Nearly Optimal One-To-Many Parallel Routing in Star Networks

Chi-Chang Chen* Jianer Chen†

Abstract

Star networks were proposed recently as an attractive alternative to the well-known hypercube models for interconnection networks. Extensive research has been performed that shows that star networks are as versatile as hypercubes. This paper is an effort in the same direction. Based on the well-known paradigms, we study the one-to-many parallel routing problem on star networks and develop an improved routing algorithm that finds $n-1$ node-disjoint paths between one node and a set of other $n-1$ nodes in the n-star network. These parallel paths are proven of minimum length within a small additive constant, and our algorithm has an optimal time complexity. This result significantly improves the previous known algorithms for the problem. Moreover, the algorithm well illustrates an application of the orthogonal partition of star networks, which was observed by the original inventors of the star networks but seems generally overlooked in the subsequent study. We should also point out that similar problems are already studied for hypercubes and have proven useful in designing efficient and fault tolerant routing algorithms on hypercube networks.

1. Introduction

The star networks have received considerable attention recently by researchers as a graph model for interconnection network. Like the hypercube, the star network has rich structural and symmetric properties, and these properties have proven very useful in the study of network computation, communication, and fault tolerance. Moreover, star networks have smaller diameter and degree while comparing with hypercubes of comparable number of vertices [1, 2].

In this paper, we investigate the one-to-many routing problem in star networks. This problem has been studied by several researchers. In particular, Dietzfelbinger, Madhavapeddy, and Sudborough [6] developed an algorithm that solves the problem based on the recursive structure of the star networks. They showed that the length of the paths generated by their algorithm for the n-star network is bounded by $5(n-2)$. Since the diameter of the n-star network is $\lfloor 3(n-1)/2 \rfloor$, the lengths of these paths are bounded by $10/3$ times the diameter of the n-star network.

The main contribution of the current paper is an improved one-to-many routing algorithm. We adopt a different approach and develop new techniques to construct $n-1$ node-disjoint paths between one node and a set of $n-1$ other nodes in the n-star network. These node-disjoint paths are proven of minimum length (plus possibly at most 6). Our algorithm has an optimal time complexity. This significantly improves the result in [6]. Moreover, our algorithm well illustrates an application of the orthogonal partition of star networks, which was observed by the original inventors of the star networks [1] but seems generally overlooked in the subsequent study. We expect that the current paper make a contribution to motivating further important applications of the orthogonal partition of star networks.

2. Background and definitions

A permutation of the symbols 1, 2, ..., n can be represented by a cycle structure [7]. For example, permutation 231546 can be given as $(2\,3\,1)(4\,5)(6)$. A cycle is *nontrivial* if it contains more than one symbol, otherwise, the cycle is *trivial*. In general, if the cycle structure of a permutation u contains a cycle of form $(\cdots j\ i \cdots)$, then i is the jth symbol in the permutation u. In particular, if (i) is a cycle in u, then i is the ith symbol in u, i.e., the symbol i is in its "correct" position. A *transposition* $\pi[1,i]$ on a permutation u is to exchange the positions of the first symbol and the ith symbol in the permutation u. More precisely, if $u = a_1 a_2 \cdots a_{i-1} a_i a_{i+1} \cdots a_n$, then $\pi[1,i](u) = a_i a_2 \cdots a_{i-1} a_1 a_{i+1} \cdots a_n$.

The *n-dimensional star network* (or simply the *n-star network*), denoted as S_n, is an undirected graph consisting of $n!$ nodes labeled with the $n!$ permutations on symbols 1, 2, $\cdots$, n. There is an edge between any two nodes u and v if and only if $\pi[1,i](u) = v$ for some $2 \le i \le n$. The node

*Department of Computer Science and Engineering, Tatung Institute of Technology, Taipei 10451, Taiwan, R.O.C.. Supported in part by the National Science Council under Grant NCS 86-2213-E-036-004. Email: ccchen@cse.ttit.edu.tw.

†Department of Computer Science, Texas A&M University, College Station, TX 77843-3112. Supported in part by the National Science Foundation under Grant CCR-9110824. Email: chen@cs.tamu.edu.

labeled with the identity permutation $\varepsilon = 12\cdots n$ will be called the *identity node*. We will use a node name in a star network to refer to either the node itself or the permutation that labels the node. The context should always make this unambiguous.

