プログラミング-言語・基礎・実践- 21－21
(1995. 3. 10)

An Efficient Algorithm for k-Pairwise Node Disjoint Path Problem in Hypercubes [1]

Qian-Ping Gu and Shietung Peng

qian@u-aizu.ac.jp
Department of Computer Software
The University of Aizu
Aizu-Wakamatsu, Fukushima, 965-80 Japan

Abstract: In this paper, we give an efficient algorithm for the following k-pairwise node disjoint path problem in n-dimensional hypercubes H_n: Given $k = \lceil \frac{n}{2} \rceil$ pairs of $2k$ distinct nodes $(s_1, t_1), \ldots, (s_k, t_k)$ in H_n, $n \geq 4$, find k node disjoint paths, each path connecting one pair of nodes. Our algorithm finds the k node disjoint paths in $O(n^2 \log^* n)$ time which improves the previous result of $O(n^2 \log n)$. The length of the paths constructed in our algorithm is at most $n + \lceil \log n \rceil + 1$ which improves the previous result of $2n$ as well. The result of this paper shows that the k-pair-diameter $d^P_{\lceil \frac{n}{2} \rceil}(H_n)$ of H_n satisfies $d^P_{\lceil \frac{n}{2} \rceil}(H_n) \leq n + \lceil \log n \rceil + 1$.

Keywords: Node disjoint path, algorithm, interconnection network, time complexity

[1]This research was partially supported by the Founding of Group Research Projects at The University of Aizu.

key words

1 Introduction

Node disjoint path problems have attracted much attention in both mathematical terms and interconnection network studies due to its numerous applications in fault tolerant routing and so on [1, 9, 8, 12, 4]. In what follows, we will use *disjoint path* for node disjoint path. In this paper, we give an efficient algorithm for the following k-pairwise disjoint path problem in n-dimensional hypercubes H_n: Given $k = \lceil \frac{n}{2} \rceil$ pairs of $2k$ distinct nodes $(s_1, t_1), \ldots, (s_k, t_k)$ in H_n, find k node disjoint paths, each path connecting one pair of nodes. For $k \leq 2$, the k paths in the k-pairwise disjoint path problem can be found in Poly($|V|$) time for arbitrary graphs $G(V, E)$ [14]. However, for $k \geq 3$, the existence of the k paths in k-pairwise disjoint path problem is NP-complete [10]. A necessary condition of a graph G having k disjoint paths for the problem is that G is $(2k-1)$-connected [15]. In interconnection network studies, much work has been done on efficient algorithms for k-pairwise disjoint path problem in practical interconnection networks [11, 2, 6, 7].

Hypercubes are interesting interconnection topologies for parallel computation and communication networks. Many commercial and experimental multi-processor systems have been built based on hypercube interconnection topologies. Several disjoint path problems in hypercubes have been studied [13, 12, 11, 3, 7, 5]. For k-pairwise disjoint path problem, Madhavapeddy and Sudborough proved that $\lceil \frac{n}{2} \rceil$ disjoint paths exist in H_n for $n \geq 4$, which is n-connected and has 2^n nodes, and gave an algorithm that finds the $\lceil \frac{n}{2} \rceil$ disjoint paths in $O(n^3 \log n)$ time [11]. The length of the paths given by the algorithm in [11] is at most $2n$. Recently, Gu and Peng gave an $O(n^2 \log n)$ time algorithm which finds $\lceil \frac{n}{2} \rceil$ disjoint paths of length at most $2n$ for k-pairwise disjoint path problem in H_n for $n \geq 4$ [7]. In this paper, we propose a new algorithm for k-pairwise disjoint path problem in H_n. Our algorithm finds $\lceil \frac{n}{2} \rceil$ disjoint paths of length at most $n + \lceil \log n \rceil + 1$ for the problem in $O(n^2 \log^* n)$ time. Our results improves the previous ones both in time complexity of the algorithm and the length of the found paths.

In the next section, we give preliminaries of this paper. The algorithm for k-pairwise disjoint path problem is given in Section 3. Section 4 concludes the paper.

2 Preliminaries

An interconnection network is presented as an undirected graph $G(V, E)$, where nodes of G represent the processors, and edges of G represent the communication channels between processors. A path in a graph is a sequence of edges of the form $(s_1, s_2)(s_2, s_3) \ldots (s_{k-1}, s_k)$ $s_i \in V$, $1 \leq i \leq k$, and $s_i \neq s_j$, $i \neq j$. The length of a path is the number of edges in the path. We sometimes denote the path from s_1 to s_k by $s_1 \rightarrow s_k$. For a path $P = s_1 \rightarrow s_k$, we also use P to denote the set $\{s_1, \ldots, s_k\}$ of nodes that appear in path P. The nodes s_1 and s_k in path $s_1 \rightarrow s_k$ are called *end nodes.* Given two paths $P = \{s_1, s_2, \ldots, s_k\}$ and $Q = \{t_1, t_2, \ldots, t_l\}$, P and Q are disjoint if $P \bigcap Q = \emptyset$, and P and Q are weakly disjoint if $(P - \{s_1, s_k\}) \bigcap (Q - \{t_1, t_l\}) = \emptyset$, where s_1 and s_k are end nodes of P, and t_1 and t_l are end nodes of Q. For any two nodes $s, t \in G$, $d(s, t)$ denotes the distance between s and t, i.e., the length of the shortest path connecting s and t. The diameter of G is defined as $d(G) = \max\{d(s,t) | s, t \in G\}$.

An n-dimensional hypercube H_n is a graph, where the nodes of H_n are in $1-1$ correspondence with the n-bit binary sequences $a_1 a_2 \ldots a_n$, and two nodes $a_1 \ldots a_n$ and $b_1 \ldots b_n$ are connected by an edge if and only if these sequences differ in exactly one bit. There are 2^n nodes in H_n, and each node has exactly n edges incident upon it. H_n is n-connected and has diameter $d(H_n) = n$. H_n can be partitioned into two disjoint $(n-1)$-dimensional subcubes by fixing the kth bits of the binary expressions of nodes in H_n into 0 and 1, respectively, for some k with $1 \leq k \leq n$. The following topological properties of H_n are important in this paper.

