Int. Computer Symposium, Dec. 15-17, 2004, Taipei, Taiwan.

Reducing the height of independent spanning trees in chordal rings

Jinn–Shyong Yang[1,3] Yue–Li Wang[1,*] Shyue-Ming Tang[2]

[1]Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan, Republic of China.
[2]Department of Psychology,Fu-Hsing-Kang College, Taipei, Taiwan , Republic of China.
[3] Department of Information Management, National Taipei College of Business, Taipei, Taiwan, ROC

Abstract

This paper is concerned with a particular family of regular 4-connected graphs, called chordal rings. Chordal rings are a variation of ring networks. By adding extra two links (or chords) at each vertex in a ring network, the reliability and fault-tolerance of the network are enhanced. Two spanning trees on a graph are said to be *independent* if they are rooted at the same vertex, say r, and for each vertex $v \neq r$, the two paths from r to v, one path in each tree, are internally disjoint. A set of spanning trees on a given graph is said to be independent if they are pairwise independent. In 1999, Y. Iwasaki et al. proposed a linear time algorithm to find four independent spanning trees on a chordal ring. In this paper, we shall give new algorithms to generate four independent spanning trees with reduced height in each tree.

Keyword: chordal rings, interconnection networks, fault-tolerant broadcasting, independent spanning trees, internally disjoint path.

1 Introduction

Chordal rings are a variation of ring networks. By adding extra two links (or chords) at each vertex in a ring network, the reliability and fault-tolerance of the network are enhanced [1, 3, 7, 8]. A number of problems on chordal rings (or called distributed loop networks) have been studied in the past two decades, including the diameter problem [1], the shortest paths problem [4], the routing and fault-tolerant routing problem [11, 12, 13, 14]. A chordal ring $CR(N, d)$ is a graph with its vertex set $V = \{0, 1, \ldots, N-1\}$ and edge set $E = \{(u, v) | [v-u]_N = 1 \text{ or } d\}$, where $[x]_N$ denotes x modulo N. To ensure every vertex has four adjacent vertices, we assume that d is less than $N/2$. An example of chordal ring for $N = 14$ and $d = 4$ is shown in Figure 1.

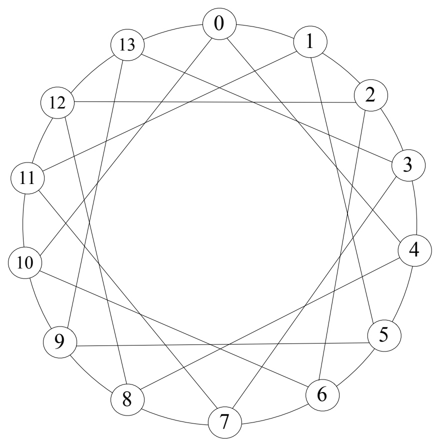

Figure 1: $CR(14, 4)$ chordal rings.

Two paths in a graph are *internally disjoint* if they have no common vertex except the two end vertices. A *spanning tree* of a graph G is a subgraph

*All correspondence should be addressed to Professor Yue–Li Wang, Department of Information Management, National Taiwan University of Science and Technology, 43, Section 4, Kee-Lung Road, Taipei, Taiwan, Republic of China (Phone: 886–02–7376768, Fax: 886–02–7376777, Email: ylwang@cs.ntust.edu.tw).

of G that contains all vertices in G and forms a tree. Two spanning trees of G are said to be *independent* if they are rooted at the same vertex, say r, and for each vertex $v \neq r$, the two paths from r to v, one path in each tree, are internally disjoint. A set of spanning trees of a graph is independent if they are pairwise independent. For example, a set of four independent spanning trees of $CR(14, 4)$ is shown in Figure 2.