The n-star network S_n is $(n-1)$-connected [2]. By Menger's theorem [8], for any given node u and any set D of $n-1$ other nodes in S_n, there are $n-1$ node-disjoint paths that connect the node u and each of the node in the set D. Moreover, every node in the n-star network S_n has degree $n-1$. Therefore, $n-1$ is the maximum number of node-disjoint paths we can expect to construct from a given node u to a given set of nodes. Finally, since the n-star network is vertex-symmetric [2], the problem of constructing $n-1$ node-disjoint paths between a given node to a set of $n-1$ nodes in the n-star network can be easily mapped to the problem of constructing $n-1$ node-disjoint paths between the identity node $\varepsilon = 12\cdots n$ to a set of $n-1$ nodes in the n-star network. Therefore, in the rest of this paper, we will concentrate on the latter problem.

Suppose that $u = c_1\cdots c_k e_1\cdots e_m$ is a node in the n-star network S_n, where c_i are nontrivial cycles and e_j are trivial cycles. A shortest path from the node u in the n-star network S_n to the identity node ε *must* be generated by repeatedly using the following three simple rules [1, 5]:

(R1) If 1 is the first symbol in u, then exchange 1 with any symbol j in a nontrivial cycle;

(R2) If i, $i \neq 1$, is the first symbol in u, then exchange i with the symbol j in the ith position;

(R3) If i, $i \neq 1$, is the first symbol in u, then exchange i with any symbol j in a nontrivial cycle that does not contain the symbol i.

We describe how these rules affect the cycle structure of the permutation u.

Suppose Rule (R1) is applied. Then the cycle structure of u should be of form $(j\cdots j')c_2\cdots c_k(1)$ (we have ignored irrelevant trivial cycles). Thus, j is the j'th symbol in u and Rule (R1) corresponds to the transposition $\pi[1,j']$ that "merges" the nontrivial cycle $(j\cdots j')$ into the trivial cycle (1): $\pi[1,j'](u) = (j\cdots j'1)c_2\cdots c_k$.

Suppose Rule (R2) is applied. Then the cycle structure of u should be of form $(ij\cdots j'1)c_2\cdots c_k$, where j is the ith symbol in u. Rule $(R2)$ corresponds to the transposition $\pi[1,i]$ that "deletes" the symbol i from the cycle containing symbol 1: $\pi[1,i](u) = (j\cdots j'1)c_2\cdots c_k$.

Finally, suppose that Rule (R3) is applied. Then the cycle structure of the node u should be of form $(i\cdots 1)(j\cdots j')c_3\cdots c_k$ (note that each cycle can be cyclically permuted and the order of the cycles is irrelevant). Rule (R3) corresponds to the transposition $\pi[1,j']$ that "merges" the nontrivial cycle $(j\cdots j')$ into the nontrivial cycle $(i\cdots 1)$: $\pi[1,j'](u) = (j\cdots j'i\cdots 1)c_3\cdots c_k$.

3. One-to-many parallel routing in star networks

In this section, we will present a one-to-many parallel routing algorithm on star networks. Given a set $D = \{v_2, v_3, \ldots, v_n\}$ of $n-1$ nodes in the n-star network S_n such that $v_i \neq \varepsilon$ for all i, the algorithm constructs $n-1$ node-disjoint paths P_2, P_3, ..., P_n such that the path P_i connects the nodes v_i and ε, for $2 \leq i \leq n$.

Definition Node u in the n-star network S_n is a (1)*-node* if 1 is the first symbol in the permutation u. Node v in S_n is a $(j1)$-node, where $2 \leq j \leq n$, if 1 is the jth symbol in the permutation v.

We point out that the above notations essentially match the notations of cycle structure of a permutation. The notation "(1)" hints the trivial cycle (1) in the cycle structure, while the notation "$(j1)$" hints a nontrivial cycle of form $(\cdots j1)$ in the cycle structure.

We first give an intuitive description of our algorithm. For each i, $2 \leq i \leq n$, we reserve a unique position p_i for the path P_i, $2 \leq p_i \leq n$, such that for all interior nodes on P_i (except possibly for at most two), the symbol 1 is at the p_ith position in the permutations. This construction ensures that the path P_i is node-disjoint from the other constructed paths. Finally, of course, we expect to keep the length of these paths as short as possible.

