided even double-exchange graphs for n less than 5. When n is 5, EdgeRegionSize is 7; for larger values of n, minimality and uniqueness of the 8-cycle is established by enumerating the possible operation sequences, so the vertex rerouting arguments of Theorem 5 may be applied again.

M

4.2.2.9. Summary

Table 4.3 summarizes the RegionSizes and RerouteCosts for local reroutability in the proposed multicomputer graph families. In particular, the table shows that the elided even double-exchange and Moebius

families are more reliable than the densest previous trivalent family, the shuffle-exchange, and as reliable as the even less dense cube-connected-cycles.

Table 4.3

Region Sizes and Reroute Costs for Graph Families

	Edge Region Size	Vertex Region Size	Reroute Cost
Graph product families	4	-	-
Tree families			
trees	-	-	-
Friedman's trees	-	-	-
Hypertree m	6	12+3m	8
Single-exchange families			
cube-connected-cycles	8	14	10
shuffle-exchange	-	-	-
elided shuffle-exchange	8	14	10
lens, degree d = 2b	4	6+d	4
multistage de Bruijn	4	6+d	4
$C_S^{'}$	4	6+d	4
de Bruijn	4	6+d	4
Saluja and Reddy	4	-	-
Double-exchange families			
elided Moebius	8	14	10
elided even double-exchange	8	14	10

4.3. Average Random Failsets

Under the worst possible set of simultaneous failures, a multicomputer graph of connectivity c can survive c-1 vertex failures, but not c failures. This worst case, however, is unlikely: An elided Moebius graph is disconnected by three failures only if they are the three neighbors of some one vertex, or if, among other constraints, they lie within distance 1 of a 3-cycle. Regardless of N, an elided Moebius graph never has more than four 3-cycles, so if the three failures are uniformly randomly distributed the probability of their disconnecting the graph is $6/(N-1)(N-2) + O(N^{-3})$. On the other extreme, the local rerouting properties of Moebius graphs show that in the best possible arrangement of failures, where all failures are at least distance 4 apart, there is no limit to the number of vertex or edge failures that can be sustained (see Lemma 13.1 below). Neither extreme in the distribution of failures is likely; a more representative measure of reliability would be the expected number of random failures required to disconnect a graph. This number is called the Average Random Failset size (ARF size) of the graph.

4.3.1. Definitions and basic results

What, exactly, does "the expected number of failures required to disconnect a graph" mean? In analogy with connectivity, the failures may be either vertices or edges; an edge-ARF concerns failed edges while a vertex-ARF concerns vertex failures. Our model assumes that the failures are uniformly randomly distributed among the nonfailed elements (vertices or edges), without replacement. A graph's ARF size is then $\geq ip(i)$, where $0 < i$ and $p(i)$ is the probability that exactly i failures are required to disconnect the graph (equivalently, that the i-th random failure disconnects the graph, while i-1 failures do not). Since a failure is held to disconnect a graph if there are two vertices in the resulting graph with no path between them, or if the resulting graph has less than two vertices (the trivial graph), the summation only goes through $i = N-1$ for vertex-ARFs and $i = |E|-(N-2)$ for edge-ARFs.

The great complexity of the probabilities $p(i)$ for nontrivial graph families has led us to three alternative approaches to measuring their ARF sizes. For a few very simple families, we can derive closed-form approximations for the sum. For more complex families, we can determine the ARF sizes for graphs with no more than a few thousand vertices by simulation; the close

correspondence between the results of the approximations and the simulations for the simple graph families provides some confidence in our implementation. A more general approach will be presented in the last section, where we derive a closed formula that is a lower bound for the ARF size of any reroutable family, in terms of its edge or vertex RegionSize.