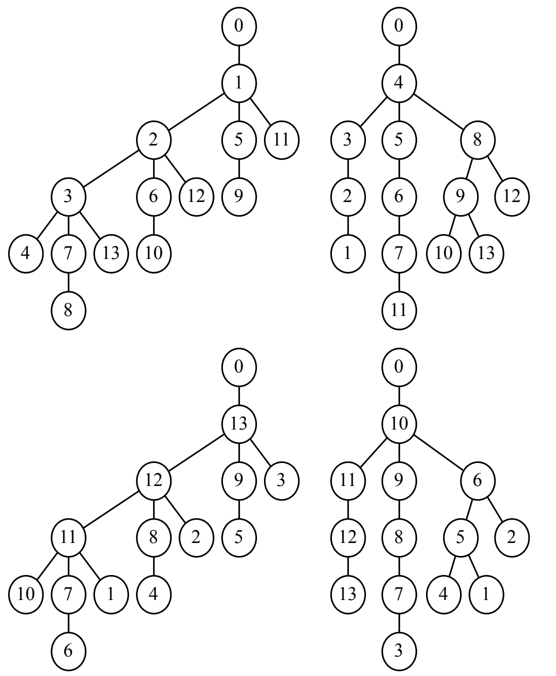

Figure 2: A set of independent spanning trees on $CR(14, 4)$.

The study of finding independent spanning trees has applications on fault-tolerant broadcasting protocol [2, 9]. The fault tolerance can be achieved by sending k copies of a message along k independent spanning trees rooted at the source node. If the source node is faultless, this scheme can tolerate up to $k-1$ faulty nodes.

In [9], Itai and Rodeh gave a linear time algorithm for finding two independent spanning trees rooted at an arbitrary vertex in a biconnected graph. In [5], Cheriyan and Maheshwari showed that, for any 3-connected graph $G = (V, E)$ and for any vertex r in G, three independent spanning trees rooted at r can be found in $O(|V||E|)$ time. In [15], Zehavi and Itai conjectured that any k-connected graph has k independent spanning trees rooted at an arbitrary vertex r. Recently, Curran presented an $O(|V|^3)$ time algorithm for finding four independent spanning trees rooted at any given vertex in a 4-connected graph [6]. This result has contribution to Zehavi and Itai's conjecture. However, the conjecture is still open for arbitrary k-connected graphs with $k \geq 5$. Although chordal rings discussed here are all 4-connected, efficient algorithms for solving the independent spanning trees problem in chordal rings are still valuable.

In [10], Iwasaki et al. gave a linear time algorithm to solve the independent spanning trees problem in a chordal ring. In Figure 2, based on their algorithm, four spanning trees T_1, T_2, T_3 and T_4 rooted at vertex 0 in $CR(14, 4)$ are constructed. Following the definition of independent spanning trees, for every vertex $v \neq 0$, the four paths from 0 to v in T_1, T_2, T_3 and T_4 are internally disjoint (or vertex-disjoint).

Let $d_G(u, v)$ denote the *distance* between vertices u and v in G. The *height* of a spanning tree T rooted at vertex r, denoted by $height(T)$, is the maximum distance of the paths from r to any other vertex in T, i.e., $height(T) = max\{d_T(r, v) | v \neq r\}$. For example, the heights of independent spanning trees T_i ($i = 1, 2, 3, 4$) shown in Figure 2 are all five. In this paper, we focus our efforts on the height of independent spanning trees. Obviously, the performance of a broadcasting protocol can be improved by reducing the height of a spanning tree rooted at the source node. We shall design new algorithms to generate four independent spanning trees with reduced height in each tree.

The remaining part of this paper is organized as follows. In Section 2, a linear time algorithm is proposed to generate height-reduced independent spanning trees rooted at one vertex in a chordal ring. In Section 3, we solve the problem for a special class of chordal rings, i.e., $CR(N, d)$ where $[N]_d = 0$. In Section 4, we prove the correctness of our algorithms. Section 5 contains our concluding remarks.

