Internode Distance and Optimal Routing in a Class of Alternating Group Networks

Baoxing Chen, Wenjun Xiao, and
Behrooz Parhami, *Fellow*, *IEEE*

Abstract—Alternating group graphs AG_n, studied by Jwo and others, constitute a class of Cayley graphs that possess certain desirable properties compared with other regular networks considered by researchers in parallel and distributed computing. A different form, AN_n, of such graphs, proposed by Youhou and dubbed alternating group networks, has been shown to possess advantages over AG_n. For example, AN_n has a node degree that is smaller by a factor of about 2 while maintaining a diameter comparable to that of AG_n, is maximally fault-tolerant, and shares some of the positive structural attributes of the well-known star graph. In this paper, we characterize the distance between any two nodes in AN_n and present an optimal (shortest-path) routing algorithm for this class of networks.

Index Terms—Alternating group graph, Cayley graph, diameter, interconnection network, internode distance, optimal routing, permutation group, regular graph, shortest-path routing.

1 Introduction

The important role played by Cayley graphs in the design and analysis of interconnection networks for parallel and distributed computing is now well understood [1], [3], [5]. Rings, toroids, hypercubes [4], [7], star graphs [1], [6], and alternating group graphs [2], [4], [9], [10] are all examples of Cayley graphs. Recently, Cayley graphs AG_n based on the alternating group A_n have been discussed in the literature [4], [9], [10]. Networks based on AG_n have a number of advantages over both the hypercube and the star graph. For example, alternating group graphs are Hamiltonian-connected, meaning that there is a Hamiltonian path between every pair of vertices. Additionally, these graphs are pancyclic, that is, they contain as subgraphs cycles of all possible lengths, from the minimum of 3 to the maximum of n (Hamiltonian cycle).

The class AN_n of alternating group networks proposed by Youhu [10] differs from the class AG_n introduced by Jwo et al. [4]. The new alternating group networks are also Cayley graphs and are thus vertex-symmetric. The diameters of AN_n and AG_n are comparable; however, the node degree of AN_n is only about half that of AG_n. Furthermore, the new graphs are maximally fault-tolerant [8] and share some of the positive structural attributes of the well-known star graphs.

The derivation of a simple and efficient routing algorithm, preferably one that is optimal (i.e., it routes via shortest paths) and has other desirable properties such as robustness, adaptivity, and freedom from deadlocks, is a requirement for practical application of any new class of interconnection networks [7]. A routing scheme (described in the Appendix) was proposed for the new alternating group networks AN_n in [10]. However, this routing scheme was not proven correct and has obvious flaws in that it does not find a path between some pairs of nodes in certain alternating group networks. For example, it fails to find a path from node 3125476 to the identity node 1234567 in AN_7 (the algorithm and a justification for the aforementioned claim can be found in the Appendix). Thus, the problem of optimal routing for AN_n remains open. In this paper, we fill this void by characterizing the distance between any two nodes in AN_n and presenting a provably optimal (shortest-path) routing algorithm for this class of networks.

- B. Chen and W.J. Xiao are with the Department of Computer Science, South China University of Technology, Guangzhou 510641, P.R. China. B. Chen is also with the Department of Computer Science, Zhangzhou Teachers' College, Zhangzhou 363000, P.R. China.
- B. Parhami is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106. E-mail: parhami@ece.ucsb.edu.

Manuscript received 6 June 2005; revised 17 June 2006; accepted 7 July 2006; published online 20 Oct. 2006.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0191-0605.

2 Background and Definitions

This section is devoted to introducing background material and notational conventions needed to understand the rest of the paper. Let G be a finite group with e as its identity element. Let $S \subset G$ be a generator set of G such that $e \notin S$ and $g^{-1} \in S$ if $g \in S$. The Cayley graph $Cay(G, S)$, defined as having G as its vertex set and $E = \{(x, y) \mid y = xg \text{ for some } g \in S\}$ as its edge set, is a regular vertex-transitive graph of node degree $|S|$. Many important and extensively studied interconnection networks, such as rings, toroids, hypercubes, star graphs, and alternating group graphs, are Cayley graphs.