Proposition 1 *For any node $s \in H_n$, $n \geq 1$, partition H_n into two disjoint $(n-1)$-dimensional subcubes $H_{n-1,1}$ and $H_{n-1,2}$ such that $s \in H_{n-1,1}$. Then there are n weakly disjoint paths of length at most 2 that connect s to n distinct nodes in $H_{n-1,2}$.*

Proof: Assume the kth bits of nodes in $H_{n-1,1}$ and $H_{n-1,2}$ are 0 and 1, respectively. Assume $s = a_1 a_2 \ldots a_n$ with $a_k = 0$. The path

$$P_k : s \rightarrow a_1 \ldots a_{k-1} \bar{a_k} a_{k+1} \ldots a_n \in H_{n-1,2}$$

is the path of length 1 that connects s to a node in $H_{n-1,2}$, where $\bar{a_k}$ is the logical negation of a_k. And the paths

$$\begin{aligned} P_j : s &\rightarrow a_1 \ldots \bar{a_j} \ldots a_k \ldots a_n \\ &\rightarrow a_1 \ldots \bar{a_j} \ldots \bar{a_k} \ldots a_n \in H_{n-1,2}, \end{aligned}$$

$1 \leq j \leq n$ and $j \neq k$, are $n-1$ paths of length 2 that connect s to $n-1$ distinct nodes in $H_{n-1,2}$. These n paths are node disjoint except at the common end node s. □

In what follows, we will say paths $P_1, \ldots, P_n$ route s into $H_{n-1,2}$.

Proposition 2 *For $s \in H_{n-1,1}$, let $P_1, \ldots, P_n$ be the n paths given in Proposition 1 that route s into $H_{n-1,2}$. Then*

1. *for any node $v \in H_n$, $v \neq s$, v can block at most one of the n paths $P_1, \ldots, P_n$;*

2. *for any path Q with length at most 2, $s \notin Q$, and $|Q \bigcap H_{n-1,2}| = 1$, Q can block at most one of the n paths $P_1, \ldots, P_n$;*

3. *for any path Q with length at most 2, $s \notin Q$, and $|Q \bigcap H_{n-1,1}| = 1$, Q can block at most one of the $n-1$ paths P_j, $1 \leq j \leq n$ and $j \neq k$, of length 2 (Q may block the path P_k of length 1 as well).*

Proof: The proof of (1) is trivial. For any nodes $x \in (P_i - \{s\})$ and $y \in (P_j - \{s\})$, $1 \le i \ne j \le n$ and $i, j \ne k$, if x, y are in the same subcube then $d(x, y) = 2$, otherwise $d(x, y) = 3$. Therefore, from the conditions of (2) and (3), (2) and (3) hold. □

The above propositions show that a node s can be routed into the opposite subcube by n different paths, and a routing path of length at most 2 for some other node can block at most 1 of the $n-1$ routing paths of length 2 for s. To simplify the descriptions of this paper, we introduce the concept of *fault cluster* which was defined in [7].

For a graph G, a cluster C of G is defined to be a connected subgraph of G. We will use C to express the cluster and the set of nodes in the cluster as well if no confusion arises. The number of nodes in C and the diameter of C are denoted as $|C|$ and $d(C)$, respectively. A cluster C is called *fault cluster* if all nodes in C are faulty. Let $\mathbf{F}$ be a set of fault clusters in a graph G. $|\mathbf{F}|$ denotes the cardinality of $\mathbf{F}$, $d(\mathbf{F}) = \max\{d(C) | C \in \mathbf{F}\}$ denotes the diameter of $\mathbf{F}$, and $F = \bigcup_{C \in \mathbf{F}} C$ denotes the set of nodes of the clusters in $\mathbf{F}$. From Proposition 2, a fault cluster C of diameter at most 1 in H_n can block at most one of the $n-1$ routing paths of length 2 for s (C may block the path P_k of length 1 as well). We call a path a *fault-free* path, if there is no fault-node in the path.

Finally, we introduce a parameter which is the optimal upper bound of the length of the paths in k-pairwise disjoint path problem. Let G be a n-connected graph, $L(P)$ be the length of a path P in G, and $\mathbf{P}$ be a set of paths in G. Define $L(\mathbf{P}) = \max\{L(P) | P \in \mathbf{P}\}$. For k-pairwise disjoint path problem, define

$$d_k^P((s_i, t_i)_{i=1}^k) = \min\{L(\mathbf{P}) | \mathbf{P} : \text{set of } k \text{ disjoint paths for } (s_i, t_i)_{i=1}^k\}.$$

The k-pair-diameter of G, $1 \le k \le \lceil \frac{n}{2} \rceil$, is defined as:

$$d_k^P(G) = \max\{s_k^P((s_i, t_i)_{i=1}^k) | s_i, t_i \in G\}.$$

Clearly, $d(G) \le d_k^P(G)$ for $1 \le k \le \lceil \frac{n}{2} \rceil$. The result of this paper shows that $d_{\lceil \frac{n}{2} \rceil}^P(H_n) \le d(H_n) + \lceil \log n \rceil + 1$.

3 Algorithm for k-Pairwise Disjoint Path Problem

Lemma 3 *Given any set F of $n-1$ fault nodes and any two non-fault nodes s and t in H_n, $n \ge 1$, a fault-free path of length at most $n+1$ connecting s and t can be constructed in $O(n)$ time.*

Proof: Partition H_n into two disjoint $(n-1)$-dimensional hypercubes $H_{n-1,1}$ and $H_{n-1,2}$ such that $s \in H_{n-1,1}$ and $t \in H_{n-1,2}$. Assume that $H_{n-1,2}$ contains at most $\frac{n-1}{2}$ fault nodes of F. Since there are at most $n-1$ fault nodes, from Proposition 2, we can find a fault-free path of length at most 2 from s to some node $s^{(1)} \in H_{n-1,2}$. Let $H_{n-1,2}$ be H_{n-1}, and repeat the above process, finding a fault-free path of length at most 2 from $s^{(i)}$ to $s^{(i+1)} \in H_{n-(i+1),2}$, until $s^{(i+1)} = t$ or $F \bigcap H_{n-(i+1),2} = \emptyset$ (Without loss of generality, we assume that when H_{n-i} is partitioned into $H_{n-(i+1),1}$ and $H_{n-(i+1),2}$, $s^{(i)} \in H_{n-(i+1),1}$, $t \in H_{n-(i+1),2}$, and $H_{n-(i+1),2}$ contains at most half of the fault nodes in H_{n-i} for all i). Since $|F \bigcap H_{n-(i+1),2}| \le |F \bigcap H_{n-i,2}|/2$ and $|F| \le n-1$, we can get a hypercube $H_{n-\lceil \log(n-1) \rceil}$ such that $F \bigcap H_{n-\lceil \log(n-1) \rceil} = \emptyset$ and $s^{(\lceil \log(n-1) \rceil)}$, denoted as s', and t are in $H_{n-\lceil \log(n-1) \rceil}$. Then s' and t can be connected by a fault-free path of length at most $d(s', t)$ in $H_{n-\lceil \log(n-1) \rceil}$ in $O(n)$ time.

