FIBONACCI NETWORKS

Rama K. Govindaraju
Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180
govindar@cs.rpi.edu

M. S. Krishnamoorthy*
Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180
moorthy@cs.rpi.edu

Narsingh Deo
Department of Computer Science, University of Central Florida, Orlando, FL 32816
deo@eola.cs.ucf.edu
(*Submitted January 1993*)

1. INTRODUCTION

Several interconnection networks have been proposed in literature for interconnecting computing elements. The interconnection network usually forms a *regular* pattern, which is exploited by the algorithms running on the network. Some of the commercially available networks are the hypercube, mesh, etc., which are highly regular. The advantage of using such regular networks is that the algorithms written for one network can be extended with minimal effort to larger versions of the same network. However, networks like the hypercube, mesh, etc, have one significant disadvantage; they do not scale in increments of one. A hypercube scales in exponents of two, and a mesh scales in order of n or k, in an $n \times k$ mesh.

A tree is the cheapest interconnection network but has unacceptably poor communication and fault-tolerant properties. On the other hand, the complete graph K_n is highly reliable but is extremely expensive. Some of the desirable properties of interconnection networks are high fault tolerance, small diameter, small degree, high connectivity, symmetry/regularity, etc. (most of which are conflicting properties).

A class of networks called *Iterative networks* were proposed to address some of the drawbacks of commercially available networks [3, 4, 7, 8, 12]. Iterative networks can be scaled in increments of one. In fact, they can scale by *any* k, where $k \geq 1$. Interconnection networks are often modeled as undirected graphs, where vertices correspond to processor-memory nodes, and edges represent full-duplex communication links between pairs of nodes. An iterative network of n nodes is a subgraph of the network with $n+1$ nodes. The algorithms running on iterative networks require minimal modifications when extended to scaled versions of the network. This is a significant advantage over networks like hypercube, mesh, etc.

Some of the proposed iterative networks that have appeared in literature are mentioned below. Stirling networks [3] are defined using Stirling numbers of the first kind. Rencontres networks [4] are defined based on rencontres numbers. Pascal networks [7] are defined using the Pascal triangle. Several others, like Steinhaus networks [12], Circulants [2], Topelitz networks [8], etc., have also been proposed in literature. All of these have some of the desirable properties of interconnection networks, but also have certain drawbacks. So the search for new interconnection networks for various classes of problems continues.

---

* Supported in part by the National Science Foundation under Grant Number CDA-8805910.

In this paper we introduce a new class of iterative networks using Fibonacci numbers, which we call *Fibonacci networks*. We investigate their graph-theoretic properties and study their suitability for implementing multicomputer systems. The paper is organized as follows: In section 2 we show how Fibonacci networks are constructed. In section 3 we explore some of the properties of these interconnection networks. We show that Fibonacci networks have most of the properties desirable in an interconnection network except that it has too many links making it expensive. We then show how the number of links can be reduced while still maintaining the basic structure of the network. We also explore the properties of the modified network and show that it still retains most of the desired properties of interconnection networks. In section 4 we show that routing can be accomplished very efficiently in Fibonacci networks. In section 5 we show how other networks can be embedded onto Fibonacci networks of comparable size. In section 6 we design some of the basic algorithms, like finding a minimum spanning tree, that can be implemented on Fibonacci networks. Finally, we present some concluding remarks. We have used standard graph-theoretic notation throughout this paper [6]. All logarithms are with respect to base 2 unless specifically mentioned otherwise.

2. FIBONACCI NETWORKS

Fibonacci networks are a class of iterative/recursive networks constructed as described below. Let $fib(q)$ denote the q^{th} Fibonacci number F_q (0 and 1 being the 0^{th} and 1^{st} Fibonacci numbers, respectively). Let $FT(r,k)=fib(k+\sum_{i=0}^{r-1} i)$ for $0<k<r$. An $n\times n$ symmetric matrix is called a *Fibonacci Matrix* $FM^p(n)$ of order n if its main diagonal entries are all 0 and its lower triangular entries (and, therefore, upper also) consist of the $\{0, 1\}$ predicate values $(FT(n-1,k)\pmod{p}\neq 0)$, where p is usually a small prime. (Later we will show how this definition can be extended when p is a set of primes.) Let

$$fm^p_{i,j}=(i,j)^{\text{th}} \text{ element of } FM^p(n)\in\{0,1\},$$

$$ft_{i,j}=(i,j)^{\text{th}} \text{ element of } FT(n,k)\in N.$$

Then, by definition,

$$fm^p_{i,j}=(ft_{i-1,j}\pmod{p}\neq 0),$$

and hence,

$$fm^p_{i,j}=\left(fib\left(j+\sum_{x=0}^{i-2}x\right)\pmod{p}\neq 0\right),$$

or, alternately,

$$fm^p_{i,j}=\left(fib\left(\frac{(i-1)(i-2)}{2}+j\right)\pmod{p}\neq 0\right),\quad j=1,2,\ldots,i-1.$$

An undirected simple (without parallel edges or self loops) graph that has $FM^p(n)$ as its adjacency matrix is called a *Fibonacci Graph* $FG^p(n)$ or order n. The vertices are numbered in the same order as the rows of $FM^p(n)$. Figure 1 depicts Fibonacci Graphs $FG^2(1)$ to $FG^2(8)$; Matrix 1 shows the matrix $FM^2(8)$.

