Journal of Interconnection Networks
Vol. 13, Nos. 1 & 2 (2012) 1250002 (48 pages)

DOI: 10.1142/ S0219265912500028

NEARLY OPTIMAL NODE-TO-SET PARALLEL ROUTING IN OTIS NETWORKS

WEIDONG CHEN
Department of Computer Science
South China Normal University
Guangzhou 510631, China

WENJUN XIAO
School of Software Engineering
South China University of Technology
Guangzhou 510641, China

BEHROOZ PARHAMI*

Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA
**parhami@ece.ucsb.edu*

Received 22 September 2009
Revised 26 November 2012

The node-to-set parallel routing problem for a k-connected network Γ is as follows: given a node s and k other nodes $\{t_1, t_2, \ldots, t_k\}$ in Γ, find k node-disjoint paths connecting s and t_i, for $1 \leq i \leq k$. From the viewpoint of applications in synthesizing fast and resilient

*Corresponding author.

collective communication operations, it is desirable to make the parallel paths as short as possible. Building such paths is a nontrivial problem for a general network. Optical transpose interconnection system (OTIS, also known as swapped) networks, a class of hierarchical structures built of n identical n-node factor networks, are known to be maximally fault-tolerant for any connected factor network, implying that they have maximal connectivity. We propose a general algorithm for the node-to-set parallel routing problem in OTIS/swapped networks that yields paths of length no greater than $D + 4$ in $O(\Delta^2 + \Delta f(n))$ time, where D and Δ represent the diameter and degree of the OTIS network and $O(f(n))$ is the time complexity of a shortest-path routing algorithm for the n-node factor network. Our node-to-set routing algorithm is shown to have optimal time complexity for certain OTIS networks of practical interest, including OTIS-Mesh and OTIS-Hypercube.

Keywords: *Connectivity; fault tolerance; node-disjoint paths; parallel routing algorithm;* OTIS *network; swapped network.*

1. Introduction

Optical transpose interconnection system (OTIS) networks are useful structures for parallel computation and communication.[1,2] An OTIS network with n^2 nodes is a two-level swapped architecture built of n copies of an n-node factor network that constitute its clusters.[3] 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, fault tolerance, and algorithmic efficiency of the resulting networks. OTIS networks have received considerable attention in recent years, and they now occupy a special place among real-world architectures for parallel and distributed systems.[4,5] A number of algorithms have been developed for routing, selection, sorting,[6,7] numerical computation,[8,9] and image processing[10] on OTIS networks.

Multiple paths in a network Γ are node-disjoint if all the nodes contained in them are distinct, except possibly for their starting and ending nodes. Finding node-disjoint paths (or parallel routing) in networks is one of the fundamental problems in design and implementation of parallel and distributed computing systems.[11,12] Because communication time can dominate processor execution time in many situations, it is important for a network to be capable of efficient routing of data among nodes. Use of node-disjoint paths for this purpose not only guarantees freedom from conflicts, and thus high performance, but also allows for successful routing in the event of node and/or link failures.[13,14] A measure of network fault tolerance is the existence of alternate paths between nodes: the more node-disjoint paths, the better. A network Γ is maximally fault-tolerant if and only if there exist at least $\delta(\Gamma)$ node-disjoint paths between two distinct nodes in Γ, where $\delta(\Gamma)$ is the minimum node degree of Γ.[15,16]

One might seek k node-disjoint or parallel paths in four distinct situations: (1) Node-to-node, where the parallel paths must lead from source node s to destination node t; (2) Node-to-set, where the parallel paths must lead from source node s to k destination nodes t_i; (3) Set-to-set, where the parallel paths must connect each one of k source nodes s_i to one (distinct, but arbitrarily chosen) destination node t_{j_i} from among k given nodes t_i; (4) k-pairs, where the parallel paths must connect node pairs s_i and t_i, $1 \le i \le k$. The set-to-set and the k-pairs parallel routing problems are NP-hard for a general network.[17–19] For the node-to-node parallel routing problem, Menger's theorem[15] guarantees the existence of at least k parallel paths between any two nodes in a k-connected network. The same guarantee of existence is provided by Menger's theorem for the node-to-set problem in a k-connected network. In both cases, however, the existence of the paths does not imply that they are easy to find, especially if one aims to minimize the longest or average path length.

A general approach for node-to-node and node-to-set parallel routing problems is the flow technique, that takes $O(N^{2.5})$ time, where N is the number of nodes.[20] The nearly cubic time complexity can render this approach impractical for general networks of very large orders. Node-to-node and node-to-set disjoint paths have been efficiently constructed for a variety of popular networks by making use of specific structural properties.[12,13,21–25] Node-to-set disjoint paths were first studied by Rabin,[26] who noted their usefulness in devising efficient and fault-tolerant randomized routing algorithms.

Previous research on node-disjoint paths in OTIS networks has been limited to constructing node-to-node disjoint paths when the OTIS network is built of a maximally fault-tolerant factor network. The methods used take advantage of the corresponding constructions of node-disjoint or parallel paths in the associated factor networks.[27–29] Recently, a general construction of a maximal number of node-to-node disjoint paths has led to the proof that any OTIS network built of a connected factor network is maximally fault-tolerant.[30] This discovery, which implies that OTIS networks have maximal connectivity[15] regardless of the factor network on which they are based, has rendered obsolete a number of earlier network-specific results.

The node-to-set parallel routing problem for OTIS networks has not been solved previously. The node-to-set problem is more difficult than the node-to-node case, because the latter can be readily reduced to the former. While it is not difficult to obtain efficient construction strategies for node-to-set disjoint paths in particular instances of OTIS networks, such as OTIS-Hypercube, by taking advantage of the specific structure of their factor networks, it would be much more productive to study general methods that are applicable to OTIS networks built of any factor network. Such a general construction of node-to-set disjoint paths in OTIS networks, based only on the interconnection rule of this class networks and with no reference to specific properties of a particular factor network, is what is being proposed here.

Let D, Δ, and N be the diameter, maximal node degree, and order (total number of nodes) of an OTIS network, built of an arbitrary connected factor network. The algorithm proposed in this paper yields the required number of node-disjoint paths of length no greater than $D + 4$ within this arbitrary OTIS network in $O(\Delta^2+\Delta f(\sqrt{N}))$ time, if the factor network of order n has a shortest-path routing algorithm of time complexity $O(f(n))$. This parallel-path construction algorithm is optimal in terms of time complexity for a number of OTIS networks of practical interest, such as OTIS-Mesh and OTIS-Hypercube.

The rest of this paper is organized as follows. In Sec. 2, OTIS networks are defined and related notions that are needed in our constructions and proofs are introduced. The algorithm for the node-to-set parallel routing problem in OTIS networks is presented in Sec. 3, with its performance analyzed in Sec. 4. Section 5 contains two example applications of the node-to-set parallel routing algorithm. Section 6 concludes the paper.

2 Preliminaries

Because networks are often modeled as graphs, we use the terms "graph" and "network" interchangeably. Let Γ be a simple undirected graph (graph, for short) with vertex (node) set $V(\Gamma)$ and edge (link) set $E(\Gamma)$. The number of nodes of Γ, $|V(\Gamma)|$, is called its order. For $v \in V(\Gamma)$, we denote by $\deg_\Gamma(v)$ the degree of v in Γ, by $N_\Gamma(v) = \{u \in V \mid (v, u) \in E(\Gamma)\}$ the open neighborhood of v, and by $N_\Gamma[v] = N_\Gamma(v) \cup \{v\}$ its closed neighborhood. The maximum degree among the vertices of Γ is denoted by $\Delta(\Gamma)$ and the minimum degree by $\delta(\Gamma)$. The distance between nodes u and v, denoted by $d_\Gamma(u, v)$, is the length of a shortest path between u and v. The diameter $D(\Gamma)$ of Γ is the maximal distance between any two nodes of Γ. Two paths from u to v are node-disjoint (also called parallel paths) if they share no internal node. The connectivity of Γ is the minimal number of nodes whose removal can cause Γ to become disconnected or trivial (degenerating to a single node). A graph is said to be k-connected if its connectivity is at least k. A graph Γ of connectivity $\delta(\Gamma)$ is maximally fault-tolerant. Other notation and terminology used in this paper follow those in Ref. 15.

Definition A. *OTIS (Swapped) network*[1,3]: *The OTIS (swapped) network OTIS-*Ω, *derived from the graph* Ω, *is a graph with the vertex set* $V(OTIS\text{-}\Omega) = \{\langle g, p\rangle \mid g, p \in V(\Omega)\}$ *and the edge set* $E(OTIS\text{-}\Omega)$, which is the union of two sets: intracluster edges $\{(\langle g, p_1\rangle, \langle g, p_2\rangle) \mid g \in V(\Omega), (p_1, p_2) \in E(\Omega)\}$ and intercluster edges $\{(\langle g, p\rangle, \langle p, g\rangle) \mid g, p \in V(\Omega), g \neq p\}$.

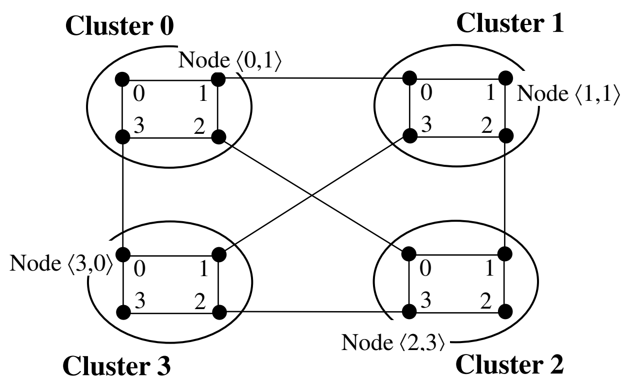


Fig. 1. OTIS-C_4, where C_4 is a cycle of order 4 and $V(C_4) = \{0, 1, 2, 3\}$.

In OTIS-Ω, the graph Ω is called the basis network or factor network. If Ω has n nodes, then OTIS-Ω is composed of n node-disjoint subnetworks called clusters, each of which is isomorphic to the factor network Ω. A node of OTIS-Ω that is labeled $\langle g, p\rangle$ constitutes node p of cluster g. An intercluster (or optical) link connects node p of cluster g to node g of cluster p, for all $p \neq g$. No intercluster link is incident to node p of cluster p. An example OTIS network appears in Fig. 1. Intuitively, if every cluster in OTIS-C_4 is viewed as a supernode, then the resulting graph of all the supernodes along with all the optical links will form the complete graph K_4.

Based on Definition A, the following basic topological parameters of OTIS-Ω are easily derived as functions of the corresponding parameters of Ω[27,29]:

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

By Definition A, one also easily knows that OTIS-Ω is connected if and only if Ω is connected. Furthermore, we have the following result.

Theorem A. *Maximal fault tolerance*[30]: *OTIS-Ω is maximally fault-tolerant if the factor network Ω is a connected graph.*

In what follows, we assume that the factor graph Ω is connected and that a shortest-path routing algorithm in Ω is known. This shortest-path routing algorithm in Ω can be used to perform shortest-path routing in OTIS-Ω. For convenience, in any cluster g, the edge from u to v is denoted by $\langle g, u \rightarrow v\rangle$, and a shortest path from x to y that is completely contained in cluster g, by $\langle g, x \Rightarrow y\rangle$. For any $g \neq p$, $\langle g, p\rangle \rightarrow \langle p, g\rangle$ denotes the intercluster link from $\langle g, p\rangle$ to $\langle p, g\rangle$. Additionally, for consistency and brevity, null links and null paths such as $u \rightarrow u$ and $x \Rightarrow x$ are allowed as segments (subpaths or components) of routing paths. In this way, when $x \Rightarrow y$ is a segment of a routing path, say, we do not have to treat the case $x = y$ separately, given that the null path segment would not contribute any distance or latency. This leads to far fewer cases in our analyses and proofs. On the null path $x \Rightarrow x$, node x is both the immediate predecessor of, and the immediate successor to, itself.