For $s^{(i)} = a_1 a_2 \ldots a_n$ ($s^{(0)} = s$) and $t = b_1 b_2 \ldots b_n$, let $D_i = \{j | a_j \ne b_j\}$. If $s^{(i)} \to s^{(i+1)}$ is a path P_{j_i} for some $j_i \in D_i$ then $d(s^{(i+1)}, t) = d(s^{(i)}, t) - L(s^{(i)} \to s^{(i+1)})$, where $L(s^{(i)} \to s^{(i+1)})$ is the length of the path $s^{(i)} \to s^{(i+1)}$. Otherwise, $d(s^{(i)}, t) = d(s^{(i+1)}, t)$. From this, $d(s^{(i+1)}, t) = d(s^{(i)}, t) - L(P_{j_i})$ if there exists a fault-free path P_{j_i}, $j_i \in D_i$, and $d(s^{(i+1)}, t) = d(s^{(i)}, t)$ if for all $j_i \in D_i$, P_{j_i}'s are faulty. Since $|D_i| = d(s^{(i)}, t)$, a fault-free path P_{j_i}, $j_i \in D_i$, can be found if $|F \bigcap H_{n-i,2}| < |D_i|$. We claim that in the path $s \to s^{(1)} \to \cdots \to s' \to t$, at most $\lceil \log \frac{n}{d(s,t)} \rceil$ paths $s^{(i)} \to s^{(i+1)}$ are the paths P_{j_i} with $j_i \notin D_i$. To prove this, let j be the integer such that $s^{(j)} \to s^{(j+1)}$ is the path P_{j_j} with $j_j \notin D_j$. Assume that in the path $s \to s^{(1)} \to \cdots s^{(j)} \to s^{(j+1)}$, m, $1 \le m \le j+1$, paths $s^{(i)} \to s^{(i+1)}$ are the paths P_{j_i} with $j_i \notin D_i$. Then $|D_{j+1}| = d(s^{(j+1)}, t) = d(s, t) - 2(j+1-m)$ and $|F \bigcap H_{n-(j+1),2}| \le \frac{n-1}{2^{j+1}}$. From the algorithm, $d(s^{(j+1)}, t) = d(s^{(j)}, t) \ge 2$. Therefore, If $m = \lceil \log \frac{n}{d(s,t)} \rceil$ then

$$\begin{aligned} \frac{n-1}{2^{j+1}} &\le \frac{n-1}{2^{\log \frac{n}{d(s,t)}}} \times \frac{1}{2^{j+1-m}} \\ &= \frac{n-1}{n} \times \frac{d(s,t)}{2^{j+1-m}} \frac{d(s,t)}{2^{j+1-m}} \\ &\le d(s,t) - 2(j+1-m). \end{aligned}$$

From this, it is easy to get $|F \bigcap H_{n-i,2}| < |D_i|$ for all $i \ge j+1$, i.e, the path P_{j_i} with $j_i \in D_i$ can be found for $i \ge j+1$. This completes the proof of the claim.

From the claim and $L(P_{j_i}) \le 2$, we can find a fault-free path of length at most

$$\begin{aligned} d(s', t) &+ \sum_{i=1}^{\lceil \log(n-1) \rceil} L(s^{(i-1)} \to s^{(i)}) \\ &= d(s', t) + \sum_{j_i \in D_i} L(P_{j_i}) + \sum_{j_i \notin D_i} L(P_{j_i}) \\ &= d(s, t) + \sum_{j_i \notin D_i} L(P_{j_i}) \\ &\le d(s, t) + 2\lceil \log \frac{n}{d(s,t)} \rceil \le n+1. \end{aligned}$$

The time for the ith iteration is $O(\frac{n}{2^i})$ since there are no more than $\frac{n}{2^{i-2}}$ nodes needed to be explored at the ith iteration. Therefore, the total time of the construction is

$$T(n) = O(n) + O(\sum_{i=0}^{\log n} \frac{n}{2^i}) = O(n).$$

□

Lemma 4 *Given any set* $\mathbf{F}$ *of fault clusters with* $|\mathbf{F}| \leq n-2$ *and* $d(\mathbf{F}) \leq 1$, *and non-fault nodes* s *and* t *in* H_n, *a fault-free path of length at most* $n+2$ *that connects* s *and* t *can be found in* $O(n)$ *time.*

Proof: Partition H_n into two disjoint subcubes $H_{n-1,1}$ and $H_{n-1,2}$ with $s \in H_{n-1,1}$ and $t \in H_{n-1,2}$. Then one subcube contains at most $|F|/2 \leq n-2$ fault nodes of F. Assume that $H_{n-1,2}$ contains at most $n-2$ fault nodes. Then from Proposition 2, we can find a fault-free path of length at most 2 in $O(n)$ time that routes s into a node $s^{(1)}$ in $H_{n-1,2}$. From Lemma 3, $s^{(1)}$ and t can be connected by a fault-free path of length at most $(n-1)+1 = n$ in $O(n)$ time in $H_{n-1,2}$. □

Theorem 5 *For* $n \geq 4$ *and* $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$, *given a set* $\mathbf{F}$ *of fault clusters with* $|\mathbf{F}| \leq n-2k$ *and* $d(\mathbf{F}) \leq 1$, *and* k *pairs of distinct non-fault nodes* $(s_1,t_1),\ldots,(s_k,t_k)$ *in* H_n, *there are* k *disjoint paths of length at most* $n+\lceil \log k \rceil + 2$ *that each path connects one pair of nodes.*

Proof: The theorem is proved by induction on k. For $k=1$, the theorem holds from Lemma 4. Assume that the theorem holds for $k-1 \geq 1$. We prove the theorem for $k \leq \lfloor \frac{n}{2} \rfloor$. We first show that the k disjoint paths can be found and then analyze the length of the found paths.

