ELSEVIER

Parallel Computing 21 (1995) 1327–1338

PARALLEL COMPUTING

Short communication

Hyper-ring connection machines

Tom Altman [a], Yoshihide Igarashi [b,*], Koji Obokata [b]

[a] *Department of Computer Science, University of Colorado at Denver, Denver, CO 80217, USA*
[b] *Department of Computer Science, Gunma University, Kiryu 376, Japan*

Received 15 May 1993; revised 8 November 1994

Abstract

A graph $G=(V, E)$ is called a hyper-ring with N nodes (N-HR for short) if $V=\{0,\dots,N-1\}$ and $E=\{\{u, v\} \mid v-u$ modulo N is a power of $2\}$. We study constructions, properties, spanners of HRs, and embeddings into HRs. A hypercube with N nodes, a grid of size $a \times b$, and a complete binary tree with N nodes can be embedded as subgraphs into an N-HR. The stretch factors of three types of spanners given in this paper are at most $\lceil \log_2 N \rceil$, $2k-1$ for any $1 \le k \le \lceil \log_2 N \rceil$, and $2k-1$ for any $2 \le k \le \lceil \log_2 N \rceil - 1$, respectively. The numbers of edges of these types of spanners are $N-1$, at most $N\lceil (\log_2 N)/k \rceil$, and at most $N(\lceil \log_2 N \rceil - k)/(2k) + Nk$, respectively. Some of these spanners are superior in both stretch factors and numbers of edges to corresponding spanners for synchronizer γ of HRs.

Keywords: Hyper-ring; Distributed system; Network; Spanner; Synchronizer

1. Introduction

A number of topologies have been proposed for interconnecting processors in large scaled distributed systems. The hypercube (HC for short) is one of the most popular and efficient networks due to its topological richness (see e.g. [10]). In this paper we present hyper-rings (HRs for short), a multi-machine organization that is, in sense, a generalization of the standard hypercube architecture. Hyper-rings and their variations have appeared in literature under several different names, including an optimal broadcasting scheme [1,6], Han-Finkel's scheme [8] and binary jumping networks [7].

* Corresponding author. Email: igarashi@cs.gunma-u.ac.jp

SSDI 0167-8191(95)00022-4

The number of links between processors in an HR is roughly twice that of an HC with the same number of processors, but the proposed organization possesses a number of advantages over that of an HC. In particular, for any N we can construct an HR with N processors, whereas a hypercube must contain exactly 2^n processors for some positive n. The fault tolerance of HRs and the optimality of their information disseminating schemes have previously been studied. For detailed discussions and references, see [6–8]. Hereafter, we use a *node* for a processor and an *edge* for a link.

We first formally introduce hyper-rings, then study their properties, and finally present the construction of spanners of HRs. A recursive scheme for construction of HRs is given in Section 2. Embeddings of HCs, grids, and complete binary trees into HRs are shown in Section 3. Spanners of a network are closely related to synchronizers for an asynchronous distributed system. In Section 4, constructions of various spanners for HRs are presented.

2. HR construction

Let us first introduce the definition of an N-HR.

Definition 1. A graph $G = (V, E)$ is called a *hyper-ring* with N nodes (N-HR for short) if $V = \{0,1,\dots,N-1\}$ and $E = \{\{u, v\} \mid [v-u]_N \text{ is a power of } 2\}$, where $[m]_r$ is m modulo r.

The number of edges in HRs does not grow monotonically with the number of nodes (examine, for example, a 7-HR and an 8-HR that have 21 edges and 20 edges, respectively). It turns out, however, that HRs do possess an interesting recursive construction. An example of a 16-HR is shown in Fig. 1. In order to study the recursive structure of HRs, we will need to examine the structure of N, itself. Unless stated otherwise, all logarithms in this paper are base 2.

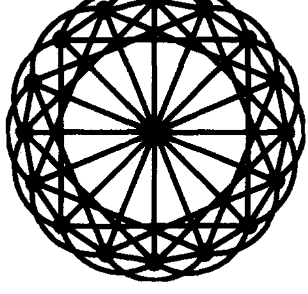

Fig. 1. A 16-HR.