From Theorem A, the connectivity of OTIS-Ω is δ(Ω), given that δ(OTIS-Ω) = δ(Ω). The node-to-set parallel routing problem in OTIS-Ω is to construct k node-disjoint paths from a given source node to k given destination nodes, where $k \leq \delta(\Omega)$.

3 The Parallel Routing Algorithm

In this section, we present an algorithm for a more general node-to-set parallel routing problem in OTIS-Ω: Given a source node $\langle g_0, s = t_0\rangle$ and a set $T = \{\langle g_1, t_1\rangle, \langle g_2, t_2\rangle, \ldots, \langle g_k, t_k\rangle\}$ of k destination nodes in OTIS-Ω, where $k \leq \min\{\deg_\Omega(t_i) \mid 0 \leq i \leq k\}$ and $\langle g_0, s\rangle \notin T$, construct k node-disjoint paths from the source node $\langle g_0, s\rangle$ to every destination node $\langle g, t\rangle \in T$; furthermore, strive to make the constructed paths as short as possible, in the sense of the longest of these paths not being much longer than the network diameter. Note that $\delta(\Omega) \leq \min\{\deg_\Omega(t_i) \mid 0 \leq i \leq k\}$. Notationally, cluster g_0 is the source cluster, cluster g_i (for $1 \leq i \leq k$) is a destination cluster, and any other cluster that is traversed by these paths constitutes an intermediate cluster.

3.1 Basic Idea

We first provide an intuitive description of our algorithm. Because the connected factor network Ω is not necessarily maximally fault-tolerant, there is no guarantee that there exist two node-disjoint paths between a pair of distinct nodes in Ω, even if $\delta(\Omega) \geq 2$ (see Fig. 2). On the other hand, from the rule for intercluster connectivity in OTIS-Ω, we know that for an arbitrary node v in a cluster g, every node of $N_\Omega[v]$ (other than node g, when $g \in N_\Omega[v]$) is connected outwards to a different cluster by an intercluster link. Thus, there are at least $|N_\Omega[v]| - 1$ node-disjoint ways for a path beginning at v to leave cluster g immediately. Based on these facts, we will aim to force the paths beginning at the source node and each destination node to leave their current clusters as quickly as possible to ensure the node-disjoint property (Note that $|N_\Omega[t_i]| = deg_\Omega(t_i) + 1 \geq k + 1$, for $0 \leq i \leq k$). We will also aim to have each intermediate cluster visited by exactly one path, as a way of ensuring nonintersecting paths. Moreover, in order to keep these paths short, we will allow no more than two intermediate clusters on a path, and construct every subpath contained in a cluster by using the given shortest routing algorithm of Ω. Note that using two intermediate clusters is necessary in some circumstances to ensure node-disjoint paths. It is readily seen, for example, that given the factor network of Fig. 2, a path parallel to (node-disjoint from) the single-step path $\langle g, u \rightarrow v\rangle$, from node u to node v of the same cluster g, must leave cluster g for an intermediate cluster g', and then reenter cluster g from another intermediate cluster g''.

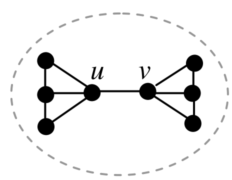


Fig. 2. An example 1-connected factor graph Ω with δ(Ω) = 2.

According to the strategy described above, for a typical destination node $\langle g, t\rangle \in T$, the path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ will begin at node s in cluster g_0, leave cluster g_0 immediately by way of a cluster-g_0 neighbor of s, and then pass through at most two intermediate clusters (called the first intermediate cluster and the second intermediate cluster, if any, respectively), until it eventually enters the destination cluster g along an intercluster link, typically incident to a neighbor of t in cluster g, prior to arriving at t. Figure 3 provides an overall view of our strategy for constructing node-to-set disjoint paths.

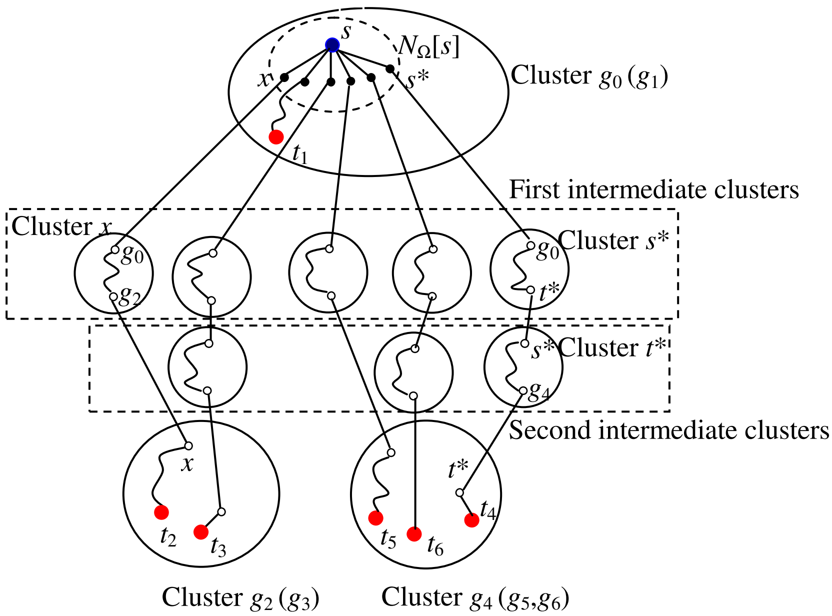


Fig. 3. Illustrating the construction of node-to-set disjoint paths in an OTIS network.

Construction of subpaths and selection of intermediate clusters so as to satisfy the following three criteria yields the required k node-disjoint paths:

(**C1**) Subpaths contained in the source cluster are node-disjoint.

(**C2**) Possible multiple subpaths contained in any destination cluster are node-disjoint.

(**C3**) The intermediate clusters of one path, if any, are different from those of any other path, and also from the source cluster and any of the destination clusters.

3.2 Routing Functions

The construction of subpaths contained in the source cluster and the destination clusters is related to the selection of intermediate clusters. In fact, we have the following results.

- A path from $\langle g_0, s\rangle$ to $\langle g_0, t\rangle$ has no intermediate cluster if and only if the path is $\langle g_0, s \Rightarrow t\rangle$.
- A path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$, $g \neq g_0$, has no intermediate cluster if and only if the path is of the form $\langle g_0, s \Rightarrow g\rangle \rightarrow \langle g, g_0 \Rightarrow t\rangle$.
- A path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$, $g \neq g_0$, has only one intermediate cluster $s^* \in N_\Omega[s]$ if and only if the path is of the form $\langle g_0, s \rightarrow s^*\rangle \rightarrow \langle s^*, g_0 \Rightarrow g\rangle \rightarrow \langle g, s^* \Rightarrow t\rangle$.
- A path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ has a first (second) intermediate cluster $s^* \in N_\Omega[s]$ ($t^* \in N_\Omega[t]$) if and only if the path is of the form $\langle g_0, s \rightarrow s^*\rangle \rightarrow \langle s^*, g_0 \Rightarrow t^*\rangle \rightarrow \langle t^*, s^* \Rightarrow g\rangle \rightarrow \langle g, t^* \rightarrow t\rangle$.

For convenience, we take $\langle g_0, s\rangle$ to be the implied source node in all cases, and use $F_1(\langle g, t\rangle)$ and $F_2(\langle g, t\rangle)$ to denote the first and the second intermediate cluster indices, respectively, selected for the path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$. Clearly, F_1 can be viewed as a function from T to $N_\Omega[s] \cup \{\Lambda\}$, and F_2 as a function from T to $V(\Omega) \cup \{\Lambda\}$, such that $F_2(\langle g, t\rangle) \in N_\Omega[t] \cup \{\Lambda\}$ for any $\langle g, t\rangle \in T$, where Λ represents null. In particular, by $F_1(\langle g, t\rangle) = F_2(\langle g, t\rangle) = \Lambda$, we mean the path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ has no intermediate cluster, and by $F_2(\langle g, t\rangle) = \Lambda$ and $F_1(\langle g, t\rangle) \neq \Lambda$, we mean the path has only one intermediate cluster. Given $F_1(\langle g, t\rangle)$ and $F_2(\langle g, t\rangle)$, it is trivial to obtain the path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$. Thus, our focus should be on computing the two functions F_1 and F_2, from which we can obtain the desired k node-disjoint paths.

Definition 1. *Let F_1 be a function from T to $N_\Omega[s] \cup \{\Lambda\}$, and F_2 a function from T to $V(\Omega) \cup \{\Lambda\}$, such that $F_2(\langle g, t\rangle) \in N_\Omega[t] \cup \{\Lambda\}$ for every $\langle g, t\rangle \in T$. We call F_1 and F_2 a pair of routing functions if the k paths obtained from F_1 and F_2 are node-disjoint.*

To compute a pair of routing functions satisfying Definition 1, we must construct appropriate subpaths contained in the source cluster and in the destination cluster for every destination node. In computing $F_2(\langle g, t\rangle)$ or $F_1(\langle g, t\rangle)$ for a new destination node $\langle g, t\rangle \in T$, we must ensure that all subpaths constructed thus far in the source cluster and in every destination cluster satisfy Criteria (C1) and (C2), and that all the intermediate clusters utilized thus far satisfy Criterion (C3). If so, we say that $F_2(\langle g, t\rangle)$ and $F_1(\langle g, t\rangle)$ are valid. Obviously, if $F_2(\langle g, t\rangle)$ and $F_1(\langle g, t\rangle)$ are valid for every node $\langle g, t\rangle \in T$, then F_1 and F_2 are the desired pair of routing functions.

To minimize the number of intermediate clusters used in routing paths, we try to pick, for each destination cluster g, a destination node in cluster g so that the path to it does not need a second intermediate cluster. Accordingly, we will first compute $F_2(\langle g, t\rangle)$ for all $\langle g, t\rangle \in T$, and then derive $F_1(\langle g, t\rangle)$. We describe the computation of F_2 and F_1, beginning with a few needed additional notations. Note that if cluster g_0 is also a destination cluster, then the path construction process is very similar to the case where the destination cluster g is different from g_0, but it contains more than one destination node. Thus, we proceed with the assumption $g \neq g_0$.

Definition 2. *For any $x \in N_\Omega[s]$, x is fixed node if it belongs to an already constructed subpath contained in the source cluster*; *otherwise x is a free node*.

Definition 3. *Let cluster g contain more than one destination node. A destination node $\langle g, \overline{t}\rangle$ is a head destination node of cluster g if $\overline{t}$ satisfies the following conditions*:

$$\overline{t} \begin{cases} \text{argmin}\{d_\Omega(g_0, t') \mid \langle g, t'\rangle \in T\}, & \textit{if} \quad g \in N_\Omega[s] \textit{ and } g \textit{ is free} \\ \text{argmin}\{d_\Omega(s, t') \mid \langle g, t'\rangle \in T\}, & \textit{otherwise} \end{cases} \quad (1)$$

where argmin, when applied to a function, yields the argument that minimizes the function. The subpath of the head destination node $\langle g, \overline{t}\rangle$ contained in cluster g is

$g_0 \Rightarrow \bar{t}$ if $g \in N_\Omega[s]$ and g is free; it is $s \Rightarrow \bar{t}$, otherwise.