Define $\langle n \rangle$ as the set $\{1, 2, \ldots, n\}$ and let $p = p_1\, p_2\, \ldots\, p_n$ be a permutation of the elements of $\langle n \rangle$, that is, $p_i \in \langle n \rangle$ and $p_i \neq p_j$ for $i \neq j$. We may refer to p_i as $p(i)$ or the ith element of p or use square brackets to delineate elements of a permutation when simply juxtaposing them would lead to ambiguities; for example, the permutation that swaps the first and ith elements may be written as $g_{1i} = i\ 2\ 3\ \ldots [i-1]1[i+1] \ldots\ n$. For two permutations σ and δ, we define their product, or composition, $\sigma\delta$, as a permutation that satisfies $\sigma\delta(i) = \sigma(\delta(i))$. In other words, if $\sigma = \sigma_1\sigma_2 \ldots \sigma_n$ and $\delta = \delta_1\delta_2 \ldots \delta_n$, we have $\sigma\delta = \sigma(\delta_1)\sigma(\delta_2) \ldots \sigma(\delta_n)$. Any permutation can be obtained from the identity permutation $e = 1\,2 \ldots n$ by a sequence of transpositions (exchanging two adjacent elements). An even (odd) permutation is one whose derivation in this manner requires an even (odd) number of steps. A transposition step applied to an even permutation produces an odd permutation and vice versa. Thus, there are $n!/2$ even (odd) permutations of $\langle n \rangle$.

Let A_n denote the set of all even permutations over $\langle n \rangle$. The set A_n, along with the operation of product (composition), constitutes a group. Let g_{kl} be the permutation that swaps elements in positions k and l and leaves all other elements undisturbed. It is easy to see that g_{kl} is an odd permutation. Define two even permutations, g_i^+ and g_i^-, as follows for $i \geq 3$:

$$g_i^+ = g_{2i}g_{12} \qquad g_i^- = g_{1i}g_{12}.$$

In other words, g_i^+ swaps the element in position 2 with the element in position i after first transposing elements 1 and 2, while g_i^- swaps the element in position 1 with the element in position i following the g_{12} transposition. Note that $g_i^+ g_i^- = e$, the identity permutation, so, g_i^+ and g_i^- are each other's inverses.

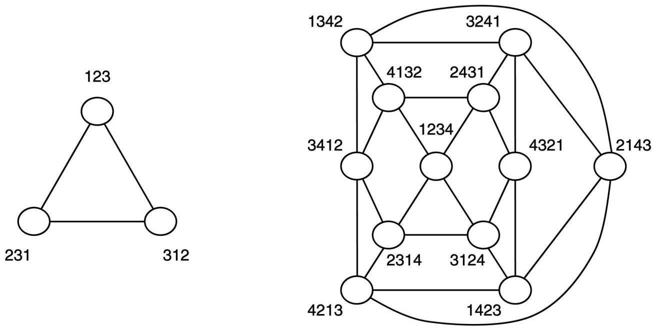
The alternating group graph AG_n is the Cayley graph $Cay(A_n, \Sigma)$ with the generator set:

$$\Sigma = \{g_3^+, g_3^-, g_4^+, g_4^-, \ldots, g_n^+, g_n^-\}.$$

The n-dimensional alternating group graph AG_n is a regular graph with $n!/2$ nodes, $n!(n-2)/2$ edges, node degree $2(n-2)$, and diameter $\lfloor 3n/2 \rfloor - 3$. Fig. 1 depicts AG_3 and AG_4.

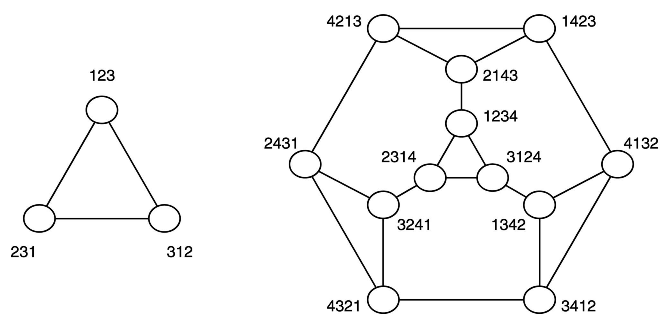
Similarly, the n-dimensional alternating group network AN_n is a regular graph with each of its nodes labeled with an even permutation of $\langle n \rangle$. Two nodes of AN_n are directly connected iff the label (permutation) of one node can be obtained from the label of the other by one of the following operations:

1. $z_i = g_{12}g_{3i}$, where $3 < i \leq n$: Swapping symbols 1 and 2 and symbols 3 and i.
2. g_L: Shifting the first (leftmost) three symbols cyclically to the left by one position.