The ARF size formula can be rewritten as $\geq q(i)$, where q(i) is the probability that at least i failures are required to disconnect the graph (equivalently, that i-1 failures do not disconnect it). For sufficiently simple families of graphs, the q(i) may be derived immediately. A line of N processors is disconnected if any edge fails, so its edge-ARF size is exactly 1. A single vertex-failure does not, however, disconnect a line if N is greater than 2 and the failed vertex is at one end of the line. For the vertex-ARF, therefore, q(1) is 1 and q(i) is q(i-1)*2/(N+2-i). The vertex-ARF size is then $\geq 2^{i-1}(N+1-i)!/N!$, or 1+2/N+O(N-2). Similarly, a ring of processors, C_N, has an edge-ARF size of 2 and a vertex-ARF size of 2+2/(N-1)+O(N-2). All trees have an edge-ARF size of 1; for an m-ary tree the vertex-ARF size is nearly m. A vertex failure disconnects the tree if and only if the vertex is not a leaf. An m-ary tree with L levels below the root has m^L leaves

out of $m^{L+1}-1/(m-1)$ vertices; when the number of vertices is large, (m-1)/m of them are leaves, and $q(i) = ((m-1)/m)^{i-1}$, so the ARF size is approximately m.

These simple results are useful because they validate the results of our simulations, which can then be used to determine the ARF sizes for graphs with more complex structure. Table 4.4 lists the graphs simulated and the ARF size found for 1000 iterations on each graph. Since each mean represents 1000 regeneration cycles, the confidence interval listed is calculated on the assumption that the means are normally distributed random variables. We have an exact summation formula for the vertex-ARF size of a line of processors, so the true ARF sizes are given in parentheses after the simulation results for lines; of the fifteen values of N tested, all of the true answers lie within the 95% confidence interval of the corresponding simulation.

Table 4.4

ARF Simulation Results

N	simulation value	exact value	confidence interval
Degree 2			
lines			
2	1.000	(1.000)	0.00
3	1.665	(1.667)	0.029
4	1.836	(1.833)	0.055
5	1.755	(1.733)	0.067
6	1.540	(1.578)	0.059
7	1.450	(1.451)	0.057
8	1.335	(1.363)	0.045
9	1.275	(1.303)	0.037
10	1.236	(1.261)	0.035
12	1.238	(1.205)	0.037
15	1.169	(1.156)	0.027
20	1.098	(1.112)	0.021
40	1.048	(1.053)	0.014
100	1.016	(1.020)	0.008

Table 4.4 (continued)

N	simulation value	confidence interval
Degree 3		
trees		
7	2.031	0.080
15	2.037	0.083
31	1.994	0.082
63	2.057	0.091
127	1.976	0.084
255	1.968	0.084
511	2.069	0.092
1023	1.998	0.088
2047	1.994	0.084
4095	1.978	0.095
cube-connected-cycles		
8	2.43	0.05
24	6.12	0.09
64	13.82	0.24
160	26.61	0.54
384	47.20	0.96
896	82.63	1.77
2048	142.42	3.01

Table 4.4 (continued)

N	simulation value	confidence interval
Degree 3		
elided shuffle-exchange		
4	1.86	0.06
8	2.61	0.10
16	3.61	0.12
32	6.71	0.19
64	10.18	0.30
128	18.24	0.54
256	29.05	0.85
512	49.41	1.36
1024	79.24	2.07
2048	132.36	3.42
4096	210.71	5.35
elided even double-exchange		
2	1.00	0.00
4	2.33	0.05
8	3.54	0.09
16	4.52	0.13
32	7.43	0.16
64	11.79	0.30
128	21.37	0.44
256	31.96	0.78
512	55.26	1.20
1024	83.17	2.10
2048	141.31	3.09
elided Moebius		
4	3.00	0.00
8	3.89	0.09
16	5.57	0.09
32	8.30	0.16
64	13.61	0.25
128	21.46	0.44
256	34.57	0.71
512	54.98	1.22
1024	88.75	2.02
2048	142.68	3.12
4096	225.86	5.18

Table 4.4 (continued)

