Complete-rotation Graph : A New Interconnection Network

Cheng-Ta Lee[1] and Chiun-Chieh Hsu[2]

[1]Department of International Trade
Fushin Institute of Technology, Ilan, Taiwan, R.O.C.
E-mail: ctlee@mail.it.fit.edu.tw

[2]Department of Information Management
National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.
E-mail: cchsu@cs.ntust.edu.tw

Abstract

This paper presents and analyzes a new interconnection network - complete-rotation graph. This network is an interesting candidate for massively parallel processing. Communication is a basic requirement of constructing a new interconnection network. Hence, we develop an optimal routing algorithm and a fault tolerant routing algorithm in complete-rotation graph. In addition, we can find that the degree and diameter of the complete-rotation graph are smaller than those of the hypercube when these two graphs have the same number of vertices.

Keyword: Interconnection network, Cayley graph, complete-rotation graph, optimal routing, trivalent Cayley graph, degree four Cayley graph, cube-connected-cycle.

1. Introduction

Interconnection networks have been an important research area for highly parallel computers which communicate by message passing. Several structures have been proposed in the literature for interconnecting a large network of computers in parallel and distributed environments. For example, hypercubes[4,5,9,12], star graphs[6-8,10], pancake graphs[7,8], rotator graphs[6,11], trivalent Cayley graphs [2,3], and degree four Cayley graph [1]. All of the above interconnection networks belong to Cayley graphs, which are very suitable for designing interconnection networks. These graphs are node symmetric, regular, and share many of the desirable properties like low diameter, low degree, and high fault tolerance etc.

Recently, Vadaplli and Srimani proposed the trivalent Cayley graphs [2,3] and the degree four Cayley graph [1]. The former is regular of degree 3 and the latter is regular of degree 4 irrespective of the size of the graph. Both of the two graphs have logarithmic diameters in the number of nodes and those are maximally fault tolerant (vertex connectivity cannot exceed the node degree). The diameter of the trivalent Cayley graphs is $2n-1$ and that of the degree four Cayley graphs is $\lfloor 3n/2 \rfloor$. Compared with cube-connected-cycle graphs, the latter graph has a higher vertex connectivity (higher fault tolerance) and it accommodates a larger number of nodes than the cube-connected-cycle graph for the same diameter.

An interconnection structure in general should have a small number of link per node (degree of a node), and a small inters node distance (diameter). Designing a network with low message traffic density and good modularity is also desirable. But in a multicomputer environment, the average internode distance, message traffic density, and fault tolerance are highly dependent on the diameter and degree of a node. There is a tradeoff between the degree of a node and the diameter. A single loop structure and a completely connected structure as described above represent the two extremes.

Henceforth, we present a new Cayley graph — complete-rotation graph and that reveals its interesting properties in this paper. The graph is symmetric directed Cayley graph. Machines with directed communication links are easy to construct, and they allow faster communication by simplifying the protocols used at link level. We will present an optimal routing algorithm and find its diameter. It will be shown that the complete-rotation graph has a smaller diameter than that of the trivalent Cayley graphs and the degree four Cayley graph for the same number of nodes, while sharing its desirable property such as maximal fault tolerance.

The rest of the paper is organized as follows: In section 2, we introduce the complete-rotation graph interconnection network, and propose some properties of the complete-rotation graph. In section 3, we present an optimal routing algorithm and fault tolerant routing algorithm. Finally, the conclusion is given in section 4.

2. Preliminaries

In this section we formally define the complete-rotation graph. We also include the definitions which will be used in later section. The terms "vertex" and "node" are used interchangeably. The term " I " is used with the same meaning as the identity permutation $t_1 t_2 \cdots t_n$. Let $a_1 a_2 \cdots a_n$ denote the label of an arbitrary node and

$a_1 = t_k^*$ for $1 \le k \le n$, where t_k^* to denote either t_k or $\bar{t}_k$. We use t_* to denote t_k and a_{c_k} to denote k th complemented symbol of an arbitrary node for $1 \le k \le l \le n$ where l is the number of the largest complemented symbol. For easy reference, we use CR_n to denote the complete-rotation graph with n dimension.

2.1 Complete-rotation Graph

A complete-rotation graph is defined as a symmetric directed graph on $N = n \times 2^n$ vertices for any integer n, $n \ge 3$. Each vertex is represented by a circular permutation of n symbols in lexicographic order where each symbol may be present in either uncomplemented or complemented form. Let t_k, $1 \le k \le n$, denote the k th symbol in the set of n symbols (we use English alphabets as symbols; thus for n , $t_1 = a$, $t_2 = b$, $t_3 = c$, and $t_4 = d$). We use t_k^* to denote either t_k or $\bar{t}_k$. Thus, for n distinct symbols, there are exactly n different cyclic permutation of the symbols in lexicographic order (disregarding the complements). And since each symbol can be present in either complemented or uncomplemented form, the vertex set of CR_n (i.e., the underlying group Γ) has a cardinality of $n \cdot 2^n$. For example, for $n = 3$, the number of vertices in CR_n is 24; $abc, cab, \bar{c}ab$ are valid nodes while acb or bac are not. Let I denote the *identity permutation* $t_1 t_2 \cdots t_n$. Since each node is some cyclic permutation of the n symbols in lexicographic order. If $a_1 a_2 \cdots a_n$ denotes the label of an arbitrary node and $a_1 = t_k^*$ for some integer k , then for all i, $2 \le i \le n$, we have $a_i = t^*_{((k+i-2) \bmod n)+1}$. Figure 2.1 shows the proposed complete-rotation graph of dimension 3.