First, let us observe a nonrecursive construction of N-HRs. The following procedure is straightforward and needs no further explanation.

for $k := 0$ **to** $N-1$ **do**
  for $i := 0$ **to** $\lfloor \log N \rfloor$ **do**
    if there is no edge between node k and node $[k+2^i]_N$
    then connect node k to node $[k+2^i]_N$

Although the above is an effective and correct procedure, it gives us no insight about the actual number of edges placed in the HR. In addition, no obvious information about recursive structure relates HRs constructed in this fashion.

Assume that an N-HR has already been constructed. Suppose we wish to construct a $2N$-HR from the given N-HR. The following procedure takes as input an N-HR_0 and returns a $2N$-HR.

(1) make a duplicate copy of N-HR_0, call it N-HR_1;
(2) relabel all nodes in N-HR_0 from i to $2i$ $(0 \le i \le N-1)$;
(3) relabel all nodes in N-HR_1 from i to $2i+1$ $(0 \le i \le N-1)$;
(4) for each i $(0 \le i \le 2N-1)$ connect node i and node j if $j \equiv i+1$ modulo $2N$;

The above procedure is called the doubling construction procedure, and it suggests a recurrence relation for the number of edges in HRs. Specifically, if $E(N)$ denotes the number of edges in an N-HR, then

$$E(2N) = 2E(N) + 2N. \quad (1)$$

Definition 2. Let N be a positive integer. The binarity of N, written $bin(N)$, is a function whose range is between 0 and $\lceil \log N \rceil$, that returns the largest integer k, such that N is divisible by 2^k. (In other words, $bin(N)$ tells us how many consecutive rightmost bits of a binary representation of N are zero.)

Observe that any positive integer N may be represented as the following product

$$N = 2^{bin(N)} M, \quad (2)$$

where M is an odd number.

We consider the problem how we determine the number of edges of an N-HR. From Eq. (2) the problem may be reduced to examination of M-HRs, where M is an odd positive integer in Eq. (2). The nonrecursive construction of N-HRs may be used for construction of such M-HRs, with a slight modification. Observe that for odd M of the form 2^k+1, we can construct the M-HR by using the nonrecursive construction procedure that ignores the last iteration of the inner **for** loop. In this case the last iteration of the inner **for** loop produces the same connections as the connections produced in the first iteration of the inner **for** loop. That is, the inner **for** loop should be iterated one less time (i.e. from 0 to $\log(M-1)-1$) to produce the M-HR. For the construction of an arbitrary N-HR one simply needs to compute $k = bin(N)$, determine the odd M in Eq. (2), construct the appropriate M-HR, and recursively apply the doubling construction procedure exactly k times.

We are now ready to give the exact estimate for $E(N)$.

Theorem 1. *The number of edges in an N-HR, $N \geq 2$, is given by the following formula*:

$$E(N) = \begin{cases} N\left(\log N - \dfrac{1}{2}\right) & \text{if the Hamming weight of } N \text{ is } 1 \\ N\lfloor \log N \rfloor & \text{if the Hamming weight of } N \text{ is } 2 \\ N\lceil \log N \rceil & \text{if the Hamming weight of } N \text{ is } 3 \text{ or more.} \end{cases}$$

Proof. If the Hamming weight of N is 1, then $N = 2^n$, $n \geq 1$. Observe that during the execution of the nonrecursive construction procedure of N-HR, no multiple edges could possibly be constructed until the very last iteration of the inner **for** loop. During that iteration, exactly half of the edges would be eliminated. It follows that the number of (nonreplicated) edges when N is a power of 2, is $N \log N - N/2 = N(\log N - \frac{1}{2})$.

A Hamming weight of 2 implies that N can be decomposed into a sum of 2^p and 2^q. Moreover, this decomposition must be unique. Because $N = 2^p + 2^q$, for each pair of nodes that were 2^p and 2^q positions apart, exactly one extra edge is eliminated due to duplication in the nonrecursive construction procedure of N-HRs. Hence, the number of edges is $N\lceil \log N \rceil - N$, in this case, is equal to $N\lfloor \log N \rfloor$.