Definition 4. *Let cluster g contain more than one destination node, $\langle g, \bar{t}\rangle$ be a head destination node of cluster g, and p be the immediate predecessor of $\bar{t}$ on the subpath leading to $\langle g, \bar{t}\rangle$ within cluster g. For any $\langle g, t\rangle \in T$ such that $t \neq \bar{t}$, let G = $\{g_0, g_1, \ldots, g_k\} \cup \{t' \mid$ cluster t′ has been selected as a second intermediate cluster$\}$, $D_t = \{t' \mid \langle g, t'\rangle \in T\} - \{t\}$. We call G the global forbidden set of $\langle g, t\rangle$,* and *$G \cup D_t \cup \{p\}$ the forbidden set of $\langle g, t\rangle$.*

(A) Computing F_2

We compute F_2 for all destination nodes in one cluster, then proceed to another cluster, and so on; if cluster g_0 is also a destination cluster, we begin with cluster g_0. For a cluster g containing only one destination node $\langle g, t\rangle$, the path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ will not need a second intermediate cluster, and $F_2(\langle g, t\rangle) = \Lambda$. For every cluster g containing more than one destination node, we first pick out a head destination node $\langle g, \bar{t}\rangle$ in cluster g that will not need a second immediate cluster. We then compute $F_2(\langle g, \bar{t}\rangle)$ first, and proceed with computing F_2 for every other destination node $\langle g, t\rangle$ in cluster g.

Computing F_2 for the head destination node $\langle g, \bar{t}\rangle$ in cluster g: If $g \in N_\Omega[s]$ and g is free, the path from $\langle g_0, s\rangle$ to $\langle g, \bar{t}\rangle$ will be $\langle g_0, s \rightarrow g\rangle \rightarrow \langle g, g_0 \Rightarrow \bar{t}\rangle$. This path requires no intermediate cluster, leading to $F_1(\langle g, \overline{t}\rangle) = F_2(\langle g, \overline{t}\rangle) = \Lambda$. Otherwise, that is, if $g \notin N_\Omega[s]$ or g is a fixed node in $N_\Omega[s]$, the path from $\langle g_0, s\rangle$ to $\langle g, \overline{t}\rangle$ does not need a second intermediate cluster, and $F_2(\langle g, \overline{t}\rangle) = \Lambda$.

Computing F_2 for every other destination node $\langle g, t\rangle$ in cluster g: The path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ leaves cluster g immediately, at node t itself or at a neighbor node of t in cluster g, implying that $F_2(\langle g, t\rangle) \in N_\Omega[t] \cup \{\Lambda\}$. Specifically, if $N_\Omega[t] - (G \cup D_t \cup \{p\}) \neq \varnothing$, we compute $F_2(\langle g, t\rangle)$ as:

$$F_2(\langle g, t\rangle) = \begin{cases} t, & \text{if } t \notin G; \\ \text{an arbitrary node} \in N_\Omega[t] - (G \cup D_t \cup \{p\}), & \text{otherwise} \end{cases} \tag{2}$$

By the definition of G, once $F_2(\langle g, t\rangle)$ has been computed, we will have $t \in G$ and $F_2(\langle g, t\rangle) \in G$; hence, all the nodes on the subpath of $\langle g, t\rangle$ contained in cluster g will belong to G. Hence, the subpath of $\langle g, t\rangle$ in cluster g is node-disjoint from all the other subpaths constructed thus far in cluster g; it is node-disjoint from the subpath of $\langle g, \overline{t}\rangle$ in cluster g, because neither node $\overline{t}$ nor node p is on it. Thus, the obtained $F_2(\langle g, t\rangle)$ is valid. On the other hand, if $N_\Omega[t] - (G \cup D_t \cup \{p\}) = \varnothing$, we use the three rules (R1), (R2), and (R3), to be introduced later, to compute $F_2(\langle g, t\rangle)$; the validity of such a choice will also be proven later.

Procedure *Routing-Function* {Compute $F_2(\langle g, t\rangle)$ and $F_1(\langle g, t\rangle)$ for all $\langle g, t\rangle \in T$}

1. Initialize:

Let $PathSet = \varnothing$, $G = \{g_0, g_1, \dots, g_k\}$, and mark every $x \in N_\Omega[s]$ as free.

2. Compute F_2 for all $\langle g, t\rangle \in T$:

2.1. If cluster g_0 is a destination cluster **Then**

Let $\bar{t} = argmin\{d_\Omega(s, t') \mid \langle g_0, t'\rangle \in T\}$, x and p be the immediate successor to s and the immediate predecessor of $\bar{t}$, respectively, on the path $s \Rightarrow \bar{t}$.

Case 1 $\{d_\Omega(s, \bar{t}) \geq 2\}$: Add $\langle g_0, s \Rightarrow \bar{t}\rangle$ to *PathSet*, mark x as fixed, and set $F_1(\langle g_0, \bar{t}\rangle) = F_2(\langle g_0, \bar{t}\rangle) = \Lambda$.

Case 2 $\{d_\Omega(s, \bar{t}) \leq 1\}$: For every $\langle g_0, t\rangle \in T$ such that $t \in N_\Omega[s]$, add $\langle g_0, s \rightarrow t\rangle$ to *PathSet*, mark t as fixed, and set $F_1(\langle g_0, t\rangle) = F_2(\langle g_0, t\rangle) = \Lambda$.

To compute $F_2(\langle g_0, t\rangle)$ for every other node $\langle g_0, t\rangle$ in cluster g_0, if any, use Eq. (2) and add $F_2(\langle g_0, t\rangle)$ to G if $N_\Omega[t] - (G \cup D_t \cup \{p\}) \neq \varnothing$; otherwise, use Rule (R1).

2.2. For every destination cluster g such that $g \neq g_0$ **Do**

If (cluster g contains more than one destination node) **Then**

Case 1 {g is a free node in $N_\Omega[s]$}: Let $\bar{t} = argmin\{d_\Omega(g_0, t') \mid \langle g, t'\rangle \in T\}$, and p be the immediate predecessor of $\bar{t}$ on the path $g_0 \Rightarrow \bar{t}$. Add $\langle g_0, s \rightarrow g\rangle \rightarrow \langle g, g_0 \Rightarrow \bar{t}\rangle$ to *PathSet*, mark g as fixed, and set $F_1(\langle g, \bar{t}\rangle) = F_2(\langle g, \bar{t}\rangle) = \Lambda$.

Case 2 {$g \notin N_\Omega[s]$ or g is a fixed node in $N_\Omega[s]$}: Let $\bar{t} = argmin\{d_\Omega(s, t') \mid \langle g, t'\rangle \in T\}$, and p be the immediate predecessor of $\bar{t}$ on the path $s \Rightarrow \bar{t}$. Set $F_2(\langle g, \bar{t}\rangle) = \Lambda$.

To compute $F_2(\langle g, t\rangle)$ for every other node $\langle g, t\rangle$ in cluster g, use Eq. (2) and add $F_2(\langle g, t\rangle)$ to G if $N_\Omega[t] - (G \cup D_t \cup \{p\}) \neq \varnothing$; otherwise use Rule (R1)**,** (R2), or (R3).

Else Set $F_2(\langle g, t\rangle) = \Lambda$ {cluster g contains only one destination node}

3. Compute F_1 for all $\langle g, t\rangle \in T$:

3.1. For every free node x in $N_\Omega[s]$ **Do**

Case 1 {$x = g_i$ for some i in the range $1 \leq i \leq k$}: Add $\langle g_0, s \rightarrow x = g_i\rangle \rightarrow \langle g_i, g_0 \Rightarrow t_i\rangle$ to *PathSet*, mark x as fixed , and set $F_1(\langle g_i, t_i\rangle) = F_2(\langle g_i, t_i\rangle) = \Lambda$.

Case 2 {$x = F_2(\langle g_0, t\rangle)$ for some $\langle g_0, t\rangle \in T$}: Add $\langle g_0, s \rightarrow x \rightarrow t\rangle$ to *PathSet*, mark x as fixed, and set $F_1(\langle g_0, t\rangle) = F_2(\langle g_0, t\rangle) = \Lambda$.

Case 3 {$x = F_2(\langle g, t\rangle)$ for some $\langle g, t\rangle \in T$ and $g \neq g_0$}: Mark x as fixed, set $F_1(\langle g, t\rangle) = x$ and $F_2(\langle g, t\rangle) = \Lambda$.

3.2. For every destination node $\langle g, t\rangle$ such that $F_1(\langle g, t\rangle)$ has not yet been computed **Do**

Pick an arbitrary free node s^* in $N_\Omega[s]$, set $F_1(\langle g, t\rangle) = s^*$, and mark s^* as fixed.

Fig. 4. The procedure *Routing-Function*, which returns the intermediate clusters $F_1(\langle g, t\rangle)$ and $F_2(\langle g, t\rangle)$, if any, used in constructing the parallel path from node $\langle g_0, s\rangle$ to node $\langle g, t\rangle$ for all $\langle g, t\rangle \in T$.

(B) Computing F_1

Once the function F_2 has been computed, it is an easy matter to compute F_1, a function from T to $N_\Omega[s] \cup \{\Lambda\}$. Note that there may exist free nodes in $N_\Omega[s]$ that belong to G. Thus, to satisfy Criterion (C3), we first need to fix these free nodes in $N_\Omega[s]$ by constructing paths needing no intermediate cluster or merging two identical intermediate clusters into one. Then, we can randomly assign a free node x in $N_\Omega[s]$ to $F_1(\langle g, t\rangle)$ for every destination node $\langle g, t\rangle$ whose F_1 value has not yet been computed because $x \notin G$. It is easy to see that if $F_2(\langle g, t\rangle)$ is valid for every node $\langle g, t\rangle \in T$, then $F_1(\langle g, t\rangle)$ will be valid for every node $\langle g, t\rangle \in T$.

The procedure *Routing-Function* for computing F_1 and F_2 is shown in Fig. 4. In these computations, if a path needs no intermediate cluster, then it will be constructed completely and directly recorded in *PathSet*. Next, we assume $N_\Omega[t] - (G \cup D_t \cup \{p\}) = \varnothing$ when computing $F_2(\langle g, t\rangle)$ for some $\langle g, t\rangle$ such that $t \neq \bar{t}$, where cluster g contains more than one destination node, and p is the immediate predecessor to $\bar{t}$ on the subpath of the head destination node $\langle g, \bar{t}\rangle$ within cluster g; a situation we refer to as "an obstacle." Let g^* be the immediate successor to s on the path $s \Rightarrow g$. Thus, we have $g^* \in N_\Omega[s]$. In this case, we apply Rules (R1)–(R3) below to compute a valid $F_2(\langle g, t\rangle)$, where $\alpha = (p \neq \bar{t} \wedge p \in N_\Omega[t] \wedge \bar{t} \notin G)$.