Definition 2.1 The edges of CR_n are defined by the following n generators in the graph:
$f_k(a_1 a_2 \cdots a_n) = a_{(k \bmod n)+1} a_{((k+1) \bmod n)+1} \cdots a_{((k+n-1) \bmod n)+1}$, for $1 \le k \le n-1$ and $g(a_1 a_2 \cdots a_n)$ = $a_2 a_3 \cdots a_n \bar{a}_1$. The former is called $f-edge$ and the latter is called $g-edge$.

Example 2.1 $f_1(abc) = bca$ and $g(abc) = bc\bar{a}$.

Lemma 2.1 $f_k(X) = Y$ *iff* $f_{n-k}(Y) = X$.

Proof. Let an arbitrary node $X = a_1 a_2 \ldots a_n$.
Then
$f_k(X) = a_{(k \bmod n)+1} a_{((k+1) \bmod n)+1} \cdots a_{((k+n-1) \bmod n)+1} = Y$ .Hence,
$f_{n-k}(Y) = a_{(k+n-k \bmod n)+1} a_{((k+n-k+1) \bmod n)+1} \cdots a_{((k+n-k+n-1) \bmod n)+1}$
$= a_1 a_2 \cdots a_n = X$. □

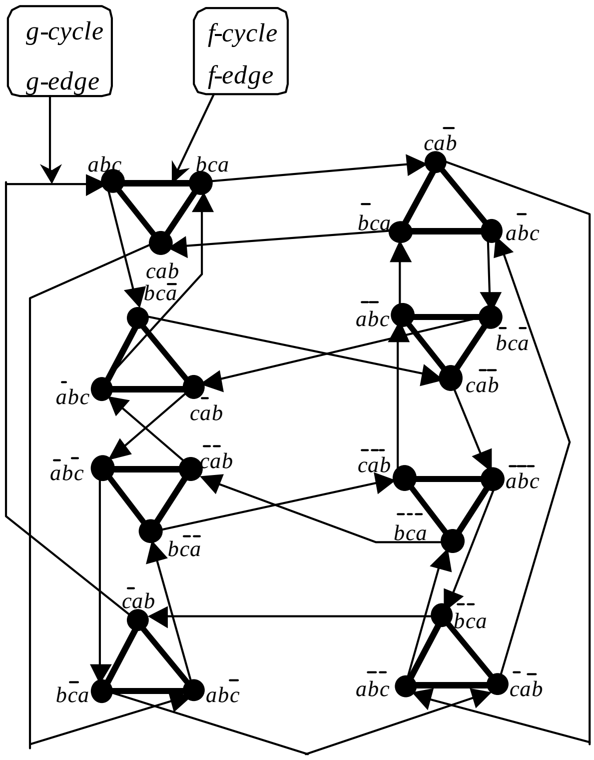

Figure 2.1 Three dimensional complete-rotation graph.

Definition 2.2 Let G be a finite group and Ω be a set of generators for G. We define the Cayley graph G with generating set Ω as follows: (1) Each element of G is a vertex of the Cayley graph. (2) For x and y in G, there is an edge from x to y if and only if $xg = y$ for some generator $g \in \Omega$.

Example 2.2 The vertex set of CR_3 (the underlying group Γ) is given by $\{abc, \bar{a}bc, a\bar{b}c, ab\bar{c}, cab, \bar{c}ab, c\bar{a}b, ca\bar{b}, bca, \bar{b}ca, b\bar{c}a, bc\bar{a}, \bar{a}\bar{b}c, a\bar{b}\bar{c}, \bar{a}b\bar{c}, \bar{c}\bar{a}b, c\bar{a}\bar{b}, \bar{c}a\bar{b}, \bar{b}\bar{c}a, b\bar{c}\bar{a}, \bar{b}c\bar{a}, \bar{a}\bar{b}\bar{c}, \bar{c}\bar{a}\bar{b}, \bar{b}\bar{c}\bar{a}\}$ and the generator set Ω is given by $\Omega = \{bca, cab, bc\bar{a}\}$. Note that since we are considering groups of permutation of distance symbols which may be complemented or uncomplemented, the generators themselves are also permutations of complemented or uncomplemented symbols.

Theorem 2.1 For any n, $n \ge 3$, the complete-rotation graph CR_n: (1) is a node symmetric regular graph of degree $n+1$ (out-degree $= n$); (2) has $n \times 2^n$ vertices; and (3) has $n * 2^n + (n(n-1)/2) \times 2^n$ edges.