Partition H_n into two disjoint subcubes $H_{n-1,1}$ and $H_{n-1,2}$ such that at least one node pair is separated. Without loss of generality, we assume k', $1 \leq k' \leq k$, node pairs are separated and $s_1,\ldots,s_{k'} \in H_{n-1,1}$ and $t_1,\ldots,t_{k'} \in H_{n-1,2}$. After the partition, we want to route s_i or t_i, $1 \leq i \leq k'$, into the opposite subcube by fault-free disjoint paths such that, after these initial routings, no node pair is separated and each subcube contains at most $k-1$ node pairs and at most $(n-1)-2(k-1)$ fault clusters of diameter at most 1.

For any separated node pair (s_i,t_i), $1 \leq i \leq k'$, we now show that there are two fault-free paths of length at most 2 that route s_i into $H_{n-1,2}$ and t_i into $H_{n-1,1}$, respectively. From Proposition 1, s_i can be routed into $H_{n-1,2}$ by n disjoint paths of length at most 2. The n paths for s_i may be blocked by the routing paths for the node pairs (s_j,t_j), $j \neq i$, and fault clusters in $\mathbf{F}$. From Proposition 2, one of the routing paths for s_j or t_j can block at most one path of length 2 for s_i. One cluster in $\mathbf{F}$ can block at most one path of length 2 for s_i as well. Therefore, at most $2(k-1)+n-2k = n-2$ paths of length 2 for s_i can be blocked. Thus, from Proposition 1, at least one routing path of length 2 that routes s_i into $H_{n-1,2}$ does not contain any node of F, any node of (s_j,t_j), $j \neq i$, or any node in the routing paths for (s_j,t_j). Similarly, t_i can be routed into $H_{n-1,1}$ by a fault-free path of length at most 2 as well.

From the argument above, the direction of routing separated node pairs can be controlled and the number of node pairs in each subcube after the routing can be balanced. Assume that for the rest $k-k'$ unseparated node pairs, k_1 pairs are in $H_{n-1,1}$ and k_2 pairs are in $H_{n-1,2}$, $0 \leq k_1,k_2 \leq k-k'$ and $k_1+k_2 = k-k'$. We further assume that $k_1 \geq k_2$. If $k_1 - k_2 \geq k'$ then we route $s_1,\ldots,s_{k'}$ to $H_{n-1,2}$ by fault-free disjoint paths of length at most 2. Otherwise, we route the nodes $s_1,\ldots,s_i$ into $H_{n-1,2}$ and the nodes $t_{i+1},\ldots,t_{k'}$ into $H_{n-1,1}$ such that, after the initial routing, each subcube contains half of the node pairs to be connected.

Assume that $s_1,\ldots,s_i$, $1 \leq i \leq k'$, are routed into $H_{n-1,2}$. Then there are at most $k-i \leq \lfloor \frac{n-1}{2} \rfloor$ node pairs to be connected in $H_{n-1,1}$. On the other hand, the routing paths for $s_1,\ldots,s_i$ should not be touched in the further routing in $H_{n-1,1}$, and thus, after the initial routing, i new fault clusters of diameter at most 1 are generated in $H_{n-1,1}$, resulting in at most $|\mathbf{F}|+i \leq n-2k+i \leq (n-1)-2(k-i)$ fault clusters in $H_{n-1,1}$. Similarly, in $H_{n-1,2}$, there are at most $k-j \leq \lfloor \frac{n-1}{2} \rfloor$, $j = k_1+k'-i \geq 1$, node pairs to be connected and at most $|\mathbf{F}|+k'-i = n-2k+k'-i \leq (n-1)-2(k-j)$ fault clusters. Therefore, by the induction hypothesis, the k disjoint paths that pairwisely connect the k node pairs in H_n can be found.

For a separated node pairs (s_i,t_i), we denote the destination node u at the routing path $s_i \to u$ for s_i by $s_i^{(1)}$ and denote the destination node of $s_i^{(g)}$ as $s_i^{(g+1)}$. From the routing strategy above, it is easy to see that there are at most $\lceil \frac{k}{2(g+h)} \rceil$ node pairs to be connected in the subcube where $s_i^{(g)}$ and $t_i^{(h)}$ reside in. Therefore, each node pair will be separated at most $\lceil \log k \rceil$ times. Thus, the length of the paths is at most

$$d(H_{n-\lceil \log k \rceil}) + 2 + 2\lceil \log k \rceil = n + \lceil \log k \rceil + 2$$

□

The above theorem implies that for even n, $k = n/2$ disjoint paths of length at most $n + \lceil \log n \rceil + 1$ can be found. To get the $\lceil n/2 \rceil$ disjoint paths for odd $n \geq 5$, some further works are needed.

Lemma 6 *For* $s \in H_{n-1,1}$, *let* P_k *be the routing path of length 1,* P_j, $1 \leq j \leq n$ *and* $j \neq k$, *be the* $n-1$ *routing paths of length 2 that route* s *into* $H_{n-1,2}$, *and for* $t \in H_{n-1,2}$ ($t \in H_{n-1,1}$), *let* Q_k *be the routing path of length 1,* Q_j, $1 \leq j \leq n$ *and* $j \neq k$, *be the* $n-1$ *routing paths of length 2 that route* t *into* $H_{n-1,1}$ *(route* t *into* $H_{n-1,2}$*). Then,*

1. *for* $t \in H_{n-1,2}$ *and* $d(s,t) \geq 2$, *at least* $2n-2$ *paths of* $P_1,\ldots,P_n,Q_1,\ldots,Q_n$ *are disjoint except at the end nodes* s *or* t*; and*
2. *for* $t \in H_{n-1,1}$ *and* $t \neq s$, *at least* $2n-2$ *paths of* $P_1,\ldots,P_n,Q_1,\ldots,Q_n$ *are disjoint except at the end nodes* s *or* t.

Proof: We only prove (1). (2) can be proved similarly. Assume $s = a_1a_2\ldots a_n$ and $b = b_1b_2\ldots b_n$ with $d(s,t) \geq 2$ and $a_k = \bar{b}_k$. If $d(s,t) \geq 4$ then obviously the $2n$ paths $P_1,\ldots,P_n$ and $Q_1,\ldots,Q_n$ are disjoint except at the end nodes s or t. For $d(s,t) = 2$, assume that $a_k = \bar{b}_k$, $a_j = \bar{b}_j$, and $a_i = b_i$, $1 \leq i \leq n$ and $i \neq j,k$. It is easy to see the path P_k meets the path Q_j, the path P_j meets the path Q_k, and none of any other pair of paths P_l and Q_m has the common node except the end nodes s or t. Thus, take $\mathbf{P} = \{P_i, Q_i, 1 \leq i \leq n,\ i \neq j\}$, the paths in $\mathbf{P}$ are disjoint except at end nodes s or t and $|\mathbf{P}| = 2n-2$.