In the case where a Hamming weight is 3 or more, the nonrecursive construction procedure of HRs does not attempt to create any duplicated edge even if the **If** condition is removed. Hence, the number of edges in this case is $N\lceil \log N \rceil$. □

3. Embeddings into HRs

Hypercubes

An n-dimensional *hypercube* (2^n-HC for short) is a graph with $N = 2^n$ nodes labeled by 2^n binary numbers from 0 to $2^n - 1$, where there exists an edge between any two nodes if and only if the binary representations of their labels differ by exactly one bit. The HC can be used to simulate many networks since it contains or nearly contains those networks as subgraphs. This is one of the main reasons why the HC is a powerful network for parallel computation. For the number of edges of an N-HC, the recurrence equation corresponding to Eq. (1) becomes

$$F(2N) = 2F(N) + N,$$

where $N = 2^n$ and $F(N)$ denotes the number of edges in the N-HC.

From the structures of N-HCs and N-HRs, it is clear that for $N = 2^n$, the N-HR contains N-HC as a subgraph, making the HR an even more powerful

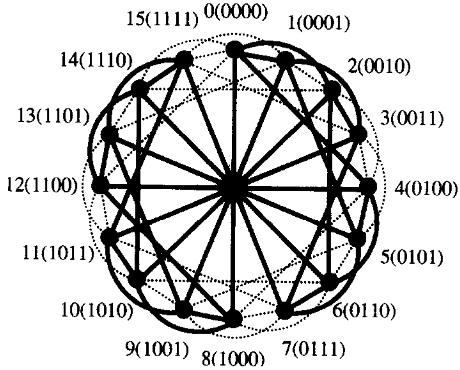

Fig. 2. An embedding of a 16-HC into a 16-HR.

network for parallel computation. Note that an N-HC is embeddable into any Q-HR with stretch factor of one as long as $Q \geq N$. An embedding of a 16-HC into a 16-HR is shown in Fig. 2.

1-dimensional grid (size $1 \times k$)

A 1-dimensional grid of size $1 \times k$ can be embedded as a subgraph into a k-HR in an obvious way. That is, there are no wasted nodes in the HR. Note that for HCs, if k is not a power of 2, e.g., $k = 2^m + 1$, then the HC will need $2^{m+1} = 2k - 2$ nodes.

2-dimensional grid (size $a \times b$)

To embed a grid of size $a \times b$ as a subgraph into a HR we need

$$\min(a \times 2^{\lceil \log b \rceil}, b \times 2^{\lceil \log a \rceil}) \text{ nodes.}$$

A possible enumeration of the nodes for the HR for a 5×6 grid is shown in Fig. 3.

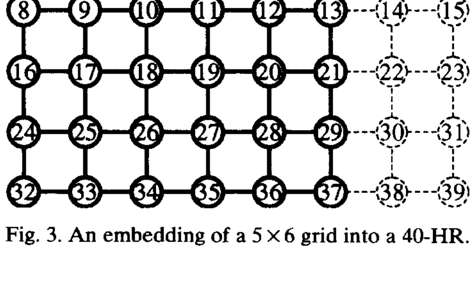

Fig. 3. An embedding of a 5×6 grid into a 40-HR.

3-dimensional grid (size $a \times b \times c$)

The number of nodes required in the HR to embed a $a \times b \times c$ grid is

$$\min(a \times 2^{\lceil \log b \rceil} \times 2^{\lceil \log c \rceil},\ b \times 2^{\lceil \log a \rceil} \times 2^{\lceil \log c \rceil},\ c \times 2^{\lceil \log a \rceil} \times 2^{\lceil \log b \rceil}).$$

In general, when embedding k-dimensional grids into HRs we have one extra degree of freedom, i.e. the size of one of dimensions can be the same as the original grid which is not necessarily a power of 2. It follows that more nodes are wasted when the grid embedding is done in HCs than in HRs (unless each of the grid dimensions is a power of 2).

Complete binary tree (size $2^n - 1$)

It is known that a complete binary tree with $2^n - 1$ nodes cannot be embedded as a subgraph into an 2^n-HC [10]. We show that it can be embedded as a subgraph into a $(2^n - 1)$-HR.

The address of each node of the complete binary tree is given by the following procedure:

(1) The address of the root is 0.
(2) If address of a node at depth k is i, then the addresses of its left child and its right child are $i + 2^k$ and $i + 2^{k+1}$, respectively.

Theorem 2. *Suppose that the addresses of each node of the complete binary tree with $2^n - 1$ nodes is given by the above procedure. Then the addresses of all nodes in the complete binary tree are distinct. Furthermore, the address of any node at depth k $(0 \le k \le n - 1)$ is between $2^k - 1$ and $2^{k+1} - 2$.*

Proof. From the procedure above, the address of any node is different from the address of any ancestor of the node. It is also immediate that the address of any node in its left subtree is different from the addresses of any node in its right subtree. Hence, all addresses in the complete binary tree are distinct.

