Pancake Problems with Restricted Prefix Reversals and some Corresponding Cayley Networks

Submitted To ICPP '98
The 27th International Conference on Parallel Processing

Douglas W. Bass and I. Hal Sudborough
Computer Science Program
The University of Texas at Dallas
Richardson, Texas USA

Please direct all correspondence to:

I. Hal Sudborough
Computer Science Program, EC 3.1
The University of Texas at Dallas
Box 830688
Richardson, Texas 75083-0688
USA

Phone: (972)-883-2184
Fax: (972)-883-2349
e-mail: hal@utdallas.edu

Abstract:

The pancake problem, which has attracted considerable attention [10, 12, 9, 15], concerns the number of prefix reversals or *flips* needed to sort the elements of an arbitrary permutation. The number of prefix reversals to sort permutations is also the diameter of the often studied n-dimensional Pancake network P_n [3, 4, 5, 8, 14, 16].

We consider restricted pancake problems, for example when only 3 of the possible n - 1 flips are allowed. Let f_i denote a flip of size i. Each flip is itself a permutation. For example, a flip of size 4, *i. e.,* f_4, on eight symbols has the effect of changing, say, 3 5 1 2 4 6 8 7 into 2 1 5 3 4 6 8 7. f_4 is the permutation 4 3 2 1 5 6 7 8. We investigate sets of permutations corresponding to flips as generators of the symmetric group S_n. Let n be the number of symbols in a permutation. We consider sets with either a constant number of generators (*i. e.,* flips) or with $\log_2 n$ generators. In special interesting cases, the corresponding Cayley networks, defined by a given set of generators and a given group of permutations, are explored. Specifically, we investigate two special families of networks:

1) The **Subcube$_n$** network, for $n = 2^k$, defined by the $\log_2 n$ generators in the set $\{f_2, f_4, f_8 \dots f_n\}$. We prove that:
    - Subcube$_n$ is isomorphic to a network obtained from an (n - 1) dimensional hypercube, Q_{n-1}, by deleting all but $\log_2 n$ of the edges incident to each of its nodes.
    - Subcube$_n$ has diameter $(3n / 2) - 2$
    - Although Subcube$_n$ has $(\log_2 n) / (n - 1)$ as many edges as Q_{n-1}, routing in Subcube$_n$ can be done by an optimal routing algorithm and via paths at worst 50% longer than optimum routes in Q_{n-1}
    - Q_{n-1} can be embedded into Subcube$_n$ with nearly optimum dilation
    - Subcube$_n$ is maximally fault tolerant
    - Subcube$_n$ can represent Q_k, for $1 \le k \le n - 1$

2) The **Triad** network of dimension n, where n is odd and $\lfloor n / 2 \rfloor$ mod 4 = 0, denoted by Triad_n, defined by the set of three generators $\{f_{\lfloor n/2 \rfloor}, f_{\lceil n/2 \rceil}, f_n\}$. We prove that:

   - Triad_n has one node for each permutation on n symbols and is regular of degree 3
   - Triad_n has diameter $\Theta(n \log_2 n)$, which is as good as any degree 3 network can have, at least to within a small constant factor.
   - Triad_n is maximally fault tolerant
   - The n-dimensional star network [2] can be emulated by Triad_n with linear slowdown
   - Both the n-dimensional shuffle-exchange [19], [25] and shuffle-exchange permutation [18] networks can be emulated by Triad_n with constant slowdown.

We show that any set of generators consisting of a constant number of flips requires $\Omega(n \log_2 n)$ to sort all permutations in S_n. We also describe necessary conditions on sets of generators to sort all permutations in $\Theta(n \log_2 n)$ steps.

Keywords: Pancake problem, sorting permutations, fixed-degree Cayley networks, diameter, routing, pruning, embeddings, shuffle-exchange network, shuffle-exchange permutation network.

I. Introduction

The pancake problem [10] concerns the number of prefix reversals or *flips* needed to sort the elements of a permutation. Let f(n) be the maximum number needed to sort any permutation on n symbols. The best bounds known for f(n) are $15n / 14 \le f(n) \le (5n + 5) / 3$ [9], [12]. A related problem, called the *burnt pancake problem*, [12], concerns the number of prefix reversals needed to sort *signed* permutations, where each symbol has an attached positive or negative sign and, each time the symbol is involved in a flip, the sign changes. Let g(n) be the maximum number of prefix reversals needed to sort any signed permutation on n symbols. The best bounds known for g(n) are $3n / 2 \le g(n) \le 2n - 3$ [9], [12]. A conjectured hardest signed permutation is -1 -2 -3 ... -n, denoted by $-I_n$ [9]. It is known that $-I_n$ can be sorted in $3(n + 1) / 2$ steps [14], so if $-I_n$ could be proved hardest, then both f(n) and g(n) would be bounded above by $3(n + 1) / 2$.

While O(n) flips are sufficient to sort using the set of all n - 1 possible flips, the degree of the associated pancake network is a linear function of the size of the permutations being sorted. Investigating the number of sorting steps needed with sets of only 3 flips is equivalent to investigating the diameter of Cayley networks of degree 3 on S_n and its subgroups. Let f_i represent a flip of size i. For example, a flip of size 4, *i. e.,* f_4, on eight symbols has the effect of changing, say, 3 5 1 2 4 6 8 7 into 2 1 5 3 4 6 8 7. f_4 is the permutation 4 3 2 1 5 6 7 8. Suppose we have a group G (for example, S_n), and a subset S of G (for example, a set of 3 flips). If every element of G can be generated as a finite product of the elements of S, then the elements of S are called *generators*, and S is called a *generating set* of G. We also say that S *generates* G. If G is a group and S generates G, then the *Cayley network Cay(G, S)* is a network where the nodes are the elements of G, and the edges are all ordered pairs (s, t) where t = sg, for some $g \in G$ and $s \in S$ [16]. Cayley graphs have been extensively studied [1], [6], [16] as bases for interconnection networks, due to their many desirable properties, including regularity, vertex-symmetry and recursive or near-recursive substructure. Recently, a number of Cayley networks of degree O(1) have been proposed. [23], [18], [17]. Some examples of fixed-degree Cayley networks are shown in Table 1, with diameter results in [24], [7] and [17].

Network	Introduced	# Nodes	Degree	Diameter
Trivalent Cayley Graph	**1995**	**$n2^n$**	**3**	**2n - 1**
Shuffle-Exchange Permutation Network	**1996**	**n!**	**3**	**$\Theta(n^2)$**
Incomplete k-ary n-cube, where k mod (n - 1) = 0	**1997**	**k^n**	**4**	**$\Theta(kn)$**

Table 1. Comparing recently introduced fixed-degree Cayley networks

If we find a set S of three flips that allows us to sort in O(f(n)) sorting steps, then we have found a Cayley network of degree 3 on S_n with O(f(n)) diameter. If we find a set of three flips that does not allow us to sort all permutations, then we have found a Cayley network of degree 3 on a subgroup of S_n. We will start by making observations about what must be true about sets of three flips that generate S_n.

II. Necessary conditions for a set of flips to generate all permutations.

Let f_i represent the flip of size i. In order for a set of flips to be able to generate all permutations, all of the following conditions must be satisfied: (These are not sufficient conditions)

1) One of the flips must be of size n. If this is not the case, then there is no way to move an element to, or from, position n.

2) One of the flips must be of size i, where $n / 2 < i < n$ if n is even, and $\lceil n / 2 \rceil < i < n$ if n is odd. If this is not the case, then there is no way to exchange an element originally in a position between 1 and n / 2, with an element originally in a position between (n / 2) + 1 and n.

3) One of the flips must be of size k, where k is even. If this is not the case, no element can be moved from an odd-numbered position to an even-numbered position, and vice versa.