(a) (b)

Fig. 1. Alternating group graphs (a) AG_3 and (b) AG_4.



(a) (b)

Fig. 2. Alternating group networks (a) AN_3 and (b) AN_4.

3. g_R: Shifting the first three symbols cyclically to the right by one position.

Clearly, the generator z_i is its own inverse, while g_L and g_R are each other's inverses. Thus, AN_n is a Cayley graph $\mathrm{Cay}(An, \Omega)$ with the generator set $\Omega = \{z_4, \ldots, z_n, g_L, g_R\}$. Fig. 2 depicts AN_3 and AN_4. From now on, we exclude the uninteresting case of AN_3, which is a 3-node ring, and focus on AN_n with $n > 3$.

The following facts about AN_n are obvious: It is a regular graph with $n!/2$ nodes, $n!(n-1)/4$ edges, and node degree $n-1$. We also know from [10] that AN_n is Hamiltonian and has a diameter of $\lceil 3n/2 \rceil - 3$, which is no more than one unit greater than the diameter $\lfloor 3n/2 \rfloor - 3$ of AG_n. As a Cayley graph, AN_n is vertex-symmetric. However, it is not edge-symmetric. The latter observation is easily verified by noting that the edge $(1\,2\,3\,4 \ldots n, 2\,3\,1\,4\ \ldots\ n)$ is in a cycle of length 3, whereas the edge $(1\,2\,3\,4\ \ldots [n-1]\,n, 2\,1\,n\,4\ \ldots [n-1]\,3)$ belongs to no such cycle. For example, in Fig. 2b, the edge $(1234, 2143)$ is not part of a cycle of length 3.

3 Finding Internode Distances

A permutation $p = p_1\,p_2 \ldots p_n$ of the elements in $\langle n \rangle$ can be represented by its cycle structure, that is, $p = c_1\ c_2 \ldots c_k\ e_1\ e_2 \ldots e_l$, where c_i is a cycle of length $|c_i| \geq 2$ for $1 \leq i \leq k$ and e_i is an invariant for $1 \leq i \leq l$. Thus, $n = |c_1| + |c_2| + \ldots + |c_k| + l$. For brevity, we may refrain from listing the invariants in the canonical cycle representation of p. For example, if $p = 5\ 1\ 2\ 4\ 3\ 7\ 6$, then the cycle structure of p can be given as (1 5 3 2) (6 7), omitting the invariant (4). Note that a cycle such as (1 5 3 2) indicates that the first element is replaced by the fifth, the fifth element is replaced by the third, the third element is replaced by the second, and, wrapping around, the second element is replaced by the first. We use $d(p,q)$ to denote the length of a shortest path from node p to node q. As before, e denotes the identity permutation $1\ 2\ 3 \ldots n$. Given a node p whose label contains a cycle c and a different node q with a label in which the elements of c are invariants (each element x appears in the corresponding position x), we say that, in going from p to q, the cycle c has been sorted. This means that c does not appear in the canonical cycle structure of q. Sorting of cycles constitutes the building block of routing algorithms, as will become clear in the rest of this paper.

Lemma 1. *Let p be a node of AN_n having the cycle structure $(1\ 2)\ c_2 \ldots c_k$, that is, $c_1 = (1\ 2)$. The cycle c_1 automatically gets sorted when all the other cycles in p have been sorted.*

Proof. Divide the cycles of p into even and odd cycles. Because p is an even permutation, it must have an even number of even cycles. So, there are an odd number of even cycles other than c_1. Sorting each odd cycle will leave the elements 1 and 2 in their original out-of-order state, given that they undergo an even number of transpositions. Sorting each even cycle, on the other hand, will transpose the elements 1 and 2 an odd number of times. After an odd number of such reorderings to sort all the other even cycles, 1 and 2 must appear in sorted order. Hence, once all the other cycles in p have been sorted, the cycle c_1 must be sorted as well. □