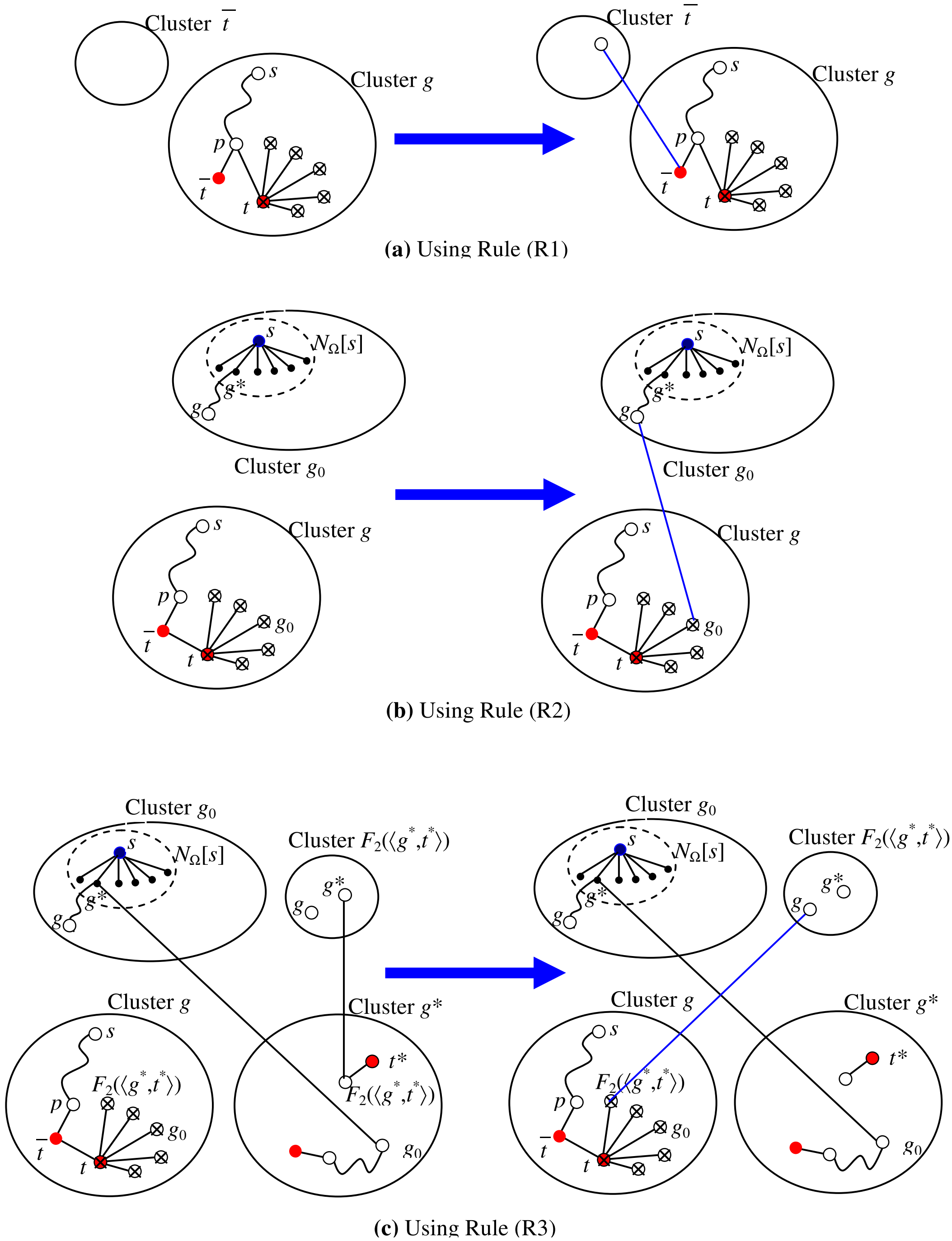


Fig. 5. Examples of applying Rules (R1), (R2), and (R3) to compute a valid $F_2(\langle g, t\rangle)$, when $N_\Omega[t] - (G \cup D_t \cup \{p\}) = \varnothing$. Nodes shown as $\otimes$ belong to the forbidden set of $\langle g, t\rangle$, excepting $\bar{t}$ and p.

(**R1**) {α is true}: We exchange the roles of $\overline{t}$ and t in cluster g, that is, we make $\langle g, t\rangle$ the head destination node in cluster g. Thus, for $\langle g, t\rangle$, the subpath contained in cluster g will be $s \Rightarrow p \to t$ ($g_0 \Rightarrow p \to t$, if g is a free node in $N_\Omega[s]$) and $F_2(\langle g, t\rangle) = \Lambda$. For $\langle g, \overline{t}\rangle$, the subpath contained in cluster g will be the single node $\overline{t}$ and $F_2(\langle g, \overline{t}\rangle) = \overline{t}$ (see Fig. 5(a)).

(**R2**) {α is false and g^* is a free node in $N_\Omega[s]$}: Assume that $g \notin N_\Omega[s]$ and $g_0 \in N_\Omega[t]$.

Case 1 {$g_0 \notin D_t$}: For $\langle g, t\rangle$, we construct the path $\langle g_0, s \Rightarrow g\rangle \to \langle g, g_0 \to t\rangle$, which needs no intermediate cluster, and set $F_2(\langle g, t\rangle) = F_1(\langle g, t\rangle) = \Lambda$ (see Fig. 5(b)).

Case 2 {$g_0 = t' \in D_t$, such that $t' \neq \overline{t}$ and $F_2(\langle g, t'\rangle) \in N_\Omega[t]$}: For $\langle g, t\rangle$, we set $F_2(\langle g, t\rangle) = F_2(\langle g, t'\rangle)$. Then, for $\langle g, t'\rangle$, we construct the new path $\langle g_0, s \Rightarrow g\rangle \to \langle g, g_0 = t'\rangle$, needing no intermediate cluster, and set $F_2(\langle g, t'\rangle) = F_1(\langle g, t'\rangle) = \Lambda$.

Case 3 {$g_0 = \overline{t}$, $p \in N_\Omega[t]$ and $p \neq \overline{t}$}: Node $\langle g, t\rangle$ is made the head destination node of cluster g, in lieu of $\langle g, \overline{t}\rangle$. That is, for $\langle g, t\rangle$, we use the cluster-g subpath $s \Rightarrow p \to t$, and set $F_2(\langle g, t\rangle) = \Lambda$. Then, for $\langle g, \overline{t}\rangle$, we construct the path $\langle g_0, s \Rightarrow g\rangle \to \langle g, g_0 = \overline{t}\rangle$, needing no intermediate cluster, and set $F_2(\langle g, \overline{t}\rangle) = F_1(\langle g, \overline{t}\rangle) = \Lambda$.

(**R3**) {α is false and g^* is a fixed node in $N_\Omega[s]$}: Assume that $g \notin N_\Omega[s]$, and let $\langle g^*, t^* \rangle$ be one destination node in cluster g^* such that $F_2(\langle g^*, t^* \rangle) \in N_\Omega[t]$.

Case 1 {$F_2(\langle g^*, t^* \rangle) \notin D_t$}: For $\langle g, t \rangle$, we set $F_2(\langle g, t \rangle) = F_2(\langle g^*, t^* \rangle)$, and then recompute $F_2(\langle g^*, t^* \rangle)$ in cluster g^* (see Fig. 5(c)).

Case 2 {$F_2(\langle g^*, t^* \rangle) = t' \in D_t$ such that $t' \neq \bar{t}$ and $F_2(\langle g, t' \rangle) \in N_\Omega[t]$}: For $\langle g, t \rangle$ and $\langle g, t' \rangle$, we set $F_2(\langle g, t \rangle) = F_2(\langle g, t' \rangle)$ and $F_2(\langle g, t' \rangle) = t'$, respectively, and then recompute $F_2(\langle g^*, t^* \rangle)$ in cluster g^*.

Case 3 {$F_2(\langle g^*, t^* \rangle) = \bar{t}$, $p \in N_\Omega[t]$ and $p \neq \bar{t}$}: Node $\langle g, t \rangle$ is made the head destination node in cluster g, in lieu of $\langle g, \bar{t} \rangle$. For $\langle g, t \rangle$, we use the cluster-g subpath $s \Rightarrow p \rightarrow t$, and set $F_2(\langle g, t \rangle)=\Lambda$. Then, we set $F_2(\langle g, \bar{t} \rangle)= \bar{t}$ and recompute $F_2(\langle g^*,t^* \rangle)$ in cluster g^*.

Note that once an obstacle occurs, there will be exactly one rule among Rules (R1)–(R3) whose conditions are met; the corresponding assumptions of the rule, if any, will be proved to be true in Sec. 4. It is readily observed that in Rules (R1)–(R3), we obtain $F_2(\langle g, t \rangle)$ via changing the head destination node or a second intermediate cluster, or by constructing a path that needs no intermediate cluster. The validity of such a choice for $F_2(\langle g, t \rangle)$ will also be proven in Sec. 4. It is worth pointing out that in using these rules, if node $\langle g, t \rangle$ becomes the head destination node in cluster g, in lieu of $\langle g, \bar{t} \rangle$, then the subpath $s \Rightarrow p \rightarrow t$ is not necessarily the same as the subpath $s \Rightarrow t$ obtained by using shortest-path routing in Ω directly, although they both are shortest paths from s to t in Ω. Therefore, in order to subsequently construct the complete path based on F_2 and F_1, we should reserve the subpath. For consistency and brevity, we still denote the subpath $s \Rightarrow p \rightarrow t$ simply by $s \Rightarrow t$. Similarly, the subpath $g_0 \Rightarrow p \rightarrow t$ is denoted by $g_0 \Rightarrow t$. Moreover, when we use these rules to compute $F_2(\langle g, t\rangle)$, we must update *PathSet* whenever a path has been completed.

3.3 Algorithm Description

Now we can describe formally the *PR-OTIS* algorithm for constructing k node-disjoint paths from the source node $\langle g_0, s = t_0\rangle$ to the destination node set $T = \{\langle g_1, t_1\rangle, \langle g_2, t_2\rangle, \ldots, \langle g_k, t_k\rangle\}$ in OTIS-Ω (see Fig. 6). The algorithm consists of two main steps. First, we compute functions F_2 and F_1 by calling the procedure *Routing-Function*, where we construct all subpaths in the source cluster and the destination clusters, and thus obtain all intermediate clusters (if a path has no intermediate cluster, then it is constructed completely). Then, based on F_2 and F_1, we construct desired paths for all $\langle g, t\rangle \in T$ such that $F_1(\langle g, t\rangle) \neq \Lambda$ or $F_2(\langle g, t\rangle) \neq \Lambda$. Recall that, for $\langle g, t\rangle \in T$, $F_1(\langle g, t\rangle) \neq \Lambda$ denotes the first intermediate cluster, implying that the subpath contained in cluster g_0 is $s \rightarrow F_1(\langle g, t\rangle)$. Similarly, $F_2(\langle g,$

$t\rangle) \neq \Lambda$ denotes the second intermediate cluster, meaning that the subpath contained in cluster g is $F_2(\langle g, t\rangle) \rightarrow t$. In addition, $F_1(\langle g, t\rangle) \neq \Lambda$ and $F_2(\langle g, t\rangle) = \Lambda$ (in this case we have $g \neq g_0$) is indicative of the path not having a second intermediate cluster. In this case, according to the methods for computing F_1 and F_2, if cluster g contains only one destination node, the subpath contained in cluster g is $F_1(\langle g, t\rangle) \Rightarrow t$; otherwise node $\langle g, t\rangle$ is the head destination node of cluster g, and the subpath must be constructed carefully to ensure that all the subpaths in cluster g are node-disjoint.

