An Efficient Construction of Node Disjoint Paths in OTIS Networks

Weidong Chen[1,2], Wenjun Xiao[1], and Behrooz Parhami[3]

[1] Department of Computer Science, South China University of Technology, Guangzhou 510641, China
[2] Department of Computer Science, South China Normal University, Guangzhou 510631, China
[3] Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA
chwd2007@hotmail.com

Abstract. We investigate the problem of constructing the maximal number of node disjoint paths between two distinct nodes in Swapped/OTIS networks. A general construction of node disjoint paths in any OTIS network with a connected basis network is presented, which is independent of any construction of node disjoint paths in its basis network. This general construction is effective and efficient, which can obtain desirable node disjoint paths of length at most D+4 in $O(\Delta^2+\Delta f(N^{1/2}))$ time if the basis network of size n has a shortest routing algorithm of time complexity $O(f(n))$, where D, Δ and N are, respectively, the diameter, the degree and the size of the OTIS network. Further, for OTIS networks with maximally fault tolerant basis networks, we give an improved version of a conventional construction of node disjoint paths by incorporating the above general construction. Finally, we show the effectiveness and efficiency of these constructions applied to OTIS-Hypercubes.

1 Introduction

Optical transpose interconnection system (OTIS) networks are interesting interconnection networks for parallel computation and communication. An OTIS network with n^2 nodes is a two-level swapped architecture built of n copies of an n-node basis network that constitute its clusters. A simple rule for intercluster connectivity (node j in cluster i connected to node i in cluster j, for all $i \neq j$) leads to regularity, modularity, packageability, fault tolerance, and algorithmic efficiency of the resulting networks. The OTIS architecture has received considerable attention in recent years and has a special place among real-world architectures for parallel and distributed systems[1,11]. A number of algorithms have been developed for routing, selection/sorting[8,10], numerical analysis[5], matrix multiplication[14], and image processing[13].

Finding node disjoint paths or parallel paths in interconnection networks is one of the fundamental issues in design and implementation of parallel and distributed

M. Xu et al. (Eds.): APPT 2007, LNCS 4847, pp. 180–189, 2007.

computing systems[4,16]. Parallel paths are useful in speeding up the transfer of large amounts of data between nodes and in providing alternative routes in cases of node or link failures [6]. From Menger's Theorem [15], there exist at least k parallel paths between any two distinct nodes in a network of connectivity k. In a general network, it is non-trivial to identify the parallel paths guaranteed by a given level of connectivity. For levels of connectivity greater than two, the identification of parallel paths is generally done using maximum flow algorithms which take $O(N^3)$ time, where N is the size of the network [16]. However, for the interconnection networks with special structures such as Hypercube networks, OTIS networks, and so on, flow techniques taking $O(N^3)$ time may be far from efficient.

Although some studies are related to general properties, including fault tolerance, of OTIS networks [3,9,17,18], so far all research work in this direction is only confined to OTIS networks with basis networks being maximally fault tolerant, and those proposed constructions of parallel paths in these OTIS networks are closely dependent upon the corresponding constructions in their basis networks [2,3,9].

In this paper, in a more general sense, we investigate the construction of the maximal number of parallel paths between two distinct nodes in any OTIS network whose basis network is connected. We propose an effective and efficient general construction of parallel paths in the OTIS network, which is independent of any construction of parallel paths in its basis network. This general construction can obtain desirable parallel paths of length at most D+4 in $O(\Delta^2+\Delta f(N^{1/2}))$ time if the basis network of size n has a shortest routing algorithm of time complexity $O(f(n))$, where D, Δ and N are, respectively, the diameter, the maximal node degree and the size of the OTIS network. Further, in the special case of a maximally fault tolerant basis network, we make an improvement over a conventional construction of parallel paths in such an OTIS network. Finally, the effectiveness and efficiency of these construction algorithms applied to OTIS-Hypercube are shown.

In the next section we describe OTIS networks. Section 3 presents the general construction of parallel paths in an OTIS network with a connected basis network. Section 4 gives the improved version of the conventional construction of parallel paths in those OTIS networks whose basis networks possess the maximally fault tolerant property. The application of these construction algorithms to OTIS-Hypercubes is discussed in Section 5. The conclusion is made in Section 6.