To achieve this, we need to solve three problems: (a) we need a procedure that constructs a short path from a node u to the node ε and keeps the symbol 1 in a fixed position along the path; (b) for each path P_i with the reserved position p_i, we need to minimize the number of nodes on P_i that are not (p_i1)-nodes; (c) all these paths P_i, $2 \leq i \leq n$, must be node-disjoint. In particular, those nodes on P_i that are not (p_i1)-nodes should not be used by other paths.

Given a node u that is not a (1)-node in the n-star network S_n, constructing a shortest path from u to ε that keeps the symbol 1 in a fixed position is relatively easy. Observe that if the node u is not a (1)-node, then applying Rule (R3) will not change the position of symbol 1 in the permutation u, and that if the cycle in u that contains symbol 1 has more than two symbols, then applying Rule (R2) does not change the position of symbol 1 in the permutation u. This observation motivates the algorithm FIX-1-ROUTER in Figure 1, which constructs a shortest path from a node u to ε and keeps the symbol 1 in a fixed position in all interior nodes along the path.

For example, let $u = (231)(45)(678)$ be a node in the

Algorithm. FIX-1-ROUTER

INPUT: a node u in S_n such that u is not a (1)-node

OUTPUT: a shortest path from u to ε in S_n

1. Repeatedly apply Rule (R3) to merge, in an arbitrary order, each nontrivial cycle into the cycle containing the symbol 1 until there is only one nontrivial cycle;
2. Repeatedly apply Rule (R2) to delete the symbol at the first position in a permutation from the nontrivial cycle containing symbol 1 until the node ε is reached.

Figure 1: The algorithm Fix-1-Router.

8-star network. The node u is not a (1)-node. A path created by the algorithm FIX-1-ROUTER may look as follows. $u = (231)(45)(678) \to (45231)(678) \to (67845231) \to (7845231) \to (845231) \to (45231) \to (5231) \to (231) \to (31) \to \varepsilon$

Now consider the path P_i with the reserved position p_i. To minimize the number of nodes on P_i that are not (p_i1)-nodes, we observe that every node in the n-star network S_n is connected to a (p_i1)-node by a path of length at most 2, as shownin the following lemma.

Lemma 3.1 (A) *Let u be a node in the n-star network S_n such that u is not a (1)-node. Then the node u has a unique neighbor that is a (1)-node. This (1)-node will be called* **the** *(1)-node associated with the node u.*

(B) *For any given p_i, $2 \le p_i \le n$, every (1)-node v in S_n has a neighbor that is a (p_i1)-node.*

PROOF. See the full paper [4]. □

Now we are ready to describe our algorithm that constructs $n-1$ node-disjoint paths from a set $D = \{v_2, v_3, \ldots, v_n\}$ of nodes in S_n to the identity node ε. For each node v_i in D, we construct a path P_i with a uniquely reserved position p_i. There are three different cases. If the node v_i is a (1)-node, then the path P_i starts with the edge from v_i to the neighbor $\pi[1, p_i](v_i)$ of v_i, which is a (p_i1)-node, and the rest nodes on P_i are all (p_i1)-nodes and are obtained by applying the algorithm FIX-1-ROUTER on the node $\pi[1, p_i](v_i)$. If the node v_i is a $(j1)$-node and the reserved position for v_i is j, then the path P_i is obtained by directly applying the algorithm FIX-1-ROUTER on the node v_i. Finally, if the node v_i is a $(j1)$-node and the reserved position p_i is not j, then we either go through the (1)-node associated with v_i then the (p_i1)-nodes, or go through a neighbor u of v_i then the (1)-node associated with u then the (p_i1)-nodes. The algorithm, called ONE-TO-MANY ROUTING, is presented formally in Figure 2.

It is easy to verify that for each i, the constructed path P_i connects the node v_i and the identity node ε. Moreover, we have the following bound on the length of the path P_i.

