

Strong Fault-Tolerance: Parallel Routing in Networks with Faults^{*}

Jianer Chen and Eunseuk Oh

Department of Computer Science, Texas A&M University,
College Station, TX 77843-3112, USA
Email: {chen,eunseuko}@cs.tamu.edu

Abstract. A d -regular network G is *strongly fault-tolerant* if for any copy G_f of G with at most $d - 2$ faulty nodes, every pair of non-faulty nodes u and v in G_f admits $\min\{d_f(u), d_f(v)\}$ node-disjoint paths in G_f from u to v , where $d_f(u)$ and $d_f(v)$ are the degrees of the nodes u and v in G_f . We show that popular network structures, such as hypercube and star networks, are strongly fault-tolerant. We develop efficient algorithms that construct the maximum number of node-disjoint paths of optimal or nearly optimal length in these networks with faulty nodes. Our algorithms are optimal in terms of their running time.

1 Introduction

Routing on large size networks with faults is an important issue in the study of computer interconnection networks. In this paper, we introduce a new measure for network fault tolerance: the *strong fault-tolerance*. Consider a network G and let G_f be a copy of G with a set S_f of faulty nodes. Let u and v be two non-faulty nodes in G_f . Based on local information, we know the degrees $d_f(u)$ and $d_f(v)$ of the nodes u and v in G_f and are interested in constructing the maximum number of node-disjoint paths between u and v in G_f . Obviously, the number of node-disjoint paths between u and v in G_f cannot be larger than $\min\{d_f(u), d_f(v)\}$. We are interested in knowing the precise bound on the size of the faulty node set S_f such that for any two non-faulty nodes u and v in G_f , there are $\min\{d_f(u), d_f(v)\}$ node-disjoint paths between u and v .

Clearly, if the network G is d -regular (i.e., all of its nodes have degree d), then in general the number of faulty nodes in the set S_f should not exceed $d - 2$ to ensure $\min\{d_f(u), d_f(v)\}$ node-disjoint paths between any two nodes u and v in G_f . This can be seen as follows. Let u and v be two nodes in G whose distance is larger than 3. Pick any neighbor u' of u and remove the $d - 1$ neighbors of u' that are not u . Note that no neighbor of u' can be a neighbor of v since the distance from u to v is at least 4. Let the resulting network be G_f . The degrees of the nodes u and v in G_f are d . However, there are obviously no d node-disjoint paths in G_f from u to v since one of the d neighbors of u in G_f , the node u' , leads to a “dead-end”. This motivates the following definition.

^{*} This work is supported in part by NSF under Grant CCR-0000206.

Definition 1. A d -regular network G is *strongly fault-tolerant* if for any copy G_f of G with at most $d - 2$ faulty nodes, every pair of non-faulty nodes u and v admits $\min\{d_f(u), d_f(v)\}$ node-disjoint paths from u to v in G_f .

Strong fault-tolerance characterizes the property of parallel routing in a network with faulty nodes. Since one of the motivations of network parallel routing is to provide alternative routing paths when failures occur, strong fault-tolerance can also be regarded as the study of fault tolerance in networks with faults.

To authors' knowledge, there has not been a systematic study on parallel routing on networks with faults. In this paper, we will concentrate on the discussion on strong fault-tolerance of two extensively studied interconnection network structures, the hypercube networks and the star networks. We first give a brief review on the previous related research on these networks.

The hypercube networks are among the earliest and still remain as one of the most important and attractive network models. A large number of fault-tolerant algorithms dealing with single-path routing in the hypercube networks have been proposed (see, for example, [8,7,12]). Parallel routing on hypercube networks without faulty nodes was studied in [18]. The general fault tolerance properties of the star networks were first studied and analyzed in [1,2,3]. Algorithms for single-path routing in star networks with faults were developed in [4,11,17]. Parallel routing algorithms on the star networks were studied in [10,9]. In particular, an efficient algorithm has been developed [6] that constructs the maximum number of node-disjoint paths of optimal length for any two nodes in the star networks. A randomized algorithm, based on the Information Dispersal Algorithm [15], for parallel routing in the star networks with faults was proposed in [16].