2 Preliminaries

Let G be a simple undirected graph (graph, for short) with vertex (node) set $V(G)$ and edge (link) set $E(G)$. For $v \in V(G)$, we denote by $deg_G(v)$ the degree of v in G, by $N_G(v) = \{u \in V \mid (v, u) \in E(G)\}$ the open neighborhood of v, and by $N_G[v] = N_G(v) \cup \{v\}$ its closed neighborhood. The maximum degree among the vertices of G is denoted by $\Delta(G)$ and the minimum degree by $\delta(G)$. The distance of between two nodes u and v, denoted by $d_G(u,v)$, is the length of a shortest path between u and v. The diameter D(G) of G is the maximal distance between any two nodes of G. Two paths from u to v are node disjoint (also called parallel paths) if they have no common internal node. The connectivity of G is the minimal number of nodes in G whose removal can cause G disconnected or trivial. A graph G of connectivity $\delta(G)$ is maximally fault tolerant. Other notation and terminology used in this paper follow those in [15]. In the remainder of this paper, we use the terms graph and network interchangeably.

Definition 1. *OTIS (Swapped) network* [7,17]: *The OTIS (swapped) network* OTIS-Ω, *derived from the graph* Ω, *is a graph with vertex set* $V(\text{OTIS-}\Omega) = \{\langle g, p\rangle \mid g, p \in V(G)\}$ *and edge set* $E(\text{OTIS-}\Omega) = \{(\langle g, p_1\rangle, \langle g, p_2\rangle) \mid g \in V(G), (p_1, p_2) \in E(G)\} \cup \{(\langle g, p\rangle, \langle p, g\rangle) \mid g, p \in V(G) \text{ and } g \neq p\}$.

In OTIS-Ω, the graph Ω is called the basis (factor) graph or network. We refer to g as the cluster address of node $\langle g, p\rangle$ and p as its processor address. In an OTIS network, an intercluster (optical) link connects processor p of cluster g to processor g of cluster p for all $p \neq g$. No intercluster link is incident to processor g of cluster g. An example of OTIS networks is shown in Fig. 1.

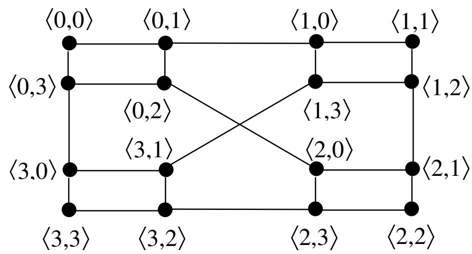

Fig. 1. An OTIS network with the basis graph C_4, a cycle of size 4

The following basis topological metrics of OTIS-Ω as functions of the corresponding metrics of Ω are derived from Definition 1 and similar expressions in [3,9]:

- $N=n^2$, where $N=|V(\text{OTIS-}\Omega)|$, $n=|V(\Omega)|$.
- $deg_{\text{OTIS-}\Omega}(\langle g, g\rangle)= deg_{\Omega}(g)$, and $deg_{\text{OTIS-}\Omega}(\langle g, p\rangle)=deg_{\Omega}(p)+1$ for $g\neq p$.
- $d_{\text{OTIS-}\Omega}(\langle g, p_1\rangle,\langle g, p_2\rangle)=d_{\Omega}(p_1, p_2)$, and for $g_1\neq g_2$, $d_{\text{OTIS-}\Omega}(\langle g_1, p_1\rangle,\langle g_2, p_2\rangle)=min\{d_{\Omega}(p_1, g_2)+d_{\Omega}(g_1, p_2)+1, d_{\Omega}(p_1, p_2)+d_{\Omega}(g_1, g_2)+2\}$.
- $\Delta(\text{OTIS-}\Omega)=\Delta(\Omega)+1$, and $\delta(\text{OTIS-}\Omega)=\delta(\Omega)$.
- $D(\text{OTIS-}\Omega)= 2D(\Omega)+1$.

The following results on parallel paths of an OTIS network have been given in [3].

Theorem 1. (*Day and Al-Ayyoub* [3]). *Let the graph Ω be connected.*

(1) *If* $g_1 \neq g_2$ *and* $deg_{\Omega}(p) = d$, *then there are d parallel paths between nodes* $\langle g_1, p\rangle$ *and* $\langle g_2, p\rangle$ *in* OTIS-Ω.