N	simulation value	confidence interval
Degree 4		
lens		
8	4.933	0.086
24	10.714	0.151
64	22.10	0.34
160	43.27	0.73
384	81.60	1.43
896	153.26	2.51
2048	278.83	15.15
de Bruijn		
4	2.84	0.02
8	4.74	0.10
16	6.98	0.15
32	11.42	0.21
64	19.84	0.36
128	33.79	0.65
256	57.56	1.10
512	97.43	1.79
1024	163.72	3.06
2048	278.81	5.13
tree		
13	2.850	0.124
40	2.933	0.140
121	2.941	0.150
364	2.995	0.151
1093	3.112	0.156
3280	3.057	0.161
hypertree 1		
15	4.944	0.138
31	6.620	0.184
63	9.573	0.28
127	13.51	0.39
255	18.45	0.57
511	26.64	0.81
1023	36.84	1.14

Table 4.4 (continued)

N	simulation value	confidence interval
Degree 5		
tree		
21	3.792	0.171
85	3.823	0.195
341	3.996	0.20
1365	3.942	0.21
5461	4.101	0.22
hypertree 2		
15	6.809	0.143
31	10.632	0.20
63	17.27	0.35
127	27.76	0.58
255	43.63	0.95
511	69.96	1.56
Unbounded Degree		
binary hypercube		
2	1.00	0.00
4	2.67	0.03
8	4.93	0.08
16	8.86	0.13
32	16.98	0.21
64	33.71	0.39
128	66.79	0.71
256	132.43	1.23
512	263.77	2.26
1024	520.62	4.11

4.3.2. An analytic lower bound

The simulation results are instructive for moderately large graphs, but fail to supply provable bounds on the ARF sizes for the interesting multicomputer graph families. Fortunately, a general formula for a lower bound of the ARF sizes for most multicomputer graph families can be derived from the ability to reroute around a single failure using purely local knowledge.

Theorem 13. A graph with VertexRegionSize R has a vertex-ARF size of at least $(\pi N/2R)^{1/2} - 1/3$.

We will derive this lower bound by defining a restricted class of sequences of failures which do not disconnect their graph (Lemma 13.1) and whose probabilities are easily bounded.

Lemma 13.1. Given a 2-connected graph G, and vertices $v_1, \ldots, v_m$ such that v_i is not in $V_R(v_j)$ $\forall$ $j < i$, $G-\{v_1,\ldots,v_m\}$ is connected.

Proof. By induction on m: If m is 1, $G-\{v_1\}$ is connected by the definition of 2-connectedness. Assume that the graph $G-\{v_1,\ldots,v_{m-1}\}$ is connected, so, in particular, for any two vertices u and w $\in V-\{v_1,\ldots,v_m\}$ there is a path P_w in G, $u = w_1 \ldots w_p = w$, such that $\forall$

$1 \leq i \leq p$, $w_i \in V-\{v_1,\ldots,v_{m-1}\}$. $G-\{v_1,\ldots,v_m\}$ is connected if there is a path between u and w all of whose vertices are in $V-\{v_1,\ldots,v_m\}$. If v_m does not appear in P_w, P_w is the required path between u and w. Otherwise, suppose that $w_i = v_m$; w_{i-1} and w_{i+1} cannot be in $V-\{v_1,\ldots,v_m\}$ because v_m cannot be a neighbor of any v_j for $j < m$. There is a path P_x in G, $w_{i-1} = x_1 \ldots x_q = w_{i+1}$, all of whose vertices lie in $V_R(v_m)$. If this path does not contain any vertices in $\{v_1, \ldots, v_{m-1}\}$, we can produce the required path between u and w by using P_x to route between w_{i-1} to w_{i+1} in P_w. If P_x does contain some forbidden vertices, we remove them from the path one at a time: Let x_r in P_x be the vertex v_j that has the highest subscript j in the sequence $v_1, \ldots, v_m$. There is a path P_y in G, $x_{r-1} = y_1 \ldots y_s = x_{r+1}$ that lies entirely in $V_R(x_r)$, and therefore lies in $G-\{v_1,\ldots,v_{j-1}\}$. We modify P_x by replacing $x_{r-1}\, x_r\, x_{r+1}$ with P_y and eliding any resulting loops; if the resulting path has any forbidden vertices, their indices in $\{v_1, \ldots v_m\}$ must be less than j. By repeating this operation at most j-1 times on the successively modified paths P_x, we produce a path P_x in $G-\{v_1,\ldots,v_m\}$ that takes w_{i-1} to w_{i+1}. The original path P_w can now be modified to avoid v_m (and all other forbidden vertices) by replacing $w_{i-1}\, v_m\, w_{i+1}$ with the final P_x and eliding