Now assume that cluster g ($g \neq g_0$) contains more than one destination node, and let the head destination node of cluster g be $\langle g, \bar{t}\rangle$. Because the subpath contained in cluster g is $s \Rightarrow \bar{t}$ (refer to Case 2 of Step 2.2 in Procedure *Routing-Function*), $s^* = F_1(\langle g, \bar{t}\rangle) \in N_\Omega[s]$, and $F_2(\langle g, \bar{t}\rangle) = \Lambda$, we construct the subpath contained in cluster g as follows: $s^* \Rightarrow \bar{t}$ (a subpath of the path $s \Rightarrow \bar{t}$, without the link $s \rightarrow s^*$) if s^* is on the path $s \Rightarrow \bar{t}$ (in other words, the path $s \Rightarrow \bar{t}$ is the path $s \rightarrow s^* \Rightarrow \bar{t}$); otherwise, $s^* \rightarrow s \Rightarrow \bar{t}$. The subpath is node-disjoint from the other subpaths in cluster g, given that the method of computing F_1 ensures $s^* \notin G$, and all the nodes on the other subpaths in cluster g belong to G (due to the method for computing F_2).

Algorithm *PR-OTIS* {Node-to-set parallel routing in OTIS-Ω}

Input: A source node $\langle g_0, s = t_0\rangle$ and a set $T = \{\langle g_i, t_i\rangle \mid 1 \le i \le k\}$ of k destination nodes, where $k \le min\{deg_\Omega(t_i) \mid 0 \le i \le k\}$.

Output: A path set *PathSet* containing k parallel paths from $\langle g_0, s\rangle$ to $\langle g_i, t_i\rangle, 1 \le i \le k$.

1. Call procedure *Routing-Function* to compute the functions F_2 and F_1, and construct those paths that pass through no intermediate cluster.

2. Construct paths that pass through at least one intermediate cluster based on F_2 and F_1. For every node $\langle g, t\rangle \in T$ such that $F_1(\langle g, t\rangle) \ne \Lambda$ or $F_2(\langle g, t\rangle) \ne \Lambda$, add the following path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ to *PathSet*.

Case 1 $\{F_2(\langle g, t\rangle) = \Lambda$, $F_1(\langle g, t\rangle) = s^*$, and cluster g holds only one destination node$\}$:

$\langle g_0, s \to s^*\rangle \to \langle s^*, g_0 \Rightarrow g\rangle \to \langle g, s^* \Rightarrow t\rangle$

Case 2 $\{F_2(\langle g, t\rangle) = \Lambda$, $F_1(\langle g, t\rangle) = s^*$, and s^* is on the path $s \Rightarrow t\}$:

$\langle g_0, s \to s^*\rangle \to \langle s^*, g_0 \Rightarrow g\rangle \to \langle g, s^* \Rightarrow t\rangle$

Case 3 $\{F_2(\langle g, t\rangle) = \Lambda$, $F_1(\langle g, t\rangle) = s^*$, and s^* is not on the path $s \Rightarrow t\}$:

$\langle g_0, s \to s^*\rangle \to \langle s^*, g_0 \Rightarrow g\rangle \to \langle g, s^* \to s \Rightarrow t\rangle$

Case 4 $\{F_1(\langle g, t\rangle) = s^*$ and $F_2(\langle g, t\rangle) = t^*\}$:

$\langle g_0, s \to s^*\rangle \to \langle s^*, g_0 \Rightarrow t^*\rangle \to \langle t^*, s^* \Rightarrow g\rangle \to \langle g, t^* \to t\rangle$

Fig. 6. The algorithm *PR-OTIS* for constructing k parallel paths from source node $\langle g_0, s\rangle$ to k destination nodes $\langle g_i, t_i\rangle$, $1 \le i \le k$, in OTIS-Ω.

According to the discussion above, if Rules (R1), (R2), and (R3) can be used to remove every obstacle that possibly occurs, so as to obtain a valid value of F_2 for the corresponding destination node in Procedure *Routing-Function*, then F_2 and F_1 are a pair of routing functions we need since $F_2(\langle g, t\rangle)$ and $F_1(\langle g, t\rangle)$ are valid for every node $\langle g, t\rangle \in T$. This claim will be proven in Sec. 4.

4 Algorithm Analysis

In this section, we prove the correctness of Algorithm *PR-OTIS* and analyze its performance with regard to running time and the length of the resulting parallel paths. For this purpose, we first need to prove that Rules (R1), (R2), and (R3) can be used to remove obstacles that possibly occur in Procedure *Routing-Function*.

Assume that we are computing $F_2(\langle g, t\rangle)$ in the procedure *Routing-Function* for some $\langle g, t\rangle \in T$, $t \neq \bar{t}$, in a cluster g containing more than one destination node. Recall that $\langle g, \bar{t}\rangle$ is the head destination node of cluster g, p is the immediate predecessor of $\bar{t}$ on the subpath of $\langle g, \bar{t}\rangle$ contained in cluster g, G is the global forbidden set of $\langle g, t\rangle$, and $D_t = \{t' \mid \langle g, t'\rangle \in D\} - \{t\}$. Furthermore, we let $D_t^* = \{t' \mid t' \in D_t$ and $F_2(\langle g, t'\rangle)$ has not yet been computed$\}$. Then, we have $G \cup D_t \cup \{p\} = G \cup D_t^* \cup \{\bar{t}\} \cup \{p\}$ according to the method for computing F_2. Therefore, if $N_\Omega[t] - (G \cup D_t^* \cup \{\bar{t}\} \cup \{p\}) = \varnothing$, we are encountering an obstacle and need to prove that Rules (R1), (R2), and (R3) can be used to remove the obstacle. That is, we now need to prove the following crucial lemma.

Lemma 1. *If* $N_\Omega[t] - (G \cup D_t^* \cup \{\bar{t}\} \cup \{p\}) = \varnothing$, *then we can use Rule* (*R1*), (*R2*), *or* (*R3*) *to compute a valid* $F_2(\langle g, t\rangle)$.

The proof outline is as follows. First we show that when an obstacle occurs, there exists exactly one rule among Rules (R1)–(R3), and one operation of the rule, whose conditions and assumptions are met, concluding that the operation is feasible. Second, we prove that during the operation, the generated subpaths and the computed F_2 values thus far satisfy Criteria (C1), (C2) and (C3). These assertions are supported by the following two main facts that are obvious by construction or will be proved to be true in the full proof of Lemma 1.

- During the use of any one of the Rules (R1)-(R3), no new intermediate cluster is selected, and all the subpaths contained in cluster g are clearly node-disjoint by construction, implying Criteria (C2) and (C3) are satisfied.
- A subpath of length more than one in cluster g_0 is potentially constructed only if cluster g_0 is a destination cluster or Rule (R2) is used. On the other hand, if cluster g_0 also is a destination cluster, then Rule (R1) can be used to remove any possible obstacle, while if cluster g_0 is not a destination cluster, then Rule (R2) will be used at most once. Therefore, cluster g_0 contains at most one subpath of length more than one, which implies that Criterion (C1) is satisfied.

Please refer to the appendix for details of the proof of Lemma 1.

Based on Lemma 1, we establish the correctness of Algorithm *PR-OTIS* in the following theorem.

Theorem 2. *Given a source node* $\langle g_0, s = t_0\rangle$ *and a set* $T = \{\langle g_1, t_1\rangle, \langle g_2, t_2\rangle, \ldots, \langle g_k, t_k\rangle\}$ *of* k *destination nodes in OTIS-Ω such that* $\langle g_0, s\rangle \notin T$, *Algorithm*

PR-OTIS provides k node-disjoint paths from the source node to every destination node in T, provided $k \leq min\{\deg_{\Omega}(t_i) \mid 0 \leq i \leq k\}$.

Proof. We first need to show that all the operations in Procedure *Routing-Function* are feasible. For this purpose, we only need to prove that the operations in Steps 2.1 and 2.2 are feasible when an obstacle does occur in computing F_2; the other operations are obvious. The assertion is true by Lemma 1. Therefore, the algorithm generates k paths from the source node to every destination node in T. Moreover, by Lemma 1, F_2 and F_1 are a pair of routing functions, as needed, since $F_2(\langle g, t\rangle)$ and $F_1(\langle g, t\rangle)$ are valid for every node $\langle g, t\rangle \in T$, as discussed previously. Hence, the node-disjoint property of these k paths is ensured, completing the proof. ■

Next, we analyze the performance of Algorithm *PR-OTIS* with regard to running time and the lengths of constructed paths. In the following, we denote by n the order of Ω, by $f(n)$ the time complexity of the given shortest-path routing in Ω, by N the order of OTIS-Ω, by D the diameter of OTIS-Ω, and by Δ the maximal node degree of OTIS-Ω.

Theorem 3. *The time complexity of Algorithm PR-OTIS* is $O(\Delta^2+\Delta f(\sqrt{N}))$ *and each path constructed by it is of length no greater than* $D + 4$.

Proof. We first consider the time complexity of this algorithm. Recall that $|N_{\Omega}[t]|$

$\leq \Delta$ for all $t \in V(\Omega)$, and $k < \Delta$. In Procedure *Routing-Function*, for every destination node $\langle g, t\rangle \in D$, computing $F_2(\langle g, t\rangle)$ or $F_1(\langle g, t\rangle)$ needs $O(\Delta)$ time; thus, the total time for computing the routing functions F_2 and F_1 is $O(k\Delta)$, since there are k destination nodes. Moreover, constructing the paths passing through no intermediate cluster in Procedure *Routing-Function*, and constructing the other paths based on F_2 and F_1 take $O(kf(n))$ time, since k paths need to be constructed, and each one requires $O(f(n))$ time to construct. Hence, the time complexity of Algorithm *PR-OTIS* is bounded by $O(\Delta^2 + \Delta f(\sqrt{N}))$ due to $k < \Delta$ and $N = n^2$.

Next, we consider the length of the paths constructed. Note that the length of the path $u \Rightarrow v$ is at most $D(\Omega)$ for any pair of nodes u and v in Ω. Since every path is explicitly constructed in this algorithm, it is easy to verify immediately that the length of a path passing through no intermediate cluster is at most $2D(\Omega) + 1$, and the length of a path passing through one or two intermediate clusters is at most $2D(\Omega) + 5$. Therefore, the length of paths constructed is at most $2D(\Omega) + 5 = D + 4$, given that $D = 2D(\Omega) + 1$. ■

We note that if $\Delta = O(f(n))$ and the given shortest-path routing algorithm in Ω is time-optimal, then the running time of Algorithm *PR-OTIS* will be bounded by $O(\Delta f(n))$, which is optimal in the sense that any node-to-set routing algorithm takes at least $\Theta(\Delta f(n))$ time in the worst case to construct $k = \Delta - 1$ node-disjoint paths in the OTIS network. For a number of OTIS networks of practical interest, such as OTIS-Mesh and OTIS-Hypercube, the factor networks do have an optimal shortest-path routing algorithm and $\Delta = O(f(n))$ does hold. Thus, Algorithm *PR-OTIS* is optimal for such OTIS networks.

On the other hand, a path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ constructed by Algorithm *PR-OTIS* is clearly optimal if its length is the distance between the source node and the destination node. From the distance formula between two nodes in OTIS-Ω in Sec. 2, we know that there exists a shortest path passing through at most one intermediate cluster between two nodes in OTIS-Ω. We can verify directly the following results about near optimality in path length for some of the k node-disjoint paths constructed by Algorithm *PR-OTIS*.