(2) *If* $\langle g_1, p_1\rangle$ *and* $\langle g_2, p_2\rangle$ *are two nodes in* OTIS-Ω *such that* $p_1 \neq p_2$ *and such that there are d parallel paths between* p_1 *and* p_2 *in* Ω, *then there are d parallel paths between* $\langle g_1, p_1\rangle$ *and* $\langle g_2, p_2\rangle$ *in* OTIS-Ω.

3 Constructing Parallel Paths in OTIS Networks with Connected Basis Graphs

In the section, we give an effective and efficient general algorithm for constructing parallel paths between two distinct nodes $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ in an OTIS network with a connected basis graph Ω.

3.1 Basis Idea

We first notice the following basis fact, which is easily derived from the rule for intercluster connectivity in OTIS networks: In cluster g_1 (g_2, respectively), every node of $N_\Omega[p_1]$ ($N_\Omega[p_2]$, respectively) is linked to one different cluster by an optical link if the node is not $\langle g_1, g_1\rangle$ ($\langle g_2, g_2\rangle$, respectively). Based on this fact, we construct parallel paths between $src=\langle g_1, p_1\rangle$ and $dst=\langle g_2, p_2\rangle$ as follows. Each of these paths begins with the source node *src*, immediately leaves cluster g_1 from a neighbor of *src* in cluster g_1 along an optical link, and then goes through successively at most two mediate clusters, until finally enters cluster g_2 at a neighbor of *dst* in cluster g_2 along an optical link prior to arriving at the destination node *dst*. If these mediate clusters are selected properly so that each mediate cluster can be passed through only by one of these paths, we will obtain desired parallel paths. See Fig. 2.

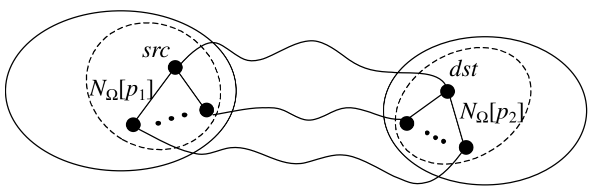

Fig. 2. An illustration of constructing parallel paths between *src* and *dst* in OTIS-Ω with Ω being connected for the case of $g_1 \neq g_2$ and $g_1 \notin N_\Omega[p_1]$ and $g_2 \notin N_\Omega[p_2]$

3.2 Algorithm

For the convenience of describing the algorithm, we need introduce some additional notations. We denote by $Path_\Omega(p, q)$ a shortest path from p to q in Ω, and by $\langle g, Path_\Omega(p, q)\rangle$ a shortest path from $\langle g, p\rangle$ to $\langle g, q\rangle$ in OTIS-Ω that is completely contained in cluster g. Let Y and Z be two disjoint subsets of $V(\Omega)$. A match M from Y to Z is a binary relation from Y to Z such that $|M|=min\{|Y|, |Z|\}$, and such that $(y, z)\neq(y', z')$ if and only if both $y\neq y'$ and $z\neq z'$ for all $(y, z), (y', z')\in M$. Obviously, a match M from Y to Z can be constructed in $O(\Delta^2(\Omega))$ time if Ω is represented by adjacency lists. In addition, we assume that a shortest routing algorithm of time complexity $O(f(n))$ in Ω is given, where n is the size of Ω.