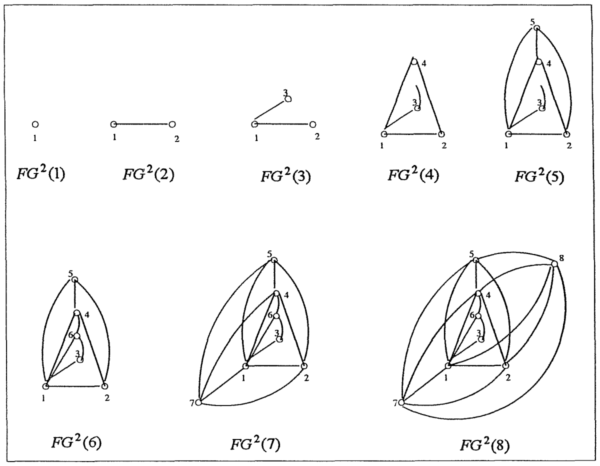

FIGURE 1. Fibonacci Graphs: $FG^2(1)-FG^2(8)$

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

MATRIX 1: $FM^2(8)$

A *Fibonacci network* with n processors and prime p is a mapping of the graph $FG^p(n)$. The vertices of the graph correspond to the processors and will be called *nodes*. The edges correspond to the communication *links* between nodes. By definition, $FG^p(n)$ is a subgraph of $FG^p(n+1)$. Hence, Fibonacci networks can be constructed incrementally. Addition of a node causes new links to be added from the new node to some of the existing nodes. None of the existing links are deleted.

Below, we list some of the symbols that are used throughout this paper.

$v_i \to v_j$ = Node v_i is adjacent to node v_j.

$v_i \mapsto v_j$ = Node v_i is not adjacent to node v_j.

$fib(n)$ = The n^{th} Fibonacci number F_n. We redefine this notation for convenience.

v_i = Vertex i in $FG^p(n)$ or node i in the corresponding network.

$\langle d_k d_{k-1} \ldots d_1 d_0 \rangle$ = Decimal representation of a positive integer. d_0 is the least significant digit and d_k is the most significant digit.

$Dia^p(n)$ = The diameter of the graph $FG^p(n)$.

$Dia_f^p(n)$ = The fault diameter of the graph $FG^p(n)$.

$deg^p(v_i) =$ The degree of vertex v_i.

$pkt_m =$ Message packet m

$dest(pkt_m) =$ Destination node of packet m.

$e^p(n) =$ The number of edges in the Fibonacci network $FG^p(n)$.

$s_p =$ The smallest integer greater than 0 such that p divides $fib(s_p)$.

Often we will omit the superscript p, in which case p is assumed to be 2.

It should be noted that this construction is different from the construction of Fibonacci Cubes [11] which also use Fibonacci numbers in their construction. However, Fibonacci Cubes are more like the hypercube and scale in increments equal to the Fibonacci numbers. The construction in [11] involves representing each node by a Fibonacci bit representation and determining adjacencies by differences in bit patterns.

3. PROPERTIES OF FIBONACCI NETWORKS

We first introduce some properties of Fibonacci numbers with respect to divisibility by primes and the degree of a vertex. The following lemma will be useful later.

Lemma 1: Prime p divides $fib(j \times s_p)$ for all $i > 0$.

Proof: We prove the lemma by induction on i. The base case is satisfied by definition of s_p. By hypothesis, let us assume that p divides $fib(j \times s_p)$ for some j. To prove that p divides $fib((j+1) \times s_p)$, we invoke the following [9]:

$$fib(n+k) = fib(k) \times fib(n+1) + fib(k-1) \times fib(n).$$

Substituting the above in $fib((j+1) \times s_p)$, we get

$$fib(j \times s_p + s_p) = fib(s_p) \times fib(j \times s_p + 1) + fib(s_p - 1) \times fib(j \times s_p).$$

Since p divides $fib(s_p)$ by base case, and p divides $fib(j \times s_p)$ by hypothesis, p divides $fib((j+1) \times s_p)$. A stronger property can be inferred immediately that p divides $fib(m)$ if and only if $m = j \times s_p$ for some integer j, since s_p is the smallest integer for which p divides $fib(s_p)$ □

We define yet another property of s_p.

Theorem 1: Let $p = (d_k d_{k-1} \dots d_1 d_0)$ be a prime less than 40; if p has t decimal digit representation, then $d_i = 0$ for all $i > (t-1)$.

$$s_p = \begin{cases} (p-1) & \text{if } d_0 = 1 \text{ or } (d_0 = 9 \text{ and } d_1 \text{ is odd}), \\ p & \text{if } d_0 = 5, \\ (p+1) & \text{if } (d_0 = 2) \text{ or } ((d_0 = 3 \text{ or } 7) \text{ and } d_i \text{ is even}), \\ (p+1)/2 & \text{if } ((d_0 = 3 \text{ or } 7) \text{ and } (d_1 \text{ is odd})), \\ (p-1)/2 & \text{if } ((d_0 = 9) \text{ and } (d_1 \text{ is even})). \end{cases}$$

Proof: We have verified the preceding relation for all primes less than 40 (using Mathematica). In this paper we will limit ourselves to primes $p < 10$ and will, therefore, assume this theorem to be true. □

Fibonacci Networks with $p = 2$

We now introduce some properties of Fibonacci networks when $p = 2$. From the previous section it is clear that $s_2 = 3$. We will assume the superscript to be 2 whenever omitted, for convenience in notation.

Proposition 1: The degree of a node v_k, $deg^2(v_k)$ in $FG^2(n)$, is given by

$$e(k)-e(k-1)+\sum_{i=k+1}^{n}\left(\left(\left(k+\sum_{j=0}^{i-2} j\right) \pmod 3\right) \neq 0\right).$$

Proof: $e(k)-e(k-1)$ sums all the "1" entries in row k from column 1 until the main diagonal of the adjacency matrix FM^2. The rest of the expression sums all the "1" entries in column k starting from row $k+1$ until row n of FM^2. Since $s_2 = 3$, we know that

$$e(k) = s(k) - \left\lfloor \frac{s(k)}{3} \right\rfloor, \text{ where } s(k) = (k \times (k-1))/2.$$

Substituting for $e(k)$ and $e(k-1)$ in the above equation and simplifying, we get

$$deg^2(v_k) = (k-1) - \left\lfloor \frac{k(k-1)}{6} \right\rfloor + \left\lfloor \frac{(k-1)(k-2)}{6} \right\rfloor + \sum_{i=k+1}^{n} \left(\left((i^2-3i+2k+2) \pmod 6\right) \neq 0\right). \quad \square$$

For $k < i$, matrix entry $f_{i,k}^2$ is "1" if and only if

$$\left(\left((i^2-3i+2+2k) \pmod 6\right) \neq 0\right).$$

We can now construct the following modulo 6 table for $k < i$.