We will first study the strong fault-tolerance for the star networks. Taking the advantage of the *orthogonal decomposition* of the star networks, we develop an efficient algorithm that constructs node-disjoint paths between any two non-faulty nodes in the n -star network S_n with at most $n - 3$ faulty nodes: for any two non-faulty nodes u and v , our algorithm constructs $\min\{d_f(u), d_f(v)\}$ node-disjoint paths of minimum length plus a small constant between u and v .

Since the hypercube networks do not have a similar orthogonal decomposition structure, the techniques in parallel routing for the star networks with faults are not applicable to the hypercube networks. In order to effectively route parallel paths in the hypercube networks with faults, we develop a new technique called "neighbor pre-matching". Based on this technique, an algorithm is developed that constructs $\min\{d_f(u), d_f(v)\}$ node-disjoint paths of optimal length for any pair of nodes u and v in the n -cube network with at most $n - 2$ faulty nodes.

Our algorithms have optimal running time.

2 Strong Fault-Tolerance of the Star Networks

The n -star network S_n is an undirected graph consisting of $n!$ nodes labeled with the $n!$ permutations on symbols $\{1, 2, \dots, n\}$. There is an edge between two nodes u and v in S_n if and only if the permutation v can be obtained from

the permutation u by exchanging the positions of the first symbol and another symbol. The n -star network is $(n - 1)$ -regular. The star networks have received considerable attention as an attractive alternative to the widely used hypercube network model [2].

Each node u in the n -star network S_n , which is a permutation on the symbols $\{1, 2, \dots, n\}$, can be given as a product of disjoint cycles (called the *cycle structure* for u) [5]. For each $2 \leq i \leq n$, an operation ρ_i on the permutations of $\{1, 2, \dots, n\}$ is defined such that $\rho_i(u)$ is the permutation obtained from the permutation u by exchanging the first symbol and the i th symbol in u .

Denote by ε the identity permutation $\langle 12 \cdots n \rangle$. Since the n -star network S_n is vertex-symmetric [2], a set of node-disjoint paths from a node w to a node v can be easily mapped to a set of node-disjoint paths from a node u to ε . Therefore, we only need to concentrate on the construction of node-disjoint paths from u to ε in S_n . Let $\text{dist}(u)$ denote the distance, i.e., the length of the shortest paths, from u to ε in S_n . The value $\text{dist}(u)$ can be easily computed using the formula given in [2]. We say that an edge $[u, v]$ (in this direction) in S_n *does not follow the Shortest Path Rules* if $\text{dist}(u) \leq \text{dist}(v)$ [6]. It is also easy to check whether an edge follows the Shortest Path Rules [6].

Lemma 1. *If an edge $[u, v]$ in S_n does not follow the Shortest Path Rules, then $\text{dist}(v) = \text{dist}(u) + 1$. Consequently, let P be a path from u to ε in which exactly k edges do not follow the Shortest Path Rules, then the length of the path P is equal to $\text{dist}(u) + 2k$.*

For the n -star network S_n , let $S_n[i]$ be the set of nodes in which the symbol 1 is at the i th position. It is well-known [1] that the set $S_n[1]$ is an independent set, and the subgraph induced by the set $S_n[i]$ for $i \neq 1$ is an $(n - 1)$ -star network.

Our parallel routing algorithm is heavily based on the concept of *bridging paths* that connect a given node to a specific substar network in the n -star network. The following lemma will serve as a basic tool in our construction.

Lemma 2. *Let u be any non-faulty node in the substar $S_n[i]$ with $k_i \leq n - 3$ faulty nodes, $i \neq 1$. A fault-free path P from u to $\rho_i(\varepsilon)$ can be constructed in $S_n[i]$ in time $O(k_i n + n)$ such that at most two edges in P do not follow the Shortest Path Rules. In case u has a cycle of form $(i1)$, the constructed path P has at most one edge not following the Shortest Path Rules.*

Definition 2. Let u be a node in the n -star network S_n and u' be a neighbor of u in the substar $S_n[i]$, $i \neq 1$. For each neighbor v of u' , $v \neq u$, a (u', j) -bridging path (of length at most 4) from u to the substar $S_n[j]$, $j \neq 1, i$, is defined as follows: if v is in $S_n[1]$ then the path is $[u, u', v, \rho_j(v)]$, while if v is in $S_n[i]$ then the path is $[u, u', v, \rho_i(v), \rho_j(\rho_i(v))]$.