any resulting loops. Therefore there is a path between any vertices in $V-\{v_1,\ldots,v_m\}$ that lies entirely in $V-\{v_1,\ldots,v_m\}$, so $G-\{v_1,\ldots,v_m\}$ is connected, completing the proof of Lemma 13.1.

∎

Proof of Theorem 13.

The vertex-ARF size formula can be written as $\Sigma q(m)$, where $1 \leq m < N$, and $q(m)$ is the probability that m-1 vertex failures do not disconnect the graph. A lower bound for each $q(m)$ therefore provides a lower bound for the vertex-ARF size. The graph G is connected, so $q(1) = 1$. If the m-1 vertex failures, $v_1, \ldots, v_{m-1}$, satisfy the condition of Lemma 13.1, the graph is connected. Enumerating such sets, we see that v_i can be any one of the $N - |\bigcup V_R(v_j)| \leq N - (i-1)R$ vertices not belonging to the rerouting regions of vertices v_j for $j < i$. Letting $\bar{N}$ denote N/R, then for i greater than 1

$$q(i) \geq q(i-1)\ (N-(i-1)R)/N \geq q(i-1)\ (\bar{N}-(i-1))/\bar{N}.$$

Solving this last recurrence shows that

$$q(i) \geq \bar{N}!/(\bar{N}^i(\bar{N}-i)!),$$

so the vertex-ARF size is greater than or equal to

$$\sum_{1\leq i\leq \bar{N}} \bar{N}!/(\bar{N}^i(\bar{N}-i)!).$$

This formula can be expressed in terms of the function

denoted Q(m) in Knuth [63, p. 112],

$$Q(m) = \sum_{1 \le i \le m} m!/(m-i)!m^i,$$

showing that the vertex-ARF size for a graph with VertexRegionSize R is greater than $Q(\bar{N})$. Q(m), however, has a well-known series expansion [63, p. 117]:

$$Q(m) = (\pi m/2)^{1/2} - 1/3 + O(m^{-1/2}),$$

where the $O(m^{-1/2})$ term is always positive. Therefore the vertex-ARF size for a graph with VertexRegionSize R is at least $(\pi N/2R)^{1/2} - 1/3$, completing the proof of Theorem 13.

∎

Theorem 14. A graph with EdgeRegionSize R has an edge-ARF size of at least $(\pi N/2R)^{1/2} - 1/3$.

Proof. The proof exactly parallels the proof for Theorem 13.

∎

The region sizes listed in Table 4.3 can now be used to derive lower bounds for the ARF sizes of all the reroutable families studied; Table 4.5 compares these bounds with the ARF sizes derived by simulation for moderately large graphs in these families.

Table 4.5

Comparison of ARF Simulation and Analytic Bound

N	simulation result	analytic bound
Degree 3		
cube-connected-cycles		
160	26.61	3.91
384	48.20	6.23
896	82.63	9.70
2048	142.42	14.83
elided shuffle-exchange		
256	29.05	5.03
512	49.41	7.25
1024	79.24	10.39
2048	132.36	14.83
4096	210.71	21.11
elided even double-exchange		
256	31.96	5.03
512	55.26	7.25
1024	83.17	10.39
2048	141.31	14.83
elided Moebius		
256	34.57	5.03
512	54.98	7.25
1024	88.75	10.39
2048	142.68	14.83
4096	225.86	21.11

Table 4.5 (continued)