- For $\langle g, t\rangle \in T$ such that $g = g_0$, if the path constructed from $\langle g_0, s\rangle$ to $\langle g_0, t\rangle$ has no intermediate cluster, then the length of the path is $d_{\text{OTIS-}\Omega}(\langle g_0, s\rangle, \langle g_0, t\rangle) = d_\Omega(s, t)$.
- For $\langle g, t\rangle \in T$ such that $g_0 \neq g$, if there exists one shortest path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ that passes through no intermediate cluster, and the constructed path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ has no intermediate cluster, then the length of the path is $d_{\text{OTIS-}\Omega}(\langle g_0, s\rangle, \langle g, t\rangle)$.
- For $\langle g, t\rangle \in T$ such that $g_0 \neq g$, if there exists one shortest path from $\langle g_0, s\rangle$ to

$\langle g, t\rangle$ that passes through one intermediate cluster, and the constructed path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ has only one intermediate cluster, then the length of the path is at most $d_{\text{OTIS-}\Omega}(\langle g_0, s\rangle, \langle g, t\rangle) + 2$.

- For $\langle g, t\rangle \in T$ such that $g_0 \neq g$, if there exists one shortest path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ that passes through no intermediate cluster, and the constructed path from $\langle g_0, s\rangle$ to $\langle g, t\rangle$ has two intermediate clusters, then the length of the path is at most $d_{\text{OTIS-}\Omega}(\langle g_0, s\rangle, \langle g, t\rangle) + 6$.

The *Rabin number* of a k-connected graph Γ is the minimum l such that for any $k + 1$ distinct nodes $s, t_1, t_2, \ldots, t_k$ of Γ, there exist k node-disjoint paths of length at most l from s to $t_1, t_2, \ldots, t_k$, respectively [26]. It has been shown that finding the Rabin number of a general graph is NP-hard. From Theorem 3, we immediately have the following corollary.

Corollary 4. *The Rabin number of OTIS-*Ω *is bounded from above by* $D + 4$.

5 Example Applications

In this section, we provide two example applications of Algorithm *PR-OTIS*. As the first application, we show that this algorithm can be used to solve efficiently the node-to-node parallel routing problem in OTIS networks.

Corollary 5. *Given a shortest-path routing algorithm of time complexity* $O(f(n))$

in an n-node connected factor network Ω, *Algorithm PR-OTIS can generate k node-disjoint paths of length at most* $D + 5$ *between two distinct nodes* $\langle g_s, s\rangle$ *and* $\langle g_t, t\rangle$ *in OTIS-*Ω *in* $O(\Delta^2+\Delta f(\sqrt{N}))$ *time, where* D, Δ, *and* N *are the diameter, maximal node degree, and order of OTIS-*Ω*, respectively, and* $k = \min\{deg_\Omega(s), \deg_\Omega(t)\}$.

Proof. Without loss of generality, we assume $|\deg_\Omega(s)| \geq |\deg_\Omega(t)| = k$, which implies that $|N_\Omega[s]| \geq |N_\Omega[t]| = k + 1$. We consider the following instance of the node-to-set parallel routing problem in OTIS-Ω: The source node is $\langle g_s, s\rangle$ and the destination node set is $\{\langle g_t, t_1\rangle, \langle g_t, t_2\rangle, \ldots, \langle g_t, t_k\rangle\}$, where $\{t_1, t_2, \ldots, t_k\}$ is a k-element subset of $N_\Omega[t]$ that includes t. By Theorems 2 and 3, Algorithm *PR-OTIS* can construct k node-disjoint paths from $\langle g_s, s\rangle$ to $\langle g_t, t_i\rangle$, $1 \leq i \leq k$, in $O(\Delta^2 + \Delta f(\sqrt{N}))$ time, and the length of each constructed path is at most $D + 4$. It is readily seen that these k node-disjoint paths can be extended by one step, if required, to obtain k node-disjoint paths of length at most $D + 5$ from $\langle g_s, s\rangle$ to $\langle g_t, t\rangle$. ■

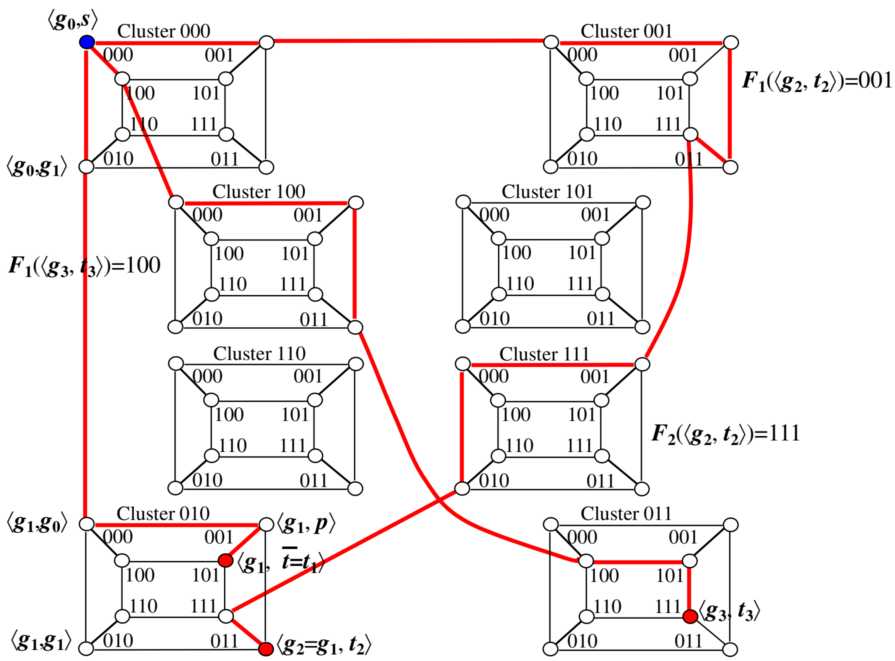
As a second application, we investigate the performance of Algorithm *PR-OTIS* when applied to OTIS networks built of binary hypercubes as factor networks. Hypercube networks and their variants, including OTIS-Hypercube, have been extensively studied.[28,31] We use Q_k to denote a k-dimensional hypercube network

(k-cube, for short) and let $n = 2^k$ denote its order. Thus, we have $k = \delta(Q_k) = \Delta(Q_k) = \log_2 n$. It is well-known that Q_k has a shortest routing algorithm of time complexity $O(\log n)$. Theorems 2 and 3 immediately yield the following result.

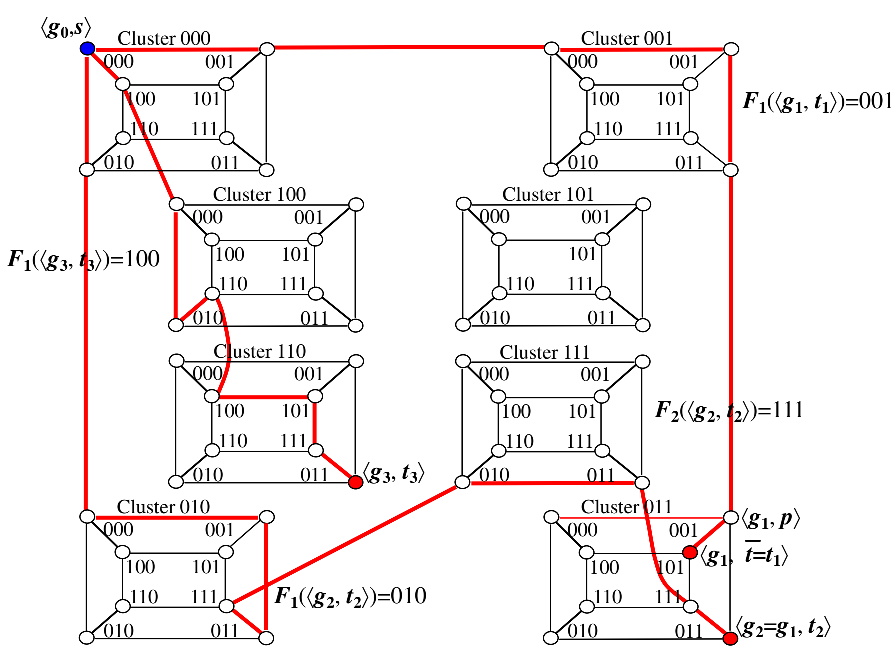
Corollary 6. *Given a source node* $\langle g_0, s\rangle$ *and k destination nodes* $\{\langle g_1, t_1\rangle, \langle g_2, t_2\rangle, \ldots, \langle g_k, t_k\rangle\}$ *in OTIS-*Q_k*, Algorithm PR-OTIS generates k node-disjoint paths of length at most* $D + 4$ *from the source node to every destination node in* $O(\log^2 N)$ *time* (*an optimal time*), where $N = 2^{2k}$ *and* $D = 2k + 1$ *are the order and the diameter of OTIS-*Q_k.

Note that the optimality of Algorithm *PR-OTIS* stated in Corollary 6 results from the fact that the worst case running time of any algorithm for constructing k paths in OTIS-Q_k is $\Theta(k \log n)$, which translates to $\Theta(\log^2 N)$, given that $k = \log_2 n$ and $N = n^2$.

Figure 7 illustrates the constructions by Algorithm *PR-OTIS* on two instances of the node-to-set parallel routing problem for *OTIS-*Q_3.



(a) An instance of $k = 3$ such that g_1 is a free node in $N_\Omega[s]$, where $F_1(\langle g_1, t_1\rangle) = F_2(\langle g_1, t_1\rangle) = □$.



(b) An instance of $k = 3$ such that no destination cluster number is a free node number in $N_\Omega[s]$.

Fig. 7. The parallel path constructions of Algorithm *PR-OTIS* (heavy lines) on two instances of OTIS-Q_3.

To evaluate the practical performance with regard to the length of paths constructed by Algorithm *PR-OTIS* for OTIS-Q_k, we conducted a simulation experiment, where for each k between 2 and 20, we selected 10,000 random combinations of the source node and the destination node set, applying the algorithm in each case to determine the mean maximum path length. Figure 8 depicts the mean maximum path length derived from these experiments, where the horizontal axis represents the dimension k of the factor network Q_k, and the vertical axis represents the average among 10,000 maximum lengths for each k.

From Fig. 8, we can conclude that for OTIS-Q_k, in practice, the maximum length of k node-disjoint paths from one source node to k destination nodes constructed by Algorithm *PR-OTIS* is generally far below the upper bound $D + 4$, and slightly above the maximal path length that results directly from the distance formula, the latter clearly providing a lower bound for the maximum length. Note that, for any specific instance of the node-to-set routing problem, the maximum length of the required paths constructed by any algorithm would generally be somewhat greater than the maximal distance from the source node to the set of destination nodes, in order to ensure the node-disjoint property.