Proof. (1) can be directly derived from Definition 2.1. (2) can be directly derived from the fact that there are exactly n circular permutations of n symbols in lexicographic order (permutation with same cyclic ordering) and in any permutations each symbol can be either in complemented or uncomplemented form. (3) can be directly derived from

(1) and (2), where there are $n \times 2^n$ g-edges and $(n(n-1)/2) \times 2^n$ f-edges. □

Theorem 2.2 *Generation Set* : { f_{k^*} *function* and g *function*};
If $a_i = t^*_{((k+i-2) \bmod n)+1}$ where $1 \le k^* \le \lceil (n-1)/i \rceil$ *for* $1 \le i \le n-1$ and $g(a_1 a_2 \cdots a_n) = a_2 \cdots a_n \overline{a_1}$, then diameter = $n+i$.
Proof. Consider the worst case that all symbols of a node u are complemented and $\overline{t}_1$ is not first symbol. In the process of u going to node v via g function generators, each symbol needs at least n steps to become uncomplemented. Because of $1 \le k^* \le \lceil (n-1)/i \rceil$, $1 \le i \le n-1$ v needs i steps routing to the identity node I. Therefore, the diameter $= n+i$. □

For the complete-rotation graph G_n, since $i=1$, diameter = $n+1$.

Example 2.3 For CR_6, node $u = \overline{b}\overline{c}\overline{d}\overline{e}\overline{f}\overline{a}$ requires 6 steps traversing to node $v = bcdefa$ where $n=6$, $i=2$, and $1 \le k^* \le 3$. Thus diameter = $n+2$, and the routing is
$\overline{b}\overline{c}\overline{d}\overline{e}\overline{f}\overline{a} \rightarrow \overline{c}\overline{d}\overline{e}\overline{f}\overline{a}b \rightarrow \overline{d}\overline{e}\overline{f}\overline{a}bc \rightarrow \overline{e}\overline{f}\overline{a}bcd \rightarrow \overline{f}\overline{a}bcde \rightarrow \overline{a}bcdef \rightarrow bcdefa \rightarrow efabcd \rightarrow abcdef$

2.2 Topological Properties of Complete-rotation Graph

In this section, some interesting topological properties of the complete-rotation graph are presented. A Cayley graph is completely specified by a set of d permutations as generators. Of course, the out-degree of the graph will be d. Every Cayley graph is vertex symmetric. A vertex symmetric graph has the desirable property that the communication load is uniformly distributed on all vertices so that there is no congestion. Recall that a stronger notion of symmetry like edge symmetry requires that every edge in the graph looks the same. Such symmetry would ensure that the communication load is uniformly distributed over all the communication links, so that there is no congestion at any link.

Definition 2.3 Any cycle in CR_n consisting of only the f-edges (induced by the symmetric functions f_1 or f_{n-1}) is called an $f-cycle$. Similarly, any cycle in CR_n consisting of only the g-edges is called a $g-cycle$.

Example 2.4 In Figure 2.1, the cycle $\{abc \leftrightarrow bca \leftrightarrow cab \leftrightarrow abc\}$ is called an *f-cycle* while the cycle $\{abc \rightarrow bc\overline{a} \rightarrow c\overline{a}\overline{b} \rightarrow \overline{a}\overline{b}\overline{c} \rightarrow \overline{b}\overline{c}a \rightarrow \overline{c}ab \rightarrow abc\}$ is called a *g-cycle*.

Definition 2.4 The complement of any vertex $u = a_1 a_2 \cdots a_n$ in CR_n is the vertex $\overline{u}$ obtained by complementing the symbols in u, i.e., $\overline{u} = \overline{a}_1 \overline{a}_2 \cdots \overline{a}_n$.

For example, complement of the vertex $u = c\overline{a}b$ in CR_3 is the vertex $\overline{u} = \overline{c}a\overline{b}$.

Lemma 2.2 For arbitrary nodes u and v in CR_n where $g(u) = v$, the complemented nodes satisfy the same relation, $g(\overline{u}) = \overline{v}$.
Proof. Let an arbitrary node $u = a_1 a_2 \cdots a_n$. v is given by $v = g(u) = a_2 \cdots a_n \overline{a}_1$. Thus, $g(\overline{u}) = g(\overline{a}_1 \overline{a}_2 \cdots \overline{a}_n) = \overline{a}_2 \cdots \overline{a}_n a_1 = \overline{v}$. □

Corollary 2.1 For any vertex v in CR_n, both v and $\overline{v}$ belong to the same g-cycle.

Lemma 2.3 All of the $n \cdot 2^n$ nodes of CR_n are partitioned into 2^{n-1} vertex disjoint g-cycles of length $2n$.
Proof. Consider an arbitrary node $v = a_1 a_2 \cdots a_n$ in CR_n. For any $i \ge 1$, let $g^i(v) = g(g^{i-1}(v))$, where $g(v) = g^1(v)$. It is easy to observe that $g^n(v) = \overline{v} = \overline{a}_1 \overline{a}_2 \cdots \overline{a}_n$ and $g^{2n}(v) = v$. Also, $g^i(v) \ne g^j(v)$ for $1 \le i, j \le 2n$. Thus, from an arbitrary vertex v, if the g function is repeatedly applied, a cycle of length $2n$ can be obtained in the graph CR_n. That these g-cycles are vertex disjoint follow from the fact that $g(u) = g(v)$, if and only if $u = v$. □