N	simulation result	analytic bound
Degree 4		
lens		
160	43.27	4.68
384	81.60	7.44
896	153.26	11.53
2048	278.83	17.61
de Bruijn		
256	57.56	6.01
512	97.43	8.64
1024	163.72	12.34
2048	278.81	17.61
hypertree 1		
255	18.45	4.84
511	26.64	6.99
1023	36.84	10.02
Degree 5		
hypertree 2		
255	43.63	4.39
511	69.96	6.35

4.4. Summary

This chapter has compared the reliability of many proposed multicomputer graphs. After showing that the even double-exchange and Moebius families can be easily modified to satisfy the conventional graph-theoretic reliability criterion of connectivity, we introduced two additional criteria: the ease of locally rerouting around failures, and the expected number of uniform random failures required to disconnect a graph (its ARF size). We have shown that these two criteria are closely related -- one measure of local (edge or vertex) reroutability, the RegionSize, provides a lower bound for the corresponding ARF size. Both the traditional standard of connectivity and our proposed reliability measures show that the elided even double-exchange and Moebius families are highly reliable, in addition to being denser than any previously proposed trivalent multicomputer graph.

Chapter 5
Conclusions

5.1. Summary

We have studied two important classes of criteria for effective multicomputer interconnection topologies, relating to density and reliability. Chapter 2 compared the known topologies in terms of density, interpolability, and routing algorithms. Here we saw that the densest families -- those with the least message transit times -- had diameters logarithmic in the number of vertices. Most of these highly dense families fell within the definitional framework of the single-exchange construction, which provided a uniform diameter bound, linear interpolability, and a simple routing algorithm. For the small degrees that we studied, still denser families were constructed using the related double-exchange operation. This method of attaining better density still allowed interpolability and simple routing; in the specific case of the even double-exchange graphs, the practical utility of the new family was further supported in Chapter 3 by demonstrating a computationally uniform emulation of the well-known shuffle-exchange network and a compact VLSI layout.

Chapter 4 began our investigation of the reliability of dense multicomputer graphs by summarizing the connectivities of known families. Eliding vertices with low degree from the densest trivalent graphs, the shuffle-exchange and two denser double-exchange families, made them 3-connecteded without harming their diameter, degree, routing, interpolability, or layout properties. The elided families were also shown to satisfy the additional reliability criterion of local reroutability, unlike other proposed modifications in which edges are added to existing vertices. Our analysis of local reroutability provided two formal measures of the cost of handling a failure by transparently detouring messages around it: the RegionSize and the RerouteCost. After deriving these costs for the dense multicomputer graph families, Chapter 4 then proved that they also provided a lower bound for our final reliability criterion, Average Random Failset size.

5.2. Future Research

Despite the progress made in designing high density graph families with small degrees, much further work remains to be done. The Moore bound still provides the only known lower bound on k/lg N for infinite families; raising this bound will surely require more powerful use of graph theory. Pragmatically, the bound suggests that there may be far denser multicomputer graph families still to be found; at degree 3, for example, we have lowered the attainable k/lg N from 2 to 1.47, but the only known limit is 1.

Linear interpolability gives the designer of multicomputers great freedom in chosing the numbers of processors used, but does not adequately answer the problem of extensibility: can processors be added cheaply to an existing network? Families with fixed degree at least guarantee that the processors will not need to be radically modified, but designs should demand only small amounts of line and routing changes for small additions. The tree variations clearly satisfy this criterion, but we have not yet found any attack on the question for the single-exchange and double-exchange families.

We seem to be close to finding an asymptotically optimal layout for the even double-exchange, using our homomorphism and the known optimal layouts for the

shuffle-exchange; if a similar homomorphism could be found for the Moebius, that family would be still more appealing.

Local reroutability not only provides a suitable measure of the cost of distributed failure recovery, but also provides a tractable lower bound for the Average Random Failset size of many desirable topologies. We expect to finish deriving local rerouting costs for other dense double-exchange families shortly. For the longer term, we are interested in the effects of multiple simultaneous failures. Local reroutability already guarantees that multiple failures can be handled when they are sufficiently far apart; if we can provide limits on the rerouting regions needed for closer failures, these may in turn allow tighter bounds to be derived for ARF sizes. In any case, ad hoc arguments may provide much more precise ARF sizes for specific families.