TABLE 1. Connectivity of FM^2

i	i^2	$-3i$	$f_k(i) = i^2-3i+2k+2$
1	1	−3	$2k$
2	4	0	$2k$
3	3	−3	$2k+2$
4	4	0	$2k$
5	5	−3	$2k$
6	0	0	$2k+2$

Proposition 2: Node v_{3i+1} is adjacent to node v_j for all $j > (3i+1)$.

Proof: We first prove that $v_i \to v_j$ for all $j > 1$. Since $s_2 = 3$, the result then follows for all v_{3i+1}. By definition, $v_1 \to v_i$ if and only if $(fib(1+\sum_{m=0}^{i-2} m) \pmod 2)) \neq 0$. Therefore,it suffices to prove that the value $val(i) = (1+\sum_{m=0}^{i-2} m)$ is not divisible by $s_2 = 3$ for any $i > 1$. We prove this by contradiction. Let us assume that $(((i-1)\times(i-2))/2+1+1)$ is divisible by 3 for some i. Thus, i^2-3i+4 must be divisible by 3 for some i. Clearly if, for some i, 3 divides (i^2+1), then 3 cannot divide i^2 (hence, cannot divide i). Thus, we have $i^2 \equiv 1 \pmod 3$ for some i by Fermat's theorem. Therefore, $i^2+1 \equiv 2 \pmod 3$; hence, 3 does not divide (i^2+1) for any integer i. □

Proposition 3: Node v_{3i+1} is adjacent to node v_j for all j, if $j \pmod 3 \neq 0$.

Proof: For $j > (3i+1)$, the proof follows from Proposition 2. For $j < (3i+1)$, the entry $fm^2_{3i+1,j}$ is "1" if and only if $\left(\left(j+\frac{(3i-1)(3i)}{2}\right) \pmod 3 \neq 0\right)$; the proof follows. □

Proposition 4: Node v_{3k+2} is adjacent to node v_i if and only if $(i \pmod 3 \neq 0)$.

Proof: For entry $fm^2_{i,3k+2}$ to be "1," $\left(\left(3k+2+\frac{(i-2)(i-1)}{2}\right) \pmod 3 \neq 0\right)$. On simplifying, we need to prove that $((i^2-3i) \pmod 3 \neq 0)$. This is clearly true if and only if $(i \pmod 3 \neq 0)$. □

Proposition 5: Node v_{3k} is adjacent to node v_i (where $i > 3k$), if and only if $i = 3j$ for some $j > k$.

Proof: For node v_{3k} to be adjacent to node v_i, $\left(\left(3k+\frac{(i-2)(i-1)}{2}\right) \pmod 3 \neq 0\right)$, when $i = 3j$ for some $j > k$. On simplifying, we need to prove that $((i^2-3i+2) \pmod 6 \neq 0)$ for $i = 3j$. On substituting for $i = 1, 2, \ldots, 6$, we find that $v_i \to v_{3k}$ if and only if $i = 3j$ for some $j > k$. □

Proposition 6: Let $t(n)$ be the number of edges added to $FG(n-1)$ to get $FG(n)$. Then

$$t(n) = \begin{cases} 2\times\left(\frac{n}{3}\right)-1 & \text{if } n \pmod 3 = 0, \\ 2\times\left\lceil\left(\frac{n}{3}\right)\right\rceil-1 & \text{if } n \pmod 3 = 2, \\ 2\times\left\lceil\left(\frac{n}{3}\right)\right\rceil-2 & \text{if } n \pmod 3 = 1. \end{cases}$$

Proof: The proof follows directly from the definition of $FG^2(n)$ and Lemma 1. □

Proposition 7: The number of edges $e(n)$ in the Fibonacci network $FG^2(n)$ with n nodes is given by

$$e(n) = \begin{cases} \frac{n}{3}\times(n-1) & \text{if } n \pmod 3 = 0, \\ e(n-1)+t(n) & \text{if } n \pmod 3 = 1, \\ e(n-2)+t(n)+t(n-1) & \text{if } n \pmod 3 = 2. \end{cases}$$

Proof: From Proposition 6 and Proposition 1, we know that $e(3k) = e(3k-3)+6(k-1)+2$. Solving this recurrence we get $3(3k) = k\times(3k-1)$. The result follows from this equation and Proposition 6. □

Proposition 8: The maximum degree of a vertex in $FG(n)$ is $n-1$.

Proof: The proof follows from Proposition 2. The degree of vertex v_1 is $n-1$. □

Proposition 9: The diameter $Dia(n) = 2$.

Proof: The $deg(v_1) = n-1$. This means that the diameter of $FG(n)$ is ≤ 2. □

Proposition 10: Node v_3 has minimum degree in $FG(n)$ for $n \geq 3$.

Proof: From Proposition 3, we know that the degree of nodes v_{3i+1} increases by at least 2 for every 3 nodes added to the network. From Proposition 4, the degree of nodes v_{3i+2} increases by 2 for every 3 nodes added in the network. From Proposition 5, the degree of nodes v_{3i} increases by only 1 for every 3 nodes added in the network, when the nodes have numbers greater than $3i$ and increases by 2 otherwise. So the node with minimum degree is the smallest v_{3i} node, which is v_3 and its degree is $\lfloor \frac{n}{3} \rfloor$. □

Proposition 11: The node connectivity of $FG(n) = deg(v_3)$.

Proof: To prove that there are at least $\lfloor \frac{n}{3} \rfloor$ node-disjoint paths between any 2 nodes (let us say, v_i and v_j) of $FG(n)$, we show that both v_i and v_j can reach all the nodes v_{3q+1}, where $q \leq (deg(v_3) - 1)$, either directly or through a nonintersecting set of node paths or $v_i \to v_j$. Let $deg(v_3) - 1 = k$.

Case 1. Integers i and j are both greater than $3k+1$. By Proposition 2, both v_i and v_j are adjacent to v_{3q+1} for $q \in \{0, 1, \ldots, k\}$. Therefore, there are at least $k+1$ node-disjoint paths between v_i and v_j.