The latter assertion of the theorem is proved by induction on k. For $k = 0$, the assertion obviously holds. Suppose that up to depth k the assertion holds. From the induction hypothesis, the range of the addresses in depth $k + 1$ is between $2^k - 1 + 2^k = 2^{k+1} - 2$ and $2^{k+1} - 2 + 2^{k+1} = 2^{k+2} - 2$. □

The address of a node in a complete binary tree is different from the address of its parent by a power of 2. From this fact and Theorem 2, a complete binary tree with $2^n - 1$ nodes can be embedded as a subgraph into a $(2^n - 1)$-HR. The embedding of a complete binary tree into the HR is shown in Fig. 4.

4. Spanners

A distributed computing network is essentially asynchronous. Asynchronous algorithms are in many cases substantially inferior in terms of their time and

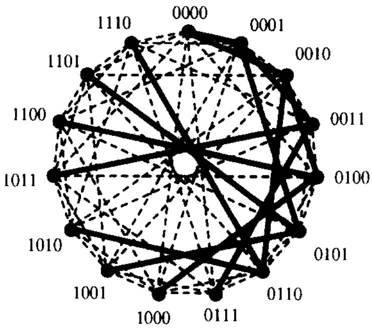

Fig. 4. An embedding of a complete binary tree with 15 nodes into a 15-HR.

communication complexity to corresponding synchronous algorithms [3,13]. It is frequently required for the nodes to obtain some common notion of time or some explicit mechanism of synchronous behavior in the asynchronous system.

Awerbuch introduced a notion of synchronizers [3]. If a synchronizer is available, the user will be allowed to write an algorithm as if it were run in a synchronous system. He proposed the construction of three types of synchronizers α, β, and γ. Since then, the construction and properties of various synchronizers have been studied extensively [2,4,5,9,12–15].

Let $T(\nu)$ and $C(\nu)$ be the time and communication requirements added by a synchronizer ν for each time step of the synchronous algorithm. Awerbuch showed that for synchronizers α, β and γ in network $G=(V, E)$, $T(\alpha)=O(1)$, $C(\alpha)=O(|E|)$, $T(\beta)=O(|V|)$, $C(\beta)=O(|E|)$, $T(\gamma)=O(\log_k|V|)$ and $C(\gamma)=O(k|V|)$, where k is a parameter in the range $2 \le k < |V|$. Given a connected simple graph $G=(V, E)$, subgraph $G'=(V, E')$ is called a t-spanner of G if for every u and v in V,

$$\frac{dist(u, v, G')}{dist(u, v, G)} \le t,$$

where $dist(u, v, H)$ denotes the distance from u to v in H. We refer to t as the stretch factor of G' against G. There is a close connection between synchronizers and the structure of t-spanners on a network.

Theorem 3 [12].
(1) *If the network G has no t-spanners with at most m edges, then every synchronizer* ν *for G requires either* $T(\nu) \ge t+1$ *or* $C(\nu) \ge tm+1$.
(2) *If the network G has a t-spanner with m edges, then it has a synchronizer* δ *with* $T(\delta)=O(t)$ *and* $C(\delta)=O(tm)$.

From the above theorem, it is important for us to determine whether there exists a family of spanners with small stretch factors and small numbers of edges for a given network family. Awerbuch proved that for some network families the best possible improvements concerning both stretch factors and numbers of edges are within constant factors from synchronizer γ [3]. However, this is not true for some networks. In fact, Peleg and Ullman constructed a 3-spanner with a linear number of edges for the HC (i.e. with $O(2^n)$ edges for the 2^n-HC). The 3-spanner is superior to synchronizer γ of the hypercube. Another example is the class of bounded-degree networks. For any bounded-degree network, the synchronizer α is optimal in both time and communication complexities [12].

4.1. Sparse spanners

For a family of graphs $G_i = (V_i, E_i)(i = 1, 2, \ldots)$, spanners of G_i's with $O(|V_i|)$ edges, are called sparse. Any spanning tree is a sparse spanner, since it has exactly $|V| - 1$ edges. Let us consider the breadth-first-search tree (BFST for short) of an HR. Then we have the next lemma.