Lemma 2. *For a node p of AN_n, let the canonical cycle structure be $c_1\ c_2 \ldots c_k$ and define $m = |c_1| + |c_2| + \ldots + |c_k|$. We derive $d(p,e)$ in the two cases when 3 is or is not an invariant.*

1. *If 3 is an invariant, then the distance $d(p,e)$ from node p to the identity node e is:*

$$
\begin{aligned}
d(p,e) &= m+k && \text{if } p_1 = 1 \text{ and } p_2 = 2\\
&= m+k-3 && \text{if } p_1 = 2 \text{ and } p_2 = 1\\
&= m+k && \text{if } |\{p_1,p_2\} \cap \{1,2\}| = 1,\\
& && \text{and 1 or 2 is an invariant}\\
&= m+k-1 && \text{if } |\{p_1,p_2\} \cap \{1,2\}| = 1,\\
& && \text{and 1 and 2 belong to the same cycle } c_i\\
&= m+k && \text{if } |\{p_1,p_2\} \cap \{1,2\}| = 0,\\
& && \text{and 1 and 2 belong to the same cycle } c_i\\
&= m+k-1 && \text{if } |\{p_1,p_2\} \cap \{1,2\}| = 0,\\
& && \text{and 1 and 2 belong to different cycles.}
\end{aligned}
$$

2. *If 3 is not an invariant, then $d(p,e)$ is two units less than the expressions given above.*

Proof. Case 1: 3 is an invariant, that is, $p_3 = 3$.

Subcase a: $p_1 = 1$ and $p_2 = 2$. To sort any cycle c_i, we need $|c_i| + 1$ steps. Thus, $d(p,e) = |c_1| + |c_2| + \ldots + |c_k| + k = m + k$.

Subcase b: $p_1 = 2$ and $p_2 = 1$, that is, the cycle structure of p includes $c_1 = (1\ 2)$. From Lemma 1, we know that we do not need to sort the cycle $c_1 = (1\ 2)$. To sort a cycle c_i other than $(1\ 2)$, we need $|c_i| + 1$ steps. Thus, $d(p,e) = |c_2| + \ldots + |c_k| + k - 1 = m + k - 3$.

Subcase c: $|\{p_1,p_2\} \cap \{1,2\}| = 1$ and 1 or 2 is an invariant. For simplicity, we assume that 2 is an invariant and 1 is in the cycle $c_i = (1\ p_1 \ldots)$. To sort this cycle, that is, to put those elements in c_i which are larger than 1 in their correct positions, we need $|c_i| + 1$ steps. Thus, $d(p,e) = |c_1| + |c_2| + \ldots + |c_k| + k = m + k$.

Subcase d: $|\{p_1,p_2\} \cap \{1,2\}| = 1$ and 1 and 2 belong to the same cycle c_i. For simplicity, we assume that 1 and 2 are in the cycle $c_i = (1\ 2\ p_2 \ldots)$. To sort this cycle, that is, to put the elements in c_i which are larger than 2 in their correct positions and the elements {1, 2} in positions 1 and 2, we need $|c_i|$ steps. Thus, $d(p,e) = |c_1| + |c_2| + \ldots + |c_k| + k - 1 = m + k - 1$.

Subcase e: $|\{p_1,p_2\} \cap \{1,2\}| = 0$, and 1 and 2 belong to the same cycle c_i. Let 1 and 2 be in the cycle $c_i = (1\ p_1\ \ldots 2\ p_2 \ldots)$. To sort this cycle, we need $|c_i| + 1$ steps. Thus, $d(p,e) = |c_1| + |c_2| + \ldots + |c_k| + k = m + k$.

Subcase f: $|\{p_1, p_2\} \cap \{1, 2\}| = 0$ and 1 and 2 belong to different cycles. Let 1 be in the cycle $c_i = (1\, p_1 \ldots)$ and 2 in the cycle $c_j = (2\, p_2 \ldots)$. To sort these two cycles, we need $|c_i| + |c_j| + 1$ steps. Thus,

$$d(p, e) = |c_1| + |c_2| + \ldots + |c_k| + k - 1 = m + k - 1.$$

Case 2: 3 is not an invariant, that is, $p_3 \neq 3$. We begin by sorting the cycle $c_j = (3\, p_3 \ldots)$ in $|c_j| - 1$ steps. The rest of the process parallels that of Case 1.