Case 2. Integers i and j are both less than $3k+1$. We have three subcases to prove.

a. If $i = 3r+1$ or $i = 3r+2$, then, from Propositions 3 and 4, we know that $v_i \to v_{3q+1}$ for $q = \{0, 1, \ldots, k\}$.

b. If $i = 3r$ and v_j is not adjacent to v_{3q+1} for some $q \leq k$, then, from Proposition 5, $v_i \to v_{3q+3}$ and $v_{3q+3} \to v_{3q+1}$. Likewise, we can prove for v_j.

c. If v_j and v_i are not adjacent to the same v_{3q+1} for some q, then, by Proposition 5, $v_i \to v_j$.

Case 3. One of i or j is less than $3k+1$ and the other is greater. This is just a subcase of Cases 1 and 2. If the node number is less than v_{3k+1}, then Case 1 holds, and if it is greater, Case 2 holds.

The proof follows from these three cases. □

Proposition 12: The fault diameter $Dia_f(n) = Dia(n) + 1$.

Proof: First, we show that the network remains connected in the event of $deg(v_3) - 1$ node failures. We then show that the diameter of the fault-free network increases by at most 1. In the worst case, nodes v_{3q+1} all fail where $q = \{0, 1, \ldots, deg(v_3) - 2\}$ since they are the nodes with maximum degree. We show that every node v_i can reach v_{3x+1} where $x = deg(v_3) - 1$. We have two cases to consider:

Case 1. $i > (3x+1)$. In this case we know, from Proposition 2, that $v_i \to v_{3x+1}$.

Case 2. $i < (3x+1)$. In this case, if $i = 3r+1$ or $3r+2$ for some $r > 0$, then, from Propositions 3 and 4, $v_i \to v_{3x+1}$. If $i = 3r$ for some $r > 0$ and $j = 3y+2$ for some $y > 0$ as shown in the connectivity proof.

Thus, the fault diameter $Dia_f(n) = Dia(n) + 1$. □

Proposition 13: $FG(n)$ is nonplanar for all $n \geq 7$.

Proof: The graph $FG(7)$ has K_5 as a subgraph (nodes: $v_1, v_2, v_4, v_5,$ and v_7). Therefore, by Kuratowski's theorem, the proof follows since $FG(n+1)$ is a subgraph of $FG(n)$ for all integers $n > 7$. □

Fibonacci Networks with p = 2 and 3

When p = 2 as shown in Proposition 7, the number of links in the network is very high (order n^2). A simple way to reduce the number of links while still retaining most of the properties of the network is to modify the definition of $FM^p(n)$, where p is a set of primes $\{p_1, p_2, \ldots, p_k\}$. So, in this case, we have

$$X = fib\left(j + \sum_{m=0}^{i-2} m\right)$$

and

$$fmp_{i,j}^p = (X \pmod{p_1} \neq 0) \times (\bmod\, p_2) \neq 0) \times \cdots \times (X \pmod{p_k} \neq 0), \text{ for } j < i.$$

The rest of the definitions remain the same. The above construction deletes some of the links in the original network. $FM^p(n)$ is a symmetric $n \times n$ matrix whose main diagonal entries are all 0, and its lower triangle (and, therefore, upper also) consists of entries $fmp_{i,j}^p$ The graph which has $FM^p(n)$ as its adjacency matrix is represented by $FG^p(n)$. The graphs $FG^{\{2,3\}}(1)$ through $FG^{\{2,3\}}(10)$ are shown in Figure 2 and the matrix $FM^{\{2,3\}}(10)$ is shown in Matrix 2. For all $n > 0$, $FG^{\{2,3\}}(n)$ is a subgraph of $FG^2(n)$.

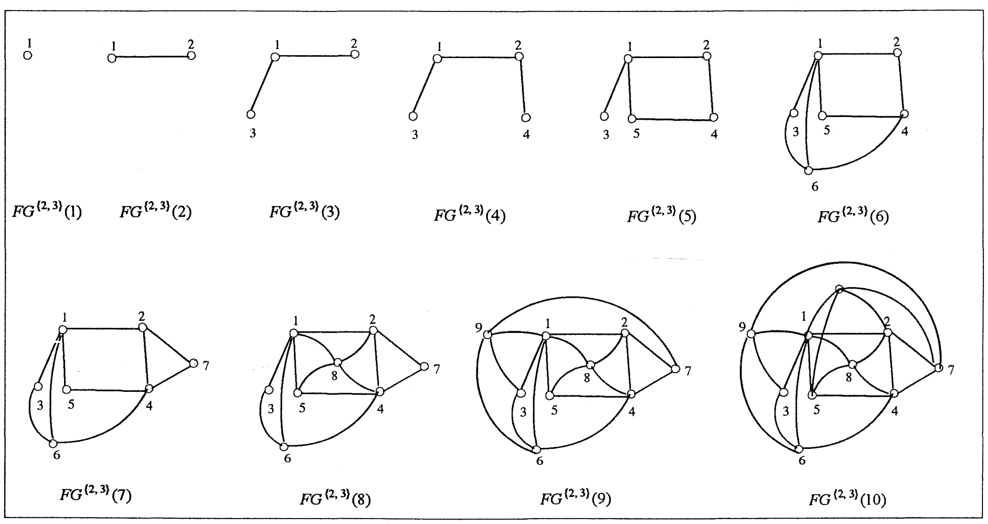

FIGURE 2. Fibonacci23 Graphs: $FG^{\{2,3\}}(1) - FG^{\{2,3\}}(10)$

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Matrix 2: $FM^{2,3}(10)$

For the remainder of this subsection, the superscript {2, 3} is assumed and is omitted for the sake of clarity. From Theorem 1, we know that $s_2 = 3$ and $s_3 = 4$. Before exploring the connectivity of this modified network, we first prove a lemma that will be useful in later proofs.

Lemma 2: $fib(n)$ is divisible by 12 if and only if n is divisible by 12.