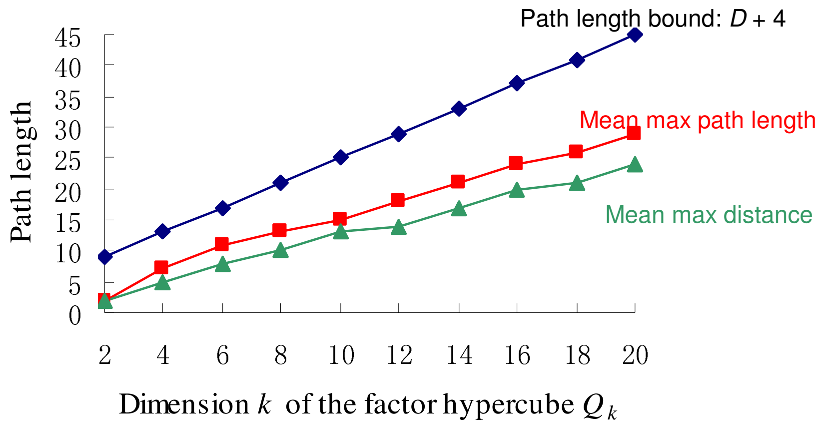


Fig. 8. Average maximum length of paths constructed by Algorithm *PR-OTIS* for OTIS-Q_k.

6 Conclusion

In this paper, we have proposed a general and efficient algorithm for the node-to-set parallel routing problem in an OTIS network built of an arbitrary connected factor network. If D, Δ, and N represent the diameter, node degree, and order of the OTIS network, the proposed algorithm constructs the requisite number of node-disjoint paths that are of length at most $D + 4$ in $O(\Delta^2 + \Delta f(\sqrt{N}))$ time, provided the factor network of order n has a shortest-path routing algorithm of time complexity $O(f(n))$. We have shown that the algorithm is optimal in terms of time complexity for a number of OTIS networks of practical interest.

Our general algorithm supersedes prior node-to-set parallel routing schemes for specific factor networks, and it can also be used for node-to-node parallel routing in OTIS networks. As an example application, the algorithm provides node-disjoint paths (that are only slightly longer than the distance-based optimal paths) in optimal time for OTIS-Hypercube.

Important open problems include finding general and efficient algorithms for the set-to-set and the k-pair parallel routing problems in OTIS networks.

Acknowledgment

This research of W. D. Chen and W. J. Xiao was supported by the National Natural Science Foundation of China (Grant Nos. 60973150 and 61170313) and the Natural Science Foundation of Guangdong Province, China (Grant No. 10451063101006313).

Appendix: Proof of Lemma 1

Please refer to Section 4 of the paper for the proof outline.

In order to prove Lemma 1, we first prove the following sequence of assertions.

Assertion 1. $$|G \cup D_t^* \cup \{\bar{t}\} \cup \{p\}| \leq \begin{cases} k+1, & \text{if the source cluster } g_0 \text{ is a destination cluster} \\ k+2, & \text{otherwise} \end{cases}$$

Furthermore, if the maximum of $k + 1$ or $k + 2$ is attained, then the four sets G, D_t^*, $\{\bar{t}\}$ and $\{p\}$ will be pairwise disjoint, and exactly one destination node in every destination cluster will not need a second intermediate cluster.

Proof. According to the method for computing F_2, if a destination cluster g' contains c destination nodes ($c \geq 1$), we need to make use of a second intermediate cluster for no more than $c - 1$ of the destinations. Thus, the total contribution of cluster g' to $|G|$ will be at most c, given that G initially contains g'. In particular, if cluster g_0 is also a destination cluster, then the contribution of cluster g_0 to $|G|$ will be no greater than the number of destinations in cluster g_0; otherwise, its contribution will be 1, owing to $g_0 \in G$. Therefore, after computing $F_2(\langle g, t\rangle)$ for all $\langle g, t\rangle \in T$, $|G|$ will be at most k if cluster g_0 is a destination cluster and at most $k + 1$, otherwise. Now, assume that we are computing $F_2(\langle g, t\rangle)$ for $\langle g, t\rangle \in T$, and that there are a destination nodes including $\langle g, t\rangle$ in cluster g whose F_2 values have not yet been computed; i.e., $|D_t^*| = a - 1$ ($a \geq 1$) due to $t \notin D_t^*$. At the moment, $|G|$ is at most $k - a$ if cluster g_0 is a destination cluster and at most $k + 1 - a$, otherwise. Hence, $|G \cup D_t^* \cup \{\bar{t}\} \cup \{p\}|$, which is at most $|G| + |D_t^*| + |\{\bar{t}\}| + |\{p\}|$, will be no greater than $k + 1$ if cluster g_0 is a destination cluster and no greater than $k + 2$, otherwise. Clearly, if the maximum of $k + 1$ or $k + 2$ is attained, then the sets G, D_t^*, $\{\bar{t}\}$, and $\{p\}$ must be pairwise disjoint, and exactly one destination node in every destination cluster will not need a second intermediate cluster. ■

Assertion 2. Let the source cluster g_0 also be a destination cluster. If $N_\Omega[t] - (G \cup D_t^* \cup \{\bar{t}\} \cup \{p\}) = \varnothing$, then $p \neq \bar{t}$, $p \in N_\Omega[t]$, and $\bar{t} \notin G$.

Proof. From Assertion 1, we have $|G \cup D_t^* \cup \{\bar{t}\} \cup \{p\}| \leq k + 1$, and that if the maximum of $k + 1$ is attained, then the sets G, D_t^*, $\{\bar{t}\}$ and $\{p\}$ are pairwise disjoint. Since $|N_\Omega[t]| \geq k + 1$, $N_\Omega[t] - (G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}) = \varnothing$ implies $N_\Omega[t] = G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}$, and the sets G, D_t^*, $\{\ \bar{t}\ \}$, and $\{p\}$ are pairwise disjoint. Hence, we have $p \neq \bar{t}$, $p \in N_\Omega[t]$, and $\bar{t} \notin G$. ■

Assertion 3. Let the subpath of $\langle g, \bar{t}\rangle$ contained in cluster g be of form $g_0 \Rightarrow \bar{t}$. If $N_\Omega[t] - (G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}) = \varnothing$, then $p \neq \bar{t}$, $p \in N_\Omega[t]$, and $\bar{t} \notin G$.

Proof. Recall that $|N_\Omega[t]| \geq k + 1$ and $|G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}\ | \leq k + 2$ by Assertion 1. We consider the two cases of $g_0 \in N_\Omega[t]$ and $g_0 \notin N_\Omega[t]$. First, if $g_0 \notin N_\Omega[t]$, then $N_\Omega[t] - (G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}) = \varnothing$ implies $|G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}| = k + 2$, and $N_\Omega[t] \cup \{g_0\} = G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}$ due to $g_0 \in G$. By Assertion 1, the sets G, D_t^*, $\{\ \bar{t}\ \}$, and $\{p\}$ are pairwise disjoint. Thus, we have $p \neq \bar{t}$, $p \in N_\Omega[t]$, and $\bar{t} \notin G$. Secondly, if $g_0 \in N_\Omega[t]$, then $d_\Omega(g_0, \bar{t}) \leq d_\Omega(g_0, t) \leq 1$ due to $d_\Omega(g_0, \bar{t}) = min\{d_\Omega(g_0, t') \mid \langle g, t'\rangle \in T\ \}$. This leads to $p = g_0$, since p is the immediate predecessor of $\bar{t}$ on the path $g_0 \Rightarrow \bar{t}$. The proof of Assertion 1, and $p = g_0 \in G$, $N_\Omega[t] - (G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}) = \varnothing$ suggest that $N_\Omega[t] = G \cup D_t^* \cup \{\ \bar{t}\ \}$, and the sets G, D_t^*, and $\{\ \bar{t}\ \}$ are pairwise disjoint. Thus, we have $p = g_0 \neq \bar{t}$, $p = g_0 \in N_\Omega[t]$, and $\bar{t} \notin G$, which concludes this proof. ■

Assertion 4. If $p \neq \bar{t}$, $p \in N_\Omega[t]$, and $\bar{t} \notin G$, then Rule (R1) can be used to compute a valid $F_2(\langle g, t\rangle)$.

Proof. Without loss of generality, we assume that the subpath of $\langle g, \bar{t}\rangle$ contained in cluster g is $s \Rightarrow \overline{t}$; the proof for the case of the subpath being $g_0 \Rightarrow \overline{t}$ is similar. Recall that the role of node p in the forbidden set is to ensure that the subpath $s \Rightarrow \overline{t}$ is node-disjoint from the other subpaths contained in cluster g, since $d_\Omega(s, \overline{t}) = min\{d_\Omega(s, t') \mid \langle g, t' \rangle \in T\}$. Considering that $p \neq \overline{t}$ and $p \in N_\Omega[t]$, we have $d_\Omega(s, t) = d_\Omega(s, \overline{t})$. Thus, to compute F_2 for other destination nodes in cluster g, we can replace $\overline{t}$ with t. In other words, we make $\langle g, t \rangle$ the head destination node in cluster g, in lieu of $\langle g, \overline{t} \rangle$. Then, for $\langle g, t \rangle$, the subpath contained in cluster g is $s \Rightarrow p \rightarrow t$ and $F_2(\langle g, t \rangle) = \Lambda$, For $\langle g, \overline{t} \rangle$, the subpath contained in cluster g is the single node $\overline{t}$, and $F_2(\langle g, \overline{t} \rangle) = \overline{t}$ due to $\overline{t} \notin G$. Hence, we can use Rule (R1) to compute a valid $F_2(\langle g, t \rangle)$. ■

Based on Assertions 1-4, we now can prove Lemma 1.

Lemma 1. *If* $N_\Omega[t] - (G \cup D_t^* \cup \{ \overline{t} \} \cup \{p\}) = \varnothing$, *then we can use Rule* (*R1*), (*R2*), *or* (*R3*) *to compute a valid* $F_2(\langle g, t \rangle)$.

Proof. Let $\alpha = (p \neq \overline{t} \wedge p \in N_\Omega[t] \wedge \overline{t} \notin G)$. If α is true, we can use Rule (R1) to compute a valid $F_2(\langle g, t \rangle)$ by Assertion 4. Next, we assume that α is false, that is, $p = \overline{t}$, $p \notin N_\Omega[t]$, or $\overline{t} \in G$. Note that $|N_\Omega[t]| \geq k + 1$ and that $N_\Omega[t] - (G \cup D_t^* \cup \{ \overline{t} \} \cup \{p\}) = \varnothing$ is given. Thus, if $p = \overline{t}$ or $p \notin N_\Omega[t]$, then $N_\Omega[t] = G \cup D_t^* \cup \{ \overline{t} \}$, and the sets G, D_t^*, and $\{ \overline{t} \}$ will be pairwise disjoint by the proof of Assertion 1. Similarly, if $\overline{t} \in G$, then $N_\Omega[t] = G \cup D_t^* \cup \{p\}$, and the sets G, D_t^*, and $\{p\}$ will be pairwise disjoint. Hence, in either case, $\bar{t} \in N_\Omega[t]$, $G \cup D_t^* \subseteq N_\Omega[t]$, and the sets G and D_t^* are disjoint. Note that because α is false, cluster g_0 is not a destination cluster by Assertion 2 and the subpath of $\langle g, \bar{t}\rangle$ contained in cluster g is $s \Rightarrow \bar{t}$ by Assertion 3; thus, $g \notin N_\Omega[s]$ by the method for computing F_2. Let g^* be the immediate successor to s on the path $s \Rightarrow g$. Clearly, $g^* \in N_\Omega[s]$, and we have the following two cases.