Thus, from each neighbor u' in $S_n[i]$ of the node u , $i \neq 1$, there are $n - 2$ (u', j) -bridging paths of length bounded by 4 that connect the node u to the substar $S_n[j]$.

Since no two nodes in $S_n[i]$ share the same neighbor in $S_n[1]$ and no two nodes in $S_n[1]$ share the same neighbor in $S_n[j]$, for any neighbor u' of u , two

(u', j) -bridging paths from u to $S_n[j]$ have only the nodes u and u' in common. Moreover, for any two neighbors u' and u'' of u in $S_n[i]$ (in this case, the node u must itself be also in $S_n[i]$), since u' and u'' have no other common neighbor except u (see, for example, [9]), a (u', j) -bridging path from u to $S_n[j]$ and a (u'', j) -bridging path from u to $S_n[j]$ share no nodes except u .

Definition 3. Let u be a node in S_n and let u' be a neighbor of u in $S_n[i]$, $i \neq 1$. A (u', j) -bridging path P from the node u to the substar $S_n[j]$ is *divergent* if in the subpath of P from u to $S_n[1]$, there are three edges not following the Shortest Path Rules.

Note that the subpath from u to $S_n[1]$ of a (u', j) -bridging path P contains at most three edges. In particular, if the subpath contains only two edges, then the path P is automatically non-divergent.

In case there are no faulty nodes in the n -star network, each divergent (u', j) -bridging path can be efficiently extended into a path from u to $\rho_j(\varepsilon)$, as shown in the following lemma. A detailed proof for the lemma can be found in [13].

Star-PRouting

Input: a non-faulty node u in S_n with at most $n - 3$ faulty nodes.

Output: $\min\{d_f(u), d_f(\varepsilon)\}$ parallel paths of length $\leq \text{dist}(u) + 8$ from u to ε .

1. **if** the node u is in $S_n[1]$
 - 1.1. **then for** each $j \neq 1$ with both $\rho_j(u)$ and $\rho_j(\varepsilon)$ non-faulty **do**
 - construct a path P_j of length $\leq \text{dist}(u) + 6$ in $S_n[j]$ from u to ε ;
 - 1.2. **else** (* the node u is in a substar $S_n[i]$, $i \neq 1$ *)
 - 1.2.1. **if** the node $\rho_i(\varepsilon)$ is non-faulty
 - then** pick a non-faulty neighbor v of u and construct a path P_v of length $\leq \text{dist}(u) + 4$ from u to ε such that all internal nodes of P_v are in $S_n[i]$ and P_v does not intersect a (u', j) -bridging path for any non-faulty neighbor $u' \neq v$ of u ;
 - 1.2.2. **if** the neighbor $u_1 = \rho_i(u)$ of u in $S_n[1]$ is non-faulty
 - then** find $j \neq 1, i$, such that both $\rho_j(u_1)$ and $\rho_j(\varepsilon)$ are non-faulty; extend the path $[u, u_1, \rho_j(u_1)]$ from $\rho_j(u_1)$ to $\rho_j(\varepsilon)$ in $S_n[j]$ to make a path of length $\leq \text{dist}(u) + 8$ from u to ε ;
2. maximally pair the non-faulty neighbors of u and ε that are not used in step 1: $(u'_1, \rho_{j_1}(\varepsilon)), \dots, (u'_g, \rho_{j_g}(\varepsilon))$;
3. **for** each pair $(u', \rho_j(\varepsilon))$ constructed in step 2 **do**
 - 3.1. **if** there is a non-divergent (u', j) -bridging path P with neither faulty nodes nor nodes used by other paths
 - then** pick this (u', j) -bridging path P
 - else** pick a divergent (u', j) -bridging path P with neither faulty nodes nor nodes used by other paths;
 - 3.2. extend the path P into a fault-free path $P_{u'}$ of length $\leq \text{dist}(u) + 8$ from u to ε such that the extended part is entirely in $S_n[j]$;