The usefulness of an implementation depends on many other criteria. For fault tolerance in particular, a network needs to be able to diagnose failures in a distributed fashion. Ideally, a diagnosability model could be designed for the general single-exchange and double-exchange constructions. Another critical issue is traffic congestion; algorithms that communicate only with nearby vertices may therefore be important, and may

make the conventional cost criteria of diameter and average distance less applicable. Our homomorphism guarantees that algorithms with good locality in the shuffle-exchange would also be local in the double-exchange; this approach needs to be extended to interrelate more multicomputer graph families.

5.3. Conclusions

Multicomputer interconnection topologies with small degree and high density are essential to achieving in practice the increased processing power promised by recent advances in microprocessor and VLSI technology. Our double-exchange families offer superior density for small degrees, without sacrificing other practical requirements, involving routing and reliability.

References

[1] D. H. Lawrie, "Access and alignment of data in an array processor," IEEE Transactions on Computers C-24, 12, pp. 1145-1155 (December 1975).

[2] T. Lang, "Interconnections between processors and memory modules using the shuffle-exchange network," IEEE Transactions on Computers C-25, 5, pp. 496-503 (May 1976).

[3] D. Hoey and C. E. Leiserson, "A layout for the shuffle-exchange network," Proc. 1980 International Conference on Parallel Processing, pp. 329-336 (August 1980).

[4] D. Kleitman, F. T. Leighton, M. Lepley, and G. L. Miller, "New layouts for the shuffle-exchange graph," Proc. 13th Annual ACM Symposium on the Theory of Computing, pp. 278-292 (11-13 May 1981).

[5] F. Harary, Graph Theory, Addison-Wesley, Reading, Massachusetts (1969).

[6] E. Chang, "An introduction to echo algorithms," Proc. 1st International Conference on Distributed Computers, pp. 193-198 (October 1979).

[7] R. S. Wilkov, "Analysis and design of reliable computer networks," IEEE Transactions on Communications COM-20, 3, pp. 660-678 (June 1972).

[8] Y. J. Park and S. Tanaka, "Reliability evaluation of a network with delay," IEEE Transactions on Reliability R-28, 4, pp. 320-324 (October 1979).

[9] F. T. Boesch, F. Harary, and J. A. Kabell, "Graphs as models of communication network vulnerability: connectivity and persistence," Networks 11, 1, pp. 57-63 (Spring 1981).

[10] A. J. Hoffman and R. R. Singleton, "On Moore graphs with diameters 2 and 3," IBM Journal of Research and Development 4, 5, pp. 497-504 (November 1960).

[11] E. Bannai and T. Ito, "On finite Moore graphs," Journal of the Faculty of Science of the University of Tokyo, Section 1A 20, 2, pp. 191-208 (August 1973).

[12] R. M. Damerell, "On Moore graphs," Proc. Cambridge Philosophical Society 74, 2, pp. 227-236 (September 1973).

[13] P. Erdos, S. Fajtlowicz, and A. J. Hoffman, "Maximum degree in graphs of diameter 2," Networks 10, 1, pp. 87-90 (Spring 1980).

[14] J. R. Goodman and A. M. Despain, "A study of the interconnection of multiple processors in a data base environment," Proc. 1980 International Conference on Parallel Processing, pp. 269-278 (August 1980).

[15] B. W. Arden and H. Lee, "Analysis of chordal ring network," Proc. Workshop on Interconnection Networks for Parallel and Distributed Processing, pp. 93-100 (April 1980).

[16] C. H. Sequin, "Doubly twisted torus networks for VLSI processor arrays," Proc. 8th Symposium on Computer Architecture, pp. 471-480 (May 1981).

[17] R. A. Finkel and M. H. Solomon, "Processor interconnection strategies," IEEE Transactions on Computers C-29, 5, pp. 360-371 (May 1980).