Algorithm 1
Case I (g_1=g_2=g):
Step 1.1: Construct a path as follows based on a shortest path from p_1 to p_2 in Ω: $\langle g, \text{Path}_\Omega(p_1, p_2)\rangle$, where $\text{Path}_\Omega(p_1, p_2)=p_1\rightarrow\text{Path}_\Omega(y_0, z_0)\rightarrow p_2$ for some $y_0 \in N_\Omega[p_1]$ and some $z_0 \in N_\Omega[p_2]$.
Step 1.2: Let $S_0=N_\Omega[p_1]\cap N_\Omega[p_2]-\{y_0, z_0\}$. For every $x\in S_0$, construct a path as follows: $\langle g, p_1\rangle\rightarrow\langle g, x\rangle\rightarrow\langle g, p_2\rangle$.
Step 1.3: Let $S_1=N_\Omega[p_1]-S_0-\{y_0, g\}$, $S_2= N_\Omega[p_2]-S_0-\{z_0, g\}$. Construct a match M between S_1 and S_2. Then, for each $(y, z)\in M$, construct a path as follows:
$\langle g, p_1\rangle\rightarrow\langle g, y\rangle\rightarrow\langle y, \text{Path}_\Omega(g, z)\rangle\rightarrow \langle z, \text{Path}_\Omega(y, g)\rangle \rightarrow\langle g, z\rangle\rightarrow\langle g, p_2\rangle$.
Case II ($g_1\neq g_2$):
Step 2.1: Construct a path as follows based on a shortest path from p_1 to g_2 and a shortest path from g_1 to p_2 in Ω: $\langle g_1,\text{Path}_\Omega(p_1, g_2)\rangle\rightarrow\langle g_2, \text{Path}_\Omega(g_1, p_2)\rangle$, where $\text{Path}_\Omega(p_1, g_2)=p_1\rightarrow\text{Path}_\Omega(y_0, g_2)$ for some $y_0 \in N_\Omega[p_1]$ and $\text{Path}_\Omega(g_1, p_2)=\text{Path}_\Omega(g_1, z_0)\rightarrow p_2$ for some $z_0\in N_\Omega[p_2]$.
Step 2.2: Let $S_0=N_\Omega[p_1]\cap N_\Omega[p_2]-\{g_1,g_2,y_0,z_0\}$. For every $x\in S_0$, construct a path as follows: $\langle g_1, p_1\rangle\rightarrow\langle g_1, x\rangle\rightarrow\langle x, \text{Path}_\Omega(g_1, g_2\rangle \rightarrow\langle g_2, x\rangle\rightarrow\langle g_2, p_2\rangle$.
Step 2.3: Let $S_1=N_\Omega[p_1]-S_0-\{g_1, y_0\}$ and $S_2=N_\Omega[p_2]-S_0-\{g_2, z_0\}$. Construct a match M between S_1 and S_2. Then, for each $(y, z)\in M$, construct a path as follows:
$\langle g_1, p_1\rangle\rightarrow\langle g_1, y\rangle\rightarrow\langle y, \text{Path}_\Omega(g_1, z)\rangle\rightarrow \langle z, \text{Path}_\Omega(y, g_2\rangle \rightarrow\langle g_2, z\rangle\rightarrow\langle g_2, p_2\rangle$.

3.3 Performance Analysis

The correctness of Algorithm 1 is stated in Theorem 2.

Theorem 2. *Let* Ω *be a connected graph*, $\langle g_1, p_1\rangle$ *and* $\langle g_2, p_2\rangle$ *be two distinct nodes in* OTIS-Ω. *Then*, *Algorithm* 1 *constructs at least d parallel paths between these two nodes in* OTIS-Ω, *where* $d = \min\{deg_\Omega(p_1), deg_\Omega(p_2)\}$.

Proof. First, it is straightforward to check that the number of paths constructed by the algorithm is at least d. Secondly, in order to show all these paths are pairwise node disjoint, we note the following two facts: (i) no cluster, except for g_1 and g_2 (g in the case of g_1=g_2=g), is visited by more than one of these paths because S_0, S_1 and S_2 are disjoint sets, and (ii) all the segments of these paths contained in clusters g_1 and g_2 (g in the case of g_1=g_2=g) are pairwise node disjoint. In any case, the pairwise node disjoint property of the constructed paths is easily derived based on the aforementioned two facts, so the details of justifications are omitted. ■

Recall that $deg_{\text{OTIS-}\Omega}(\langle g, g\rangle)=deg_\Omega(g)$, and $deg_{\text{OTIS-}\Omega}(\langle g, p\rangle)=deg_\Omega(p)+1$ for $g\neq p$. From Theorem 2, we know that the number of parallel paths constructed by Algorithm 1 attains the maximum or less one than the maximum.

The performance of Algorithm 1 is given in the following theorem.

Theorem 3. *Let n and N be*, *respectively*, *the size of* Ω *and the size of* OTIS-Ω, $\Delta=\Delta(\text{OTIS-}\Omega)$, $d=d_{\text{OTIS-}\Omega}(\langle g_1,p_1\rangle,\langle g_2,p_2\rangle)$, *and l be the length of any path constructed by Algorithm* 1. *Then*,

(1) *the time complexity of Algorithm* 1 *is* $O(\Delta^2+\Delta f(N^{1/2}))$,

(2) $l \le D$(OTIS-Ω)+4, *and*

(3) *if* $g_1 \ne g_2$ *and* $d = d_\Omega(p_1, g_2) + d_\Omega(g_1, p_2) + 1$, *then* $d \le l \le d+6$ *holds for all the constructed paths*, *with at most* $|S_0|$ *exceptions in Step* 2.2.