For $d(s,t) = 3$, assume $a_k = \bar{b}_k$, $a_j = \bar{b}_j$, $a_l = \bar{b}_l$, and $a_i = b_i$, $1 \le i \le n$ and $i \ne j,k,l$. Then in the $2n$ paths $P_1, \ldots, P_n$ and $Q_1, \ldots, Q_n$, path P_l meets path Q_j and path P_j meets path Q_l, and none of any other pair of paths has the common node except the end nodes s or t. Thus, there are $2n-2$ paths of the above $2n$ paths are disjoint except at the end nodes s or t. □

Lemma 7 *Given a set* $\mathbf{F}$ *of fault clusters with* $|\mathbf{F}| \le n-3$, $d(\mathbf{F}) \le 1$, *and* $|F| \le 2n-7$, *and two pairs of non-fault distinct nodes* (s_1,t_1) *and* (s_2,t_2) *in* H_n *with* $n \ge 4$, $s_1, s_2 \in H_{n-1,1}$ *and* $t_1, t_2 \in H_{n-1,2}$, *we can find two fault-free disjoint paths of length at most 2, one path route* s_1 *or* s_2 *into* $H_{n-1,2}$ *and the other path route* t_2 *or* t_1 *into* $H_{n-1,1}$ *such that each subcube contains one node pair after the routing.*

Proof: If s_1 and t_2 can be routed into $H_{n-1,2}$ and $H_{n-1,1}$, respectively, by fault-free disjoint paths of length at most 2, then the Lemma holds. If all the n paths for one of the nodes s_1 or t_2 are blocked, then we will prove that s_2 and t_1 can be routed into $H_{n-1,2}$ and $H_{n-1,1}$, respectively. Assume all the n paths for s_1 are blocked. Then it must be the case that one cluster of diameter 1 blocks one path of length 1 and one path of length 2 and the other $n-2$ paths of length 2 are blocked by the rest $n-4$ fault clusters and the nodes s_2 and t_2. From this, $d(s_1,t_1) \ge 2$ and at least $n-4+2 = n-2$ fault nodes appear in the routing paths for s_1. From Lemma 6, there are at least $n-2$ routing paths of length at most 2 for t_1 that are disjoint with the routing paths for s_1. The nodes in the routing paths for s_1 can not appear in the $n-2$ routing paths for t_1. The number of fault nodes in fault clusters that do not appear in the routing paths for s_1 is at most $2n-7-(n-2) = n-5$. Therefore, at most $n-3$ routing paths for t_1 can be blocked (fault clusters block $n-5$ paths, s_2 blocks 1, and t_2 blocks 1). From this, t_1 can be routed into $H_{n-1,1}$ by a fault-free disjoint path of length at most 2. Similarly, there are at least $n-2$ routing paths of length at most 2 for s_2 that are disjoint with the routing paths for s_1. And at most $n-3$ routing paths for s_2 can be blocked (fault clusters block $n-5$ paths and the routing path for t_1 blocks 2). Thus, the lemma holds. □

Theorem 8 *For even* $n \ge 4$, *given a fault node* f *and* $k = n/2$ *pairs of non-fault distinct nodes* $(s_1,t_1), \ldots, (s_k,t_k)$ *in* H_n, *and for odd* $n \ge 5$, *given* $k = \lceil \frac{n}{2} \rceil$ *pairs of distinct nodes* $(s_1,t_1), \ldots, (s_k,t_k)$ *in* H_n, *there are* k *fault-free disjoint paths of length at most* $n + \lceil \log n \rceil + 1$ *that each path connects one pair of nodes.*

Proof: We prove the theorem for $n \ge 7$ here. For $4 \le n \le 6$, the theorem is proved by an enumerate argument (see Appendix). Similar to the proof of Theorem 5, we partition H_n into two disjoint subcubes $H_{n-1,1}$ and $H_{n-1,2}$ such that at least one node pair is separated. Assume that $s_1, \ldots, s_{k'} \in H_{n-1,1}$ and $t_1, \ldots, t_{k'} \in H_{n-1,2}$, $1 \le k' \le k$. Then we show that s_i or t_i, $1 \le i \le k'$, can be routed into the opposite subcube by fault-free disjoint paths such that, after the initial routing, no node pair is separated and each subcube contains at most $k-1$ node pairs. The proof is divided into two cases:

Case 1: $k' = 1$ and $s_i, t_i \in H_{n-1,1}$ for $2 \le i \le k$.

For odd n, since $H_{n-1,2}$ contains only t_1, s_1 can be routed into $H_{n-1,2}$ by the path of length 1. After the routing, $H_{n-1,2}$ contains only one node pair to be connected and the connection is trivial. $H_{n-1,1}$ contains $k-1$ node pairs and one fault node (node s_1). Since $n-1$ is even and $k-1 = (n-1)/2$, the theorem can be recursively applied to $H_{n-1,1}$.

For even n, if the fault node f is in $H_{n-1,2}$, then, from Propositions 1 and 2, s_1 can be routed into $H_{n-1,2}$ by a fault-free disjoint path of length at most 2, since at most $2(k-1)+1 = n-1$ routing paths for s_1 can be blocked by f, and the $2(k-1)$ nodes in node pairs (s_i,t_i), $2 \le i \le k$. After the routing, obviously, Theorem 5 can be applied to $H_{n-1,2}$. $H_{n-1,1}$ has $k-1$ node pairs to be connected and one fault cluster (the routing path for s_1) of diameter at most 1. Since $k-1 = \lfloor \frac{n-1}{2} \rfloor$ and $(n-1) - 2(k-1) = 1$, Theorem 5 can be applied to $H_{n-1,1}$. Assume that $f \in H_{n-1,1}$. Then s_1 can be routed into $H_{n-1,2}$ by the routing path of length 1. After the routing, the connection of the node pair in $H_{n-1,2}$ is trivial. $H_{n-1,1}$ has $k-1$ node pairs to be connected and two fault nodes (the given one and node s_1). Treating the two fault nodes as one node pair to be connected, there are k node pairs to be connected in $H_{n-1,1}$. Since $k = \lceil \frac{n-1}{2} \rceil$, the theorem can be recursively applied to $H_{n-1,1}$.

Case 2: $k' \ge 2$ or $k' = 1$ and $\exists(s_l,t_l) \in H_{n-1,1}$ and $\exists(s_m,t_m) \in H_{n-1,2}$.