2 A New Algorithm for Generating Independent Spanning Trees

Chordal rings are vertex-symmetric [7]. Without loss of generality, we simply consider independent spanning trees rooted at vertex 0 of a chordal ring. Let T_1, T_2, T_3 and T_4 denote the four spanning trees. Since the four adjacent vertices of vertex v in $CR(N, d)$ are $[v+1]_N$, $[v-1]_N$, $[v+d]_N$ and $[v-d]_N$, vertices $1, d, N-1$ and $N-d$ can be assigned as the only child of the root in T_1, T_2, T_3 and T_4 , respectively. The first algorithm we proposed can generate four independent spanning trees rooted at vertex 0 in $CR(N, d)$, where $2 \leq d < N/2$. The algorithm contains three phases. At the first phase, we generate

Procedure Gen_T1(N, d)
begin
1. Calculate the $span\_column$.
   Let $node = d \times \lfloor (N-(d-1))/d \rfloor - 1$.
   Let $span\_column = [node - N]_d$.
   If $span\_column < d-1$ then
     $do\_move = True$
2. For $i = 1$ to $d-1$ do
   Set parent of vertex i to $i-1$.
3. For $i = 1$ to $d-1$ do
   If $i > \lfloor d/2 \rfloor$ then
     Set parent of vertex $[i-d]_N$ to $[i-d]_N$-1.
   Else // $i \leq \lfloor d/2 \rfloor$
     Set parent of vertex $[i-d]_N$ to i.
4. For $i = 1$ to $d-1$ do
   Let $x = i$.
   While $x \leq N-2d$ do
     If $(d-i) > span\_column$ then
       If $do\_move{=}True$ and $[N-(x+d)]_d{=}0$
       and $\lfloor (N-(x+d))/d \rfloor \leq \lfloor N/2d \rfloor$
         Set parent of vertex $x{+}d$ to $x{+}d{+}1$.
       Else
         Set parent of vertex $x+d$ to x.
     Else // $(d-i) \leq span\_columnd$
       Set parent of vertex $x+d$ to $x+2d$.
     Let $x = x+d$.
5. For $i = 1$ to $\lfloor (N-d)/d \rfloor$ do
   Let $x = i \times d$.
   If $do\_move = True$ and $i \leq \lfloor N/2d \rfloor$ then
     Set parent of vertex x to $x+1$.
   Else
     Set parent of vertex x to $x-1$.

end.

Procedure Gen_T2(N, d)
begin
1. Calculate the $span\_column$.
   Let $node = d \times \lfloor (N-(d-1))/d \rfloor - 1$.
   Let $span\_column = [node - N]_d$.
   If $span\_column < d-1$ then
   $do\_move = True$
2. For $i = 1$ to $\lfloor (N-d)/d \rfloor$ do
   Let $x = i \times d$.
   Set parent of vertex x to $x-d$.
3. For $i = d$ to $N-d-1$ and $[i+1]_d \neq 0$ do
   If $d-[i-1]_d > span\_column$ then
     If $do\_move{=}True$ and $[N-(i{+}1)]_d{=}0$
     and $\lfloor (N-(i+1))/d \rfloor \leq \lfloor N/2d \rfloor$
       Set parent of vertex $i+1$ to $i{+}1{-}d$.
     Else
       Set parent of vertex $i+1$ to i.
   Else // $d-[i-1]_d \leq span\_column$
     Set parent of vertex $i+1$ to $i+2$.
   Let $x = x+d$.
4. For $i = 1$ to $d-1$ do
   If $i \leq \lfloor (d-1)/2 \rfloor$ then
     Set parent of vertex i to $i+d$.
   Else // $i > \lfloor (d-1)/2 \rfloor$
     Set parent of vertex i to $i+1$.
5. For $i = 1$ to $d-1$ do
   If $i > \lfloor d/2 \rfloor$ then
     Set parent of vertex $[i-d]_N$ to i.
   Else // $i \leq \lfloor d/2 \rfloor$
     Set parent of vertex $[i-d]_N$ to
     $[i-d]_N - d$.

end.

T_1 using Procedure **Gen_T1**. Then, we generate T_2 using Procedure **Gen_T2**. We generate T_3 and T_4 from T_1 and T_2 directly by replacing the label of each non-root vertex v with $N-v$.

Then, we describe our algorithm for generating four independent spanning trees rooted at vertex 0 in $CR(N, d)$.