4) One of the flips must correspond to an odd number of transpositions. If $\{f_i, f_j, f_n\}$ is a set of flips where each flip corresponds to an even number of transpositions, then the best the flips can do is generate the even permutations, *i. e.*, the alternating

group [22]. A flip f_i corresponds to an even number of transpositions when i mod 4 = 0 or i mod 4 = 1.

5) The flip sizes must be relatively prime. If k > 1 is the greatest common divisor of the flip sizes, then elements in blocks of k contiguous elements can never be separated from each other.

III. Subgroups of S_n generated by sets of 3 flips.

Suppose a set of 3 flips doesn't meet all the conditions listed above. If this is the case, then the set generates a proper subgroup of S_n, and the corresponding Cayley network has less than n! nodes. Table 2 shows the sizes of networks generated by sets of 3 flips where the largest flip size is 10, and the sets that generate them. The length of the permutation is the largest flip size of the generating set. It should be noted that the many of the networks mentioned in column 3 are of degree greater than 3.

Consider, as an example, the network generated by the set of flips $\{f_2, f_5, f_8\}$. Each node is represented by a permutation on 8 symbols. We write the 8 symbols as (a b)x(c d)y(e f), as none of the three flips ever separates the adjacencies between a and b, between c and d, and between e and f. A flip of size 2 reverses the order of the elements in whichever block occupies positions 1 and 2. We can therefore represent each block as a signed integer. Let +1 represent (a b) and -1 represent (b a). Then the permutation representing (a b)x(c d)y(e f) would be represented by (+1)x(+2)y(+3), and each flip of size 2 simply toggles the sign of the first block. A flip of size 5 exchanges the first and second blocks, toggling the sign of both and leaving the symbol between untouched. A flip of size 8 reverses the order of all three blocks, toggles all three signs, and exchanges the order of the two symbols x and y. Hence, one can get with these flips all signed permutations of three symbols and the two orderings of the symbols x and y. That is, one has a degree 3 spanning subnetwork of P_2 X BP_3, the product of the pancake network of dimension 2 and the burnt pancake network of dimension 3. This network has a diameter of 9.

A network of size…	Generated by the sets…	Is a degree 3 spanning subnetwork of…
48	$\{f_2, f_4, f_6\}$, $\{f_3, f_6, f_9\}$	Burnt Pancake$_3$ (BP$_3$)
72	$\{f_2, f_3, f_6\}$, $\{f_2, f_3, f_7\}$, $\{f_2, f_3, f_8\}$, $\{f_2, f_3, f_9\}$,$\{f_2, f_3, f_{10}\}$, $\{f_3, f_5, f_6\}$	Pancake$_3$ (P$_3$) X P$_3$ X P$_2$
96	$\{f_2, f_5, f_8\}$	BP$_3$ X P$_2$
128	$\{f_2, f_4, f_8\}$, $\{f_2, f_4, f_9\}$, $\{f_2, f_4, f_{10}\}$, $\{f_3, f_4, f_8\}$, $\{f_3, f_4, f_9\}$, $\{f_3, f_4, f_{10}\}$	Q$_7$ (See Section IV)
144	$\{f_3, f_5, f_7\}$	P$_3$ X P$_4$
192	$\{f_2, f_6, f_{10}\}$	Even permutations of BP$_4$
200	$\{f_4, f_5, f_{10}\}$	Cycle$_{10}$ (C$_{10}$) X C$_{10}$ X P$_2$
240	$\{f_4, f_8, f_{10}\}$	1/16 of the nodes of BP$_5$
288	$\{f_3, f_5, f_{10}\}$	P$_3$ X P$_3$ X P$_2$ X P$_2$ X P$_2$
320	$\{f_2, f_8, f_{10}\}$	Burnt Cycle$_5$ (BC$_5$)
384	$\{f_4, f_6, f_8\}$	BP$_4$
576	$\{f_4, f_7, f_{10}\}$	½ of the nodes of (P$_4$ X BP$_3$)
960	$\{f_3, f_5, f_9\}$, $\{f_3, f_7, f_9\}$	P$_2$ X P$_2$ X P$_2$ X P$_5$
1,152	$\{f_3, f_5, f_8\}$, $\{f_3, f_7, f_8\}$, $\{f_5, f_7, f_8\}$	Unknown
1,296	$\{f_2, f_6, f_9\}$	P$_3$ X P$_3$ X P$_3$ X P$_3$
2,880	$\{f_5, f_7, f_9\}$	P$_4$ X P$_5$
3,840	$\{f_4, f_6, f_{10}\}$, $\{f_6, f_8, f_{10}\}$	BP$_5$
10,080	$\{f_2, f_5, f_9\}$	½ of the even permutations of P$_8$
14,400	$\{f_5, f_9, f_{10}\}$	P$_5$ X P$_5$
20,160	$\{f_4, f_5, f_8\}$	Even permutations of P$_8$
28,800	$\{f_2, f_7, f_{10}\}$, $\{f_3, f_6, f_{10}\}$, $\{f_3, f_7, f_{10}\}$ $\{f_3, f_9, f_{10}\}$, $\{f_5, f_7, f_{10}\}$, $\{f_7, f_9, f_{10}\}$	P$_5$ X P$_5$ X P$_2$
181,440	$\{f_4, f_5, f_9\}$, $\{f_4, f_8, f_9\}$, $\{f_5, f_8, f_9\}$	Even permutations of P$_9$

Table 2. Cayley networks on subgroups of S_n generated by 3 flips

While each network in Table 1 has its own significance, there is a particular network from Table 1 with some properties we wish to examine further:

IV. Subcube$_n$, a network isomorphic to a subnetwork of a binary hypercube.

The set $\{f_2, f_4, f_8\}$, where $n = 8$ generates a subgroup of 128 elements. As we have already seen from condition 5) of Section II, this set of generators can never separate individual elements in blocks of 2. That is, $\{f_2, f_4, f_8\}$ preserves adjacencies between elements originally in positions $2i - 1$ and $2i$ for all i. The flips that must be made to route from the identity to a permutation in this subgroup can be represented by a bit string of 7 values b_1, b_2 ..., b_7, which is constructed from left to right by the following:

- $b_1 = 1 \Leftrightarrow$ Elements 1 through 8 need to be flipped by f_8
- $b_2 = 1 \Leftrightarrow$ Elements 1 through 4, wherever they are, need to be flipped
- $b_3 = 1 \Leftrightarrow$ Elements 5 through 8, wherever they are, need to be flipped
- $b_4 = 1 \Leftrightarrow$ Elements 1 and 2, wherever they are, need to be flipped
- $b_5 = 1 \Leftrightarrow$ Elements 3 and 4, wherever they are, need to be flipped
- $b_6 = 1 \Leftrightarrow$ Elements 5 and 6, wherever they are, need to be flipped
- $b_7 = 1 \Leftrightarrow$ Elements 7 and 8, wherever they are need to be flipped

Consider the permutation 6 5 8 7 3 4 2 1. To obtain this permutation from the identity, elements 1 through 8 of the identity must be flipped, resulting in 8 7 6 5 4 3 2 1, so $b_1 = 1$. Elements 1 through 4 don't have to be flipped, so $b_2 = 0$. Elements 5 through 8 must be flipped, giving 5 6 7 8 4 3 2 1, so $b_3 = 1$. Elements 3 and 4, elements 5 and 6, and elements 7 and 8 must be flipped, yielding 6 5 8 7 3 4 2 1, and a bit string of 1 0 1 0 1 1 1. This mapping from permutations in S_8 to binary bit strings of length 7 is, in fact, an isomorphism between the network generated by $\{f_2, f_4, f_8\}$, (which we call the *Subcube network* of dimension 8, or Subcube_8), and a particular spanning subnetwork of Q_7, the binary hypercube of dimension 7. As they are isomorphic we shall interchangeably refer to Subcube_8 as a network with permutation labeling processors and a network with binary bit strings labeling processors.