In this case, we show that Theorem 5 can be applied to $H_{n-1,1}$ and $H_{n-1,2}$ after the first initial routing. Assume that for the rest $k-k'$ unseparated node pairs, k_1 pairs are in $H_{n-1,1}$ and k_2 pairs are in $H_{n-1,2}$, where $0 \le k_1, k_2 \le k-k'$ and $k_1 + k_2 = k - k'$. We further assume that $k_1 \ge k_2$. If $k_1 - k_2 \ge k'$ then we route $s_1, \ldots, s_{k'}$ to $H_{n-1,2}$ by fault-free disjoint paths of length at most 2 as follows: For odd n and arbitrary node s_i, $1 \le i \le k'$, at most $2(k-1) = n-1$ of the n routing paths can be blocked by the $2(k-1)$ nodes in node pairs (s_j,t_j), $1 \le j \le k$ and $j \ne i$ (Proposition 2 assures that the routing path for s_j, $1 \le j \le k'$ and $j \ne i$, can block at most one of the n routing paths of s_i). Similarly, for even n and s_i, at most $2(k-1)+1 = n-1$ of the n routing paths can be blocked by the $2(k-1)$ nodes in (s_j,t_j) and the fault node f. After the initial routing, $H_{n-1,2}$ contains $k'+k_2 \le \lfloor k/2 \rfloor$ node pairs and at most 1 fault node. Since $k'+k_2 \le \lfloor \frac{n-1}{2} \rfloor$ and $(n-1)-2(k'+k_2) \ge 1$, Theorem 5 can be applied to $H_{n-1,2}$. $H_{n-1,1}$ contains $k-(k_2+k')$ node pairs and at most k' fault clusters for odd n (k' routing paths for $s_1, \ldots, s_{k'}$) and at most $k'+1$ fault clusters (plus the fault node f) for even n of diameter at most 1. From the condition of Case 2, $k_2 + k' \ge 2$. Therefore, $k-(k_2+k') \le \lfloor \frac{n-1}{2} \rfloor$, $(n-1) - 2(k-(k_2+k')) \ge k'$ for odd n, and $(n-1) - 2(k-(k_2+k')) \ge k'+1$ for even n. Thus, Theorem 5 can be applied to $H_{n-1,1}$ as well.

Assume that $k_1 - k_2 < k'$. In this case, we route l_1 s_i's and l_2 t_j's in the separated node pairs into $H_{n-1,2}$ and $H_{n-1,1}$, respectively, such that each subcube contains half of the node pairs to be connected. The routing is done as follows: We first pair up the separated node pairs into l_2 groups such that each group has two separated node pairs, one is to be routed into $H_{n-1,2}$ and the other is to be routed into $H_{n-1,1}$. Next we route the node s_i's in the rest $k'-2l_2$ separated node pairs into $H_{n-1,2}$ as we did in the case $k_1-k_2 \geq k'$. Finally, we route the two node pairs in each group into $H_{n-1,1}$ and $H_{n-1,2}$, respectively, as shown in Lemma 7. While routing the two node pairs (s_1,t_1) and (s_2,t_2) in some group, there are at most $k-2$ routing paths for other node pairs (s_i,t_i), say for s_i's, $k-2$ nodes t_i's, and one fault node f if n is even, which may block the routing paths for (s_1,t_1) or (s_2,t_2). This results in $2(k-2)=n-3$ fault clusters for odd n or results in $2(k-2)+1=n-3$ fault clusters for even n, total at most $2(k-2)+(k-2)+1 \leq 2n-7$ fault nodes for either cases. After the initial routing, $H_{n-1,1}$ contains $k_1+l_2=\lfloor k/2 \rfloor$ node pairs and at most l_1 fault clusters for odd n and at most l_1+1 fault clusters for even n, and $H_{n-1,2}$ contains $k_2+l_1=\lceil k/2 \rceil$ node pairs and at most l_2 fault clusters for odd n and at most l_2+1 fault clusters for even n. From $n \geq 7$ and $k=\lceil n/2 \rceil$, $k_1+l_2=k-(k_2+l_1) \leq \lfloor \frac{n-1}{2} \rfloor$. On the other hand, for odd $n \geq 7$ and $k=\lceil n/2 \rceil$, $(n-1)-2(k_1+l_2)=(n-1)-2\lfloor k/2 \rfloor \geq \lceil k/2 \rceil \geq l_1$, and for even $n \geq 8$ and $k=n/2$, $(n-1)-2(k_1+l_2)=(n-1)-2\lfloor k/2 \rfloor \geq \lceil k/2 \rceil+1 \geq l_1+1$. Thus, Theorem 5 can be applied to $H_{n-1,1}$. Similarly, Theorem 5 can be applied to $H_{n-1,2}$.

We have shown that the k disjoint paths can be found. If the separated node pair is routed in Case 1 then the path for this pair is at most $n+1$. If the separated node pairs are routed in Case 2, following a similar argument in Theorem 5, the length of the paths for those node pairs is at most $n+\lceil \log k \rceil+2 \leq n+\lceil \log n \rceil+1$. □

Theorems 5 and 8 implies an algorithm for k-pairwise disjoint path problem in H_n. We now show the time complexity of the algorithm.

Lemma 9 *For a node $u \in H_n$, let $N(u)=\{v|v \in H_n,\ d(u,v) \leq 1\}$. For any set of nodes $X \subseteq H_n$ define $f(x)=|\{y|x \in N(y), y \in X\}|$, and $F(X)=\sum_{x\in X} f(x)$. Then $F(X) \leq |X|\log|X|$.*

Proof: The lemma is proved by induction on $|X|$. It is trivial to show that the lemma holds for $|X| \leq 4$. Now we assume that the lemma holds for $|X| \leq m-1$, and prove the case of $|X|=m$. We partition H_n into two subcubes $H_{n-1,1}$ and $H_{n-1,2}$ such that $H_{n-1,1}$ and $H_{n-1,2}$ contain l and $m-l$ nodes of X, respectively, with $1 \leq l \leq m-1$. From the induction hypothesis we have $F(X \bigcap H_{n-1,1}) \leq l\log l$ and $F(X \bigcap H_{n-1,2}) \leq (m-l)\log(m-l)$. It is clearly that each node of $H_{n-1,1}$ is a neighbor node of exactly one node in $H_{n-1,2}$ and vice versa. Therefore, we have

$$F(X) \leq l\log l+(m-l)\log(m-l)+2\min\{l,m-l\}.$$

It is not difficult to show that

$$l\log l+(m-l)\log(m-l)+2\min\{l,m-l\} \leq m\log m$$

for $1 \leq l \leq m-1$. Thus, the lemma holds. □

Lemma 10 *Let $X,Y \subseteq H_n$ with $|X| \geq |Y|$. For $x \in X$, define $f(x,Y)=|\{y|x \in N(y), y \in Y\}|$ and $F(X,Y)=\sum_{x\in X} f(x,Y)$. Then $F(X,Y)=O(|X|\log|Y|)$.*