Fig. 1. Parallel routing on the star network with faulty nodes

Lemma 3. *There is an $O(n)$ time algorithm that, for a divergent (u', j) -bridging path P from a node u to a substar $S_n[j]$, extends P in $S_n[j]$ into a path Q from u to $\rho_j(\varepsilon)$, in which at most 4 edges do not follow the Shortest Path Rules. Moreover, for two divergent (u', j) -bridging paths P_1 and P_2 , the two corresponding extended paths Q_1 and Q_2 have only the nodes u , u' , and $\rho_j(\varepsilon)$ in common.*

Based on the above discussion, we are now ready to present our parallel routing algorithm on star networks with faults, which is given in Figure 1.

Step 1 of the algorithm constructs certain number of paths between non-faulty neighbors of the node u and non-faulty neighbors of the node ε , Step 2 of the algorithm maximally pairs the rest non-faulty neighbors of u with the rest non-faulty neighbors of ε . It is easy to see that the number of pairs constructed in Step 2 plus the number of paths constructed in Step 1 is exactly $\min\{d_f(u), d_f(\varepsilon)\}$. Since Step 3 of the algorithm constructs a path from u to ε for each pair constructed in Step 2, the algorithm **Star-PRouting** constructs exactly $\min\{d_f(u), d_f(\varepsilon)\}$ paths from u to ε . In fact, we can prove the following theorem (a detailed proof of the theorem can be found in [13]).

Theorem 1. *If the n -star network S_n has at most $n - 3$ faulty nodes and the node ε is non-faulty, then for any non-faulty node u in S_n , in time $O(n^2)$ the algorithm **Star-PRouting** constructs $\min\{d_f(u), d_f(\varepsilon)\}$ node-disjoint fault-free paths of length bounded by $\text{dist}(u) + 8$ from the node u to the node ε .*

The following example shows that the bound on the path length in Theorem 1 is actually almost optimal. Consider the n -star network S_n . Let the source node be $u = (21)$, here we have omitted the trivial cycles in the cycle structure. Then $\text{dist}(u) = 1$. Suppose that all neighbors of u and all neighbors of ε are non-faulty. By Theorem 1, there are $n - 1$ node-disjoint fault-free paths from u to ε . Thus, for each i , $3 \leq i \leq n$, the edge $[u, u_i]$ leads to one P_i of these node-disjoint paths from u to ε , where $u_i = (i21)$. Note that the edge $[u, u_i]$ does not follow the Shortest Path Rules. Now suppose that the node $(i2)(1)$ is faulty, for $i = 3, 4, \dots, n - 1$ (so there are $n - 3$ faulty nodes). Then the third node on the path P_i must be $v_i = (ji21)$ for some $j \neq 1, 2, i$, and the edge $[u_i, v_i]$ does not follow the Shortest Path Rules. Since the only edge from v_i that follows the Shortest Path Rules is the edge $[v_i, u_i]$, the next edge $[v_i, w_i]$ on P_i again does not follow the Shortest Path Rules. Now since all the first three edges on P_i do not follow the Shortest Path Rules, by Lemma 1, $\text{dist}(w_i) = \text{dist}(u) + 3 = 4$, and the path P_i needs at least four more edges to reach ε . That is, the length of the path P_i is at least $7 = \text{dist}(u) + 6$. Thus, with $n - 3$ faulty nodes, among the $n - 1$ node-disjoint paths from u to ε , at least $n - 3$ of them must have length larger than or equal to $\text{dist}(u) + 6$.

The situation given above seems a little special since the distance $\text{dist}(u)$ from u to ε is very small. In fact, even for nodes u with large $\text{dist}(u)$, we can still construct many examples in which some of the node-disjoint fault-free paths connecting u and ε must have length at least $\text{dist}(u) + 6$ (see [13] for details).