Lemma 2.4 All of the $n \cdot 2^n$ nodes of CR_n can be partitioned into 2^n vertex disjoint f-cycles of length n.
Proof. Similar to the proof of Lemma 2.3. □

Lemma 2.5 In CR_n, all nodes of each f-cycle constitute a complete graph K_n.
Proof. Consider two arbitrary nodes $u = a_1 a_2 \cdots a_n$ and $v = a_{(k \bmod n)+1} a_{((k+1) \bmod n)+1} \cdots a_{((k+n-1) \bmod n)+1}$ in same f-cycle where $1 \le k \le n-1$. We can use $f_k(u) = v$ and $f_{n-k}(v) = u$ generate graph K_n. □

Lemma 2.6 Consider a vertex v in CR_n where $f_1(v) = u$ and $g(v) = w$. There exists a vertex x such that $g(x) = u$ and $f_1(x) = w$, and v, u, w, and x are all distinct.

Proof. Let $v = a_1a_2\cdots a_n$. Then $u = f_1(v) = a_2\cdots a_na_1$ and $w = g(v) = a_2\cdots a_n\bar{a}_1$. Let $x = f_{n-1}(w) = \bar{a}_1a_2\cdots a_n$. Thus, $g(x) = u$ and $f_1(x) = a_2\cdots a_n\bar{a}_1 = w$. It is obvious that the four nodes are distinct. □

Lemma 2.6 can be verified in Figure 2.2.

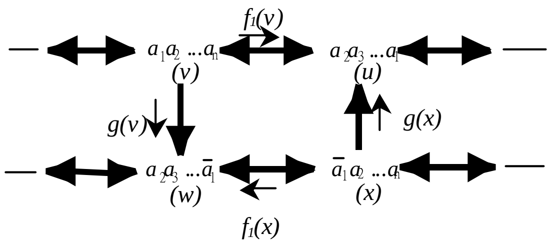

Figure 2.2 Example of v, u, w, and x are all distinct.

Lemma 2.7 In CR_n, if two f-cycles F_i and F_j are *adjacent*, then there are exactly a pair of g-edges connecting F_i and F_j.

Proof. Since F_i and F_j are adjacent, we assume that there exists a vertex $v \in F_i$ and $g(v) = w$, where $w \in F_j$. Let $f_1(v) = u$. Then by Lemma 2.6, $g(x) = u$ and $f_1(x) = w$. Thus x is a node in F_j. □

Lemma 2.8 Each f-cycle in CR_n is adjacent to n different f-cycles.

Proof. Consider an arbitrary f-cycle with the node $v = a_1a_2\cdots a_n$. We can gain $y_0 = g(v) = a_2a_3\cdots\bar{a}_1$, $y_1 = gf_1(v) = a_3a_4\cdots\bar{a}_2$, ... , and $y_{n-1} = gf_{n-1}(v) = a_1a_2\cdots\bar{a}_n$ belonging to different f-cycles and hence any f-cycle is adjacent to n different f-cycles in CR_n. □

Corollary 2.2 Consider an f-cycle F_i for a given $i = (b_{n-1}b_{n-2}\cdots b_0)_2$, $0 \le i < 2^{n-1}$. The f-cycle F_i or $f_{(b_{n-1}b_{n-2}\cdots b_0)_2}$ is adjacent to the n f-cycles $f_{(\bar{b}_{n-1}b_{n-2}\cdots b_0)_2}$, $f_{(b_{n-1}\bar{b}_{n-2}\cdots b_0)_2}$, ... , *and* $f_{(b_{n-1}b_{n-2}\cdots\bar{b}_0)_2}$.

Lemma 2.9 In CR_n, if each f-cycle is reduced to one node, a hypercube H_n is obtained.

Proof. Follow from Lemma 2.5 , 2.7, 2.8 and Corollary 2.2. □

Lemma 2.9 can be verified in Figure 2.3.

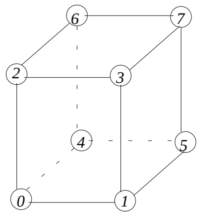

Figure 2.3 The reduce graph of CR_3.

Theorem 2.3 For $n \ge 3$, the bisection width of the complete-rotation graph CR_n is 2^n.

Proof. Following from Lemma 2.7, Lemma 2.9, and the bisection width of the hypercube H_n is 2^{n-1}, we conclude that bisection width of the CR_n is 2^n. □

Availability of hamiltonian cycle can be extensively applied, such as indexing, embedding of linear arrays and ring.

Theorem 2.4 The graph CR_n, $n \ge 3$ is hamiltonian.

Proof. Follow from Lemma 2.6 and 2.7, consider two arbitrary adjacent f-cycles, say F_i and F_j. There exist nodes $u, v \in F_i$ and nodes $w, x \in F_j$ such that $w = g(v) = f_1(x)$ and $u = f_1(v) = g(x)$. A larger cycle can be constructed involving all nodes of F_i and F_j by using these two g-edges as shown in Figure 2.4. This couples with the facts that f-cycles in CR_n are vertex-disjoint and each f-cycle is adjacent to exactly n distinct f-cycles, which lead to the desired result. □

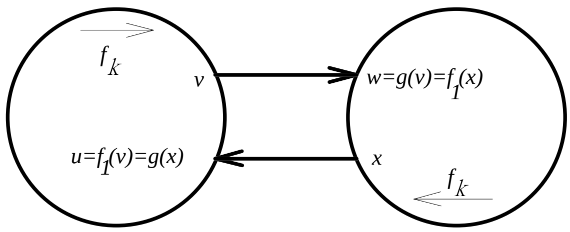

Figure 2.4 Combining two f-cycle to produce a large cycle.