Algorithm. ONE-TO-MANY ROUTING

INPUT: a set $D = \{v_2, v_3, \ldots, v_n\}$ of $n-1$ nodes in the n-star network S_n

OUTPUT: $n-1$ node-disjoint paths connecting the nodes in D and the identity node ε in S_n

1. mark all nodes in D and mark the node ε;
2. **for** $j = 2$ **to** n **do**

   let D_j be the set of all $(j1)$-nodes in D; if $D_j \ne \phi$, pick a node w_j in D_j such that $dist(w_j)$ is the minimum among all nodes in D_j (if there is a tie, pick any of them); call w_j the *representative node* of the set D_j
3. for each node v_i in the set D, reserve a position p_i for v_i with $2 \le p_i \le n$, such that $v_i \ne v_t$ implies $p_i \ne p_t$, and that for each $D_j \ne \phi$, the representative node w_j gets the position j;

   {Now we start constructing a path P_i from v_i to ε for each i. Without loss of generality, assume that all representative nodes appear before all other nodes in the list $D = \{v_2, v_3, \ldots, v_n\}$.}
4. **for** $i = 2$ **to** n **do**

   - **case 1.** v_i is a (1)-node

     then the first interior node on the path P_i is $\pi[1, p_i](v_i)$, the rest of the path P_i is obtained by calling the algorithm FIX-1-ROUTER on the node $\pi[1, p_i](v_i)$;
   - **case 2.** v_i is the representative node for a set D_j

     then construct the path P_i by calling the algorithm FIX-1-ROUTER on the node v_i;
   - **case 3.** v_i is a $(j1)$-node for some j but not the representative node for set D_j

     if the (1)-node u associated with v_i is unmarked

     then the first two interior nodes on the path P_i are u and $\pi[1, p_i](u)$, and the rest of the path P_i is obtained by calling the algorithm FIX-1-ROUTER on the node $\pi[1, p_i](u)$

     else find an unmarked neighbor u of v_i such that the (1)-node u' associated with u is also unmarked; the first three interior nodes on the path P_i are u, u', and $\pi[1, p_i](u')$, and the rest of the path P_i is obtained by calling the algorithm FIX-1-ROUTER on the node $\pi[1, p_i](u')$

   mark all nodes on the path P_i;

Figure 2: The algorithm One-To-Many Routing.

Lemma 3.2 *For each i, $2 \le i \le n$, the length of the path P_i constructed by the algorithm* ONE-TO-MANY ROUTING *is bounded by $dist(v_i) + 6$.*

PROOF. For each i, $2 \le i \le n$, there is a node u_i on the path P_i such that the subpath of P_i from u_i to ε is obtained by applying the algorithm FIX-1-ROUTER on the node u_i. Moreover, the subpath from the node v_i to the node u_i has length bounded by 3. Therefore, $dist(u_i) \le dist(v_i) + 3$. Finally, since the algorithm FIX-1-ROUTER creates a shortest path from the node u_i to ε thus the path has length exactly $dist(u_i)$. In conclusion, the length of the path P_i is bounded by $3 + dist(u_i)$, which is bounded by $dist(v_i) + 6$. □

The rest of this section is for a proof of the correctness of the algorithm ONE-TO-MANY ROUTING. We start with a few simple facts on the algorithm. These facts can be verified easily.

Fact 3.1. If v_i is a (1)-node, then all interior nodes on the path P_i are (p_i1)-nodes, where p_i is the position reserved for the node v_i.

Fact 3.2. If v_i is the representative node w_j of a set D_j, then all interior nodes, plus the node v_i, are $(j1)$-nodes. Note that in this case, j is the position reserved for the node v_i.

Fact 3.3. If v_i is a $(j1)$-node but not the representative node w_j of the set D_j, then the path P_i starts with zero or one interior node that is a $(j1)$-node, followed by a (1)-node. The rest of the interior nodes on P_i are all (p_i1)-nodes, where p_i $(\neq j)$ is the position reserved for the node v_i.

We say that a node sequence $\{u_1, u_2, \ldots, u_s\}$ in the n-star network S_n is a *simple circle* if all nodes u_i are distinct and $[u_s, u_1]$, $[u_i, u_{i+1}]$, $1 \leq i \leq s-1$, are all edges in S_n. The following lemma will be very useful.

Lemma 3.3 *There is no simple circle of length less than* 6 *in the n-star network S_n.*

PROOF. See the full paper [4]. □

A crucial step in Algorithm ONE-TO-MANY ROUTING is the **case 3** in step 4, in which for the given $(j1)$-node v_i that is not the representative node w_j of the set D_j, if the (1)-node associated with v_i has been marked, we must show that the node v_i has a neighbor u such that the neighbor u and the (1)-node u' associated with u are both unmarked. Note that each marked node is a node contained in a previously constructed path P_t, $2 \leq t \leq i-1$.