Proof: Partition X into $r=|X|/|Y|$ disjoint subsets $X_1,\ldots,X_r$ with the same size. Then from Lemma 9,

$$\begin{aligned} F(X,Y) &= \sum_{i=1}^{r} F(X_i,Y) \\ &\leq \sum_{i=1}^{r} F(X_i \bigcup Y)=O(|X|\log|Y|). \end{aligned}$$

□

Theorem 11 *Given $k=\lceil \frac{n}{2} \rceil$ pairs of distinct nodes $(s_1,t_1),\ldots,(s_k,t_k)$ in H_n, k disjoint paths of length at most $n+\lceil \log n \rceil+1$, each path connecting one pair of nodes, can be found in $O(n^2\log^* n)$ time.*

Proof: Theorems 5 and 8 implies an algorithm that finds the k disjoint paths of length at most $n+\lceil \log n \rceil+1$ for k-pairwsie disjoint path problem. We now analyze the time complexity of the algorithm. There are two basic works in the algorithm. One is to connect a node pair in a subcube which can be done in $O(n)$ time (Lemma 4), and the other is to partition H_n into two disjoint subcubes and to find the routing paths for the k' separated node pairs, after the partition. Each partition takes $O(n)$ time. In each partition, for a separated node pair (s_i,t_i), if m_i of the $2n$ routing paths are blocked then it takes $O(m_i)$ time to find the routing path for (s_i,t_i), and it takes $O(\sum_{i=1}^{k'} m_i)$ time to complete the initial routing for all the separated node pairs. Let Y be the set of nodes in separated node pairs and X be the set of nodes which may block the routing paths needed. Then $f(x,Y)$ defined in Lemma 10 gives an upper bound on the number of the routing paths for separated node pairs that are blocked by the node $x \in X$. Therefore, from Lemma 10,

$$\sum_{i=1}^{k'} m_i \leq F(X,Y)=O(|X|\log|Y|).$$

From this, $|X|=O(n)$, and $|Y|=O(k')$, It takes $O(n\log k')$ time to find the routing paths for the k' separated node pairs in each partition. We claim that there are at most $O(n/k')$ partitions that generate at least k' separated node pairs in the algorithm. To prove this, we look at the *strictly binary* partition tree T_P of H_n given by the algorithm. [1] The root of T_P is H_n with k node pairs. Each tree-node denotes a subcube with certain node pairs to be connected in it. The two children of a non-leaf tree-node H denote subcubes obtained by the partition and initial routing in H. For any tree-node with at least k' node pairs separated in the partition of the tree-node, from the initial routing strategy of the algorithm, each of its two children contains at least $k'/2$ node pairs. Now, we delete the tree-nodes with less than $k'/2$ node pairs from the partition tree T_P. In the resulting subtree of T_P,

[1] A strictly binary tree is a binary tree that each non-leaf tree-node has two non-empty children.

if we delete every tree-node of outdegree 1 and connect its child to its father then we can get a strictly binary tree T_R such that each tree-node in T_P with at least k' node pairs separated appears in T_R (the tree-node with outdegree 1 has less than k' node pairs separated, since one of its children has less than $k'/2$ node pairs). Since there are at most $k = \lceil n/2 \rceil$ node pairs in the leaves of T_R, T_R has $O(n/k')$ tree-nodes. From this, our claim holds.

Let $n_0 = n$ and $n_i = \log n_{i-1}$, $i \geq 1$. Then the total time used in the partition and initial routing that involve $n_i \leq k' < n_{i-1}$ separated node pairs is $O(n/n_i \times n \log n_{i-1}) = O(n^2)$. Let j be the minimum number such that $n_j \leq 2$. Then $j = \log^* n$ and the total time of the algorithm is $\sum_{i=1}^{j} O(n^2) = O(n^2 \log^* n)$. □

4 Conclusional Remarks

In this paper, we an efficient algorithm for the k-pairwise disjoint path problem in n-dimensional hypercubes H_n. Our algorithm finds the k node disjoint paths in $O(n^2 \log^* n)$ time which improves the previous result of $O(n^2 \log n)$. A trivial time lower bound on the k-pairwise disjoint path problem in H_n is $O(n^2)$. The length of the paths constructed in our algorithm is at most $n + \lceil \log n \rceil + 1$ which improves the previous result of $2n$ as well. The result of this paper shows that the k-pair-diameter $d^P_{\lceil \frac{n}{2} \rceil}(H_n)$ of H_n satisfies $d^P_{\lceil \frac{n}{2} \rceil}(H_n) \leq n + \lceil \log n \rceil + 1$. A trivial lower bound on $d^P_{\lceil \frac{n}{2} \rceil}(H_n)$ is $d(H_n) + 1 = n + 1$. Finding better upper bounds on $d^P_{\lceil \frac{n}{2} \rceil}(H_n)$ are worth further research attention.

References

[1] J. C. Bermond. Interconnection networks. *Discrete Math.*, 1992.

[2] M. Dietzfelbinger, S. Madhavapeddy, and I. H. Sudborough. Three disjoint path paradigms in star networks. In *Proc. IEEE Symposium on Parallel and Distributed Processing*, pages 400–406, 1991.

[3] Q. Gu, S. Okawa, and S. Peng. Disjoint paths in hypercubes and star graphs. In *Proc. Joint Symposium on Parallel Processing*, pages 49–56, 1994.

[4] Q. Gu and S. Peng. Efficient algorithms for disjoint paths in star graphs. Technical Report 93-1-015, Univ. of Aizu, Japan, 1993. To appear in the 6th International Transputer/Occam Conference.

[5] Q. Gu and S. Peng. Advanced fault tolerant routing in hypercubes. In *Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'94)*, pages 189–196, 1994.

[6] Q. Gu and S. Peng. Efficient algorithms for disjoint paths in star graphs. In *Proc. of the 6th Transputer/Occam International Conference*, pages 53–65, 1994.