Subcube_8, is what Q_7, would look like if a particular set of 4 of the edges incident to each node were removed. More generally, let Subcube_n, where n is a power of 2, be defined as the network generated by the set $\{f_2, f_4, f_8, ..., f_n\}$. The degree of Subcube_n is $\log_2 n$. The number of nodes in Subcube_n is 2^{n-1}. Subcube_n is what Q_{n-1} would look like if a particular set of $n - 1 - (\log_2 n)$ of the edges incident to each node were removed. Figure 1 shows the relationship between Subcube_4 and Q_3. As we shall see, the diameter of Subcube_n is described by the recurrence relation $D(n) = 2D(n / 2) + 2$, $D(2) = 1$, whose solution is $(3n / 2) - 2$. Table 3 shows how Subcube_n

compares with Q_{n-1}. Since $Subcube_n$ has 2^{n-1} nodes, its diameter is $\Theta(\log_2 N)$, where N is the number of nodes.

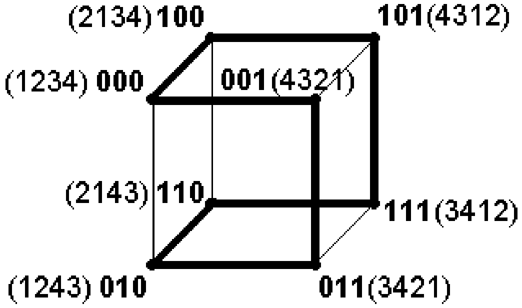

Figure 1. $Subcube_4$ is isomorphic to Q_3 with 1 edge removed from each node

Measure	$Subcube_n$	Q_{n-1}
Degree as a function of the dimension n	**$\log_2 n$**	**n - 1**
Diameter as a function of the dimension n	**(3n / 2) - 2**	**n - 1**
Degree as a function of the number of nodes N	**$\log_2((\log_2 N) + 1)$**	**$\log_2 N$**
Diameter as a function of the number of nodes N	**$\frac{3}{2}((\log_2 N) + 1) - 2$**	**$\log_2 N$**

Table 3. Comparing $Subcube_n$ with Q_{n-1}

Suppose we are trying to route from a permutation π in $Subcube_n$ to a permutation π'. Algorithm ROUTE returns the sequence of flips required to route from π to π', which through the isomorphism can also be viewed as a sequence of bit toggles to route between nodes in Q_{n-1}. The length of that sequence is described by the same recurrence relation used to compute the diameter, and is thus an optimal routing algorithm.

If n = 2, then at most one flip is necessary. If $n = 2^k$, then the number of flips needed D(n) is no more than 2D(n / 2) + 2. The solution of this recurrence relation is D(n) = (3n / 2) - 2.

Suppose we are routing from π to π' in $Subcube_n$, and that the first n / 2 elements of π are the same as the first n / 2 elements of π', only not in the same order. If D(n / 2) is the diameter of $Subcube_{n/2}$, then it may take D(n / 2) flips to put the first n / 2 elements of π in the same order as the first n / 2 elements of π'. To change the order of the last n / 2 elements, we must do a flip of

```
Algorithm ROUTE($\pi$, $\pi'$, n)

    begin
        if n = 2 then
            begin
                if $\pi_1 \neq \pi'_1$ then
                    $f_2$
            end
        else
            if the first n / 2 elements of $\pi$ are same as the first n / 2 elements of $\pi'$ then
                begin
                    ROUTE($\{\pi_1 \ldots \pi_{size / 2}\}$, $\{\pi'_1 \ldots \pi'_{size / 2}\}$, n / 2)
                    Apply $f_n$ to $\pi$
                    ROUTE($\{\pi_1 \ldots \pi_{size / 2}\}$, $\{\pi'_{size}\ \pi'_{size - 1} \ldots \pi'_{(size / 2) + 1}\}$, n / 2)
                    Apply $f_n$ to $\pi$
                end
            else
                begin
                    ROUTE($\{\pi_1 \ldots \pi_{size / 2}\}$, $\{\pi'_{size}\ \pi'_{size - 1} \ldots \pi'_{(size / 2) + 1}\}$, n / 2)
                    Apply $f_n$ to $\pi$
                    ROUTE($\{\pi_1 \ldots \pi_{size / 2}\}$, $\{\pi'_1 \ldots \pi'_{size / 2}\}$, n / 2)
                end
    end
```

Figure 2. An optimal routing algorithm for Subcube$_n$

size n. It may take D(n / 2) flips to put the last n / 2 elements of π in the same order, only reversed, as the last n / 2 elements of π'. We then do a flip of size n again to put the last n / 2 elements of π in the last n / 2 positions. Thus the diameter D(n) is described by the recurrence relation D(n) = 2D(n / 2) + 2, D(2) = 1, and Algorithm ROUTE is an optimal routing algorithm.

Example: Suppose we wished to route from 3 4 1 2 7 8 6 5 to 8 7 6 5 1 2 4 3. The algorithm would proceed as shown in Table 4. ROUTE returns the sequence $\{f_4, f_2, f_4, f_8, f_4, f_2\}$ to route from {3 4 1 2 7 8 6 5} to {8 7 6 5 1 2 4 3}, *i. e.*, from {0 1 1 1 1 0 1} to {1 1 0 0 1 0 0} in the hypercube.

Theorem 1: Let M_1 be the isomorphism between the indicated spanning subnetwork of Q_{n-1} and Subcube$_n$. Then M_1 embeds *all* of Q_{n-1} into Subcube$_n$ with dilation $2(\log_2 n) - 1$.

Proof: We need to show that toggling any bit of Q_{n-1} can be done with at most $2(\log_2 n) - 1$ flips in the network Subcube$_n$. Changing a bit value in Q_{n-1} is done by changing the parity of a particular

Step	Permutation π	Bit String
ROUTE({3 4 1 2 7 8 6 5}, {8 7 6 5 1 2 4 3}, 8)	3 4 1 2 7 8 6 5	0 1 1 1 1 0 1
ROUTE({3 4 1 2}, {3 4 2 1}, 4)		
ROUTE({3 4}, {3 4}, 2)		
f_4	2 1 4 3 7 8 6 5	0 0 1 1 1 0 1
ROUTE({1 2}, {2 1}, 2)		
f_2	1 2 4 3 7 8 6 5	0 0 1 0 1 0 1
f_4	3 4 2 1 7 8 6 5	0 1 1 0 1 0 1
f_8	5 6 8 7 1 2 4 3	1 1 1 0 1 0 1
ROUTE({5 6 8 7}, {8 7 6 5}, 4)		
ROUTE({5 6}, {5 6}, 2)		
f_4	7 8 6 5 1 2 4 3	1 1 0 0 1 0 1
ROUTE({7 8}, {8 7}, 2)		
f_2	8 7 6 5 1 2 4 3	1 1 0 0 1 0 0

Table 4. An example of the routing algorithm for Subcube$_n$

pair of elements in Subcube$_n$, while leaving the rest of the permutation exactly the same. In order to change the parity of a pair, it must be brought into the first two positions. If a pair is in two of the positions between 1 and 2^k, then it takes at most one flip to move it into two of the positions between 1 and 2^{k-1}. Since $k \le \log_2 n$, and the pair is being brought to positions 1 and 2, the pair can be brought into the first two positions in at most $(\log_2 n) - 1$ flips. Once the pair is in the first two positions, one flip changes the parity. We then use the sequence of flips that brought the pair to the first two positions, in reverse order. This produces the original permutation, except that the parity of one pair has been altered. Altogether the sequence is of length $(\log_2 n) - 1 + 1 + (\log_2 n) - 1 = 2(\log_2 n) - 1$.