3 Strong Fault-Tolerance of the Hypercube Networks

The construction of parallel routing paths in the star networks in the previous section heavily takes the advantage of the orthogonal decomposition structure $\{S_n[1], S_n[2], \dots, S_n[n]\}$ of the star networks. In particular, the (u', j) -bridging paths route a node u through the substar $S_n[1]$ to the substar $S_n[j]$ so that extension of the paths can be recursively constructed in the substar $S_n[j]$. Unfortunately, the hypercube networks do not have a similar orthogonal decomposition structure so the above techniques are not applicable. We need to develop new techniques for parallel routing in the hypercube networks with faults. Because of space limit, some details are omitted. Interested readers are referred to [14].

Recall that an n -cube network Q_n is an undirected graph consisting of 2^n nodes labeled by the 2^n binary strings of length n . Two nodes are adjacent if they differ by exactly one bit. An edge is an i -edge if its two ends differ by the i th bit. The (Hamming) distance $dist(u, v)$, i.e., the length of the shortest paths, from u to v is equal to the number of the bits in which u and v differ. Since the n -cube network Q_n is vertex-symmetric, we can concentrate, without loss of generality, on the construction of node-disjoint paths from a node of form $u = 1^r 0^{n-r}$ to the node $\varepsilon = 0^n$. Define $dist(u) = dist(u, \varepsilon)$.

The node connected from the node u by an i -edge is denoted by u_i , and the node connected from the node u_i by a j -edge is denoted by $u_{i,j}$. A path P from u to v can be uniquely specified by a sequence of labels of the edges on P in the order of traversal, denoted by $u\langle i_1, \dots, i_r \rangle v$. This notation can be further extended to a set of paths. Thus, for a set S of permutations, $u\langle S \rangle v$ is the set of paths of the form $u\langle i_1, \dots, i_r \rangle v$, where $\langle i_1, \dots, i_r \rangle$ is a permutation in S .

Our parallel routing algorithm is based on an effective pairing of the neighbors of the nodes u and ε . We first assume that the nodes u and ε have no faulty neighbors. We pair the neighbors of u and ε by the following strategy.

prematch-1

pair u_i with ε_{i-1} for $1 \leq i \leq r$,¹ and pair u_j with ε_j for $r+1 \leq j \leq n$.

Under the pairing given by **prematch-1**, we construct parallel paths between the paired neighbors of u and ε using the following procedure.

procedure-1

1. for each $1 \leq i \leq r$, construct $n-2$ paths from u_i to ε_{i-1} , $r-2$ of them are of the form $u_i\langle S_1 \rangle \varepsilon_{i-1}$, where S_1 is the set of all cyclic permutations of the sequence $(i+1, \dots, r, 1, \dots, i-2)$; and $n-r$ of them are of the form $u_i\langle h, i+1, \dots, r, 1, \dots, i-2, h \rangle \varepsilon_{i-1}$, where $r+1 \leq h \leq n$.
2. for each $r+1 \leq j \leq n$, construct $n-1$ paths from u_j to ε_j , r of them are of the form $u_j\langle S_2 \rangle \varepsilon_j$, where S_2 is the set of all cyclic permutations of the sequence $(1, 2, \dots, r)$, and $n-r-1$ of them are of the form $u_j\langle h, 1, 2, \dots, r, h \rangle \varepsilon_j$, where $h \neq j$, and $r+1 \leq h \leq n$.

¹ The operations on indices are by mod r . Thus, ε_0 is interpreted as ε_r .

The paths constructed by cyclic permutations of a sequence are pairwise disjoint (see, for example [18]). It is easy to verify that for each pair of neighbors of u and ε , the paths constructed between them are pairwise disjoint.

Lemma 4. *Let (u_h, ε_g) and (u_s, ε_t) be two pairs given by **prematch-1**. Then, there is at most one path in the path set constructed by **procedure-1** for the pair (u_h, ε_g) that shares common nodes with a path in the path set constructed by **procedure-1** for the pair (u_s, ε_t) .*

Since the n -cube network Q_n may have up to $n - 2$ faulty nodes, there is a possibility that for a pair (u_i, ε_{i-1}) constructed by **prematch-1**, where $1 \leq i \leq r$, all $n - 2$ paths constructed by **procedure-1** from u_i to ε_{i-1} are blocked by faulty nodes. In this case, we can directly construct n node-disjoint paths from u to ε , as follows.