Algorithm IST_CR
Input : $CR(N, d)$
Output : T_1,T_2,T_3 and T_4
begin
1. Call **Procedure Gen_T1**(N, d).
2. Call **Procedure Gen_T2**(N, d).
3. Generate T_3 from T_1 by replacing the label of each non-root vertex v in T_1 with $N-v$.
4. Generate T_4 from T_2 by replacing the label of each non-root vertex v in T_2 with $N-v$.

end.

For example, we generate T_1, T_2, T_3 and T_4 on $CR(14, 4)$ as shown in Figure 3. Notice that the height of each tree in Figure 3 is reduced from five to four by comparing with the corresponding tree in Figure 2. Constructing the four independent spanning trees also takes linear time, as did in [10].

We are now at a position to compare the results of our algorithms with Iwasaki's algorithms. Using Iwasaki's algorithms, the height of each spanning tree can be expressed by a simple formula, i.e., $height(T_i) = d + \lfloor (N-2d)/d \rfloor$, where $i = 1, 2, 3, 4$. Taking a look at our algorithm, in procedures **Gen_T1**(N, d) and **Gen_T2**(N, d), variable $span\_column$ plays an important role to determine the height of independent spanning trees generated by Algorithm **IST_CR**. For the sake of conciseness, we omit the detail analysis. In case of $span\_column = \lfloor d/2 \rfloor$, the height of independent spanning trees is

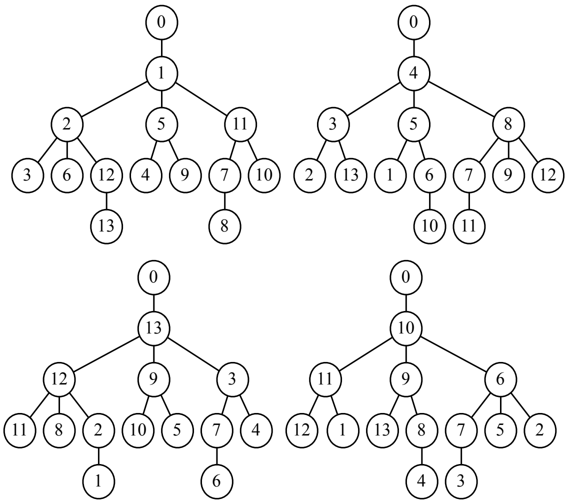

Figure 3: T_1, T_2, T_3 and T_4 on $CR(14, 4)$.

$d - \lfloor d/2 \rfloor + \lfloor (N-2d)/d \rfloor + 1$. In the best situation ($\lfloor N/2d \rfloor > 1$), the heights of T_1 and T_3 can be further reduced to $d - \lfloor d/2 \rfloor + \lfloor (N-2d)/d \rfloor$. In the worst case, the height of independent spanning trees can not be reduced any more by using our algorithms. For example, the height of independent spanning trees on $CR(14, 2)$ is $2+\lfloor (14-2\times 2)/2 \rfloor = 7$ by using Iwasaki's algorithms. This height can not be reduced. Consequently, we have the following theorem.

Theorem 1 *Algorithm* **IST_CR** *can reduce the height of independent spanning trees in $RC(N, d)$ to an extent of $\lfloor d/2 \rfloor$ by comparing with Iwasaki's algorithms.*

3 Another Algorithm for Constructing Independent Spanning Trees

In this section, we propose another algorithm to generate independent spanning trees for a special class of chordal rings. That is, this algorithm is designed for chordal rings $CR(N, d)$ where N is dividable by d (i.e., $[N]_d = 0$) and $d > 2$. For this class of chordal rings, Algorithm **IST_CR** does not help in independent spanning trees problem. We give the algorithm as follows.

Then, we describe our algorithm for generating four independent spanning trees rooted at vertex 0 in $CR(N, d)$. Clearly, constructing the four independent spanning trees also takes linear time.