Since the dilation of any embedding of a network of degree n - 1 into a network of degree $\log_2 n$ is $\Omega(\log_2 n / \log_2 \log_2 n)$, M_1 embeds Q_{n-1} into Subcube$_n$ with a dilation that is close to optimal.

A generating set G is called **minimal** if and only if removing a generator causes the set to generate a smaller subgroup [16].

Lemma 1: Subcube$_n$ has a minimal generating set.

Proof: If f_2 is removed from the generating set of Subcube_n, then one can never separate individual elements of blocks of four contiguous elements, so the result is a smaller subgroup. If f_k is removed, where $2 < k < n$, then individual elements in the range k / 2 through 3k / 2 cannot be separated from each other, resulting in a smaller subgroup. If f_n is removed, then the elements in the range n / 2 through n cannot be moved, resulting in a smaller subgroup.

Theorem 2: Subcube_n is ($\log_2 n$) edge-connected and and maximally fault tolerant.

Proof: This follows from Lemma 1 and the fact that all vertex-symmetric networks are maximally edge connected [26] [20] and all Cayley networks with minimal generating sets are maximally fault tolerant [13].

Even though Subcube_n is only defined when n is a power of 2, it can still represent Q_k for $k \leq n - 1$. We do this by using the mapping M_1 as before, and simply discarding the rightmost unwanted bits. In general, Q_n can be represented by Subcube_m, where $m = 2^{\lfloor \log n \rfloor + 1}$.

V. Properties of generating sets

If the inverse of every generator is again in the generating set, then the Cayley network will be undirected. Cayley networks generated by sets of flips are always undirected, since every flip is its own inverse.

Theorem 3: Let f(n) be the diameter of any undirected Cayley network on S_n generated by a set of O(1) generators. Then, $f(n) = \Omega(n \log_2 n)$.

Proof: Start with the identity permutation and apply all k generators. k + 1 nodes have now been reached. Apply all k generators again to all the nodes that have been reached. No more than $k + 1 + (k * (k - 1)) = k^2 + 1$ nodes could be reached, because one of the generators is the inverse of a previously applied generator. Repeat this process d times. No more than k + 1 + (k *(k - 1)$^{d - 1}$) nodes could be reached. When k + 1 + (k * (k - 1)$^{d - 1}$) = n!, then all nodes are reached. k + 1 + (k * (k - 1)$^{d - 1}$) = Θ(k^{d}). If n! = Θ(k^{d}), then log$_2$ n! = log$_2$(Θ(k^{d})) = log$_2$ k^{d} + Θ(1), which is d log$_2$ k + Θ(1). Since log$_2$ n! = Ω(n log$_2$ n), d = Ω(n log$_2$ n) .

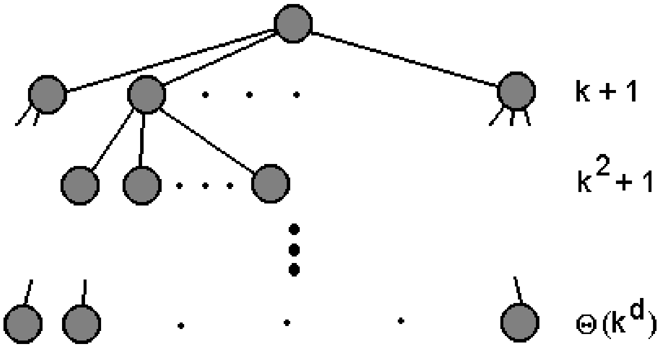

Figure 3. The maximum number of nodes that can be reached by d sorting steps

Consider the elements of a permutation π arranged in order around a circle so that π_n is adjacent to π_1. Let d(i, j) be the minimum circular distance between elements i and j. Let π and σ be two permutations on n symbols. Then let the **displacement** D(π, σ) = $\sum_{i=1}^{n-1}$ d(σ_i, σ_{i+1}) - $\sum_{i=1}^{n-1}$ d(π_i, π_{i+1}). If σ is the permutation that results from applying the flip of size i to π, then D(i) = D(π, σ). Notice that D(n) = 0, since there is no change in the relative circular positions of the elements. In general, D(i) = O(i^2) for i ≤ n / 2, and D(i) = O((n - i)^2) for i > n / 2. Consider the permutation π = 1 3 5 ... n - 1 2 4 6 ... n. Then D(π, I_n) = (n / 2)(n / 2) + ((n / 2) - 1)((n / 2) - 1) - (n - 1) = n^2 / 2 - 2n + 2. In order to sort in k steps, a set of flips must contain a flip with a displacement of at least (n^2 / 2 - 2n + 2) / k. Substituting k = O(n log$_2$ n), it follows that a set of flips must contain a flip with a displacement of Ω(n / log$_2$ n).

Theorem 4: A generating set of flips must contain a flip of size i where O($\sqrt{n/\log n}$) ≤ i ≤ n - O($\sqrt{n/\log n}$), if it allows sorting in Θ(n log$_2$ n) flips.

While the following sets of flips are able to sort an arbitrary permutation, they take $\Omega(n^2)$ steps, because the maximum displacement of any flip in any of these sets is O(1): $\{f_2, f_{n-1}, f_n\}$, $\{f_{n-2}, f_{n-1}, f_n\}$, $\{f_{n-3}, f_{n-2}, f_n\}$, $\{f_3, f_{n-1}, f_n\}$ where n is odd, $\{f_2, f_{n-2}, f_n\}$ where n is odd, $\{f_{n-3}, f_{n-1}, f_n\}$, where n is odd, and $\{f_3, f_{n-2}, f_n\}$, where n is even.

VI. The Triad network Triad_n.

Triad_n, where n is odd, is the network on S_n where the generating set is $\{f_{\lfloor n/2 \rfloor}, f_{\lceil n/2 \rceil}, f_n\}$. Triad_n has n! nodes and 3n! / 2 edges. By [26], [20], Triad_n has an edge connectivity of 3, and is maximally edge connected. Triad_n is maximally fault tolerant as it has a minimal generating set. [13]. This network is an attempt to keep the benefits of fixed-degree Cayley networks on S_n, (vertex-symmetry, n! nodes, fixed-degree, routing equivalent to sorting) while attaining a network of optimal asymptotic diameter. Triad_n has much better diameter, *i. e.*, $\Theta(n \log_2 n)$, than the recently studied shuffle-exchange permutation network, SEP_n, which has $\Theta(n^2)$ diameter [7].

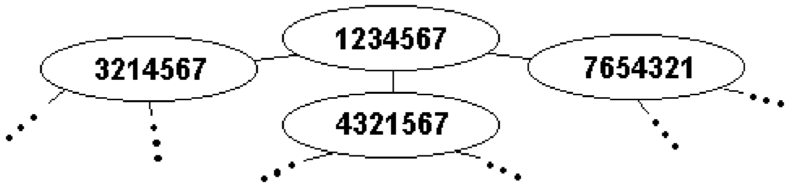

Figure 4. A portion of Triad_7

Let LEFT(i, j, k) be defined as a left shuffle (cyclic shift) of elements π_i through π_j by k positions. Let RIGHT(i, j, k) be defined similarly, but for a right shuffle, and let EXCHANGE(i, i + 1) denote LEFT(i, i + 1, 1). If $(G_1, G_2, \ldots, G_n)$ is a sequence of generators, then let $(G_1, G_2, \ldots, G_n)^i$ represent i successive applications of that sequence.

Theorem 5: Triad_n is a Cayley network on S_n for $n \geq 5$ where n is odd and $\lfloor n / 2 \rfloor \bmod 4 \neq 0$.