The node v_i has $n-1$ neighbors, one is a (1)-node and the others are $(j1)$-nodes. Without loss of generality, let the $n-1$ neighbors of v_i be u_2', u_3, u_4, …, u_n, where u_2' is the (1)-node, and u_t, $3 \leq t \leq n$, are $(j1)$-nodes. By Lemma 3.1(A), each node u_t, $3 \leq t \leq n$, has a unique associated (1)-node u_t'. All these (1)-nodes are distinct — if $u_t' = u_s'$ for $t \neq s$, then the sequence $\{v_i, u_s, u_s', u_t\}$ would form a simple circle of length 4 in the n-star network, contradicting Lemma 3.3. Similarly, the neighbor u_2' of v_i is distinct from all (1)-nodes u_t', $3 \leq t \leq n$, since the n-star network S_n has no circles of length 3. Let $N_i = \{u_3, \ldots, u_n\}$ be the set of all neighbors of v_i that are $(j1)$-nodes, and let $A_i = \{u_2', u_3', \ldots, u_n'\}$ be the set of all (1)-nodes associated with the nodes in $N_i \cup \{v_i\}$.

Lemma 3.4 *Let v_i be a $(j1)$-node but not the representative node of set D_j and let the sets N_i and A_i be defined as above. Then each P_t of the paths P_2, …, P_{i-1} contains at most one node in the set N_i and at most one node in the set A_i. Moreover, if the path P_t contains a node u in the set N_i and a node u' in the set A_i, then u' must be the (1)-node associated with the node u.*

PROOF. See the full paper [4]. □

Now we are ready to show that Algorithm ONE-TO-MANY ROUTING is always valid in **case 3** of step 4 in the algorithm.

Lemma 3.5 *Let v_i be a node in* **case 3** *of step 4 in Algorithm* ONE-TO-MANY ROUTING. *If the (1)-node associated with v_i is marked, then there is a neighbor u of v_i such that the node u and the (1)-node u' associated with u are both unmarked.*

PROOF. See the full paper [4]. □

Now we are ready to show that all paths constructed by the algorithm ONE-TO-MANY ROUTING are node-disjoint. For this, the following lemma is sufficient.

Lemma 3.6 *Let P_i be the path from the node v_i to ε constructed in Algorithm* ONE-TO-MANY ROUTING, *then the path P_i is node-disjoint with all paths P_2, $\cdots$, P_{i-1} constructed by the algorithm.*

PROOF. See the full paper [4]. □

Now we can conclude with the following theorem.

Theorem 3.7 *Given a set $D = \{v_2, \ldots, v_n\}$ of $n-1$ distinct nodes in the n-star network S_n, the algorithm* ONE-TO-MANY ROUTING *constructs $n-1$ node-disjoint paths P_2, …, P_n in time $O(n^2)$ such that for $2 \leq i \leq n$, the path P_i connects the node v_i and the identity node ε in the n-star network S_n and has length at most $dist(v_i) + 6$.*

PROOF. The fact that the constructed paths satisfy the conditions stated in the theorem has been proved by Lemma 3.2 and Lemma 3.6. What remains is to show the complexity of the algorithm.

Calculating the distance from a node to the identity node can be done in time $O(n)$ using the formula given in Section 2. Thus, steps 1, 2, 3 of the algorithm ONE-TO-MANY ROUTING totally take time $O(n^2)$. Since the diameter of the n-star network S_n is bounded by $O(n)$ [2], the algorithm FIX-1-ROUTER constructs a shortest path from a given node to the identity node ε in time $O(n)$. For each node v_i in step 4, we may need to search among the $n-1$ neighbors of v_i to find a neighbor u such that u and the (1)-node associated with u are both unmarked. This searching takes time $O(n)$ for each node. After this searching, the algorithm FIX-1-ROUTER is applied and takes time $O(n)$. In conclusion, step 4 of the algorithm ONE-TO-MANY ROUTING takes time $O(n^2)$. Summarizing all these together, we conclude that the algorithm ONE-TO-MANY ROUTING runs in time $O(n^2)$. □

We point out that the algorithm ONE-TO-MANY ROUTING has an optimal time complexity. In fact, even printing out $n-1$ paths whose length can be as long as $\Omega(n)$ already takes time $\Omega(n^2)$.