Procedure Div_T1(N, d) ($d > 2$,$[N]_d = 0$.)
begin
1. For $i = 1$ to $d-1$ do
    Set parent of vertex i to $i-1$.
2. For $i = 1$ to $d-1$ do
    Let $x = i$.
    While $x \leq N-d$ do
      Case 1: $i < \lfloor d/2 \rfloor$
        If $x/d < N/2d$ then
          Set parent of vertex $x+d$ to x.
        Else
          Set parent of vertex $x+d$ to $[x+2d]_N$.
      Case 2: $i = \lfloor d/2 \rfloor$
        If $x/d < N/2d$ then
          Set parent of vertex $x+d$ to x.
        Else
          If $[d]_2 \neq 0$ or $[x+2d]_N < d$
            Set parent of vertex $x+d$ to $[x+2d]_N$.
          Else
            Set parent of vertex $x+d$ to $[x+d-1]_N$.
      Case 3: $i > \lfloor d/2 \rfloor$
        Set parent of vertex $x+d$ to $x+d-1$.
      Let $x = x+d$.
3. For $i = 1$ to $(N/d - 1)$ do
    Let $x = i \times d$.
    Set parent of vertex x to $x+1$.
end.

Algorithm DIV_CR
Input : $CR(N, d)$ ($d > 2$,$[N]_d = 0$.)
Output : T_1,T_2,T_3 and T_4
begin
1. Call **Procedure Div_T1**(N, d).
2. Call **Procedure Div_T2**(N, d).
3. Generate T_3 and T_4 using the same method as steps 3 and 4 of **Algorithm IST_CR**.
end.

For example, we generate T_1, T_2, T_3 and T_4 on $CR(35, 5)$ as shown in Figure 4. The heights of T_1 and T_3 in Figure 4 are both 7, while the heights of T_2 and T_4 are both 8. Notice that the height of independent spanning trees constructed by Iwasaki's algorithm is 10. The result of Algorithm **IST_CR** is also 10.

Using Procedure **Div_T1**, the height of the spanning tree can be expressed by a simple formula, i.e., $d-1+\lfloor N/2d \rfloor$. That is, $height(T_1)= height(T_3)=d-1+\lfloor N/2d \rfloor$. Meanwhile, the height of the spanning tree generated by procedure **Div_T2** is $N/d - 1 + \lfloor d/2 \rfloor$. That is, $height(T_2)=height(T_4)= N/d - 1 + \lfloor d/2 \rfloor$. By comparing with Iwasaki's algorithms, the reduced height of each spanning tree is either $d + \lfloor (N-2d)/d \rfloor - (d-1+\lfloor N/2d \rfloor) = \lfloor (N/d-1)/2 \rfloor$

Procedure Div_T2(N, d) $(d > 2, [N]_d = 0.)$
begin
1. For $i = 1$ to $(N/d - 1)$ do
    Let $x = i \times d$.
    Set parent of vertex x to $x - d$.
2. For $i = 1$ to $N/d - 1$ do
    Let $x = i \times d$.
    If $i \leq \lfloor (N/d - 1)/2 \rfloor$ then
        For $j = 1$ to $\lfloor d/2 \rfloor$ do
            Set parent of vertex $x + j$ to $x + j - 1$.
        For $j = 1$ to $\lfloor (d-1)/2 \rfloor$ do
            Set parent of vertex $x - j$ to $x - j + 1$.
    Else // $i > \lfloor (N/d - 1)/2 \rfloor$
        For $j = 1$ to $\lfloor d/2 \rfloor$ do
            If $i = \lfloor (N/d)/2 \rfloor$ and $[N/d]_2 = 0$ then
                Set parent of vertex $x+j$ to $x+j-1$.
            Else
                Set parent of vertex $x+j$ to $x+j-d$.
        For $j = 1$ to $\lfloor (d-1)/2 \rfloor$ do
            Set parent of vertex $x - j$ to $x - j - d$.
3. For $i = 1$ to $\lfloor (d-1)/2 \rfloor$ do
    Set parent of vertex i to $i + d$.
    Set parent of vertex $N - i$ to $[N - i + d]_N$.
  If $[d]_2 = 0$ then
    The parent of vertex $d/2$ to $d/2 + 1$.
end.

or $d + \lfloor (N-2d)/d \rfloor - (N/d - 1 + \lfloor d/2 \rfloor) = \lfloor (d-1)/2 \rfloor$. As a result, we have Theorem 2.