Proof: We prove the lemma by induction. The base case is clearly true since $fib(12) = 144$. By hypothesis, let $fib(12 \times k)$ be divisible by 12. We must prove that 12 divides $fib(12 \times (k+1))$. But from [9] we have $fib(12 \times (k+1)) = 144 \times fib(12 \times k + 1) + 89 \times fib(12 \times k)$. Since 12 divides $fib(12 \times k)$ by hypothesis, the lemma follows. □

Proposition 14 Let $s(n) = \sum_{i=0}^{n-1} i$, then

$$e(n) = s(n) - \left\lfloor \frac{s(n)}{3} \right\rfloor - \left\lfloor \frac{s(n)}{4} \right\rfloor + \left\lfloor \frac{s(n)}{12} \right\rfloor.$$

Proof: The total number of edges is equal to the number of "1" entries in the lower triangle of $FM^{\{2,3\}}(n)$. Since $s_2 = 3$ and $s_3 = 4$, the above expression follows from the principle of inclusion and exclusion. □

Proposition 15: The degree of a node v_k, $deg^{\{2,3\}}(v_k)$ in a network $FG^{\{2,3\}}(n)$, is given by

$$e(k) - e(k-1) + \sum_{i=k+1}^{n} ((X \pmod 3 \neq 0) \;\&\; (X \pmod 4 \neq 0)),$$

where $X = \left(k + \sum_{j=0}^{i-2} j\right)$.

Proof: This follows using the same outline as shown in the proof of Proposition 1. □

For $k < i$, the matrix entry $fmp_{i,k}^{2,3}$ is "1" if and only if

$$\left(((i^2 - 3i + 2 + 2k) \pmod 6 \neq 0) \;\&\; ((i^2 - 3i + 2 + 2k) \pmod 8 \neq 0)\right).$$

The expression inside the summation forms a field modulo 24 and the degree of nodes increases symmetrically with the addition of every 24 nodes (see Table A-1 in the Appendix).

Proposition 16: If a node $v_j \mapsto v_{3i+1}$ for some $j > (3i+1)$, then $v_j \mapsto v_{3i+4}$ and vice versa.

Proof: Let $X = \left((3i+1) + \Sigma_{k=1}^{j-2} k\right)$. We need to prove that $X + 3$ is not divisible by 4 if $v_j \mapsto v_{3i+1}$. We know that if $v_j \mapsto v_{3i+1}$ for some $j > (3i+1)$, then, by Proposition 2, X must be divisible by 4. Since X is divisible by 4, $X + 3$ cannot be divisible by 4. Hence, $v_j \mapsto v_{3i+4}$ if $v_j \mapsto v_{3i+1}$. The vice versa proof follows similarly. □

Proposition 17: The maximum degree of a node in $FG(n) = deg(v_1) = deg(v_4)$.

Proof: This follows from Proposition 15. The degree of node v_1 for every 24 nodes is $deg(v_1) = deg(v_4) = 17k$, where $k = \left\lfloor \frac{n}{24} \right\rfloor$. □

Proposition 18: The diameter of $FG(n)$, $Dia(n) = 3$.

Proof: The proof follows from Proposition 16.

Proposition 19: The minimum degree of a node in $FG(n) = deg(v_3)$.

Proof: This follows from Proposition 15, using the same argument as in the proof of Proposition 10. The degree of node v_3 for every 24 nodes is $deg(v_3) = deg(v_9) = 1 + 6k$, where $\left\lfloor \frac{n}{24} \right\rfloor$. □

4. ROUTING

Routing in Fibonacci networks can be preformed very efficiently because of their high connectivity. We consider the case in which $p = 2$. We exploit the fact that nodes $v_{3i+1} \mapsto v_j$ for all $j > (3i+1)$.

Input: A one-to-one permutation showing source and destination nodes.

Output: A path for each packet to be routed.

<u>Step 1</u>. Each node v_j routes its packet to node v_i, where $((i = \max(3k+1)) \le j)$.

<u>Step 2</u>. Each v_i that receives a packet in Step 1 routes the packet pkt_m to v_ℓ such that $((\ell = \max(3r+1)) \le dest(pkt_m))$.

<u>Step 3</u>. Each v_i that receives a packet in Step 2 routes the packet pkt_m to $dest(pkt_m)$.

The algorithm clearly runs in constant time. The number of packets at any node at any given instance of time is at most 3, assuming that each processor node works in synchronous lock step.

When $p = \{2, 3\}$, the routing algorithm requires only a minor modification, as shown below. If $v_j \mapsto v_i$, where $((i = \max(3k+1)) \le j)$, then, by Proposition 12, $v_j \mapsto v_{3k+4}$. So, if $v_j \mapsto v_{3k=1}$ for some v_j in the previous algorithm, it reroutes through v_{3k+4}. This increases the routing complexity by 2 steps for certain packets and the maximum number of packets queued at any node at any given time is at most 6. The algorithm still runs in constant time, with constant queue lengths. We have shown that a network with $p = 2$ can be simulated by a network with $p = \{2, 3\}$ with a loss of speed by a constant factor only.

5. REDUCING THE TOTAL NUMBER OF LINKS

In Section 3 we showed how we could reduce the total number of links by using a higher prime number to prune some of the links. By using prime '3,' the number of links was reduced by 17%. In this section we describe three methods of further reducing the total number of links while maintaining the basic structure of the network.

1. Using higher primes: We follow the same technique as described in the construction of Fibonacci networks with primes 2 and 3. The following table shows the effect of using higher primes on the total number of links.

TABLE 2. Effect of Using Larger Primes

Primes Used	*Percentage of Links Pruned*	*New Diameter*
3	17	3
3, 5	27	4
3, 5, 13	32	6
3, 5, 13, 7	35	7

The number of links reduces by 35% from *FN* by using four more primes. The number of links pruned is computed using the principle of inclusion and exclusion, as shown in the proof of Proposition 14. The diameter results follow, using the argument given in the proof of Proposition 18. It should be noted that the primes p were selected based on the smallest s_p values ($s_{13} < s_7 < s_{11}$). The diameter, which reflects the slow-down in the routing time, increases almost linearly with the number of primes used. Therefore, the routing time slows down by a factor of 7, while 35% of the links in *FN* have been pruned. We observe that using primes higher than 13 results in diminishing returns.