Proof. (1) We first prove that the time complexity of the algorithm is $O(\Delta^2 + \Delta f(N^{1/2}))$. On the one hand, generating the sets S_0, S_1 and S_2 requires $O(\Delta^2)$ time, and then obtaining the match M requires $O(\Delta^2)$ time. Based on these sets, on the other hand, constructing all required paths takes $O(\Delta f(n))$ time since at most Δ parallel paths need to be constructed. So, the total running time of the algorithm is $O(\Delta^2 + \Delta f(n))$, namely, $O(\Delta^2 + \Delta f(N^{1/2}))$ due to $N = n^2$.

(2) In the algorithm, obviously, any path contains at most two sub-paths like $Path_\Omega(g,p)$ for some $g, p \in V(\Omega)$, at most three optical links, and at most two other links (a link from $\langle g_1,p_1\rangle$ to its a neighbor in cluster g_1, a link from $\langle g_2,p_2\rangle$ to its a neighbor in cluster g_2). Considering $Path_\Omega(x,y) \le D(\Omega)$ for all $x, y \in V(\Omega)$, we have $l \le 2D(\Omega)+5 = D(\text{OTIS-}\Omega)+4$.

(3) In the case of $g_1 \ne g_2$, we consider the length l of any path constructed in Step 2.1 and Step 2.3. When $d = d_\Omega(p_1, g_2) + d_\Omega(g_1, p_2) + 1$, we have $l = d_\Omega(p_1, g_2) + d_\Omega(g_1, p_2) + 1 = d$ for the path constructed in Step 2.1, and $l \le d_\Omega(y,g_2) + d_\Omega(g_1,z) + 5 \le d_\Omega(p_1,g_2) + d_\Omega(g_1, p_2) + 7 = d+6$ for the path constructed in Step 2.3. Note that the last inequation is based on $y \in N_\Omega[p_1]$ and $z \in N_\Omega[p_2]$. Thus, we have $d \le l \le d+6$, as claimed. ■

Theorem 2 means there exist at least δ(OTIS-Ω) parallel paths between any two distinct nodes in OTIS-Ω, since δ(OTIS-Ω)=δ(Ω). From Menger's Theorem[15], we can derive that OTIS-Ω is maximally fault tolerant. From Theorem 3(2), moreover, we can obtain an upper bound of the fault diameter of OTIS-Ω, the diameter of the resulting graph from OTIS-Ω by removing at most δ(OTIS-Ω)−1 nodes.

Corollary 4. *Let* Ω *be a connected graph. Then*, OTIS-Ω *is maximally fault tolerant*, *and the fault diameter of* OTIS-Ω *is at most* D(OTIS-Ω)+4.

4 Constructing Parallel Paths in OTIS Networks with Maximally Fault Tolerant Basis Graphs

In this section, we consider how to effectively and efficiently construct parallel paths in an OTIS network with a maximally fault tolerant basis graph.

4.1 A Conventional Algorithm

Provided that a parallel path construction method in the basis graph Ω is known, there is a straightforward construction of parallel paths in OTIS-Ω according to [3]. The idea of the construction in OTIS-Ω is as follows. If the source node and the destination node are in the same cluster, the construction of parallel paths between these two nodes is trivial since the construction of parallel paths in Ω is given. Otherwise, for each parallel path $\pi_i(p_1,p_2)$ from p_1 to p_2 in Ω, if the last 2rd node x_i on the path (namely, $x_i \in N_\Omega(p_2)$) such that $x_i \notin \{g_1,g_2\}$ then a path from $\langle g_1,p_1\rangle$ to $\langle g_2,p_2\rangle$ in OTIS-Ω is constructed as follows: $\langle g_1,\pi_i(p_1,x_i)\rangle \rightarrow \langle x_i, Path_\Omega(g_1,g_2)\rangle \rightarrow \langle g_2,x_i\rangle \rightarrow \langle g_2, p_2\rangle$, where $\pi_i(p_1, x_i)$ is a