Theorem 2 *Algorithm* **DIV_CR** *can reduce the height of independent spanning trees in* $RC(N, d)$ *by an amount of* $\lfloor (N/d - 1)/2 \rfloor$ *(in* T_1 *and* T_3*) or* $\lfloor (d-1)/2 \rfloor$ *(in* T_2 *and* T_4*) by comparing with Iwasaki's algorithms.*

4 Correctness of the algorithms

In this section, we shall concisely prove that T_1, T_2, T_3 and T_4 generated by both Algorithms **IST_CR** and **DIV_CR** are independent spanning trees rooted at 0 in $CR(N, d)$.

Lemma 3 T_1, T_2, T_3 *and* T_4 *generated by both Algorithms* **IST_CR** *and* **DIV_CR** *are spanning trees of* $CR(N, d)$.

Proof. By analyzing the steps of Algorithms **IST_CR** and **DIV_CR**, T_i (i=1,2,3,4) consists of N vertices and $N-1$ edges. Meanwhile, T_i is connected. Therefore, T_1, T_2, T_3 and T_4 are four spanning trees of $CR(N, d)$. □

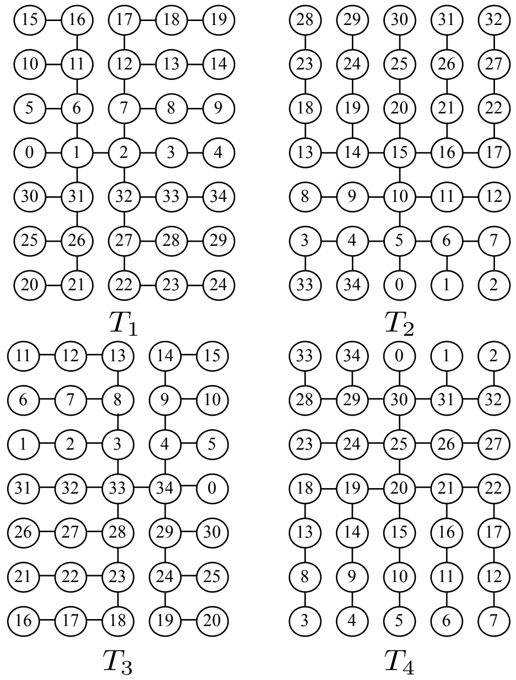

Figure 4: T_1, T_2, T_3 and T_4 on $CR(35, 5)$.

To prove that T_1, T_2, T_3 and T_4 are pairwise independent, we define the *ancestor set* of a vertex v in T_i $(i = 1, 2, 3, 4)$, denoted by $ancestor(v, i)$, as the vertex set of the path from *root* vertex 0 to the *parent* vertex of v in T_i. By the definition of independent spanning trees, we figure out that T_i and T_j $(i \neq j)$ are independent if and only if for every vertex v in $CR(N, d)$, $v \neq 0$, $ancestor(v, i) \bigcap ancestor(v, j) = 0$. This property is the main idea in proving the following lemma.

Lemma 4 T_1, T_2, T_3 *and* T_4 *generated by both Algorithms* **IST_CR** *and* **DIV_CR** *are mutually independent.*

Proof. By analyzing the ancestor set of every vertex v $(v \neq 0)$ with respect to the four spanning trees generated by **IST_CR** or**DIV_CR**, we can prove that $ancestor(v, 1) \bigcap ancestor(v, 2) \bigcap ancestor(v, 3) \bigcap ancestor(v, 4) = 0$. That is, T_1, T_2, T_3 and T_4 are mutually independent. □

We summarize Lemmas 3 and 4 as Theorem 5.

Theorem 5 *Algorithm* **IST_CR** *and Algorithm* **DIV_CR** *can correctly generate four independent spanning trees rooted at vertex* 0 *in* $CR(N, d)$.