Subcase a: $p_1 = 1$ and $p_2 = 2$. To sort any cycle c_i that does not contain 3, we need $|c_i| + 1$ steps. Thus, $d(p, e) = |c_1| + |c_2| + \ldots + |c_k| + k - 2 = m + k - 2$.

Subcase b: $p_1 = 2$ and $p_2 = 1$, that is, the cycle structure of p includes $c_1 = (12)$. From Lemma 1, we know that we do not need to sort the cycle $c_1 = (1\,2)$. To sort any other cycle c_i that does not include 3, we need $|c_i| + 1$ steps. Thus, $d(p, e) = |c_2| + \ldots + |c_k| + k - 3 = m + k - 5$.

Subcase c: $|\{p_1, p_2\} \cap \{1, 2\}| = 1$ and 1 or 2 is an invariant. For simplicity, we assume that 2 is an invariant and 1 is in the cycle $c_i = (1\, p_1 \ldots)$. To sort this cycle, we need $|c_i| + 1$ steps when c_i does not include 3 and we need $|c_i| - 1$ steps when it does. Thus, $d(p, e) = |c_1| + |c_2| + \ldots + |c_k| + k - 2 = m + k - 2$.

Subcase d: $|\{p_1, p_2\} \cap \{1, 2\}| = 1$ and 1 and 2 belong to the same cycle c_i. For simplicity, we assume that 1 and 2 are in the cycle $c_i = (1\; 2\; p_2 \ldots)$. To sort this cycle, that is, to put the elements in c_i which are larger than 2 in their correct positions and the elements {1, 2} in positions 1 and 2, we need $|c_i|$ steps when c_i does not include 3 and we need $|c_i| - 2$ steps when it does. Thus, $d(p, e) = |c_1| + |c_2| + \ldots + |c_k| + k - 3 = m + k - 3$.

Subcase e: $|\{p_1, p_2\} \cap \{1, 2\}| = 0$ and 1 and 2 belong to the same cycle c_i. Let 1 and 2 be in the cycle $c_i = (1\, p_1 \ldots 2\, p_2 \ldots)$. To sort this cycle, we need $|c_i| + 1$ steps when c_i does not include 3 and $|c_i| - 1$ steps when it does. Thus, $d(p, e) = |c_1| + |c_2| + \ldots + |c_k| + k - 2 = m + k - 2$.

Subcase f: $|\{p_1, p_2\} \cap \{1, 2\}| = 0$ and 1 and 2 belong to different cycles. Let 1 be in the cycle $c_i = (1\, p_1 \ldots)$ and 2 in the cycle $c_j = (2\, p_2 \ldots)$. To sort these two cycles, we need $|c_i| + |c_j| + 1$ steps when $c_i \cup c_j$ does not include 3 and we need $|c_i| + |c_j| - 1$ steps when it does. Thus, $d(p, e) = |c_1| + |c_2| + \ldots + |c_k| + k - 3 = m + k - 3$. □

4 An Optimal Routing Algorithm

In this section, we provide, and prove correct, a shortest-path routing algorithm $Route(p)$ for AN_n. Because AN_n is vertex-symmetric, a path from any node x to an arbitrary node y can be easily deduced, given a path from $p = y^{-1}\, x$ to the identity node $e = 1\,2\,3 \ldots n$. The latter is the path that our algorithm provides.

$Route(p = p_1\, p_2\, \ldots p_n)$: returns p', the first node on a shortest path from p to e

Case 1: if $p_3 > 3$ then $p' = p\, z_{p_3}$ endif
Case 2: if $p_3 = 3$
    then if $p_4 = 4, p_5 = 5, \ldots, p_n = n$ then stop endif
    else $p' = p\, z_t$, where $t > 3$ and $p_t \neq t$
    endif
Case 3: if $p_3 < 3$
    then if $p_1 < 3$ or $p_2 < 3$
        then if $p_1 < 3$ then $p' = p\, g_R$ else $p' = p\, g_L$ endif
        endif
    else if ($p_3 = 1$ and 2 is not in the cycle $(3\,1 \ldots)$)
            or ($p_3 = 2$ and 1 is in the cycle $(3\,2 \ldots)$)
        then $p' = p\, g_R$
        else $p' = p\, g_L$
        endif
    endif

Note that, in Algorithm *Route*, the notation "cycle $(i\, j \ldots)$" represents the cycle that contains both i and j and $p\, g_R$ or $p\, g_L$ is the group product of p and one of the generators g_R or g_L, defined in Section 2.