2. Bounding maximum degree to log(n): The second technique that can be used is to bound the maximum degree of each node to $\log(n)$ (or any predefined constant) for $n > c$ (where c is a suitable constant). Therefore, for a network of size less than c, the network is identical to *FN*. For $n > c$ "1" entries in the matrix are set to "0" if the degree of the corresponding node has already reached $\log(n)$. It can easily be shown that the diameter of this network is $O(\log(n))$ and the total number of links is $e(n) < n \times \log(n)$. This network is quite similar to the hypercube. However, this technique does not preserve the basic structure of the Fibonacci network. The routing algorithm will have to be appropriately modified.

3. Cube connected Fibonacci network: The third technique is to replace each node in *FN* by a cycle of length equal to the degree of the node (just as is done in Cube Connected Cycles). This will increase the diameter of the network while reducing the overall degree. However, a problem with this approach is that the network is no longer scalable by one node.

6. EMBEDDING OF VARIOUS TOPOLOGIES

Claim 1: A complete binary tree of k levels (containing $2^k - 1$ nodes) is a subgraph of $FG(3 \times (2^{k-1} - 1))$.

Proof: We show that a complete binary tree of k levels can be mapped on $FG(3\times(2^{k-1}-1))$. From Proposition 2, we know that $v_{3i+1}\to v_j$ for all $j>(3i+1)$. As shown in Figure 3, we assign the nodes of level 1 through level $k-1$ processor nodes v_{3j+1} in order, where $j\in\{0,1,\dots,(2^{k-1}-2)\}$ Each node v_{3j+1} in level $k-1$ is adjacent to nodes v_{3j+2} and v_{3j+3}, which form the leaf nodes. The number of processors required up to $k-1$ levels is $3\times(2^{k-1}-1)-2$. Therefore, the last leaf node processor required is $3\times(2^{k-1}-1)$

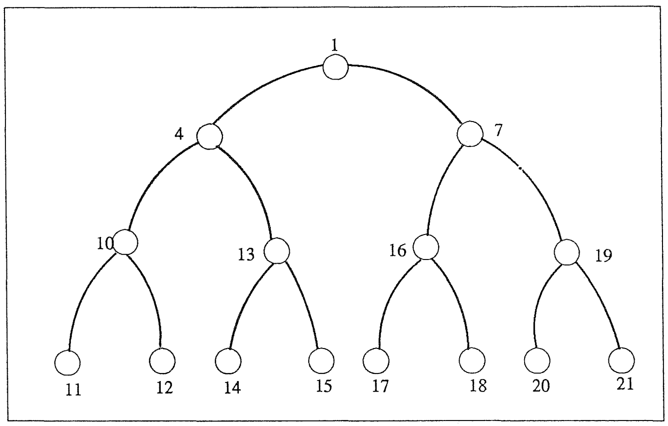

FIGURE 3. Embedding a Complete Binary Tree on *FN*

Claim 2: A complete ringed binary tree of k levels (containing 2^k-1 nodes) is a subgraph of $FG(3\times(2^k-1)-2)$.

Proof: We follow the same outline as in the previous proof. We construct all k levels the same way as we construct $k-1$ levels in the previous proof. From Proposition 2, we know that $v_{3j+1}\to v_j$ for all $j>(3i+1)$. As shown in Figure 4, we assign the nodes of level 1 through level k, processor nodes v_{3j+1} in order, where $j=\{0,1,\dots,(2^k-2)\}$.

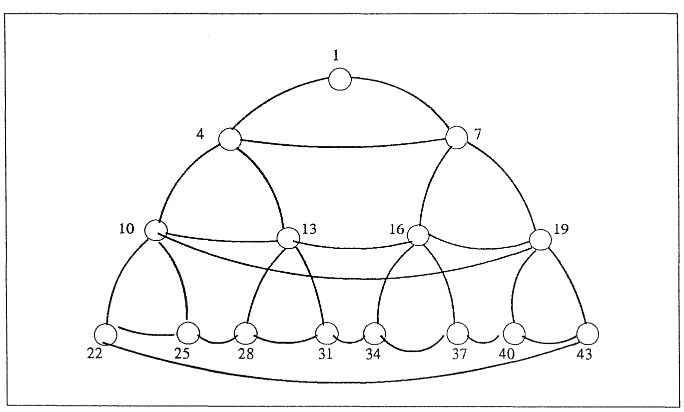

FIGURE 4. Embedding a Ringed Binary Tree of *FN*

Claim 3: A rectangular mesh of size $\ell + k$ is a subgraph of $FG\left(\left\lceil \frac{3(\ell \times k)}{2} \right\rceil\right)$.

Proof: We show how the mesh can be embedded on $FG(n)$. All nodes v_i such that $i \pmod 3 \neq 0$ can be arranged in increasing order, row-wise. The horizontal adjacencies are guaranteed by Propositions 2 and 5 above, and the vertical adjacencies are guaranteed by Proposition 2. Only every third node v_{3j} is not used in the embedding. Therefore, the number of nodes used is $\left\lceil \frac{3(\ell \times k)}{2} \right\rceil$. An example embedding of a 4×4 mesh is shown in Figure 5.

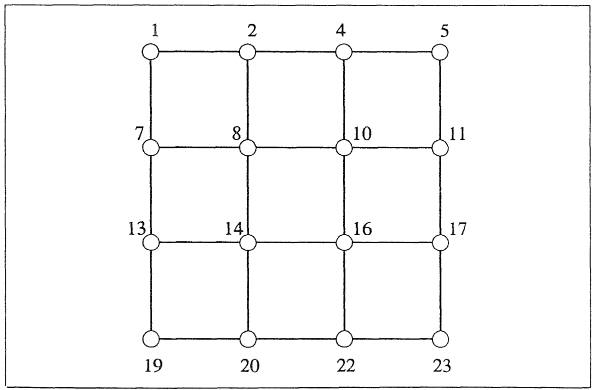

FIGURE 5. Embedding a 4 × 4 Mesh on *FN*

Claim 4: A complete bipartite graph $K_{n,n}$ is a subgraph of $FG(n)$.