[18] H. Sullivan, T. R. Bashkow, and K. Klappholz, "A large scale, homogeneous, fully distributed parallel machine," Proc. 4th Symposium on Computer Architecture, pp. 105-124 (March 1977).

[19] J. C. Bermond and B. Bollobas, "The diameter of graphs - a survey," Proceedings of the 12th Southeastern Conference on Combinatorics, Graph Theory, and Computing, (December 1981).

[20] L. N. Bhuyan and D. P. Agrawal, "A general class of processor interconnection strategies," Proc. 9th Symposium on Computer Architecture, pp. 90-98 (April 1982).

[21] L. D. Wittie, "Communication structures for large networks of microcomputers," IEEE Transactions on Computers C-30, 4, pp. 264-273 (April 1981).

[22] H. D. Friedman, "A design for (d,k) graphs," IEEE Transactions on Electronic Computers EC-15, 4, pp. 253-254 (April 1966).

[23] B. E. Arden and H. Lee, "A multi-tree-structured network," Proceedings of Compcon, Computer Communications Networks, pp. 201-210 (Fall 1978).

[24] J. R. Goodman and C. H. Sequin, "Hypertree, a multiprocessor interconnection topology," IEEE Transactions on Computers, (1979) Submitted for publication.

[25] S. W. Golomb, "Permutations by cutting and shuffling," SIAM Review 3, pp. 293-297 (October 1961).

[26] H. S. Stone, "Parallel processing with the perfect shuffle," IEEE Transactions on Computers C-20, 2, pp. 153-161 (February 1971).

[27] F. P. Preparata and J. Vuillemin, "The cube-connected-cycles: A versatile network for parallel computation," CACM 24, 5, pp. 300-309 (May 1981).

[28] R. A. Finkel and M. H. Solomon, "The Lens Interprocessor Connection Strategy," IEEE Transactions on Computers C-30, 12, pp. 960-965 (December 1981).

[29] G. Farhi, Diametres dans les graphes et numerotations gracieuses, Ph.D. Thesis, l'Universite de Paris Sud (30 April 1981).

[30] D. S. Parker, Jr., "Notes on shuffle/exchange-type switching networks," IEEE Transactions on Computers C-29, 3, pp. 213-222 (March 1980).

[31] H. J. Siegel and S. D. Smith, "Study of multistage SIMD interconnection networks," Proc. 5th Annual Symposium on Computer Architecture, pp. 223-229 (April 1978).

[32] T. Lang and H. S. Stone, "A shuffle-exchange network with simplified control," IEEE Transactions on Computers C-25, 1, pp. 55-65 (January 1976).

[33] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," 1st Annual Symposium on Computer Architecture, pp. 21-28 (December 1973).

[34] M. C. Pease, "The indirect binary n-cube microprocessor array," IEEE Transactions on Computers C-26, 5, pp. 458-473 (May 1977).

[35] D. G. de Bruijn, "A combinatorial problem," Koninklijke Nederlandsche Academie van Wetenschappen te Amsterdam, Proc. Section of Sciences 49, 7, pp. 758-764 (1946).

[36] H. S. Stone, "Dynamic memories with enhanced data access," IEEE Transactions on Computers C-21, 4, pp. 359-366 (April 1972).

[37] Y. V. Golunkov, "Automation program realization of substitutions of symmetric semigroups II," Kibernetika 5, pp. 35-42 (1975).

[38] C. W. H. Lam, "On some solutions of A^k = dI + lamda J," Journal of Combinatorial Theory, Series A 23, pp. 140-147 (1977).

[39] M. Imase and M. Itoh, "Design to minimize diameter on building-block network," IEEE Transactions on Computers C-30, 6, pp. 439-442 (June 1981).

[40] D. K. Pradhan, "Interconnection topologies for fault-tolerant parallel and distributed architectures," Proc. 1981 International Conference on Parallel Processing, pp. 238-244 (25 August 1981).

[41] R. M. Storwick, "Improved construction techniques for (d,k) graphs," IEEE Transactions on Computers C-19, 12, pp. 1214-1216 (December 1970).