3. Data Communications

We know that complete-rotation graph is vertex symmetric. Let $c\bar{a}b$ be the source node and $bc\bar{a}$ be the destination node. We can map the destination node to the identity node abc by renaming the symbols as $b \to a, \bar{b} \to \bar{a}, c \to b, \bar{c} \to \bar{b}, \bar{a} \to c$ *and* $a \to \bar{c}$. Under this mapping the source node becomes $\bar{b}\bar{c}a$. Then the paths between the original source and destination nodes become isomorphic to the paths between the node $\bar{b}\bar{c}a$ and the identity node abc in the renamed graph.

Thus in our subsequent discussion about a path from a source node to a destination node, the destination node is always assumed to be the identity node I without loss of generality. The following algorithms Simple_Route and Optimal_Route compute a path from an arbitrary source node $a_1a_2\cdots a_n$ in CR_n to the identity node I.

3.1 Simple Routing, Optimal Routing, and Diameter

Algorithm Simple_Route.

Step 1 Initialize $k=0$, compute minimum k, $1 \le k \le n$, such that $a_k = a_{c_1}$;
Step 2 If $k=0$ then { compute j, $1 \le j \le n$, such that $a_j = t_1$;
if $j=1$ then routing finished;
else traverse along $f_{j-1}-edge$;}
else { if $k=1$ then traverse along $g-edge$;
else {traverse along $f_{k-1}-edge$ and traverse along $g-edge$;}
go to step 1;}

Example 3.1 Consider the node $c\bar{d}a\bar{b}$ in CR_4. The simple routing is

$$c\bar{d}a\bar{b} \xrightarrow{f_1} \bar{d}a\bar{b}c \xrightarrow{g} a\bar{b}cd \xrightarrow{f_1} \bar{b}cda \xrightarrow{g} cdab \xrightarrow{f_2} abcd.$$

Lemma 3.1 The algorithm of Simple_Route correctly computes a path from an arbitrary node $a_1a_2\cdots a_n$ to the identity node $I = t_1t_2\cdots t_n$.

Proof. The integer k always exists (step 1) since the node $a_1a_2\cdots a_n$ is a cyclic permutation of the lexicographic ordering of n symbols (each symbol is in complemented or uncomplemented form). Such a move obviously exists for each iteration and at the end of the algorithm we can reach the destination node $t_1t_2\cdots t_n$. □

Theorem 3.1 For an arbitrary node $a_1a_2\cdots a_n$ in CR_n, the Simple_Route algorithm generates a path of length $\le n+1$.

Proof. See Simple_Route algorithm. An arbitrary vertex u have at most k complemented symbols and at most $n-k$ uncomplemented symbols. In the worst case, the algorithm has to pass through k g-edges and $n-k+1$ f-edge. Hence, the Simple_Route algorithm generates a path of length $\le n+1$. □

Observation 3.1 Based on Simple_Route algorithm, if $u \in CR_n$, then

$$\text{distance}(u, I) = \begin{cases} k-1 & \text{if } a_{c_l} = \bar{t}_n \text{ and } a_{c_l} \ne a_n \\ k+1 & \text{if } a_n = \bar{t}_* \text{ and } a_n \ne \bar{t}_n \\ k & \text{otherwise} \end{cases}$$

where k = complemented symbol's number of u + uncomplemented substring(s) symbol's number of u.

Example 3.2 Base on Simple_Route algorithm, let $u = abcdefg$, $v_1 = bcd\bar{e}f\bar{g}a$, $v_2 = \bar{g}a\bar{b}cde\bar{f}$, $v_3 = f\bar{g}a\bar{b}cde$ and $v_4 = a\bar{b}cde\bar{f}\bar{g}$, we can obtain:
distance $(u, v_1) = k-1 = 5-1 = 4$,
distance $(u, v_2) = k+1 = 5+1 = 6$,
distance $(u, v_3) = k = 5$ and
distance $(u, v_4) = k = 5$.

Routing in interconnection networks is the vital factor which decides the efficiency and throughput of the interconnection networks. Minimal path routing is preferred because the number of hops traveled by the messages are important in the interconnection networks.

Algorithm Optimal_Route.