Example 1. A shortest path from $p = 4\,5\,1\,2\,3$ to $e = 1\,2\,3\,4\,5$ in AN_5:

$$\begin{array}{ll} 4\,5\,1\,2\,3 & \to g_L \\ 5\,1\,4\,2\,3 & \to z_4 \\ 1\,5\,2\,4\,3 & \to g_R \\ 2\,1\,5\,4\,3 & \to z_5 \\ 1\,2\,3\,4\,5 & \end{array}$$

Example 2. A shortest path from $p = 3\,1\,2\,5\,4\,7\,6$ to $e = 1\,2\,3\,4\,5\,6\,7$ in AN_7:

$$\begin{array}{ll} 3\,1\,2\,5\,4\,7\,6 & \to g_L \\ 1\,2\,3\,5\,4\,7\,6 & \to z_4 \\ 2\,1\,5\,3\,4\,7\,6 & \to z_5 \\ 1\,2\,4\,3\,5\,7\,6 & \to z_4 \\ 2\,1\,3\,4\,5\,7\,6 & \to z_6 \\ 1\,2\,7\,4\,5\,3\,6 & \to z_7 \\ 2\,1\,6\,4\,5\,3\,7 & \to z_6 \\ 1\,2\,3\,4\,5\,6\,7 & \end{array}$$

Theorem 1. *The algorithm $Route(p)$ is optimal.*

Proof. We only need to prove that $d(p', e) = d(p, e) - 1$. Suppose that the cycle structure of p is $c_1\, c_2 \ldots c_k$, where $|c_i| \geq 2$ for $1 \leq i \leq k$. Let $m = |c_1| + |c_2| + \ldots + |c_k|$. Suppose that the cycle structure of p' is $h_1\, h_2 \ldots h_{k'}$, where $|h_i| \geq 2$ for $1 \leq i \leq k'$. Let $m' = |h_1| + |h_2| + \ldots + |h_{k'}|$. The three cases identified in the algorithm must be considered: 1) $p_3 > 3$, 2) $p_3 = 3$, and 3) $p_3 < 3$. In the following, we only consider the case of $p_3 < 3$ as the first two cases can be proven similarly. Furthermore, because $p_3 = 2$ and $p = (3\,2 \ldots)c_2 \ldots c_k$ is similar to $p_3 = 1$ and $p = (3\,1 \ldots)c_2 \ldots c_k$, we only supply a proof for the latter case.

Case a: If $|\{p_1, p_2\} \cap \{1, 2\}| = 1$ and 1 or 2 is an invariant, then 2 is an invariant, $d(p, e) = m + k - 2$ and $p' = p\, g_L$.

Subcase a1: If $p_1 = 3$, then $p' = (1\,2)c_2 \ldots c_k$. By Lemma 2, we have

$$d(p', e) = 2 + |c_2| + \ldots + |c_k| + k - 3 = m + k - 3 = d(p, e) - 1.$$

Subcase a2: If $p_1 > 3$, then $p' = (1\,2)(3\, p_1\, \ldots)c_2 \ldots c_k$. By Lemma 2, we have

$$\begin{aligned} d(p', e) &= m' + k' - 5 = m + 1 + k + 1 - 5 \\ &= m + k - 3 = d(p, e) - 1. \end{aligned}$$

Case b: If $|\{p_1, p_2\} \cap \{1, 2\}| = 1$ and 1 and 2 belong to the same cycle c_i, then $p_1 = 2, d(p, e) = m + k - 3$ and $p' = p\, g_R$.

Subcase b1: If $p_2 = 3$, then $p' = c_2 \ldots c_k$. By Lemma 2, we have $d(p', e) = m' + k' = m - 3 + k - 1 = d(p, e) - 1$.