Proof: Since SEP_n is known to be a Cayley network on all of S_n, we show Triad_n to be a Cayley network on all of S_n by showing that a sequence of Triad_n generators produce the generators of

SEP$_n$, namely {LEFT(1, n, 1), RIGHT(1, n, 1), EXCHANGE(1, 2)}. When n is odd, LEFT(1, n, 1) can be accomplished by the sequence $\{f_{\lceil n / 2 \rceil}, f_n, f_{\lfloor n / 2 \rfloor}, f_{\lceil n / 2 \rceil}, f_n, f_{\lfloor n / 2 \rfloor}\}$. A right shuffle can be simulated by applying the left shuffle sequence in reverse order. LEFT(1, n, k) = LEFT(1, n, 1)^k. When $\lfloor n / 2 \rfloor \bmod 4 \neq 0$, EXCHANGE($\lfloor n / 2 \rfloor$, $\lceil n / 2 \rceil$) is effected by the sequence ($f_{\lceil n / 2 \rceil}$, LEFT(1, n, 2), ($f_{\lfloor n / 2 \rfloor}$, LEFT (1, n, 1))$^{\lfloor n / 2 \rfloor}$). Elements π_1 and π_2 can be transposed by the sequence {RIGHT(1, n, $\lfloor n / 2 \rfloor$ - 1), EXCHANGE($\lfloor n / 2 \rfloor$, $\lceil n / 2 \rceil$), LEFT(1, n, $\lfloor n / 2 \rfloor$ - 1)}. As the three generators of SEP$_n$ can be simulated by sequences of Triad$_n$ generators, Triad$_n$ is a Cayley network on S_n where n is odd and $\lfloor n / 2 \rfloor \bmod 4 \neq 0$.

If n is even, then the generators of SEP$_n$ cannot be simulated by the above method. If $\lfloor n / 2 \rfloor \bmod 4 = 0$, then $f_{\lfloor n / 2 \rfloor}$, $f_{\lceil n / 2 \rceil}$, f_n all correspond to even numbers of transpositions. When this is the case, EXCHANGE($\lfloor n / 2 \rfloor$, $\lceil n / 2 \rceil$) cannot be done, and the largest set that can be generated is the alternating group.

In order to prove the $O(n \log_2 n)$ bound for the diameter of Triad$_n$, we describe operations that can be simulated by an O(1) length sequence of Triad$_n$ generators. We can use these operations as a macro to prove our bound. If LEFT(i, j, k) can be simulated by a given sequence, then RIGHT(i, j, k) can be simulated by applying that sequence in reverse order.

1) *RIGHT(1, $\lceil n / 2 \rceil$, 1)* : $f_{\lfloor n / 2 \rfloor}$, $f_{\lceil n / 2 \rceil}$.
2) *LEFT($\lceil n / 2 \rceil$, n, 1)* : f_n, RIGHT(1, $\lceil n / 2 \rceil$, n), f_n.
3) *LEFT(1, n, 1)* : $f_{\lceil n / 2 \rceil}$, f_n, $f_{\lfloor n / 2 \rfloor}$, $f_{\lceil n / 2 \rceil}$, f_n, $f_{\lfloor n / 2 \rfloor}$.
4) *LEFT($\lfloor n / 2 \rfloor$, $\lceil n / 2 \rceil$ + 1, 1)* : LEFT($\lceil n / 2 \rceil$, n, 1), RIGHT(1, $\lceil n / 2 \rceil$, 1), RIGHT($\lceil n / 2 \rceil$, n, 1), LEFT(1, $\lceil n / 2 \rceil$, 1).
5) *LEFT($\lceil n / 2 \rceil$ + 1, n, 2)* : RIGHT($\lceil n / 2 \rceil$, n, 1), RIGHT($\lfloor n / 2 \rfloor$, $\lceil n / 2 \rceil$ + 1, 1), RIGHT($\lceil n / 2 \rceil$, n, 1), RIGHT($\lfloor n / 2 \rfloor$, $\lceil n / 2 \rceil$ + 1, 1).
6) *LEFT(1, $\lfloor n / 2 \rfloor$, 2)* : f_n, LEFT($\lceil n / 2 \rceil$ + 1, n, 2), f_n.

7) $\pi_1\ \pi_2\ \dots\ \pi_n \longrightarrow \pi_1\ \pi_2\ \dots\ \pi_{\lfloor n / 2 \rfloor}\ \pi_{(\lfloor n / 2 \rfloor - 1)}\ \pi_{(\lceil n / 2 \rceil + 1)}\ \pi_{\lceil n / 2 \rceil} \dots\ \pi_n$: RIGHT(1, $\lceil n / 2 \rceil$, 1), LEFT($\lceil n / 2 \rceil$, n, 1), RIGHT(1, $\lceil n / 2 \rceil$, 1), LEFT($\lceil n / 2 \rceil$, n, 1), LEFT(1, n, 2), RIGHT($\lceil n / 2 \rceil$, n, 2).

We will partition elements p through r by shuffling π_q into position $\lceil n / 2 \rceil$. Since we have RIGHT(1, $\lceil n / 2 \rceil$, 1), and LEFT($\lceil n / 2 \rceil$, n, 1), we can use the right and left ends of the permutation as the tops of two "stacks" for elements p through q and elements q + 1 through r, respectively. The element in position $\lceil n / 2 \rceil$ will be pushed onto one of these two stacks.

When k < n elements are being partitioned, sometimes the last element between p and q is moved to its stack while there are still elements between q + 1 and r in positions $\lceil n / 2 \rceil + 1$, $\lceil n / 2 \rceil + 2$, …, $\lceil n / 2 \rceil + k$. In the same way, sometimes the last element between q + 1 and r is moved to its stack while there are still elements between p and q in positions $\lfloor n / 2 \rfloor$, $\lfloor n / 2 \rfloor - 1$, … $\lfloor n / 2 \rfloor - k$. To resolve this, we apply RIGHT(1, $\lfloor n / 2 \rfloor$, 2), or LEFT($\lceil n / 2 \rceil + 1$, n, 2), as needed.

```
SORT(π, p, r, n)

begin
    if (r - p) ≥ 1 then
    begin
        q ⇓  (p + r) / 2
        Partition π so that elements p through q are in positions p ± k through
            q ± k in arbitrary order, and elements q + 1 through r are in
            positions q + 1 ± k through r ± k, for some k ≤ n / 2
        SORT(π, p, q, n)
        SORT(π, q + 1, r, n)
    end
    if (r - p + 1 = n) then
        Shuffle 1 into position 1 in as few flips as possible
end
```

Figure 5. A routing algorithm for Triad$_n$

At this point, the elements p through q are in positions 1 through q - p + 1, and the elements q + 1 through r are in positions n - r + q + 1 through n. We must put the partitioned halves back into their original positions without disturbing the relative positions of the other elements. We shuffle the elements p through q into positions $\lceil n / 2 \rceil - p + q$ through $\lceil n / 2 \rceil$. We move the elements q + 1 through r into positions $\lceil n / 2 \rceil + 1$ through $\lceil n / 2 \rceil + r$ by the sequence (RIGHT($\lceil n / 2 \rceil$, n, 1), RIGHT($\lfloor n / 2 \rfloor$, $\lceil n / 2 \rceil + 1$, 1))$^{r - q}$. It doesn't matter that the relative position of some of the elements between p and q are being changed by this sequence, because the partition around q is not being disrupted.