Lemma 1. *The depth of the BFST of an N-HR is at most* $\lceil \log N/2 \rceil$.

Proof. Without loss of generality we may assume that node 0 is the root of the BFST. For $N \leq 8$ the assertion of the lemma obviously holds. Let $n = \lceil \log N \rceil$, and let $D(N)$ denote the depth of the BFST of N-HR. Since the depth of any node between 2^{n-2} and $3 \times 2^{n-2}$ is at most $D(2^{n-2}) + 1$, we have the following recurrence inequalities:

$$D(N) \leq D(2^n) \leq D(2^{n-2}) + 1,$$

$$D(2^2) = 1 \quad \text{and} \quad D(2^3) = 2.$$

From the above inequalities and the values of $D(2^2)$ and $D(2^3)$ we have $D(N) \leq D(2^n) \leq \lceil n/2 \rceil$. □

Theorem 4. *The stretch factor of the BFST of an N-HR is at most* $\lceil \log N \rceil$.

Proof. Let $n = \lceil \log N \rceil$. If n is even, then from Lemma 1 its stretch factor is at most n. Suppose that n is odd. Consider a 2^n-HR and two subgraphs A_n and B_n of the 2^n-HR, where A_n consists of all even numbered nodes and their induced edges, and B_n consists of all odd numbered nodes and their induced edges. Then from Lemma 1 the distance between any pair of nodes in A_n or any pair of nodes in B_n is at most $n - 1$. The distance between any node in A_n and any node in B_n is at most $2(n-1)/2 + 1 = n$. Since $D(N) \leq D(2^n)$, the theorem holds. □

The stretch factors of the BFSTs of N-HRs up to $N = 130$ are shown in Table 1. We note that for many values of N the bound given in Theorem 3 is tight and that for some values of N the stretch factors are less than $\lceil \log N \rceil$ by 2.

4.2. Other constructions

Let N-HR(k) be a spanner of an N-HR defined as follows: N-HR$(k) = (V, E)$, where $V = \{0, 1, \ldots, N-1\}$ and $E = \{\{u, v\} \mid [u - v]_N$ is 2^i for some i such that i is

Table 1
Stretch factors of BFST's of N-HRs

a \ b	1	2	3	4	5	6	7	8	9	10
0	1	1	2	2	2	3	2	3	3	3
1	4	4	4	3	4	4	4	4	4	4
2	4	5	4	5	4	5	4	4	4	5
3	4	5	5	5	4	5	4	5	5	5
4	5	6	4	6	6	6	6	6	6	6
5	6	6	6	6	6	5	6	6	6	6
6	6	6	6	6	6	6	6	6	6	6
7	6	6	6	6	6	6	6	6	6	6
8	6	6	6	6	6	6	6	7	6	7
9	6	6	6	7	6	7	6	7	6	6
10	6	7	6	7	6	7	6	6	6	6
11	6	6	6	6	6	6	6	7	6	7
12	6	7	6	6	6	7	6	7	7	7

The value in entry (a, b) denotes the stretch factor of the BFST of $(10a + b)$-HR (i.e., $N = 10a + b$)

0 modulo k, $0 \le i \le \lceil \log N \rceil - 1\}$. Then the stretch factor of an N-HR(k) is obviously at most 2^{k-1}, and the number of edges of this spanner is at most $N\lceil(\log N)/k\rceil$.

For $k = 1$, this spanner is the N-HR itself and the spanner for synchronizer α [3]. For $k = 2$, it is a 2-spanner with at most $N\lceil(\log N)/2\rceil$ edges. At present we do not know whether there exists a 2-spanner of an N-HR with a smaller number of edges than the number of edges in N-HR(2).

We next consider spanner construction by partition. Let N-HR $= (V, E)$, and let $K = 2^k$, $2 \le k \le \lceil \log N \rceil - 1$. In the rest of this subsection we assume that N is a multiple of K. For each $i(0 \le i \le K-1)$ let $G_i = (V_i, E_i)$, where $V_i = \{v \mid 0 \le v \le N-1$ and $v = i$ modulo $K\}$ and E_i is the set of all induced edges by V_i in E of the N-HR. Then G_i is isomorphic to an N/K-HR, where each node u in the N/K-HR corresponds to node $uK + i$ in G_i. As an example, we show 4 partitioned subgraphs $G_i(0 \le i \le 3)$ of a 20-HR in Fig. 5, where $K = 2^2$.