Proof: We show how the complete bipartite graph can be embedded on $FG(n)$. We group the nodes v_i, where $i \neq 3k$ into two halves such that the lower half of the processor nodes are in one group and the upper half of the processors are in the other group. Each processor node in one group is adjacent to each processor node in the second group by Propositions 2 and 5.

Claim 5: An n-cube is a subgraph of $FG(3 \times 2^{n-1})$.

Proof: This follows immediately from the previous embedding proof. An example embedding of Q_3 (3-cube) is shown in Figure 6.

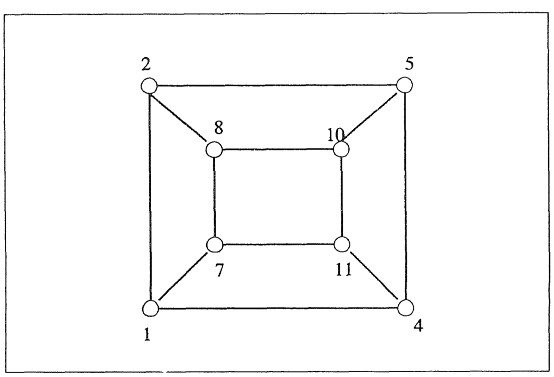

FIGURE 6. Embedding a Hypercube Q_3 on *FN*

7. IMPLEMENTATION OF DISTRIBUTED ALGORITHMS

In the previous section we showed how some of the common topologies can be mapped onto $FG(n)$. The algorithms that run on various topologies can be implemented on $FG(n)$ with minor modifications. Below, we show how a minimum weight spanning tree can be computed on *FN*.

Minimum Weight Spanning Tree

The problem is to find a spanning tree with minimum sum of edge weights in a given undirected, connected, weighted graph *G*, with *N* nodes. We show how this problem can be implemented efficiently on $FG(n)$. We implement Prim-Dijkstra's algorithm on $FG(n)$. A set *T* contains the set of nodes currently in the spanning tree, and a set *E* contains the set of edges currently in the spanning tree. We adapt the procedure outlined in [3], as follows:

Input: A graph *G* with *N* nodes and an adjacency matrix.

Output: A set of edges marked as belonging to the minimum weight spanning tree.

Step 1. $T \leftarrow \phi$. $E \leftarrow \phi$.

Step 2. Partition the nodes of *G* equally among the *n* processor nodes of $FG(n)$ so that each processor node is responsible for $\lceil N/n \rceil$ nodes.

Step 3. $T \leftarrow$ *vertex one* of G.

Step 4 Each processor examines its subset of nodes not in *T* and selects closest neighbor to *T* (*closest in terms of edge weight*).

Step 5. Processor P_1 finds the globally closest neighbor, say v_k.

Step 6. $T \leftarrow T \cup v_k$. $E \leftarrow E \cup edge(T, v_k)$.

Step 7. Processor P_1 broadcasts v_k to all processors.

Step 8. Each processor updates its closest neighbor information.

Repeat Steps 4 through 8 until all nodes have been included in *T*.

Steps 1, 2, and 3 require one time unit. Step 4 requires $O(\lceil N/n \rceil)$ time units in parallel. Step 5 requires $O(\lceil N/n \rceil)$ time units by processor one. Steps 6, 7, and 8 require one time unit. Steps 4 through 8 are repeated *N* times. The overall complexity of the algorithm is $O(N^2/n)$ The sequential algorithm takes $O(N^2)$; hence, this algorithm is optimal.

8. COMPARISON WITH OTHER ITERATIVE NETWORKS

We computed various structural properties of known iterative networks from *Path* to *Complete* networks of 35 nodes. Table 3 below shows these properties. Let

tot-deg = The total number of links in the network.
non-plan = smallest network size for which the network is nonplanar.
min-node = The node with the minimum number of links
min-deg = The degree of *min-node*.
max-node = The node with the maximum number of links.
max-deg = The degree of *max-node*.
inc-deg = Increase in degree with the addition of a node.
dia = The diameter of the network
f-dia = The fault fiameter of the network

inf = Disconnected network
$SG(n)$ = Stirling network of n nodes.
$PG(n)$ = Pascal network of n nodes.
$RG(n)$ = Rencontres network of n nodes.

TABLE 3. Comparison of Iterative Networks

Network	*tot-deg*	*non-plan*	*min-node*	*min-deg*	*max-node*	*max-deg*	*dia*	*f-dia*
Path	34	*inf*	1	1	2	2	34	*inf*
Stirling	169	8	1	2	31	17	6	9
Rencontres	166	7	34	2	2	18	3	3
Pascal	291	7	26	7	1	34	2	3
Fibonacci23	298	10	3	9	1	26	3	4
Fibonacci	397	7	3	11	1	34	2	3
Complete	595	5	1	34	1	34	1	2

The Path network has very low connectivity and is not fault-tolerant. The number of links in the Rencontres network, the Stirling network, and the Pascal network does not scale uniformly. These networks are not symmetric either. Fibonacci networks have too many links, making them prohibitively expensive.

In [5] it was shown that a *full ringed binary tree* with $2^k - 1$ nodes is a subgraph of $SG(2^k - 1)$ for $k \geq 2$, a *full ringed tree machine* of $3(n/4) - 2$ nodes when $n = 2^k - 1$ for $k \geq 3$ is contained in $SG(n)$ for any $\ell \leq k$, a *rectangular mesh* of size $2^\ell \times 2^{k-\ell}$ is embedded in a sub-network induced by the nodes 2^k through 2^{k+1} of $SG(n)$, and a *binary hypercube* is a homeomorphic subgraph of $SG(2^{t+1} - 1)$ for $t \geq 3$.