The algorithm will eventually get to the point of partitioning two elements. We move the two elements into positions $\lfloor n / 2 \rfloor - 1$ and $\lfloor n / 2 \rfloor$. We then apply the sequence that changes $\pi_1\ \pi_2\ \ldots\ \pi_n$ into $\pi_1\ \pi_2\ \ldots\ \pi_{\lfloor n / 2 \rfloor}\ \pi_{(\lfloor n / 2 \rfloor - 1)}\ \pi_{(\lceil n / 2 \rceil + 1)}\ \pi_{\lceil n / 2 \rceil}\ \ldots\ \pi_n$. It doesn't matter that the elements in positions $\lceil n / 2 \rceil$ and $\lceil n / 2 \rceil + 1$ are being transposed, because the algorithm will resolve them later. This strategy will not work for putting n - 1 and n in order, because it would disrupt the position of 1 and 2. In the proof of Theorem 3, we mentioned that the sequence $\{f_{\lceil n / 2 \rceil}$, LEFT(1, n, 2), $(f_{\lfloor n / 2 \rfloor}$, LEFT(1, n, 1)$)^{\lfloor n / 2 \rfloor}\}$ will transpose the elements in positions $\lfloor n / 2 \rfloor$ and $\lceil n / 2 \rceil$ in O(n) flips. We will use this sequence if and only if n and n - 1 need to be transposed. This completes the description of the partitioning algorithm, which is given in full detail below. Selected statements are numbered to describe the number of moves required by the algorithm.

Theorem 6: The diameter of Triad_n is $\Theta(n \log_2 n)$.

Proof: When PARTITION(π, p, q, r) is called, and less than n elements are being partitioned, then either 1), the right end of the elements to be partitioned are in position $\lceil n / 2 \rceil$, or 2) the left end of the elements to be partitioned are in position $\lceil n / 2 \rceil + 1$. Statement 5) therefore requires O(n) flips, where n is the number of elements being partitioned. Statements 6) through 12) require O(1) flips and are executed O(n) times, therefore requiring O(n) flips. Statements 1) and 2) require O(1) flips. Statements 3) and 4) require O(n) flips, but are only executed once throughout all the calls to PARTITION. Therefore, if there are n elements to be partitioned, then PARTITION requires O(n) flips. The number of flips required by SORT is described by the recurrence $T(n) = 2T(n / 2) + O(n)$, , because PARTITION requires O(n) flips, and it takes O(n) flips to put n - 1 and n in the proper order, and shuffle 1 into position 1 at the end. The solution of the recurrence is $O(n \log_2 n)$. Since the diameter of Triad_n is $\Omega(n \log_2 n)$ by Theorem 4, the diameter of Triad_n is $\Theta(n \log_2 n)$.

```
PARTITION(π, p, q, r)

begin
    if (r = p) then
        do nothing
    else if (r - p) = 1 then
    begin
        if r is before p then
            if p ≠ n - 1 or n - 2 then
            begin
1)              Move r and p to positions $\lfloor n/2 \rfloor - 1$ and $\lfloor n/2 \rfloor$
2)              Change $\pi_1 \pi_2 \ldots \pi_n$ into $\pi_1 \pi_2 \ldots \pi_{\lfloor n/2 \rfloor} \pi_{(\lfloor n/2 \rfloor - 1)} \pi_{(\lceil n/2 \rceil + 1)} \pi_{\lceil n/2 \rceil} \ldots \pi_n$
            end
            else
            begin
3)              Move p and r to positions $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$
4)              Apply $\{f_{\lceil n/2 \rceil}, \mathrm{LEFT}(1, n, 2), (f_{\lfloor n/2 \rfloor}, \mathrm{LEFT}(1, n, 1))^{\lfloor n/2 \rfloor}\}$
            end
    end
    else
    begin
5)      Move the element corresponding to q to position $\lceil n/2 \rceil$
        while there are elements between p and q in positions q + 1 to r do
        begin
            if the element in position $\lceil n/2 \rceil$ is between p and q then
6)              RIGHT(1, $\lceil n/2 \rceil$, 1)        // 2 flips
            else
7)              LEFT($\lceil n/2 \rceil$, n, 1)         // 4 flips
        end
        while there are elements between p and q not on the right end do
8)          RIGHT(1, $\lfloor n/2 \rfloor$, 2)              // 34 flips
        while there are elements between q + 1 and r not on the left end do
9)          LEFT($\lceil n/2 \rceil$ + 1, n, 2)           // 32 flips
        while there are elements between p and q on the right end do
10)         LEFT(1, $\lceil n/2 \rceil$, 1)               // 2 flips
        while there are elements between q + 1 and r on the left end do
        begin
11)         RIGHT($\lceil n/2 \rceil$, n, 1)              // 4 flips
12)         RIGHT($\lfloor n/2 \rfloor$, $\lceil n/2 \rceil$ + 1, 1)// 12 flips
        end
    end
end
```

Figure 6. A partitioning algorithm for Triad$_n$

Corollary 1: Triad_n is a Cayley network of degree 3 on S_n with asymptotically optimal diameter (up to constant factors).

Example: Consider the permutation 7162543. We will first partition the entire permutation: 7162543 $\xrightarrow{(2)}$ 2716543 $\xrightarrow{(4)}$ 2715436 $\xrightarrow{(4)}$ 2714365 $\xrightarrow{(2)}$ 4271365 $\xrightarrow{(2)}$ 1427365 $\xrightarrow{(4)}$ 1423657. Now we will partition 1 through 4. 1423657 $\xrightarrow{(12)}$ 5714236 $\xrightarrow{(4)}$ 5712364 $\xrightarrow{(2)}$ 2571364 $\xrightarrow{(2)}$ 1257364 $\xrightarrow{(32)}$ 1257436 $\xrightarrow{(4)}$ 5712436. 1 and 2 are in order, therefore put 3 and 4 in order. 5712436 $\xrightarrow{(18)}$ 2436571 $\xrightarrow{(30)}$ 2345671 $\xrightarrow{(6)}$ 1234567. While this permutation was selected at random, putting 3 and 4 in order had the happy consequence of putting 5 and 6 in order as well.

In the algorithm PARTITION, statement 5 takes 6(q - p) flips, or roughly 3n flips, where n is the number of elements being partitioned. Statements 6 and 7 take at most 4n flips. Statements 8 and 9 will be executed no more than n / 2 times, adding at most 17n flips. Statement 10 will be executed n / 2 times, adding n flips. Statements 11 and 12 will be executed n / 2 times, adding 8n flips, for a total of 3n + 4n + 17n + n + 8n = 33n. The constant is less when only 2 elements are being partitioned. Therefore the recurrence equation is T(n) = 2T(n / 2) + 33n, T(1) = 0, whose solution is $33(n \log_2 n)$.

VII. Embeddings Into Triad_n.

Since the efficiency of many parallel algorithms in dependent on the topology of the interconnection network, it is useful to determine a rough upper bound on the number of Triad_n flips required to simulate moves in other networks.

The star network is bipartite, with the bipartition based on permutations with even and odd numbers of transpositions. Let the mapping M_2 be as follows:

- If π is an even permutation, then $M_2(\pi) = \pi$
- If π is odd, then $M_2(\pi) = (\pi_2\ \pi_3 \ldots \pi_{(\lfloor n/2 \rfloor - 1)}\ \pi_1\ \pi_{\lceil n/2 \rceil} \ldots \pi_n)$

Theorem 7: M_2 embeds S_n into Triad_n with dilation $n + O(1)$

Proof: Suppose $\pi = (\pi_1\ \pi_2 \ldots \pi_n)$ is even. Then π', the result of any move of S_n, is odd. Suppose we are making the move of S_n that exchanges π_1 and π_k.

Case I. $k = 2$. $\pi' = (\pi_1\ \pi_3 \ldots \pi_{(\lfloor n/2 \rfloor - 1)}\ \pi_2\ \pi_{\lceil n/2 \rceil} \ldots \pi_n)$. Apply RIGHT(1, $\lceil n/2 \rceil$, 1), moving π_2 to position $\lceil n/2 \rceil$ in 2 flips. Apply RIGHT(1, n, 1), moving π_2 to position $\lceil n/2 \rceil + 1$ in 6 flips. Apply LEFT(1, $\lceil n/2 \rceil$, 2), moving π_1 to position $\lceil n/2 \rceil$ in 4 flips. Apply LEFT($\lceil n/2 \rceil$, n, 2), moving π_1 and π_2 to positions $n - 1$ and n, respectively, in 6 flips. Apply RIGHT(1, n, 2), moving π_1 and π_2 to positions 1 and 2, respectively, in 12 flips. Apply LEFT($\lceil n/2 \rceil$, n, 1), moving π_n to position n in 4 flips, for a total of O(1) flips.