[42] W. E. Leland, R. A. Finkel, Li Qiao, M. H. Solomon, and L. Uhr, "High density graphs for processor interconnection," Information Processing Letters 12, 3, pp. 117-120 (13 June 1981).

[43] J. C. Bermond, C. Delorme, and G. Farhi, "Large graphs with given degree and diameter II," Rapport de recherche 105, Universite de Paris-sud, Centre d'Orsay, Laboratoire de Recherche en Informatique (June 1981).

[44] J. C. Bermond, C. Delorme, and G. Farhi, "Large graphs with given degree and diameter III," Rapport de recherche 104, Universite de Paris-sud, Centre d'Orsay, Laboratoire de Recherche en Informatique (September 1981).

[45] J. J. Quisquater, New constructions of large graphs with fixed degree and diameter, To appear.

[46] J. C. Bermond, C. Delorme, and J. J. Quisquater, "Grands graphes non diriges de degre et diametre fixes," Rapport de recherche 113, Universite de Paris-sud, Centre d'Orsay, Laboratoire de Recherche en Informatique (December 1981).

[47] W. E. Leland and M. H. Solomon, "Dense trivalent graphs for processor interconnection," IEEE Transactions on Computers C-31, 3, pp. 219-223 (March 1982).

[48] M. C. Pease, "An adaptation of the Fast Fourier Transform for parallel processing," Journal of the ACM 15, 2, pp. 252-264 (April 1968).

[49] J. T. Schwartz, "Ultracomputers," ACM Transactions on Programming Languages and Systems 2, 4, pp. 484-521 (October 1980).

[50] J. P. Fishburn and R. A. Finkel, "Quotient networks," IEEE Transactions on Computers C-31, 4, (April 1981).

[51] C. D. Thompson, "Area-time complexity for VLSI," Proc. 11th Annual ACM Symposium on the Theory of Computing, pp. 81-88 (May 1979).

[52] C. D. Thompson, A Complexity Theory for VLSI, Ph.D. Thesis, Carnegie-Mellon University Computer Science Department (September, 1979).

[53] H. Frank and I. T. Frisch, "Analysis and design of survivable networks," IEEE Transactions on Communications Technology COM-18, pp. 501-519 (October 1970).

[54] I. Rubin, "On reliable topological structures for message-switching communication networks," IEEE Transactions on Communications COM-26, 1, pp. 62-74 (January 1978).

[55] D. Minoli and E. H. Lipper, "Cost implications of survivability of terrestrial networks under malicious failure," IEEE Transcations on Communications COM-28, 9, pp. 1668-1674 (September 1980).

[56] K. Steiglitz, P. Wiener, and D. J. Kleitman, "The design of minimum-cost survivable networks," IEEE Transactions on Circuit Theory 16, pp. 455-460 (November 1969).

[57] F. T. Boesch and R. E. Thomas, "On graphs of invulnerable communication networks," IEEE Transactions on Communications Technology COM-18, pp. 484-489 (May 1970).

[58] K. K. Saluja and S. M. Reddy, "A class of undirected graphs," Proc. 15th Annual Conference on Information Systems and Sciences, pp. 388-393 (March 1981).

[59] B. Bollobas, "A problem of the theory of communication networks," pp. 29-36 in Theory of Graphs, Akad. Kaido, Budapest (1968) edited by G. Kalona and P. Erdos.

[60] J. Bartman and I. Rubin, "On diameter stability of graphs," pp. 247-254 in Theory and Applications of Graphs, Springer-Verlag; Lecture Notes 642 (1978) edited by Y. Alavi and D. R. Lick.

[61] A. M. Farley, "Networks immune to isolated failures," Networks 11, 3, pp. 255-268 (Fall 1981).

[62] J. R. Goodman and C. H. Sequin, "Hypertree, a multiprocessor interconnection topology," Technical Report 427, University of Wisconsin--Madison Computer Sciences (April 1981).

[63] D. E. Knuth, The Art of Computer Programming Volume 1--Fundamental Algorithms (second edition), Addison-Wesley (1973).