[7] Q. Gu and S. Peng. k-pairwise cluster fault tolerant routing in hypercubes. In *Proc. of the 5th International Symposium on Algorithms and Computation ISAAC'94*, pages 345–353, 1994.

[8] D. F. Hsu. Interconnection networks and algorithms. *Networks*, 1993.

[9] D. F. Hsu. On container with width and length in graphs, groups, and networks. *IEICE Trans. on Fundamental of Electronics, Information, and Computer Sciences*, (4):668–680, 1994.

[10] R. M. Karp. On the computational complexity of combinatorial problems. *Networks*, pages 45–68.

[11] S. Madhavapeddy and I. H. Sudborough. A topological property of hypercubes: Node disjoint paths. In *Proc. 2nd IEEE Symp. on Parallel and Distributed Processing*, pages 532–539, 1990.

[12] M. A. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. ACM*, pages 335–348, 1989.

[13] Y. Saad and M. H. Shultz. Topological properties of hypercubes. *IEEE Trans. on Computers*, pages 867–872, 1988.

[14] Y. Shiloach. The two paths problem is polynomial. Technical Report STAN-CS-78-654, Stanford University.

[15] M. Watkin. On the existence of certain disjoint arcs in graphs. *Duke Math, Journal*, 1968.

Appendix

Lemma 12 *Given a fault cluster of diameter at most 1 and a fault node, and two distinct non-fault nodes s and t in a 3-dimensional hypercube H_3, a fault-free path of length at most 4 that connects s and t can be found in $O(1)$ time.*

Proof: Trivial. □

Lemma 13 *Given a fault cluster C of diameter at most 1 and two pairs of distinct non-fault nodes (s_1, t_1) and (s_2, t_2) in an H_4, 2 fault-free disjoint paths of length at most 6, each connecting a pair of nodes, can be found in $O(1)$ time.*

Proof: Partition H_4 into two disjoint subcubes $H_{3,1}$ and $H_{3,2}$ such that if $d(C) = 1$ (i.e., $C = (f_1, f_2)$) then $f_1 \in H_{3,1}$ and $f_2 \in H_{3,2}$ and if C is a single fault node f then $f \in H_{3,1}$.

Case 1: No node pair is separated.

Assume that $s_1, t_1, s_2, t_2 \in H_{3,1}$. Connect s_1 and t_1 by a fault-free path in $H_{3,1}$. If the path can not be found then it must be the case that f_1 (or f) is the neighbor of s_1 (or t_1). Viewing the nodes s_1 and f_1 as a fault cluster of diameter 1 and t_1 as a fault node, from Lemma 12, s_2 and t_2 can be connected by a fault-free path of length at most 4 in $H_{3,1}$. Obviously, s_1 and t_1 can be routed into s'_1 and t'_1 in $H_{3,2}$, respectively, by fault-free disjoint paths of length 1 and then s'_1 and t'_1 can be connected in $H_{3,2}$ by a fault-free path of length at most 3. The case that $d(C) = 1$ and $s_1, t_1, s_2, t_2 \in H_{3,2}$ can be proved similarly. Assume that $f \in H_{3,1}$ and $s_1, t_1, s_2, t_2 \in H_{3,2}$. Since $H_{3,1}$ has only one fault node, the nodes s_1 and t_1 can be routed into $H_{3,1}$ by two fault-free paths, one of length 1 the other of length at most 2. After the routing, $H_{3,2}$ has one node pair, one fault cluster of diameter at most 1, and one fault node. Then the two node pairs can be connected in $H_{3,1}$ and $H_{3,2}$ by fault-free paths of length at most 3 and 4, respectively.

Case 2: One node pair is separated.

Assume that $s_1 \in H_{3,1}$ and $t_1, s_2, t_2 \in H_{3,2}$. Obviously, t_1 can be routed into $H_{3,1}$ by a fault-free path of length at most 2. Thus, from Lemma 12, the two node pairs can be connected in $H_{3,1}$ and $H_{3,2}$, respectively.

Case 3: Two node pairs are separated.

Assume $s_1, s_2 \in H_{3,1}$ and $t_1, t_2 \in H_{3,2}$. Obviously, t_1 and s_2 can be routed into $H_{3,1}$ and $H_{3,2}$ by fault-free disjoint paths of length at most 2. After the routing, each subcube has one node pair, one fault cluster of diameter at most 1, and one fault node. Thus, from Lemma 12, the connection can be done.

Obviously, it takes $O(1)$ time to find the paths and the length of the found paths is at most 6. □

Lemma 14 *Give three pairs of distinct non-fault nodes (s_1, t_1), (s_2, t_2), and (s_3, t_3) in an H_5, 3 fault-free disjoint paths of length at most 8, each connecting a pair of nodes, can be constructed in $O(1)$ time.*

Proof: Partition H_5 into subcubes $H_{4,1}$ and $H_{4,2}$ such that at least one node pair is separated. Assume $s_1 \in H_{4,1}$ and $t_1, s_2, t_2, s_3, t_3 \in H_{4,2}$. Then route t_1 into $H_{4,1}$ by a fault-free path of length 1. Assume $s_1, s_2 \in H_{4,1}$ and $t_1, t_2, s_3, t_3 \in H_{4,2}$. Then route t_1 and t_2 into $H_{4,1}$ by fault-free disjoint paths of length at most 2. Assume $s_1, s_2, s_3 \in H_{4,1}$ and $t_1, t_2, t_3 \in H_{4,2}$. First, route t_1 into $H_{4,1}$ and then route t_2 and s_3 (or t_3 and s_2) into $H_{4,1}$ and $H_{4,2}$, respectively, by Lemma 7 (treat the routing path for t_1 as a fault cluster of diameter 1 and s_1 as a fault node). It is easy to see that Lemma 13 can be applied to both $H_{4,1}$ and $H_{4,2}$ after the routing. □

Lemma 15 *Give a fault node f and three pairs of distinct non-fault nodes (s_1, t_1), (s_2, t_2), and (s_3, t_3) in an H_6, 3 fault-free disjoint paths of length at most 10, each connecting a pair of nodes, can be constructed in $O(1)$ time.*

Proof: Partition H_6 into two subcubes $H_{5,1}$ and $H_{5,2}$. Assume $f \in H_{5,1}$. Then, we route all the nodes to be connected that reside in $H_{5,1}$ into $H_{5,2}$ by disjoint, fault-free paths of length at most 2 and connect them pairwisely in $H_{5,2}$ by disjoint fault-free paths of length at most 8 using Lemma 14. □