Case II. $3 \le k \le \lfloor n/2 \rfloor$. $\pi' = (\pi_2\ \pi_3 \ldots \pi_{k-1}\ \pi_1\ \pi_{k+1} \ldots \pi_{(\lfloor n/2 \rfloor - 1)}\ \pi_k\ \pi_{\lceil n/2 \rceil} \ldots \pi_n)$. Apply RIGHT(1, $\lceil n/2 \rceil$, 1), moving π_k to position $\lceil n/2 \rceil$ in 2 flips. Apply RIGHT(1, n, 1), moving π_k to position $\lceil n/2 \rceil$ in 6 flips. Shift elements 1 through $\lceil n/2 \rceil$ so that π_{k-1} is in position $\lceil n/2 \rceil$ in at most $\lceil n/2 \rceil$ flips. Apply LEFT(1, n, 1), moving π_1 to position n in 6 flips. Shift elements 1 through $\lceil n/2 \rceil$ so that π_2 is in position 1 in at most $\lceil n/2 \rceil$ flips. Apply RIGHT(1, n, 1), moving π_1 to position 1 in 6 flips. Apply LEFT($\lceil n/2 \rceil$, n, 1), moving π_n to position n in 4 flips, for a maximum total of $n + 25 = n + O(1)$ flips.

Case III. $k = \lceil n/2 \rceil$. $\pi' = (\pi_2\ \pi_3 \ldots \pi_{\lceil n/2 \rceil}\ \pi_1\ \pi_{\lceil n/2 \rceil + 1} \ldots \pi_n)$ Apply RIGHT(1, $\lceil n/2 \rceil$, 1), moving π_1 to position 1, for a total of O(1) flips.

Case IV. $k > \lceil n/2 \rceil$. $\pi' = (\pi_2\ \pi_3 \ldots \pi_{(\lfloor n/2 \rfloor - 1)}\ \pi_k\ \pi_{\lceil n/2 \rceil} \ldots \pi_{k-1}\ \pi_1\ \pi_{k+1} \ldots \pi_n)$. Shift elements $\lceil n/2 \rceil$ through n until π_{k+1} is in position $\lceil n/2 \rceil$ in at most $\lceil n/2 \rceil + 2$ flips. Apply RIGHT(1, n, 1), moving π_1 to position 1 in 6 flips. Shift elements $\lceil n/2 \rceil$ through n until $\pi_{\lceil n/2 \rceil}$ is in position $\lceil n/2 \rceil$ in at most $\lceil n/2 \rceil + 2$ flips, for a maximum total of $n + 11 = n + O(1)$ flips. The proof for when π is odd is similar.

Theorem 8: M_2 embeds SEP_n into $Triad_n$ with dilation O(1)

Proof: Suppose $\pi = (\pi_1\ \pi_2\ \ldots\ \pi_n)$ is even. Then $\pi_L = (\pi_2\ \pi_3\ \ldots\ \pi_n\ \pi_1)$ is also even, and $M_2(\pi_L)$ can be reached from $M_2(\pi)$ in 6 flips by applying LEFT(1, n, 1). $\pi_R = (\pi_n\ \pi_1\ \pi_2\ \ldots\ \pi_{n-1})$ is also even, and $M_2(\pi_R)$ can be reached from $M_2(\pi)$ in 6 flips by applying RIGHT(1, n, 1). $\pi_E = (\pi_2\ \pi_1\ \pi_3\ \ldots\ \pi_n)$ is odd, and $M_1(\pi_E) = (\pi_1\ \pi_3\ \ldots\ \pi_{(\lfloor n/2 \rfloor - 1)}\ \pi_2\ \pi_{\lceil n/2 \rceil}\ \ldots\ \pi_n)$. $M_2(\pi_E)$ can be reached from $M_2(\pi)$ in 34 = O(1) flips by the method described in Case I of the proof of Theorem 8. When π is odd, $M_2(\pi_L)$ can be reached from $M_2(\pi)$ by the sequence (RIGHT($\lceil n/2 \rceil$, n, 1), RIGHT(1, n, 1), RIGHT($\lfloor n/2 \rfloor$, $\lceil n/2 \rceil + 1$, 1), LEFT(1, n, 1), RIGHT(1, $\lceil n/2 \rceil$, 1)), $M_2(\pi_R)$ can be reached from $M_2(\pi)$ by the sequence (LEFT(1, $\lceil n/2 \rceil$, 1), RIGHT(1, n, 1), RIGHT($\lfloor n/2 \rfloor$, $\lceil n/2 \rceil + 1$, 1), LEFT(1, n, 1), LEFT($\lceil n/2 \rceil$, n, 1)). $M_2(\pi_E)$ can be reached from $M_2(\pi)$ by the same way as when π is even.

We will construct an embedding of the shuffle-exchange network of dimension n, SE_n into $Triad_n$ by constructing an embedding of SE_n into SEP_n. Let G = (V, E) and H = (V', E') be two networks, and let f be a function mapping nodes in V to subsets of V' other than the empty set. f is called a **dilation d one-to-many embedding** if for every pair of adjacent nodes u and v in V, 1) f(u) and f(v) share no nodes in common, and 2) there is a node x in f(u) and a node y in f(v) such that the distance in H between x and y is ≤ d [11], [21].

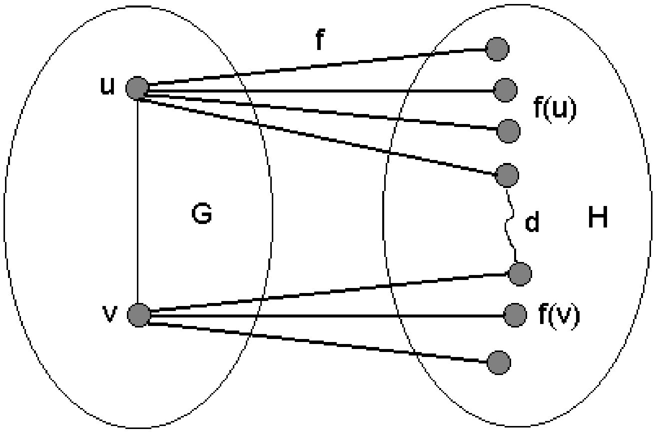

Figure 7. A dilation d one-to-many embedding of G into H

Let M_3 be a one-to-one mapping between a bit string B of SE_n and the permutations $M_3(B)$ of SEP_n as follows:

- $B_1 = 0 \Leftrightarrow$ 1 is to the left of 2 in $M_3(B)$
- $B_2 = 0 \Leftrightarrow$ the rightmost element of {1, 2} is to the left of the leftmost element of {3, 4} in $M_3(B)$
- $B_3 = 0 \Leftrightarrow$ 3 is to the left of 4 in $M_3(B)$
- $B_4 = 0 \Leftrightarrow$ the rightmost element of {3, 4} is to the left of the leftmost element of {5, 6} in $M_3(B)$

*

*

*

- $B_{n-1} = 0 \Leftrightarrow n - 1$ is to the left of n in $M_3(B)$.
- $B_n = 1 \Leftrightarrow$ the element in position n is swapped with the element in position 1 in $M_3(B)$

Consider the bit string B = 0 1 1 1 1 0 1. $B_1 = 0$, so 1 is to the left of 2 in $M_3(B)$ (1 2 * * * * *). B_2 and $B_3 = 1$, so 4 is to the left of 3 in $M_3(B)$, and 2 (the rightmost element of {1, 2}) is to the right of 4 (the leftmost element of {3, 4}) in $M_3(B)$ (1 4 2 * * * *). B_4, B_5 and B_6 yield 1 4 2 6 3 5 7. Since $B_7 = 1$, the elements in positions 1 and 7 are swapped, and $M_3(B)$ = 7 4 2 6 3 5 1.