Subcase b2: If $p_2 > 3$, then $p = (3\,1\,2\, p_2 \ldots)c_2 \ldots c_k$, $p' = (3\, p_2 \ldots)c_2 \ldots c_k$. By Lemma 2, we have

$$d(p', e) = m' + k' - 2 = m - 2 + k - 2 = m + k - 4 = d(p, e) - 1.$$

Case c: If $|\{p_1, p_2\} \cap \{1, 2\}| = 0$ and 1 and 2 belong to the same cycle c_i, then $p_1 > 2, p_2 > 2, d(p, e) = m + k - 2$ and $p' = p\, g_L$.

Subcase c1: If $p_2 = 3$, then $p = (3\ 1\ p_1 \ldots 2)c_2 \ldots c_k$. Thus, $p' = (1\ 3\ p_1 \ldots 2)c_2 \ldots c_k$. By Lemma 2, we have

$$d(p', e) = m' + k' - 3 = m + k - 3 = d(p, e) - 1.$$

Subcase c2: If $p_2 > 3$, then $p_1 > 3$,

$$p = (3\ 1\ p_1 \ldots 2\ p_2 \ldots j)c_2 \ldots c_k.$$

Thus, $p' = (3\ p_1 \ldots 2\ 1\ p_2 \ldots j)c_2 \ldots c_k$. By Lemma 2, we have

$$d(p', e) = m' + k' - 3 = m + k - 3 = d(p, e) - 1.$$

Case d: If $|\{p_1, p_2\} \cap \{1, 2\}| = 0$ and 1 and 2 belong to different cycles, then we have $p_1 > 2$, $p_2 > 2$, $p = (3\ 1\ p_1 \ldots i)(2\ p_2 \ldots j)c_3 \ldots c_k, d(p, e) = m + k - 3$ and $p' = p\ g_R$. Thus, $p' = (3\ p_2 \ldots j\ 2\ p_1 \ldots i)c_3 \ldots c_k$ and 1 is an invariant of p'. By Lemma 2, we have

$$d(p', e) = m' + k' - 2 = m + k - 4 = d(p, e) - 1.$$

□

5 Conclusion

Given that the design space for interconnection networks is quite vast, identifying structures that lend themselves to theoretical analyses constitutes an important area of research. Cayley graphs in general, and networks based on alternating groups in particular, provide a rich area of study for analyzing both well-known and new interconnection networks. In this paper, we derived the distance between any pair of nodes in the alternating group networks AN_n and used the results in the construction and proof of an optimal (shortest-path) routing algorithm for this new class of networks. In so doing, we solved an open problem due to the lack of a complete, and provably correct, routing algorithm for AN_n. We have also built a foundation for further studies of these and other Cayley graphs based on alternating groups that offer advantages over well-known, and extensively studied, interconnection networks such as the hypercube and the star graph.

The work reported here can be extended in several directions. Some important open problems include: 1) enumerating and constructing node-disjoint, or parallel, paths for the evaluation of fault tolerance, 2) finding the exact value of the network's fault diameter to assess its robustness, and 3) deriving a fault-tolerant routing algorithm to allow continued operation in the presence of faults. We are now investigating some aspects of these and other problems of practical interest.

Appendix

A Routing Algorithm for AN_n

The following routing algorithm for AN_n, called *YRoute*, is reproduced from [10] for comparison purposes and for verifying our claim that it fails to find a route in some cases. The algorithm *YRoute* supplies a route from node p in AN_n whose cycle structure is $c_1\ c_2 \ldots c_k$ to node e.

YRoute($p = c_1\ c_2 \ldots c_k$): p' is the first node on a shortest path from p to e

Step 1: if $p_3 = 3$
  then if p contains a cycle $(i_1\ i_2 \ldots i_s)$ with $s \geq 2$ and $i_j > 3$
    then $p' = p\ z_{i_1}$
    else $p' = p(132)$ or $p' = p(123)$
      /* there must be a cycle $(i_1 i_2 \ldots i_s\ 1 \ldots)$
      or $(i_1\ i_2 \ldots i_s\ 2 \ldots)$ with $s \geq 1$ and $i_j > 3$ */
    endif
  else /* $p_3 \neq 3$ */
    if p contains a cycle $(i_1\ i_2 \ldots i_s\ 3 \ldots)$ with $s \geq 1$ and
      $i_j > 3$
    then $p' = p\ z_{i_s}$
    endif
    if p contains $(i_1\ i_2 \ldots i_s\ 1\ 3 \ldots)$ (2) or
      $(i_1\ i_2 \ldots i_s\ 2\ 3 \ldots)$ (1) with $s \geq 1$ and $i_j > 3$
    then $p' = p(132)$ or $p' = p(123)$
    endif
    if p contains $(1\ 3 \ldots)$ $(2\ i_1\ i_2 \ldots i_s)$ or $(2\ 3 \ldots)$
      $(1\ i_1\ i_2 \ldots i_s)$ with $s \geq 1$ and $i_j > 3$
    then $p' = p(123)$ or $p' = p(132)$
    endif
  endif