Assumption. there is a pair (u_i, ε_{i-1}) given by **prematch-1**, $1 \leq i \leq r$, such that all $n - 2$ paths constructed by **prematch-1** for the pair are blocked by faulty nodes.

prematch-2

1. pair u_i with ε_{i-2} , u_{i-1} with ε_i , and u_{i+1} with ε_{i-1} ;
2. for other neighbors of u and ε , pair them as in **prematch-1**;

procedure-2

1. from u_i to ε_{i-2} : $u_i \langle i - 1, \dots, r, 1, \dots, i - 3 \rangle \varepsilon_{i-2}$;
2. from u_{i-1} to ε_i : $u_{i-1} \langle i + 1, \dots, r, 1, \dots, i - 2 \rangle \varepsilon_i$;
3. from u_{i+1} to ε_{i-1} : $u_{i+1} \langle i + 2, \dots, r, 1, \dots, i - 2, i \rangle \varepsilon_{i-1}$;
4. for $g \neq i - 1, i, i + 1$, $1 \leq g \leq r$: $u_g \langle g + 1, \dots, r, 1, \dots, g - 2 \rangle \varepsilon_{g-1}$;
5. for $r + 1 \leq j \leq n$: if $i = 1$ then $u_j \langle 2, \dots, r, 1 \rangle \varepsilon_j$ else $u_j \langle 1, \dots, r \rangle \varepsilon_j$.

Lemma 5. *Under the conditions of **prematch-2**, the algorithm **procedure-2** constructs n fault-free parallel paths of length $\leq \text{dist}(u) + 4$ from u to ε .*

Now we consider the case in which the nodes u and ε have faulty neighbors. We apply the following pairing strategy, which pairs the edges incident on the neighbors of the nodes u and ε , instead of the neighbors of u and ε .

prematch-3

for each edge $[u_i, u_{i,i'}]$ where both u_i and $u_{i,i'}$ are non-faulty **do**

1. if $1 \leq i, i' \leq r$ and $i' = i + 1$, then pair $[u_i, u_{i,i'}]$ with the edge $[\varepsilon_{i-1, i-2}, \varepsilon_{i-1}]$, if $\varepsilon_{i-1, i-2}$ and ε_{i-1} are non-faulty;
2. if $1 \leq i, i' \leq r$ and $i' = i - 1$, then pair $[u_i, u_{i,i'}]$ with the edge $[\varepsilon_{i'-1, i'-2}, \varepsilon_{i'-1}]$, if $\varepsilon_{i'-1, i'-2}$ and $\varepsilon_{i'-1}$ are non-faulty;
3. otherwise, pair $[u_i, u_{i,i'}]$ with the edge $[\varepsilon_{j, j'}, \varepsilon_j]$ if $\varepsilon_{j, j'}$ and ε_j are non-faulty, where the indices j and j' are such that **prematch-1** pairs the node $u_{i'}$ with ε_j , and the node u_i with $\varepsilon_{j'}$.

Note that it is possible that an edge $[u_i, u_{i,i'}]$ with both u_i and $u_{i,i'}$ non-faulty is not paired with any edge because the corresponding edge in **prematch-3** contains faulty nodes. For each pair of edges given by **prematch-3**, we construct a path as follows.

procedure-3

1. for $1 \leq i, i' \leq r$, $i' = i + 1$, and paired edges $[u_i, u_{i,i'}]$, $[\varepsilon_{i-1,i-2}, \varepsilon_{i-1}]$, construct the path $u_i \langle i', \dots, i-2 \rangle \varepsilon_{i-1}$;
2. for $i \leq i, i' \leq r$, $i' = i - 1$, and paired edges $[u_i, u_{i,i'}]$, $[\varepsilon_{i'-1,i'-2}, \varepsilon_{i'-1}]$, construct a path by flipping i and i' in the path $u_{i'} \langle i, \dots, i-2 \rangle \varepsilon_{i'-1}$;
3. Otherwise, for paired edges $[u_i, u_{i,i'}]$, $[\varepsilon_{j,j'}, \varepsilon_j]$, if $i < i'$, construct a path by flipping j and j' in the path $u_i \langle i', \dots, j \rangle \varepsilon_{j'}$; if $i > i'$, construct a path by flipping i and i' in the path $u_{i'} \langle i, \dots, j' \rangle \varepsilon_j$.