Let M_4 be a one-to-many mapping between the bit strings of SE_n and the permutations of SEP_n as follows. $M_4(B)$ contains $M_3(B)$ for all bit strings B in SE_n. If B and B' are two bit strings, and B' is obtained by applying k right shuffles to B, then take $M_3(B)$, apply k right shuffles, and add the resulting permutation to $M_4(B')$. $M_4(B)$ contains n elements for all bit strings B. For example, since the bit string 0 1 1 1 1 0 1 is obtained by applying a right shuffle to 1 1 1 1 0 1 0, then M_4(0 1 1 1 1 0 1) includes M_3(1 1 1 1 0 1 0) = (2 4 1 5 3 7 6), shuffled right by 1 position (6 2 4 1 5 3 7).

Theorem 9: M_4 is a dilation O(1) one-to-many embedding of SE_n into SEP_n.

Proof: Let B be a bit string of length n, B_L = B shuffled one position left, B_R = B shuffled one position right, and B_E = B with B_1 toggled. Then $M_3(B)$ is a permutation. Let $(M_3(B))_L = M_3(B)$ shuffled one position left and $(M_3(B))_R = M_3(B)$ shuffled one position right. It takes 1 SEP move to represent a left or right shuffle, since $M_4(B_R)$ contains $(M_3(B))_R$ and $M_4(B_L)$ contains $(M_3(B))_L$. We will toggle B_1 by routing from $M_3(B)$ to $M_3(B_E)$. The number of moves needed to route from $M_3(B)$ to M3(B_E) depend on the position of 1 and 2 in $M_3(B)$, which in turn depend on the values of B_1, B_2 and B_n.

Case I. $B_1 = 0$, $B_2 = 0$, $B_n = 0$. $M_3(B) = 1\ 2\ \ldots\ n$. Setting $B_1 = 1$ ⌠ Exchanging 1 and 2 in $M_3(B)$, which takes 1 SEP move.

Case II. $B_1 = 0$, B_2, $= 0$, $B_n = 1$. $M_3(B) = x\ 2\ \ldots y\ 1$, where x can be n - 1 or n, and if x is n, then y is n - 1, and vice versa. Exchanging 1 and 2 takes 5 SEP moves.

Case III. $B_1 = 0$, $B_2 = 1$, $B_n = 0$. $M_3(B) = 1\ x\ 2\ \ldots\ n$, where x can be 3 or 4. Exchanging 1 and 2 takes 5 SEP moves.

Case IV. $B_1 = 0$, $B_2 = 1$, $B_n = 1$. $M_3(B) = x\ y\ 2\ \ldots\ z\ 1$, where x can be n - 1 or n, y can be 3 or 4, and if x is n, then z is n - 1, and vice versa. Exchanging 1 and 2 takes 11 SEP moves.

Theorem 10: M_4 composed with M_2 is a dilation O(1) one-to-many embedding of SE_n into $Triad_n$.

Acknowledgments

The authors would like to thank Linda Morales of Nortel for her helpful comments on the draft of this paper.

References

[1] S. B. Akers and B. Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38(4):555-566, April 1989.

[2] S. B. Akers and B. Krishnamurthy. The star graph: an attractive alternative to n-cube. In *Proceedings of International Conference on Parallel Processing (ICPP-87)*, pages 393-400, St. Charles, Illinois, August 1987

[3] S. G. Akl and K. Qiu. A Novel Routing Scheme on the Star and Pancake Networks and Its Applications. *Parallel Computing* 19(1):95-101, 1993

[4] S. G. Akl, K. Qiu and I. Stojmenovic. Data Communication and Computational Geometry on the Star and Pancake Interconnection Networks. In *Proceedings of the Third IEEE Symposium on Parallel and Distributed Processing*, 415-422, 1991

[5] S. G. Akl, K. Qiu and I. Stojmenovic. Fundamental Algorithms for the Star and Pancake Interconnection Networks with Applications to Computational Geometry. *Networks*, special issue on Interconnection Networks and Algorithms

[6] F. Annexstein, M. Baumslag and A. L. Rosenberg. Group action graphs and parallel architectures. *SIAM Journal on Computing*, 19(3):544-569, June 1990

[7] D. Bass and I. H. Sudborough. On the Shuffle-Exchange Permutation Network. In *Proceedings of the 3rd International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '97)*, Los Alamitos, California, December 1997

[8] P. Berthome, A. Ferreira, and S. Perennes. Optimal information dissemination in star and pancake networks. In *Proceedings of the 5th Annual IEEE Symposium on Parallel and Distributed Processing,* IEEE Press, 1993.

[9] D. S. Cohen and M. Blum. Improved bounds for sorting pancakes under a conjecture. Manuscript, Computer Science Division, University of California at Berkeley, 1992.

[10] H. Dweighter, Amer. Math. Monthly 82 (1975), 1010.

[11] L. Gardner, Z. Miller, D. Pritikin and I. H. Sudborough. Embedding Hypercubes into Pancake, Cycle Prefix and Substring Reversal Networks. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences,* Los Alamitos, California, January 1995

[12] W. H. Gates and C. H. Papadimitriou. Bounds for sorting by prefix reversal. *Discrete Math.*, vol. 27, pages 47-57, 1979.

[13] C. D. Godsil. Connectivity of minimal Cayley graphs, *Arch. Math.* 37(1981), 473-476.

[14] M. Heydari and I. H. Sudborough. On the diameter of the pancake network. *Journal of Algorithms*, vol. 25, pages 67-94, 1997

[15] M. Heydari and I. H. Sudborough. Pancake sorting is NP-complete, submitted for publication

[16] M.-C. Heydemann, Cayley graphs and interconnection networks. In G. Hahn and G. Sabidussi (eds.), Graph Symmetry. Kluwer Academic Publishers, The Netherlands, 1997.

[17] D. M. Kwai and B. Parhami. A Class of Fixed-Degree Cayley-Graph Interconnection Networks Derived by Pruning k-ary n-cubes. In *Proceedings of the 26th International Conference on Parallel Processing*, Los Alamitos, California, August 1997

[18] S. Latifi and P. K. Srimani. A new fixed degree regular network for parallel processing. In *Proceedings of the Eighth IEEE Symposium on Parallel and Distributed Processing*, pages 152-159, Los Alamitos, California, October 1996.

[19] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann Publishers, San Mateo, California, 1992.

[20] W. Mader, Minimale n-fach kantenzusammenhangende Graphen, Math. Ann., vol. 191, pages 21-28, 1971

[21] Z. Miller, D. Pritikin and I. H. Sudborough. Near embeddings of hypercubes into Cayley graphs on the symmetric group. *IEEE Transactions on Computers*, vol. 43(1) 1994, 13-22.

[22] D. J. S. Robinson, *A Course in the Theory of Groups*, 2nd edition, Springer-Verlag, 1996.

[23] P. K. Srimani and P. Vadapalli. Trivalent Cayley graphs for interconnection networks. *Information Processing Letters*, vol. 54, pages 329-335, 1995.

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

[25] H. Stone. Parallel processing with the perfect shuffle. *IEEE Transactions on Computers*, C-20(2):153-161, February 1971

[26] M. E. Watkins. Connectivity of transitive graphs, J. Combin. Theory, vol. 8, pages 23-29