4. Orthogonal partition of star networks

Like most well-known interconnection network models, star networks enjoy a recursive structure that has been proven very useful in network computation and communication.

Fix a position $p > 1$ in the permutations on 1, 2, $\ldots$, n (e.g., $p = n$ is the last position in the permutations). For each i, $1 \leq i \leq n$, the nodes in the n-star network S_n whose pth symbol is i form an $(n-1)$-dimensional substar network $S_{p,i}$. The n-dimensional star network S_n is partitioned into n node-disjoint $(n-1)$-dimensional substar networks $S_{p,1}$, $S_{p,2}$, $\ldots$, $S_{p,n}$ [1]. This partition of the n-star network is well-known and is called the *standard partition* of the n-star network. The standard partition of star networks has found wide applications in routing, broadcasting, emulation, and fault tolerance of star networks. In particular, the parallel one-to-many routing algorithm by Dietzfelbinger, Madhavapeddy, and Sudborough [6] that constructs $n-1$ node-disjoint paths between a given node and a set of $n-1$ other nodes in the n-star network is based on the standard partition of star networks.

If instead of fixing a position p, we fix the symbol 1 (or any of the other symbols), and consider the $n-1$ node-disjoint $(n-1)$-dimensional substar networks $S_{2,1}$, $S_{3,1}$, $\ldots$, $S_{n,1}$. Using our notation in the previous sections, the substar network $S_{j,1}$ consists of all $(j1)$-nodes in S_n. The rest nodes in S_n that are not contained in any of these $(n-1)$-dimensional substar networks are the (1)-nodes in S_n. We say that the $(n-1)!$ (1)-nodes of S_n from a "virtual $(n-1)$-dimensional substar network $S_{1,1}$" in S_n. This partition of the n-star network is called the *orthogonal partition* of the n-star network. Figure 3 illustrates the orthogonal partition of the 4-star network. The orthogonal partition of star networks was observed by Akers, Harel, and Krishnamurthy in their seminal paper on star networks [1]. However, this important property of star networks seems overlooked in the subsequent study of star networks.

The (1)-nodes in the virtual $(n-1)$-substar network $S_{1,1}$ of the n-star network S_n form a very interesting set of nodes in S_n. This set is a maximal independent set (i.e., no two (1)-nodes are adjacent) as well as a minimal dominating set (i.e., every other node in S_n is adjacent to a (1)-node). Moreover, each (1)-node v provides a nice "bridge" for the $n-1$ node-disjoint $(n-1)$-substar networks $S_{2,1}$, $S_{3,1}$, $\ldots$,

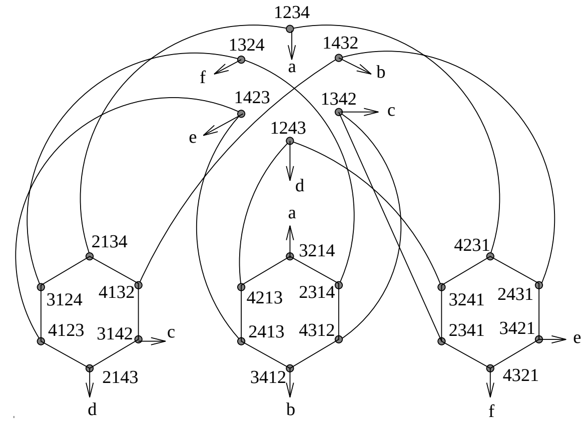

Figure 3: The orthogonal partition of the 4-star network.

$S_{n,1}$: v is adjacent to a node in each of the $(n-1)$-substar networks (these properties of n-star network were proven in Lemma 3.1). These properties seem very useful in star network communication and computation.

Our algorithm ONE-TO-MANY ROUTING is a good example that well illustrates an application of the orthogonal partition of star networks. In order to make the $n-1$ constructed paths node-disjoint, the algorithm reserves for each node v_i in the set D a distinct $(n-1)$-substar network S'_i in the orthogonal partition of the n-star network S_n so that the path P_i connecting v_i and ε is essentially in the substar S'_i. The substar network S'_i is actually the substar network $S_{p_i,1}$ consisting of all (p_i1)-nodes in S_n, where p_i is the position reserved for the node v_i in the algorithm.