Now we are ready to present our main algorithm for parallel routing in the hypercube networks with faults. The algorithm is given in Figure 2.

Cube-PRouting

input: non-faulty nodes $u = 1^r 0^{n-r}$ and $\varepsilon = 0^n$ in Q_n with $\leq n - 2$ faults.
output: $\min\{d_f(u), d_f(\varepsilon)\}$ parallel paths of length $\leq \text{dist}(u) + 4$ from u to ε .

1. **case 1.** u and ε have no faulty neighbors
for each pair (u_i, ε_j) given by **prematch-1** **do**
 - 1.1. **if** all paths for (u_i, ε_j) by **procedure-1** include faulty nodes
then use **prematch-2** and **procedure-2** to construct n parallel paths from u to ε ; **STOP**.
 - 1.2. **if** there is a fault-free unoccupied path from u_i to ε_j by **procedure-1**
then mark the path as occupied by (u_i, ε_j) ;
 - 1.3. **if** all fault-free paths constructed for (u_i, ε_j) include occupied nodes
then pick any fault-free path P for (u_i, ε_j) , and for the pair $(u_{i'}, \varepsilon_{j'})$ that occupies a node on P , find a new path;
2. **case 2.** there is at least one faulty neighbor of u or ε
for each edge pair $([u_i, u_{i,i'}], [\varepsilon_{j,j'}, \varepsilon_j])$ by **prematch-3** **do**
 - 2.1. **if** there is a fault-free unoccupied path from u_i to ε_j by **procedure-3**
then mark the path as occupied by the pair $([u_i, u_{i,i'}], [\varepsilon_{j,j'}, \varepsilon_j])$;
 - 2.2. **if** all fault-free paths for the pair include occupied nodes
then pick any fault-free path P for the edge pair, and for the edge pair that occupies a node on P , find a new path.

Fig. 2. Parallel routing on the hypercube network with faulty nodes

Lemma 5 guarantees that step 1.1 of the algorithm **Cube-PRouting** constructs n fault-free parallel paths of length $\leq \text{dist}(u) + 4$ from u to ε . Step 1.3 of the algorithm requires further explanation. In particular, we need to show that for the pair $(u_{i'}, \varepsilon_{j'})$, we can always construct a new fault-free path from $u_{i'}$ to $\varepsilon_{j'}$ in which no nodes are occupied by other paths. This is ensured by the following lemma.

Lemma 6. *Let (u_i, v_j) and $(u_{i'}, v_{j'})$ be two pairs given by **prematch-1** such that two paths constructed for (u_i, v_j) and $(u_{i'}, v_{j'})$ share a node. Then the algorithm **Cube-PRouting** can always find fault-free paths for (u_i, v_j) and $(u_{i'}, v_{j'})$, in which no nodes are occupied by other paths.*

A similar analysis shows that step 2.2 of the algorithm **Cube-PRouting** can always construct a new fault-free path without nodes occupied by other paths.

Let us summarize all these discussions in the following theorem.

Theorem 2. *If the n -cube network Q_n has at most $n - 2$ faulty nodes, then for each pair of non-faulty nodes u and v in Q_n , in time $O(n^2)$ the algorithm **Cube-PRouting** constructs $\min\{d_f(u), d_f(v)\}$ node-disjoint fault-free paths of length bounded by $\text{dist}(u, v) + 4$ from u to v .*