5 Concluding remarks

In this paper, we present two algorithms for constructing four independent spanning trees rooted at an arbitrary vertex in a chordal ring. By comparing with Iwasaki's algorithms, Algorithm **IST_CR** can reduce the height of each spanning tree to an extent of $\lfloor d/2 \rfloor$, while Algorithm **DIV_CR** can reduce the height of each spanning tree by an amount of $\lfloor (N/d-1)/2 \rfloor$ (in T_1 and T_3) or $\lfloor (d-1)/2 \rfloor$ (in T_2 and T_4). To provide a clear comparison, we aggregate the results of programming efforts as shown in Table 1. We use two criteria in Table 1, *total height*(TH) and *total path length*(TPL). The former is the summation of *height*(T_i)($i = 1, 2, 3, 4$), the latter is the summation of path length in each tree.

Table 1: Comparison of different algorithms.

		Iwasaki		**IST_CR**		**DIV_CR**	
N	d	TH	TPL	TH	TPL	TH	TPL
7	2	12	52	12	52	N/A	N/A
32	5	36	608	30	520	N/A	N/A
31	7	36	588	28	480	N/A	N/A
9	3	16	80	14	78	12	76
30	5	36	556	34	532	28	468
35	5	40	716	38	688	30	594
36	6	40	740	38	708	32	614
48	8	48	1180	46	1120	38	958
99	11	72	3604	70	3454	54	2834

References

[1] Bruce W. Arden, Hikyu Lee, Analysis of chordal ring network, *IEEE Transactions on Computers*, Vol. C-30, No. 4, April 1981, pp. 291-295.

[2] F. Bao, Y. Igarashi, S.R. Ohring, Reliable broadcasting in product networks, *IEICE Technical Report COMP*, 95(18), 1995, pp. 57-66.

[3] J.-C.Bermond, F. Comellas, D. F. Hsu, Distributed loop computer networks: A survey, *Journal of Parallel and Distributed Computing*, Vol. 24, 1995, pp. 2-10.

[4] Nimmagadda Chalamaiah, Badrinath Ramamurthy, Finding shortest paths in distributed loop networks, *Information Processing Letters*, Vol. 67, 1998, pp. 157-161.

[5] J. Cheriyan, S. N. Maheshwari, Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs, *Journal of Algorithms* 9, 1988, pp. 507-537.

[6] S. Curran, Independent Trees in 4-connected graphs, *Doctoral thesis* at School of Mathematics, Georgia Institute of Technology, 2003.

[7] D.-Z. Du, D. F. Hsu, Qiao Li, Junming Xu,A combinational problem related to distributed loop networks, *Networks*, Vol. 20, 1990, pp. 173-180.

[8] Paul Erdos, Frank D. Hsu,Distributed loop network with minimum transmission delay, *Theoretical Computer Science*, Vol. 100, 1992, pp. 223-241.

[9] A. Itai, M. Rodeh,The multi-tree approach to reliability in distributed networks, *Information and Computation* 79, 1988, pp. 43-59.

[10] Yukihiro Iwasaki, Yuka Kajiwara, Koji Obokata, Yoshihide Igarashi, Idependent spanning trees of chordal rings, *Information Processing Letters* 69, 1999, pp. 155-160.

[11] Krishnendu Mukhopadhyaya, Bhabani P. Sinha, Fault-tolerant routing in distributed loop networks, *IEEE Transactions on Computers*, Vol. 44, No. 12, 1995, pp. 1452-1456.

[12] L.Narayanan, J. Opatrny, Compact routing on chordal rings of degree 4, *Algorithmica*, Vol. 23, 1999, pp. 72-96.

[13] Behrooz Parhami, Ding-Ming Kwai, Periodically regular chordal rings, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 10, No. 6, 1999, pp. 658-672.

[14] Guy W. Zimmerman, Acdol-Hossein Esfahanian, Chordal rings as fault-tolerant loops, *Discrete Applied Mathematics*, Vol. 37/38, 1992, pp. 563-573.

[15] A. Zehavi, A. Itai, Three tree-paths, *Journal of Graph Theory* 13, 1989, pp. 175-188.