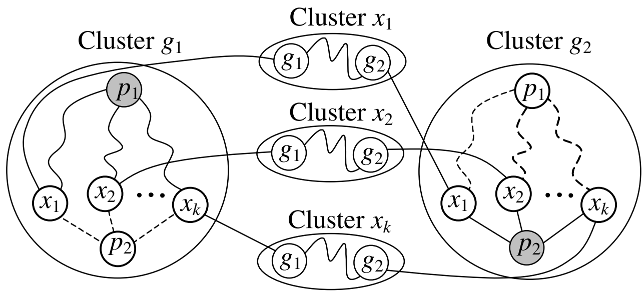

Fig. 3. An illustration of constructing parallel paths(shown solid) from $\langle g_1,p_1\rangle$ to $\langle g_2,p_2\rangle$ for the case of $g_1 \neq g_2$ and $p_1 \neq p_2$ and all $x_i \notin \{g_1,g_2\}$ in Algorithm 2

sub-path of $\pi_i(p_1,p_2)$. See Fig. 3. Notice that even if there exists i such that $x_i \in \{g_1,g_2\}$, one desired path can be constructed by cleverly using π_i as well as p_1 and/or p_2.

The construction described in [3] is here called Algorithm 2, which constructs k parallel paths between $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ in OTIS-Ω, where k is the number of parallel paths from p_1 to p_2 in Ω generated by the given construction of parallel paths in Ω. See [3] for the detailed description and proof of the correctness of the algorithm.

Now we give the performance of Algorithm 2. Assume the given construction of parallel paths in Ω requires $O(g(n))$ time for each path, and the given shortest routing algorithm requires $O(f(n))$ time, where n is the size of Ω. Let $\sigma+d_\Omega(p_1, p_2)$ be an upper bound of the length of any path between nodes p_1 and p_2 in Ω generated by the parallel path construction in Ω. The following theorem establishes the performance of Algorithm 2.

Theorem 5. *Let N be the size of* OTIS-Ω, $\Delta=\Delta(\text{OTIS-}\Omega)$, $d=d_{\text{OTIS-}\Omega}(\langle g_1,p_1\rangle,\langle g_2,p_2\rangle)$, *l be the length of any path constructed by Algorithm* 2. *We have*,

(1) *the time complexity of Algorithm* 2 *is* $O(\Delta g(N^{1/2})+\Delta f(N^{1/2}))$,

(2) $l \leq max\{D(\text{OTIS-}\Omega)+1+\sigma, D(\text{OTIS-}\Omega)+2\}$, *and*

(3) *if* $g_1=g_2$ *or*, $g_1 \neq g_2$ *and* $d=d_\Omega(p_1, p_2)+d_\Omega(g_1, g_2)+2$, *then* $d \leq l \leq d+\sigma$ *holds for all the constructed paths.*

Proof. A proof of Theorem 5 is similar to the one of Theorem 3, and therefore is omitted. ■

4.2 An Improved Algorithm

From Theorem 3 and Theorem 5, we can see that there is no much difference between the performance of Algorithm 1 and one of Algorithm 2. However, we can improve Algorithm 2 in the length of constructed paths by combining it with Algorithm 1, so that the resulting algorithm can offer a guarantee that the length l of any constructed path is not much longer than the distance d between the source node and the destination node with a few possible exceptions for all the cases.

Recall that the distance d between two nodes $\langle g_1,p_1\rangle$ and $\langle g_2,p_2\rangle$ is $d_\Omega(p_1, p_2)$ for $g_1=g_2$, and $min\{d_\Omega(p_1, g_2)+d_\Omega(g_1, p_2)+1, d_\Omega(p_1, p_2)+d_\Omega(g_1, g_2)+2\}$ for $g_1 \neq g_2$. The two items in the minimum function are independent from each other. From Theorem 3 and

5, we think Algorithm 1 is more effective than Algorithm 2 in the case of $g_1 \neq g_2$ and $d_\Omega(g_1, g_2)+d_\Omega(p_1, p_2)+2>d_\Omega(p_1,g_2)+d_\Omega(g_1,p_2)+1$, whereas Algorithm 2 is more effective than Algorithm 1 for the other cases. Therefore, Algorithm 2 can be improved by incorporating Algorithm 1, described as Algorithm 3.

Algorithm 3
If $g_1 \neq g_2$ and $d_\Omega(g_1, g_2)+ d_\Omega(p_1, p_2)+2>d_\Omega(p_1, g_2)+d_\Omega(g_1, p_2)+1$, then construct parallel paths by Algorithm 1; otherwise, construct parallel paths by Algorithm 2.