Let $E' = E - (E_1 \cup E_2 \cup \cdots E_{K-1})$, and let $G' = (V, E')$. Then G' is a spanner of N-HR. For any edge $\{u, v\} \in E_1 \cup E_2 \cup \cdots \cup E_{K-1}$, $dist(u, v, G') \le 2(k-1) + 1 = 2k - 1$. Therefore, the stretch factor of G' is at most $2k - 1$ and the number of edges of G' is at most $N(\lceil \log N \rceil - k)/2^k + Nk$. Spanners constructed in this way for the 20-HR are shown in Fig. 6.

The product of t and m of a t-spanner with m edges is an interesting measure of the spanner. The question whether for an N-HR there exists a t-spanner with m edges such that $mt = O(N \log N)$ has an affirmative answer. Specifically, if we choose $k = \log \log N$, for the spanner constructed by partition, the product of the stretch factor and the number of edges will be $O(N(\log \log N)^2)$.

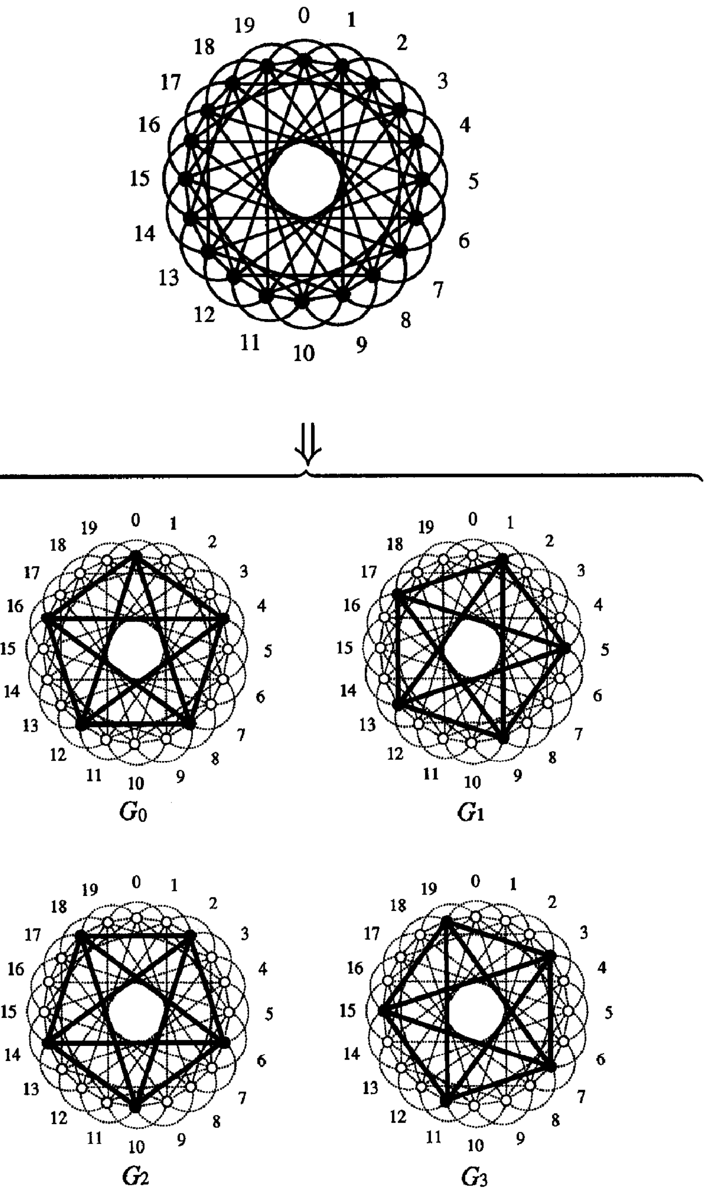

Fig. 5. Four partitioned subgraphs of a 20-HR.