Step 1 initialize flag=0;
Step 2 if $a_1 \ne a_{c_1}$ and exist $\bar{t}_n$ and $a_n \ne \bar{t}_n$ and exist t_1
then { compute minimum i, $2 \le i \le n$, such that $\bar{t}_i = a_l$ where $1 \le l \le n$; traverse along $f_{l-1}-edge$; and traverse along $g-edge$; flag $=1$;}
Step 3 initialize $k=0$, compute minimum k, $1 \le k \le n$, such that $a_k = a_{c_1}$;
if $k=0$ then { compute j, $1 \le j \le n$, such that $a_j = t_1$;
if $j=1$ then routing finished;
else traverse along $f_{j-1}-edge$;}
else { if $k=1$ then traverse along $g-edge$;
else {traverse along $f_{k-1}-edge$; and traverse along ;}
if flag $=0$ then go to step 2;
else go to step 3;}

Example 3.3 Consider the node $c\bar{d}a\bar{b}$ in CR_4. The optimal routing is

$c\bar{d}a\bar{b} \xrightarrow{f_3} \bar{b}c\bar{d}a \xrightarrow{g} c\bar{d}ab \xrightarrow{f_1} \bar{d}abc$
$\xrightarrow{g} abcd$.

Lemma 3.2 In CR_n, for an arbitrary node $a_1a_2\cdots a_n$ route to identity node $I = t_1t_2\cdots t_n$. The Simple_Route algorithm at most increases by 1 step than the Optimal_Route algorithm.

Proof. From Simple_Route and Optimal_Route algorithm, if $a_1 = a_{c_1}$, then using g function. In Optimal_Route algorithm, if $a_1 \neq a_{c_1}$ and exist $\bar{t}_n$ and $a_n \neq \bar{t}_n$ and exist t_1, then Optimal_Route algorithm can choose the best function (Step 2 of Optimal_Route algorithm) to generate the optimal path. It makes f-edge at most minus one than Simple_Route algorithm. □

Lemma 3.3 The algorithm of Optimal_Route correctly computes the shortest path from an arbitrary node $a_1a_2\cdots a_n$ to the identity node $I = t_1t_2\cdots t_n$.

Proof. The integer k always exists (step 3) since the node $a_1a_2\cdots a_n$ is a cyclic permutation of the lexicographic ordering of n symbols (each symbol is in complemented or uncomplemented form). Such a move obviously exists for each iteration and at the end of the algorithm we can reach the destination node $t_1t_2\cdots t_n$. From Lemma 3.2, if $a_1 = a_{c_1}$, then use g function, else we choose the best function to generate the optimal path. □

Theorem 3.2 For an arbitrary node $a_1a_2\cdots a_n$ in CR_n, the algorithm Optimal_Route generates a path of length $\leq n+1$.

Proof. See Optimal_Route algorithm. An arbitrary vertex u has at most k complemented symbols and at most $n-k$ uncomplemented symbols. In the worst case, the algorithm has to pass through k g-edges and $n-k+1$ f-edge. Hence, the Optimal_Route algorithm generates a path of length $\leq n+1$. □

Theorem 3.3 The diameter of the graph CR_n is given by $D(G_n) = n+1$.

Proof. Consider the node $x = \bar{t}_2\bar{t}_3\cdots\bar{t}_1$. We see that $D(x) = n+1$. This, coupled with Theorem 3.2, establishes the result. □

Example 3.4 Based on the Optimal_Route algorithm, let $u = abcdefg$, $v_1 = bcd\bar{e}f\bar{g}a$, $v_2 = \bar{g}a\bar{b}cde\bar{f}$, $v_3 = \bar{f}ga\bar{b}cde$ and $v_4 = a\bar{b}cde\bar{f}\bar{g}$, we can obtain: distance $(u,v_1) = 4$, distance $(u,v_2) = 6$, distance $(u,v_3) = 4$ and distance $(u,v_4) = 5$.

3.2 Routing Tree and Fault Tolerant Routing

Routing a message from a node x to a node y in any Cayley graph is the same as routing from node xy^{-1} to the identity permutation[13]. The routing tree of CR_n is given in Figure 3.1 base on Simple_Route algorithm. The identity node is at level zero of the routing tree, and there are $n+1$ levels ranging from 0 to $(n+1)$. The links of every node in CR_n are marked by g and f_k, for $1 \leq k \leq n-1$, where communication takes place along links.

Definition 3.1 In CR_n, that is define the returnable symbol is defined by

$$\begin{cases} a_n & if \quad no \ symbol \ complemented \\ \bar{a}_{c_l} & otherwise \quad \forall \ nodes \in CR_n \end{cases},$$

where l is the number of largest complemented symbol.

Example 3.5 From Definition 3.1, let $u = bca$ and $v = a\overline{bc}$. The returnable symbol of u is a and the returnable symbol of v is b.

For example, in Figure 3.2 the link from $ab\bar{c}$ to $\bar{c}ab$ is marked as f_2, since the function applied to the node $ab\bar{c}$ to reach the node $\bar{c}ab$ is f_2. The returnable symbol of the subtree is c.

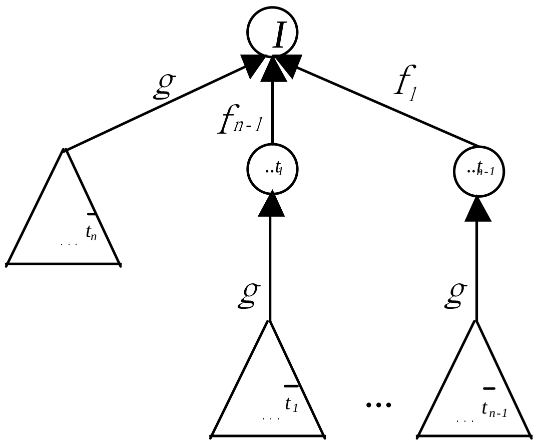

Figure 3.1 Routing tree.