The performance comparison among the three algorithms is shown in Table 1. In Table 1, N is the size of OTIS-Ω, D=D(OTIS-Ω), l is the length of any constructed parallel paths, d is the distance between nodes $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$, and $\sigma+d_\Omega(p_1, p_2)$ is an upper bound of the length of any path between nodes p_1 and p_2 in Ω generated by the given parallel path construction in Ω. In addition, Case A refers to the case of $g_1 \neq g_2$ and $d_\Omega(g_1, g_2)+ d_\Omega(p_1, p_2)+2> d_\Omega(p_1, g_2)+d_\Omega(g_1, p_2)+1$, and Case B to the other cases.

Table 1. Performance comparison among three algorithms

Alg.	Case A	Case B	Upper Bound on l	Time Complexity
Alg.1	$d \leq l \leq d+6$*	---	$D+4$	$O(\Delta^2+\Delta f(N^{1/2}))$
Alg.2	---	$d \leq l \leq d+\sigma$	$D+1+ max\{1,\sigma\}$	$O(\Delta g(N^{1/2})+\Delta f(N^{1/2}))$
Alg.3	$d \leq l \leq d+6$*	$d \leq l \leq d+\sigma$	$D+1+ max\{3,\sigma\}$	$O(\Delta g(N^{1/2})+\Delta f(N^{1/2}))$

Note. In Table 1, these two inequations with asterisk hold with at most $|S_0|$ exceptions, where $S_0=N_\Omega[p_1] \cap N_\Omega[p_2]-\{g_1,g_2,y_0,z_0\}$ in Step 2.2 of Algorithm 1.

5 An Example—Constructing Parallel Paths in OTIS-Hypercubes

In order to show the effectiveness and efficiency of these algorithms applied to an OTIS network with a specific basis network, in the section, we investigate these algorithms in the context of OTIS-Hypercubes, whose basis graphs are hypercube networks [12]. Hypercube networks and their many variants, including OTIS-Hypercubes, are popular graphs as the models of many interconnection networks. Some research works on OTIS-Hypercubes are reported [2,18]. We use Q_k to denote an k-dimensional hypercube network. Let n is the size of Q_k, namely, $n=2^k$ .It is known that Q_k has a shortest routing algorithm of time complexity $O(\log n)$. Moreover, it has been shown in [12] that Q_k is maximal fault tolerant, and there are k node-disjoint paths between any two nodes x and y in Q_k, each of which can be constructed in $O(\log n)$ time. Among these k paths, $d_{Qk}(x, y)$ paths are of optimal length $d_{Qk}(x, y)$ and k-$d_{Qk}(x, y)$ paths are of length $d_{Qk}(x, y)+2$.

From Theorem 3 and Theorem 5, the time complexity of each of these three algorithms applied to OTIS-Q_k is $O(\log^2 N)$, since $f(n)=O(\log n)$ and $g(n)=O(\log n)$ as well as $\Delta=\log n$. The performance comparison among these algorithms applied to OTIS-Q_k is given in Table 2, which straightforwardly comes from Table 1 due to $\sigma=2$. All notations in Table 2 are the same as ones in Table 1.

From Table 2, we can see that these algorithms are the same efficient in the context of OTIS-Q_k. However, Algorithm 3 slightly outperforms Algorithm 1 and Algorithm 2 with regard to the length of constructed parallel paths.

Table 2. Performance comparison among three algorithms for OTIS-Q_k

Alg.	Case *A*	Case *B*	Upper Bound on *l*	Time Complexity
Alg.1	$d \le l \le d+6$*	---	$D+4$	$O(\log^2 N)$
Alg.2	---	$d \le l \le d+2$	$D+3$	$O(\log^2 N)$
Alg.3	$d \le l \le d+6$*	$d \le l \le d+2$	$D+4$	$O(\log^2 N)$

6 Conclusion

In this paper, we have proposed an effective and efficient construction algorithm for the node-to-node disjoint path problem in an OTIS network with a connected basis network, which can find desired parallel paths of length at most $D+4$ in $O(\Delta^2+\Delta f(N^{1/2}))$ time if the basis network of size n has a shortest routing algorithm of time complexity $O(f(n))$, where D, Δ and N are, respectively, the diameter, the degree and the size of the OTIS network. Obviously, if the basis network with logarithmic degree has a shortest routing algorithm of logarithmic time complexity then the time complexity of the algorithm is $O(\log^2 N)$. The number of parallel paths constructed by the algorithm attains the maximum or less one than the maximum. In addition, in the special case of maximally fault tolerant basis networks, we make an improvement over a conventional construction of node disjoint paths in OTIS networks by incorporating the above algorithm. These obtained algorithms can replace a number of parallel path constructions in OTIS networks for specific basis networks. As an application of these algorithms to OTIS-Hypercubes, desirable node disjoint paths are obtained in $O(\log^2 N)$ time.