Step 2: One by one, place the elements $i_s, i_{s-1}, \ldots, i_1$ in p' at their correct positions

Step 3: Sort the cycles $c_1, c_2, \ldots, c_k$ in turn by using Steps 1 and 2 as required;
Finally, put the elements 1, 2, and 3 in their correct positions.

Now, consider the example of routing from the node $p = 3125476$ to the identity node 1234567 in AN_7, cited in Section 1. Thus, $p_3 = 2$ and the cycle expression of p is (1 3 2) (4 5) (6 7). Because $p_3 \neq 3$, p does not satisfy any of the following conditions:

1. containing a cycle $(i_1\ i_2 \ldots i_s\ 3 \ldots)$ with $s \geq 1$ and $i_j > 3$,
2. containing cycles $(i_1\ i_2 \ldots i_s\ 1\ 3 \ldots)$ (2) or $(i_1\ i_2 \ldots i_s\ 2\ 3 \ldots)$ (1) with $s \geq 1$ and $i_j > 3$,
3. containing cycles $(1\ 3 \ldots)$ $(2\ i_1\ i_2 \ldots i_s)$ or $(2\ 3\ \ldots)$ $(1\ i_1\ i_2 \ldots i_s)$ with $s \geq 1$ and $i_j > 3$.

Algorithm *YRoute* fails to compute p'. In other words, the three subcases under the "else" part of Step 1 (corresponding to $p_3 \neq 3$) do not cover all possibilities. Clearly, the example above is not the only one for which Algorithm *YRoute* fails to provide a routing path. Routing from any node whose cycle structure contains (1 3 2) will create the same problem.

Acknowledgments

This research was supported in part by the Natural Science Foundation of China and Fujian Province (No. F0110012) and by the Scientific Research Foundation of Fujian Provincial Education Department (No. JA03144).

References

[1] S.B. Akers and B. Krishnamurthy, "A Group-Theoretic Model for Symmetric Interconnection Networks," *IEEE Trans. Computers*, vol. 38, pp. 555-566, 1989.
[2] E. Cheng and M.J. Lipman, "Orienting Split-Stars and Alternating Group Graphs," *Networks*, vol. 35, no. 2, pp. 139-144, 2000.
[3] Y.O. Hamidoune, A.S. Llado, and O. Serra, "The Connectivity of Hierarchical Cayley Digraphs," *Discrete Applied Math.*, vols. 37/38, pp. 175-180, 1992.
[4] J.S. Jwo, S. Lakshmivarahan, and S.K. Dhall, "A New Class of Interconnection Networks Based on the Alternating Groups," *Networks*, vol. 23, pp. 315-326, 1993.
[5] S. Lakshmivarahan, J.S. Jwo, and S.K. Dhall, "Symmetry in Interconnection Networks Based on Cayley Graphs of Permutations: A Survey," *Parallel Computing*, vol. 19, pp. 361-407, 1993.
[6] S. Latifi, "On the Fault-Diameter of the Star Graph," *Information Processing Letters*, vol. 46, pp. 143-150, 1993.
[7] B. Parhami, *Introduction to Parallel Processing: Algorithms and Architectures.* Plenum Press, 1999.
[8] D.K. Pradhan and F.J. Meyer, "Communication Structures in Fault-Tolerant Distributed Systems," *Networks*, vol. 23, pp. 379-389, 1993.
[9] Y. Roichman, "Expansion Properties of Cayley Graphs of the Alternating Groups," *J. Combinatorial Theory*, vol. 79(A), pp. 281-279, 1997.
[10] J. Youhu, "A New Class of Cayley Networks Based on the Alternating Groups," *Applied Math.—JCU*, vol. 14(A), no. 2, pp. 235-239, 1998 (in Chinese).