Case 1. $\{g^*$ is a free node in $N_\Omega[s]\}$: Recall that $D_t = \{t' \mid \langle g, t'\rangle \in T\} - \{t\}$, $D_t^* = \{t' \mid t' \in D_t$ and $F_2(\langle g, t'\rangle)$ has not yet been computed$\}$, and $g_0 \in N_\Omega[t]$ due to $g_0 \in G \subseteq N_\Omega[t]$. We have the following two subcases.

Case 1.1. $\{g_0 \notin D_t\}$: Clearly, Case 1 of Rule (R2) can be used to compute $F_2(\langle g, t\rangle)$.

Case 1.2. $\{g_0 = t' \in D_t\}$: In this case, $F_2(\langle g, t'\rangle)$ has been computed, since we know that G and D_t^* are disjoint, and $g_0 \in G$. Note that at the moment, $\bar{t}$ is the only destination node in cluster g not needing a second intermediate cluster, that is, $F_2(\langle g, \bar{t}\rangle) = \Lambda$. Therefore, if $F_2(\langle g, t'\rangle) \neq \Lambda$, then $t' \neq \bar{t}$, and $F_2(\langle g, t'\rangle) \in G \subseteq N_\Omega[t]$; thus Case 2 of Rule (R2) can be used to compute $F_2(\langle g, t\rangle)$. Otherwise, that is, for $F_2(\langle g, t'\rangle) = \Lambda$, we have $t' = \bar{t} = g_0 \in G$, leading to $N_\Omega[t] = G \cup D_t^* \cup \{p\}$, and the sets G, D_t^*, and $\{p\}$ are pairwise disjoint, as discussed above. Hence, we have $g_0 = \bar{t}$, $p \in N_\Omega[t]$, and $p \neq \bar{t}$, which suggests that Case 3 of Rule (R2) can be used to compute $F_2(\langle g, t\rangle)$.

Case 2. $\{g^*$ is a fixed node in $N_\Omega[s]\}$: According to the method for computing F_2, we know that cluster g^* contains more than one destination node, and F_2 has been computed for every destination node in cluster g^*(see Case 1 of Step 2.2 in Procedure *Routing-Function*). In this case, there exists a destination node $\langle g^*, t^*\rangle$ such that $F_2(\langle g^*, t^*\rangle) \neq \Lambda$. Hence, we have $F_2(\langle g^*, t^*\rangle) \in G$, and thus $F_2(\langle g^*, t^*\rangle) \in N_\Omega[t]$, since $G \cup D_t^* \subseteq N_\Omega[t]$. We have the following two subcases.

Case 2.1. $\{F_2(\langle g^*, t^*\rangle) \notin D_t\}$: Clearly, Case 1 of Rule (R3) can be used to compute $F_2(\langle g, t\rangle)$.

Case 2.2. $\{F_2(\langle g^*, t^*\rangle) = t' \in D_t\}$: In this case, $F_2(\langle g, t'\rangle)$ has been computed, since we know that G and D_t^* are disjoint, and $t' = F_2(\langle g^*, t^*\rangle) \in G$. Note that at the moment, $\bar{t}$ is the only destination node in cluster g not needing a second intermediate cluster, that is, $F_2(\langle g, \bar{t}\rangle) = \Lambda$. Therefore, if $F_2(\langle g, t'\rangle) \neq \Lambda$, then $t' \neq \bar{t}$ and $F_2(\langle g, t'\rangle) \in G \subseteq N_\Omega[t]$; thus Case 2 of Rule (R3) can be used to compute $F_2(\langle g, t\rangle)$. Otherwise, that is, for $F_2(\langle g, t'\rangle) = \Lambda$, we have $t' = \bar{t} = F_2(\langle g^*, t^*\rangle) \in G$, and thus $N_\Omega[t] = G \cup D_t^* \cup \{p\}$, and the sets G, D_t^*, and $\{p\}$ are pairwise disjoint, as discussed above. Hence, we have $F_2(\langle g^*, t^*\rangle) = \bar{t}$, $p \in N_\Omega[t]$, and $p \neq \bar{t}$, which suggests that Case 3 of Rule (R3) can be used to compute $F_2(\langle g, t\rangle)$.

Note that the subpath of the head destination node in cluster g^* is of form $g_0 \Rightarrow \vec{t}^*$, where $\vec{t}^* = argmin\{d_\Omega(g_0, t') \mid \langle g^*, t'\rangle \in T\}$, implying that it is feasible to recompute a valid $F_2(\langle g^*, t^*\rangle)$ via Rule (R1) by Assertions 3 and 4.

We next prove the validity of $F_2(\langle g, t\rangle)$ obtained via Rule (R2) or (R3). Since no new intermediate cluster is selected, and all the subpaths contained in cluster g are clearly node-disjoint by construction, Criteria (C2) and (C3) are satisfied. In order to prove that Criterion (C1) is also satisfied, considering that g^* is a free node in $N_\Omega[s]$, we only need to prove that before we use Rule (R2) to compute $F_2(\langle g, t\rangle)$, all constructed subpaths in cluster g_0 are of length at most one, which suggests all the nodes on these subpaths are within $N_\Omega[s]$ of cluster g_0. Note that, according to the method for computing F_2, a subpath of length more than one in cluster g_0 is potentially constructed only if cluster g_0 is a destination cluster or Rule (R2) is used. Now that we have proven that cluster g_0 is not a destination cluster, we only need to prove that Rule (R2) can be used at most once. Note that once Rule (R2) has been used in some cluster, at least two destination nodes in that cluster will not need a second intermediate cluster, which implies, by the proof of Assertion 1, that $|G \cup D_t^* \cup \{\ \bar{t}\ \} \cup \{p\}| \le k + 1$ will hold and that if the maximum of $k + 1$ is attained, then the sets G, D_t^*, $\{\ \bar{t}\ \}$, and $\{p\}$ will be pairwise disjoint. In this case, similar to Assertion 2, we can easily prove that α is true if $N_\Omega[t] - (G \cup D_t^* \cup \{\ \bar{t}\}$

$\cup \{p\}) = \varnothing$; thus Rule (R1) can remove any obstacle that might possibly occur later, which implies that Rule (R2) can be used at most once, as claimed. ■

References

[1] G. Marsden, P. Marchand, P. Harvey, and S. Esener, "Optical transpose interconnection system architecture," *Optical Letters*, **18** (1993), 1083-1085.

[2] F. Zane, P. Marchand, R. Paturi, and S. Esener, "Scalable network architectures using the optical transpose interconnection system (OTIS)," *J. Parallel and Distributed Computing*, **60** (5) (2000), 521–538.

[3] C.-H. Yeh, and B. Parhami, "Swapped networks: unifying the architectures and algorithms of a wide class of hierarchical parallel processors," Proc. Int'l Conf. Parallel and Distributed Systems (1996), pp. 230–237.

[4] J. Rayn, "WDM: North American development trend," *IEEE Communications*, **32** (2) (1998), 40-44.

[5] S. Chatterjee, and S. D. Pawlowski, "All-optical networks," *Communications of the ACM*, **42** (6) (1999), 74-83.

[6] S. Rajasekaran, and S. Sahni, "Randomized routing, selection, and sorting on the OTIS-mesh," *IEEE Trans. Parallel and Distributed Systems*, **9** (9) (1998), 833-840.

[7] C.-F. Wang, and S. Sahni, "Basic operations on the OTIS-mesh optoelectronic computer," *IEEE Trans. Parallel and Distributed Systems*, **9** (12) (1998), 1226-1236.

[8] C.-F. Wang, and S. Sahni, "Matrix multiplication on the OTIS-mesh optoelectronic computer," *IEEE Trans. Computers*, **50** (7) (2001), 635–646.

[9] P. K. Jana, "Polynomial interpolation and polynomial root finding on OTIS-mesh," *Parallel Computing*, **32** (2006), 301–312.

[10] C.-F. Wang, and S. Sahni, "Image processing on the OTIS-mesh optoelectronic computer," *IEEE Trans. Parallel and Distributed Systems*, **11** (2) (2000), 97–109.

[11] D. F. Hsu ,and Y. D. Lyuu, "A graph-theoretical study of transmission delay and fault tolerance," *Int'l J. Mini and Microcomputers*, **16** (1) (1994), 35–42.

[12] C.-N. Lai, *One-to-Many Disjoint Paths in the Hypercube and Folded Hypercube*, (PhD Dissertation, National Taiwan University, June 2001).

[13] C.-C. Chen, and J. Chen, "Nearly optimal one-to-many parallel routing in star networks," *IEEE Trans. Parallel and Distributed Systems*, **8** (12) (1997), 1196–1202.

[14] J. Kim, S. Moh, I. Chung, and C. Yu, "Robust multipath routing to exploit maximally disjoint paths for wireless ad hoc networks," Proc. Int'l Workshop Sensor Networks, LNCS Vol. 3842 (Springer, 2006), pp.306-309.

[15] D. B. West, *Introduction to Graph Theory*, (Prentice-Hall, 2001).

[16] T. Takabatake, and T. Nakamigawa, "Node-disjoint paths in a level block of generalized hierarchical completely connected networks," *Theoretical Computer Science*, **465** (2012), 28-34.

[17] R. M. Karp. "On the computational complexity of combinatorial problems," *Networks*, **5** (1975), 45-68.

[18] J. Kleinberg, *Approximation Algorithms for Disjoint Paths Problems*, (PhD Dissertation, MIT, May 1996).

[19] Q. Gu, and S. Peng, "An efficient algorithm for the *k*-pairwise disjoint paths problem in hypercubes," *J. Parallel and Distributed Computing*, **60** (2000), 764 -774.

[20] S. Even, and R. E. Tarjan, "Network flow and testing graph connectivity," *SIAM J. Computing*, **4** (4) (1975), 507-518.

[21] Q. Gu, and S. Peng, "Node-to-set disjoint paths problem in star graphs," *Information Processing Letters*, **62** (4) (1997), 201-207.

[22] K. Kaneko, "An algorithm for node-to-set disjoint paths problem in bi-rotator graphs," *IEICE Trans. Information and Systems*, **E89-D** (2) (2006), 647-653.

[23] R.-Y. Wu, et al., "Node-disjoint paths in hierarchical hypercube networks," *Information Sciences*, **177** (2007), 4200-4207.

[24] S. Zhou, W. Xiao, and B. Parhami, "Construction of vertex-disjoint paths in alternating group networks," *The Journal of Supercomputing*, **54** (2) (2010), 206–228.

[25] L. Lipták, E. Cheng, J.-S. Kim, S. W. Kim, "One-to-many node-disjoint paths of hyper-star networks," *Discrete Applied Mathematics*, **160** (13–14) (2012), 2006-2014.

[26] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *J. ACM*, **36** (2) (1989), 335-348.

[27] K. Day, and A. Al-Ayyoub, "Topological properties of OTIS-networks," *IEEE Trans. Parallel and Distributed Systems*, **14** (4) (2002), 359-366.

[28] K. Day, "Optical transpose *k*-ary *n*-cube networks," *J. Systems Architecture*, **50** (2004), 697-705.

[29] B. Parhami, "Swapped interconnection networks: topological, performance, and robustness attributes," *J. Parallel and Distributed Computing*, **65** (2005), 1443-1452.

[30] W. Chen, W. Xiao, and B. Parhami, "Swapped (OTIS) networks built of connected basis networks are maximally fault tolerant," *IEEE Trans. Parallel and Distributed Systems*, **20** (3) (2009), 361-366.

[31] Y. Saad, and M. H. Schultz, "Topological properties of hypercubes," *IEEE Trans. Computers*, **37** (1988), 867–872.