4 Conclusion

Network strong fault-tolerance is a natural extension of the study of network fault tolerance and network parallel routing. In particular, it studies the fault tolerance of large size networks with faulty nodes. In this paper, we have demonstrated that the popular interconnection networks, such as the hypercube networks and the star networks, are strongly fault-tolerant. We developed algorithms of running time $O(n^2)$ that for two given non-faulty nodes u and v in the networks, constructs the maximum number (i.e., $\min\{d_f(u), d_f(v)\}$) of node-disjoint fault-free paths from u to v such that the length of the paths is bounded by $\text{dist}(u, v) + 8$ for the star networks, and bounded by $\text{dist}(u, v) + 4$ for the hypercube networks. The time complexity of our algorithms is optimal since each path from u to v in the network S_n or Q_n may have length as large as $\Theta(n)$, and there can be as many as $\Theta(n)$ node-disjoint paths from u to v . Thus, even printing these paths should take time $O(n^2)$. We have shown that the length of the paths constructed by our algorithm for the star networks is almost optimal. For the n -cube network Q_n , the length of the paths constructed by our algorithm is bounded by $\text{dist}(u, v) + 4$. It is not difficult to see that this is the best possible, since there are node pairs u and v in Q_n with $n - 2$ faulty nodes, for which any group of $\min\{d_f(u), d_f(v)\}$ parallel paths from u to v contains at least one path of length at least $\text{dist}(u, v) + 4$.

We should mention that Rescigno [16] recently developed a randomized parallel routing algorithm on star networks with faults, based on the Information Disersal Algorithm (IDA) [15]. The algorithm in [16] is randomized thus it does not always guarantee the maximum number of node-disjoint paths. Moreover, in terms of the length of the constructed paths and running time of the algorithms, our algorithms seem also to have provided significant improvements.

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 Proc.*, (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. S. B. AKERS AND B. KRISHNAMURTHY, The fault tolerance of star graphs, *Proc. 2nd International Conference on Supercomputing*, (1987), pp. 270-276.

4. N. BAGHERZADEH, N. NASSIF, AND S. LATIFI, A routing and broadcasting scheme on faulty star graphs, *IEEE Trans. on Computers* 42, (1993), pp. 1398-1403.
5. G. BIRKHOFF AND S. MACLANE, *A Survey of Modern Algebra*, The Macmillan Company, New York, 1965.
6. C. C. CHEN AND J. CHEN, Optimal parallel routing in star networks, *IEEE Trans. on Computers* 46, (1997), pp. 1293-1303.
7. M.-S. CHEN AND K. G. SHIN, Adaptive fault-tolerant routing in hypercube multi-computers, *IEEE Trans. Computers* 39 (1990), pp. 1406-1416.
8. G. -M. CHIU AND S. -P. WU, A fault-tolerant routing strategy in hypercube multicomputers, *IEEE Trans. Computers* 45 (1996), pp. 143-154.
9. 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.
10. M. DIETZFELBINGER, S. MADHAVAPEDDY, AND I. H. SUDBOROUGH, Three disjoint path paradigms in star networks, *Proc. 3rd IEEE Symposium on Parallel and Distributed Processing*, (1991), pp. 400-406.
11. Q.-P. GU AND S. PENG, Fault tolerant routing in hypercubes and star graphs, *Parallel Processing Letters* 6, (1996), pp. 127-136.
12. T. C. LEE AND J. P. HAYES, Routing and broadcasting in faulty hypercube computers, *Proc. 3rd Conf. Hypercube Concurrent Computers and Applications* (1988), pp. 625-630.
13. E. OH AND J. CHEN, Strong fault-tolerance: parallel routing in star networks with faults *Tech. Report*, Dept. Computer Science, Texas A&M University (2001).
14. E. OH AND J. CHEN, Parallel routing in hypercube networks with faulty nodes, *Tech. Report*, Dept. Computer Science, Texas A&M University (2001).
15. M. O. RABIN, Efficient dispersal of information for security, load balancing, and fault tolerance, *Journal of ACM* 36, (1989), pp. 335-348.
16. A. A. RESCIGNO, Fault-tolerant parallel communication in the star network, *Parallel Processing Letters* 7, (1997), pp. 57-68.
17. A. A. RESCIGNO AND U. VACCARO, Highly fault-tolerant routing in the star and hypercube interconnection networks, *Parallel Processing Letters* 8, (1998), pp. 221-230.
18. Y. SAAD, M. H. SCHULTZ, Topological properties of hypercubes, *IEEE Transactions on Computers*, 37, (1988), pp. 867-872.