It is interesting to find efficient general algorithms for other disjoint path problems, such as node-to-set disjoint paths problem, set-to-set disjoint paths problem and k-pair nodes disjoint path problem, in OTIS networks.

Acknowledgement. Research of the first two authors was supported by the Natural Science Foundation of Guangdong Province, China(No.04020130).

References

1. Chatterjee, S., Pawlowski, S.: All Optical Networks. Comm. ACM 42(6), 74–83 (1999)
2. Day, K.: Optical Transpose k-ary n-cube Networks. J. Systems Architecture 50, 697–705 (2004)
3. Day, K., Al-Ayyoub, A.: Topological Properties of OTIS-Networks. IEEE Trans. Parallel and Distributed Systems 14(4), 359–366 (2002)
4. Hsieh, H.-J., Duh, D.-R.: Constructing Node-Disjoint Paths in Enhanced Pyramid Networks. In: Jesshope, C., Egan, C. (eds.) ACSAC 2006. LNCS, vol. 4186, pp. 380–386. Springer, Heidelberg (2006)

5. Jana, P.K.: Polynomial Interpolation and Polynomial Root Finding on OTIS-Mesh, Parallel Computing 32, 301–312 (2006)
6. Kim, J., Moh, S., Chung, I., Yu, C.: Robust Multipath Routing to Exploit Maximally Disjoint Paths for Wireless Ad Hoc Networks. In: Shen, H.T., Li, J., Li, M., Ni, J., Wang, W. (eds.) Advanced Web and Network Technologies, and Applications. LNCS, vol. 3842, pp. 306–309. Springer, Heidelberg (2006)
7. Marsden, G., Marchand, P., Harvey, P., Esener, S.: Optical Transpose Interconnection System Architecture. Optical Letters 18, 1083–1085 (1993)
8. Osterloh, A.: Sorting on the OTIS-Mesh. In: Proc. 14th Int'l Parallel and Distributed Processing Symp., pp. 269–274 (2000)
9. Parhami, B.: Swapped Interconnection Networks: Topological, Performance, and Robustness Attributes. J. Parallel and Distributed Computing 65, 1443–1452 (2005)
10. Rajasekaran, S., Sahni, S.: Randomized Routing, Selection, and Sorting on the OTIS-Mesh. IEEE Trans. Parallel and Distributed Systems 9(9), 833–840 (1998)
11. Rayn, J.: WDM: North American Development Trend. IEEE Comm. 32(2), 40–44 (1998)
12. Saad, Y., Schultz, M.: Topological properties of hypercubes. IEEE Transactions on Computers 37, 867–871 (1988)
13. Wang, C.-F., Sahni, S.: Image Processing on the OTIS-Mesh Optoelectronic Computer. IEEE Trans. Parallel and Distributed Systems 11(2), 97–109 (2000)
14. Wang, C.-F., Sahni, S.: Matrix Multiplication on the OTIS-Mesh Optoelectronic Computer. IEEE Trans. Computers 50(7), 635–646 (2001)
15. West, D.B.: Introduction to Graph Theory. Prentice-Hall, Englewood Cliffs, NJ (2001)
16. Wu, R.-Y., et al.: Node-disjoint paths in hierarchical hypercube networks. Information Sciences 177, 4200–4207 (2007)
17. Yeh, C.-H., Parhami, B.: Swapped Networks: Unifying the Architectures and Algorithms of a Wide Class of Hierarchical Parallel Processors. In: Proc. Int'l Conf. Parallel and Distributed Systems, pp. 230–237 (1996)
18. Zane, F., Marchand, P., Paturi, R., Esener, S.: Scalable Network Architectures Using the Optical Transpose Interconnection System (OTIS). J. Parallel and Distributed Computing 60(5), 521–538 (2000)