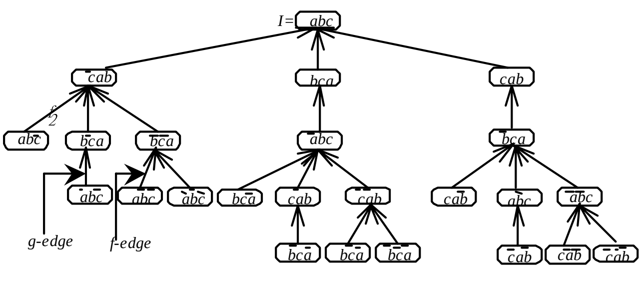

Figure 3.2 Routing tree of CR_3.

From the simple routing algorithm. The returnable symbol of the routing path's nodes are all the same excepting identity node. Thus, we have the following properties:

Property 3.1 Returnable symbols of each subtree's nodes are all the same.

Property 3.2 Returnable symbols of different subtree's nodes are not alike.

Property 3.3 For arbitrary node x in CR_n, returnable symbols of xg and $xf_k g$ are not alike where $1 \le k \le n-1$.

Property 3.4 Returnable symbols of identity node's in-degree incident nodes are not alike.

Property 3.5 Paths of different subtree are node-disjoint excepting identity node.

The Fault_Tolerant Static Routing algorithm can tolerate at least $n-1$ node faults and the faulty nodes are incident identity node or source node. The routing distance at most increases by 6 steps than optimal routing distance.

Algorithm Fault_Tolerant Static Route.

1 Step 1 Find the symbol $a_j \ne$ {returnable symbol$|$ for {incident nodes of identity node and Sg are faulty}$\cup$ {$Sf_k g \mid Sf_k$ are faulty nodes} where S is source node and $1 \le k \le n-1$} where $1 \le j \le n$;

2 Step 2 If $a_j = \bar{t}_*$ then { traverse along $f_j - edge$; Simple_Route();}

3 else { traverse along $f_{j-1} - edge$; traverse along g-edge; Simple_Route();}

Example 3.6 Consider the node $b\bar{c}d\bar{e}\bar{a}$ in CR_5, and let faulty node set = {$\bar{c}d\bar{e}\bar{a}\bar{b}, \bar{c}d\bar{e}\bar{a}b, \bar{e}\bar{a}b\bar{c}d, \bar{e}abcd$}. Hence, {{incident nodes of identity node and Sg are faulty}$\cup$ {$Sf_k g \mid Sf_k$ are faulty nodes}} = {$\bar{c}d\bar{e}\bar{a}\bar{b}$, $d\bar{e}\bar{a}bc$, $\bar{a}b\bar{c}de$, $\bar{e}abcd$}. The fault-tolerant routing is $b\bar{c}d\bar{e}\bar{a} \to d\bar{e}\bar{a}b\bar{c} \to \bar{e}\bar{a}b\bar{c}\bar{d} \to \bar{a}b\bar{c}\bar{d}e \to$ $b\bar{c}\bar{d}ea \to \bar{c}\bar{d}eab \to \bar{d}eabc \to eabcd \to abcde$.

Lemma 3.4 The algorithm of Fault_Tolerant Static Route correctly computes a path from an arbitrary node $a_1 a_2 \cdots a_n$ to the identity node I.

Proof. From Property 3.3 and Property 3.4, the Fault_Tolerant Static Route algorithm correctly computes a path from an arbitrary node $a_1 a_2 \cdots a_n$ to the identity node I. Follow from Lemma 3.1, the integer k always exists (step 1) since the node $a_1 a_2 \cdots a_n$ is a cyclic permutation of the lexicographic ordering of n symbols (each symbol is in complemented or uncomplemented form). Such a move obviously exists for each iteration and at the end of the algorithm. Thus we can reach the destination node $t_1 t_2 \cdots t_n$. □

Theorem 3.4 The algorithm of Fault_Tolerant Static Route can tolerate at least $n-1$ node faults and the routing distance at most increases by 6 steps than optimal routing distance.

Proof. Refer to the routing tree in Figure 3.1. We can tolerate at least $n-1$ node faults in the tree. For an arbitrary node u, it can route to the identity node I via a subtree which has no faulty node. Refer to the algorithm of Fault_Tolerant Static Route. The worst case in line 4 needs one step and line 5 needs one step. The line 5 leads to k value increasing one according to Observation 3.1. $a_n = \bar{t}_*$ and $a_n \ne \bar{t}_n$ make to distance increase two according to Observation 3.1. If simple algorithm need $k-1$ steps. Consequently, the fault tolerant algorithm at most increases by 5 steps than the simple routing algorithm. Because Lemma 3.2, the fault tolerant algorithm at most increases by 6 steps than the optimal routing algorithm. □

4. Conclusions

The complete-rotation network has smaller diameter than that of the trivalent Cayley graph and the degree four Cayley graph. In particular, when we construct the network using the same nodes as those of a hypercube, the complete-rotation has smaller degree and diameter than those of the hypercube. Tables 4.1 and 4.2 show comparison among complete-rotation graph, cube-connected-cycle, trivalent Cayley graph, degree four Cayley graph, and hypercube. Regardless of its small diameter can be executed in complete-rotation with the same time complexity as in hypercube, mesh, cube-connected-cycles, trivalent Cayley graph and degree four Cayley graph. In addition, we can find that the degree, and diameter of the complete-rotation graph is smaller than those of the hypercube when these two graphs have the same number of vertices. Thus the complete-rotation graph is a good alternative for massively parallel system.