4.3. *Spanners as synchronizer γ*

As described in [3], synchronizer γ is a combination of the two synchronizers α and β. The spanners for synchronizer α and β of a network are the network itself and the BFST of the network, respectively. Awerbuch demonstrated that spanners

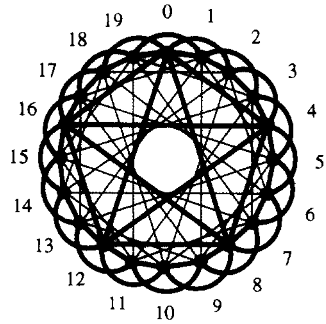

Fig. 6. A 3-spanner of a 20-HR.

for synchronizer γ are optimal for some networks in the sense of time and communication complexity within a constant factor [3]. However, as shown in [12], for some networks there exist better spanners than spanners for synchronizer γ. We have implemented Awerbuch's spanner construction algorithm for HRs and observed that many spanners constructed in the ways in this section are superior to spanners of HRs for synchronizer γ.

5. Concluding remarks

We have shown construction procedures for HRs, the exact evaluation for the number of edges in an N-HR, embeddings of HCs, grid graphs, and complete binary trees into HRs, and various spanners of HRs. All embeddings in this paper are subgraphs of HRs.

We do not know at present whether spanners given in Section 4.2 are optimal within a constant factor in the sense of the smallest value for the product of the stretch factor and the number of edges for an appropriate value of k. We are interested in deriving a nontrivial lower bound on the product of the stretch factor and the number of edges.

References

[1] N. Alon, A. Barak and U. Mauber, On disseminating information reliably without broadcasting, in: *7th Int. Conf. on Distributed Computing Systems* (1987) 74–81.

[2] H. Attiya and M. Mavronicolas, Efficiency of semi-synchronous versus asynchronous networks, Center for Research in Computing Technology, Harvard University, TR-21-90, 1990.

[3] B. Awerbuch, Complexity of network synchronization, *J. ACM* 32 (1985) 804–823.

[4] M. Beck, T.K. Srikanth and S. Toueg, Implementation issues in clock synchronization, in: *Fault-Tolerant Distributed Computing, Lecture Notes in Computer Science* 448 (Springer-Verlag, New York, 1987) 95–107.

[5] A. Fekete, N. Lynch and L. Shrira, A modular proof of correctness for a network synchronizer, in: *Distributed Algorithms, Lecture Notes in Computer Science* 312 (Springer-Verlag, New York, 1987) 219–256.

[6] Y. Han and R. Finkel, An optimal scheme for disseminating information, in: *Proc. 1988 Int. Conf. on Parallel Processing*, St. Charles (1988) 198–203.
[7] Y. Han, Y. Igarashi, K. Kanai and K. Miura, Fault-tolerant broadcasting in binary jumping networks, in: *Algorithms and Computation, ISAAC '92*, Nagoya, *Lecture Notes in Computer Science* 650 (Springer-Verlag, New York, 1992) 145–154.
[8] Y. Igarashi, K. Kanai, K. Miura and S. Osawa, Optimal schemes for disseminating information and their fault tolerance *IEICE Trans. Inf. and Syst.* E75-D (1992) 22–29.
[9] K.B. Lakshmanan and K. Thulasiraman, On the use of synchronizers for asynchronous communication networks, in: *Distributed Algorithms*, *Lecture Notes in Computer Science* 312 (Springer-Verlag, New York, 1987) 257–277.
[10] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures* (Morgan Kaufman, San Mateo, 1992).
[11] D. Peleg and A.A. Schäffer, Graph spanners, *J. Graph Theory* 13 (1989) 99–116.
[12] D. Peleg and J.D. Ullman, An optimal synchronizer for the hypercube, *SIAM J. Comput.* 18 (1989) 740–747.
[13] L. Shabtay and A. Segall, Message delaying synchronizers, in: *Distributed Algorithms*, *Lecture Notes in Computer Science* 579 (Springer-Verlag, New York, 1991) 309–318.
[14] B. Simons, J. Welch and N. Lynch, An overview of clock synchronization, in: *Distributed Algorithms*, *Lecture Notes in Computer Science* 312 (Springer-Verlag, New York, 1987) 84–96.
[15] G. Tel, *Topics in Distributed Algorithms* (Cambridge Univ. Press, Cambridge, 1991).
[16] S. Ueno and M. Yamazaki, Graph spanners and connectivity, in: *Algorithms and Computation, ISAAC '92*, Nagoya, *Lecture Notes in Computer Science* 650 (Springer-Verlag, New York, 1992) 126–134.