Based on the orthogonal partition of the n-star network, Algorithm ONE-TO-MANY ROUTING can be interpreted as follows. If the node v_i is a (1)-node (**case 1** in step 4), then since v_i is adjacent to a node in each of the substar networks in the orthogonal partition of S_n, the path P_i can go from v_i by a single edge to the substar network S'_i and the rest part of P_i is entirely within S'_i. This case is illustrated by Case 1 in Figure 4. If the node v_i is the representative node of the set D_{p_i} (**case 2** in step 4), then the path P_i is entirely contained in the substar network S'_i. This case is illustrated by Case 2 in Figure 4. Finally, if the node v_i is a $(j1)$-node but not the representative node of the set D_j, then either v_i is adjacent to an unused (1)-node or v_i has an unused neighbor that is adjacent to an unused (1)-node (this is formally proved by Lemma 3.5 in Section 3). Thus, we can get out of the current substar network $S_{j,1}$ in one or two steps and arrive the virtual substar network without hitting any used nodes. Now again in one step we can get into the reserved substar network $S'_i = S_{p_i,1}$ and do the rest of the routing. This case is illustrated by Case 3 in Figure 4

We believe that the orthogonal partition of star networks has a great potential for further important applications in the study of computation and communication on star networks.

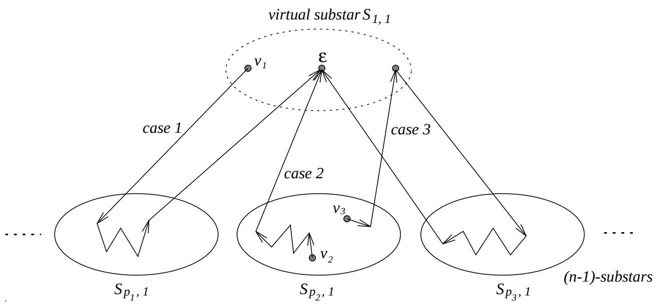

Figure 4: The algorithm One-To-Many Routing illustrated by the orthogonal partition of S_n. Case 1: node v_1 is a (1)-node; Case 2: node v_2 is the representative node for set D_{p_2}; Case 3: node v_3 is a $(p_2 1)$-node but not the representative node for set D_{p_2}.

Another interesting application of the orthogonal partition of star networks was explained in [3], in which it is shown that based on the orthogonal partitions, star networks can be represented in a much more condensed way (the n-star network is represented by a graph of n nodes) and this representation can be used in the study of emulations of star networks. We expect that the current paper make a contribution to motivating further applications of the orthogonal partition of star networks.

References

[1] S. B. Akers, D. Harel, and B. Krishnamurthy, The star graph: an attractive alternative to the n-cube, *Proc. Intl. Conf. of Parallel Processing*, (1987), pp. 393-400.

[2] S. B. Akers and B. Krishnamurthy, A group-theoretic model for symmetric interconnection networks, *IEEE Trans. on Computers 38*, (1989), pp. 555-565.

[3] C. C. Chen, Combinatorial and algebraic methods in star and de Bruijn networks, *Ph.D. Dissertation*, Dept. of Computer Science, Texas A&M University, 1995.

[4] C. C. Chen and J. Chen, Nearly Optimal One-to-Many Parallel Routing in Star Networks, *Tech. Rep. 96-003*, Dept. of Computer Science, Texas A&M University, (1997).

[5] K. Day and A. Tripathi, A comparative study of topological properties of hypercubes and star graphs, *IEEE Trans. Parallel, Distrib. Syst. 5*, (1994), pp. 31-38.

[6] M. Dietzfelbinger, S. Madhavapeddy, and I. H. Sudborough, Three disjoint path paradigms in star networks, *Proc. 3nd IEEE Symposium on Parallel and Distributed Processing*, (1991), pp. 400-406.

[7] C. L. Liu, *Elements of Discrete Mathematics*, McGraw-Hill Book Company, 1986.

[8] K. Menger, Zur allgemeinen kurventheorie, *Fund. Math. 10*, (1927), pp. 96-115.