Table 4.1 A comparative study of complete-rotation graph, cube-connected-cycles, trivalent Cayley graph and degree four Cayley graph

	Complete-rotation graph	Cube-connected-cycles	Trivalent Cayley Graph	Degree Four Cayley Graph
dimension	n	n	n	n
vertices	$n\times 2^n$	$n\times 2^n$	$n\times 2^n$	$n\times 2^n$
degree	$n+1$	3	3	4
out-degree	n	3	3	4
edges	$n\times 2^n+\frac{n(n-1)}{2}\times 2^n$	$3n\times 2^{n-1}$	$3n\times 2^{n-1}$	$n\times 2^{n+1}$
diameter	$n+1$	6 if $n=3$; $2n+\left\lfloor\frac{n}{2}\right\rfloor-2$ if $n>3$	$2n-1$	$\left\lfloor\frac{3n}{2}\right\rfloor$
bisection width	2^n	-	-	-

Table 4.2 A comparative study of complete-rotation graph and hypercube

	Complete-rotation graph	Hypercube	Complete-rotation graph	Hypercube	Complete-rotation graph	Hypercube
dimension	n	$n+\log n$	4	6	8	11
vertices	$n\times 2^n$	$n\times 2^n$	64		2048	
degree	$n+1$	$n+\log n$	5	6	9	11
out-degree	n	$n+\log n$	4	6	8	11
edges	$n\times 2^n+\frac{n(n-1)}{2}\times 2^n$	$(n+\log n)\times 2^{n+\log n-1}$	160	192	9216	11264
diameter	$n+1$	$n+\log n$	5	6	9	11
bisection width	2^n	$2^{n+\log n-1}$	16	32	256	1024
fault tolerant routing	at least tolerant $n-1$ node faults and at most add 6 steps	-	at least tolerant 3 node faults and at most add 6 steps	-	at least tolerant 7 node faults and at most add 6 steps	-

5. References

[1] P. Vadapalli and P.K. Srimani, "A New Family of Cayley Graph Interconnection Networks of Constant Degree Four, "*IEEE Tran. on Parallel and Distributed Systems*, Vol. 7 No. 1, January pp.26-32, 1996.

[2] P. Vadapalli and P.K. Srimani, "Trivalent Cayley graph for interconnection networks, "*Information Processing Letters* 54, pp.329-335, 1995.

[3] P. Vadapalli and P.K. Srimani, "Shortest routing in trivalent Cayley graph network, "*Information Processing Letters* 57, pp.183-188, 1996.

[4] D.R. Duh, G.H. Chen, D.F. Hsu, "Combinatorial properties of generalized hypercube graphs,"*Information Processing Letters* 57, pp.41-45, 1996.

[5] S.B. Akers and B. Krishnamurthy, "A group-network theoretical model for Symmetrical interconnection network,"*IEEE Trans. Computers,* 38, 4, Apr., pp.555-566, 1989.

[6] S. Ponnuswamy and V. Chaudhary, "A Comparative Study Star Graphs And Rotator Graphs,"*International Conference on Parallel Processing*, pp.I32-I50, 1994.

[7] K. Qiu, S.G. Akl, and H. Meuer, "On Some Properties and Algorithms for the Star and Pancake Interconnection Networks,"*Journal of Parallel and Distributed Computing* 22, pp.16-25, 1994.

[8] S.G. Akl and K. Qiu, "Fundamental algorithms for the star and pancake interconnection networks with applications to computational geometry,"*Networks*. 23, pp.215-225, 1993.

[9] F.T. Leighton, "Introduction to Parallel Algorithms and Architectures: Array, Trees, Hypercubes,"*Morgan Kaufmann*, 1992.

[10] J. S. Jwo, S. Lakshmivarahan and C. Y. Hsieh, "On Relatives of Star Graph,"*Proceedings of International Computer Symposium*, pp.271-276, 1994.

[11] Peter F. Corbett, "Rotator graphs: an efficient topology for point-to -point multiprocessor networks,"*IEEE Trans. on Parallel and Distributed Systems*. Vol 3, No 5, Sep., pp.622-626, 1992.

[12] S. Lakshmivarahan, J.S. Jwo and S.K. Dhall, "Symmetry in interconnection networks based on Cayley graphs of permutation groups: a survey,"*Parallel Comput.* 19, pp.361-407, 1993.

[13] S.B. Akers and B. Krishnamurthy, "A group-network theoretical model for Symmetrical interconnection network,"*IEEE Trans. Computers*, 38, 4, Apr., pp.555-566, 1989.