Embeddability of the Rencontres network and the Pascal network have not been studied extensively. However, in [4] it was shown that $RG(n)$ contains a Hamiltonian circuit of n nodes and the *Complete bipartite network* $K_{n,n}$ is a subgraph of $RG(2^n)$. In [7] it was shown that $PG(n)$ contains a *startree* for all $n \geq 1$, that $PG(n)$ contains a Hamiltonian circuit $[1, 2, \ldots, n-1, n, 1]$, and that $PG(n)$ contains $W_n - x$ (wheel of order n minus an edge).

In section 6 we showed that various popular topologies can be embedded onto *FN*. It is clear from the above that we need to be able to fine-tune a network design that has characteristics almost midway between the Path networks and the Complete networks.

9. CONCLUDING REMARKS

Fibonacci networks have many properties desirable in interconnection networks. They have a small diameter, high fault tolerance, rich connectivity, small fault diameter, simple and fast routing, etc. A major disadvantage of the network is its high coast because of the large number of links $(O(n^2))$. We have suggested several ways of reducing the number of links symmetrically so that the basic structure of the network is still maintained. This method of reduction has been shown to cause only constant factor loss of speedup (especially in routing). Broadcasting can be accomplished in constant time assuming that node v_1 has enough buffer space to queue messages. Yet another method of reduction which could be used is to prune links least used by the routing algorithm. Several basic algorithms can be mapped onto Fibonacci networks. We are currently working on embedding other interconnection networks on Fibonacci networks and improving the efficiency of some basic algorithms running on Fibonacci networks.

10. APPENDIX: TABLES A-1 AND A-2

Let

k_{num} = the number of nonzero entries in f^1 and f^2 in Table A-1.
$rk_{num}(j)$ = the number of nonzero entries in the first j entries in Table A-1.
rem = n (mod 24).

The expression for degree of a node v_k in $FG^{2,3}(n)$ is given by

$$deg(v_k) = k_{num} \times \left\lfloor \frac{n}{24} \right\rfloor + rk_{num}(rem).$$

It should be noted that Table A-1 can be used only for the construction of the lower triangle of the adjacency matrix. Therefore, Table A-1 is true *only* when $k < i$. Since the adjacency matrix is symmetric, the upper tirangle is just a copy of the lower triangle.

TABLE A-1. Connectivity in $FM^{\{2,3\}}$

i	$f_k^1(i) = (i^2 - 3i + 2 + 2k) \pmod 6$	$f_k^2(i) = (i^2 - 3i + 2 + 2k) \pmod 8$
1	$2k$	$2k$
2	$2k$	$2k$
3	$2 + 2k$	$2 + 2k$
4	$2k$	$2k - 2$
5	$2k$	$2k - 4$
6	$2 + 2k$	$4 + 2k$
7	$2k$	$2k - 2$
8	$2k$	$2 + 2k$
9	$2 + 2k$	$2k$
10	$2k$	$2k$
11	$2k$	$2 + 2k$
12	$2 + 2k$	$2k - 2$
13	$2k$	$2k - 4$
14	$2k$	$4 + 2k$
15	$2 + 2k$	$2k - 2$
16	$2k$	$2 + 2k$
17	$2k$	$2k$
18	$2 + 2k$	$2k$
19	$2k$	$2 + 2k$
20	$2k$	$2k - 2$
21	$2 + 2k$	$2k - 4$
22	$2k$	$4 + 2k$
23	$2k$	$2k - 2$
24	$2 + 2k$	$2 + 2k$

The degree of a node v_k increases as follows (see Table A-2) for every 24 nodes added to the network.

TABLE A-2. Increase in Degree of Nodes for Every 24 Nodes

rem	1	2	3	4	5	6	7	8	9	10	11	12
inc. in deg.	17	11	7	17	11	8	14	12	7	14	12	8
rem	13	14	15	16	17	18	19	20	21	22	23	24
inc. in deg.	14	12	8	14	10	9	14	11	11	13	11	11

ACKNOWLEDGMENT

The authors would like to thank Dr. Oren Patashnik for pointing out an error in an earlier version of the paper.

REFERENCES

1. A. Boals, A. Gupta, & N. Sherwani. "On Optimal Embeddings into Incomplete Hypercubes." *Proc. 5th Intl. Parallel Processing Symposium*, 1991.
2. F. Boesch & R. Tindell. "Circulants and Their Connectivities." *J. Graph Theory* (1984): 487-99.
3. S. K. Das & Narsingh Deo. "Stirling Networks and Their Properties." *Congressus Numerantium* **54** (1986):5-20.
4. S. K. Das & Narsingh Deo. "Rencontres Graphs: A Family of Bipartite Graphs." *The Fibonacci Quarterly* **25.3** (1987):250-62.
5. S. K. Das, Joydeep Ghosh, & Narsingh Deo. "Stirling Networks: A Versitile Combinatorial Topology for Multiprocessor Systems." *Discrete Applied Math.* **37/38** (1992):119-46.
6. Narsingh Deo. *Graph Theory with Applications to Engineering and Computer Science.* Englewood Cliffs, NJ: Prentice Hall, 1974.
7. Narsingh Deo & M. J. Quinn. "Pascal Graphs and Their Properties." *The Fibonacci Quarterly* **21.3** (1983):203-14.
8. Narsingh Deo & M. S. Krishnamoorthy. "Topelitz Networks and Their Properties." *IEEE Transactions on Circuits and Systems* **36.8** (1989):1089-92.
9. R. L. Graham, D. Knuth, & O. Patashnik. *Concrete Mathematics.* New York: Addison Wesley, 1989.
10. J. Hong, K. Melhorn, & A. Rosenberg. "Cost-Tradeoffs in Graph Embeddings." *Journal of the ACM* (1983):709-28.
11. W. Hsu. "Fibonacci Cubes—New Interconnection Topology." *IEEE Transactions on Paralel and Distributed Systems* **4.1** (1993):3-12.
12. J. C. Molluzzo. "Steinhaus Graphs." In *Theory and Applications of Graphs.* New York: Springer Verlag, 1978.
13. B. Monien & I. H. Sudborough. "Comparing Interconnection Networks." *Lecture Notes in Computer Science* **324** (1988):138-53.

AMS Classification Numbers: 05C75, 05C90, 68M10

❖❖❖