Enhanced Cluster *k*-Ary *n*-Cube, A Fault-Tolerant Multiprocessor

Baback A. Izadi, *Member*, *IEEE*, and Füsun Özgüner, *Member*, *IEEE*

Abstract—In this paper, we present a strongly fault-tolerant design for the *k*-ary *n*-cube multiprocessor and examine its reconfigurability. Our design augments the *k*-ary *n*-cube with $(\frac{k}{j})^n$ spare nodes. Each set of j^n regular nodes is connected to a spare node and the spare nodes are interconnected as either a $(\frac{k}{j})$-ary *n*-cube if $j \neq \frac{k}{2}$ or a hypercube of dimension n if $j = \frac{k}{2}$. Our approach utilizes the capabilities of the wave-switching communication modules of the spare nodes to tolerate a large number of faulty nodes. Both theoretical and experimental results are examined. Compared with other proposed schemes, our approach can tolerate significantly more faulty nodes with a low overhead and no performance degradation.

Index Terms—Fault tolerance, *k*-ary *n*-cube, hypercube, spare allocation, reconfiguration, augmented multiprocessor, wave switching.

1 Introduction

A number of multiprocessors have been built using networks that are either *k*-ary *n*-cube or are isomorphic to one. Examples include meshes, tori, and hypercubes. Some of the architectures based on the mesh and the torus include *Paragon XP/S* and *DELTA* (*Intel*) [1], *J-Machine* (*MIT*) [2], and *MPP-T3E* (*Cray*) [3].

As the size of the *k*-ary *n*-cube multicomputer grows, due to its complexity, the probability of node failures becomes high. Therefore, it is crucial that such systems be able to withstand a large number of faults for a reasonable amount of time. To sustain the same level of performance, some researchers have investigated hardware schemes for the *k*-ary *n*-cube where spare nodes are used to replace the faulty ones. A system so reconfigured, however, may no longer be able to efficiently support the original task partition. Furthermore, it may have to use a slower adaptive router instead of continued use of a faster oblivious one. Therefore, a fault-tolerant design that retains the same service level in the presence of faults and does not require any alteration of the computation or communication algorithms is preferred. Most of such schemes in the literature do so by preserving the physical topology of the system and, in general, are referred to as strongly fault-tolerant schemes [4], [5]. The scheme in this paper retains the same capabilities by preserving the logical topology of the system. Therefore, we have expanded the definition of the strongly fault-tolerant system to preservation of either the physical or the logical topology of the system. Under the extended definition, a strongly fault-tolerant system must not require any modification of the communication or computation algorithms. Our scheme does not add any congestion penalty on the communication paths. However, it does impose a higher dilation, resulting in small additional internode communication delays. Therefore, the service level could be degraded slightly.

Two classes of hardware schemes have been proposed in the literature to tolerate faulty nodes. Some researchers have examined local reconfiguration techniques where a spare node can only replace a faulty node within a given subset [6], [7], [8], [9], [10], [11], [12], [13], [14]. The switch complexity of these schemes is similar to ours. However, they suffer from low utilization of spare nodes and, in general, are not strongly fault-tolerant.

The second class of approaches uses global reconfiguration schemes. One group of approaches is based on creating a supergraph of the target topology such that if a node and its associated links are removed from the supergraph, the remaining graph would be isomorphic to the target topology [15], [16], [17], [18], [19], [20]. These schemes are therefore strongly fault-tolerant. However, most of the graph-based schemes are only node-minimal and suffer from prohibitively large node degree. For example, the f-fault-tolerant scheme for n-dimensional mesh in [15], which tolerates f faulty nodes, requires a node degree of $(f+1)n$ and $(f+2)n$ for even and odd f, respectively. Another node-minimal approach for an f-fault-tolerant two-dimensional mesh [16] requires f rows of switches between every pair of processing elements rows. A non-node-minimal f-fault-tolerant two-dimensional mesh [19] has a node degree of 13, but requires $8f^3 + 16f^2$ spare nodes.

A second group of global reconfiguration schemes utilizes rows and columns of spare nodes to tolerate the faulty nodes [21], [22], [23], [24]. The main objective of row-column bypass schemes is to increase yield (defect tolerance) in integrated microelectronic systems. The main distinction among most of the proposed schemes is the complexity of the switch elements and the way they are implemented. To achieve a reasonable probability of

- *B.A. Izadi is with the Department of Electrical and Computer Engineering, State University of New York, 75 S. Manheim Blvd., New Paltz, NY 12561. E-mail: bai@engr.newpaltz.edu.*
- *F. Özgüner is with the Department of Electrical Engineering, Ohio State University, 2015 Neil Ave., Columbus, OH 43210-1277. E-mail: ozguner@ee.eng.ohio-state.edu.*

Manuscript received 11 July 2001; revised 19 Apr. 2002; accepted 30 Jan. 2003.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 114510.

survival, most of the schemes require a large number of spare rows or spare columns.

In this paper, we propose a global reconfiguration scheme that utilizes wave-switching communication [25] to make the k-ary n-cube multicomputer strongly fault-tolerant. Wave-switching implements circuit-switching and wormhole-switching concurrently; permanent connections and long messages use the circuit-switched segment, while short messages are transmitted using the wormhole-switching. In our scheme, one spare node is assigned to each group of regular nodes, called a cluster; each spare node is connected to every regular node of its cluster via an intracluster spare link. Furthermore, the spare nodes of neighboring clusters are interconnected using intercluster spare links; two clusters are declared neighbors if there exists at least one regular node in each with a direct link between them. Our approach differs from others in the way we link the spare nodes to the faulty regular nodes. Once a node becomes faulty, most schemes discard the faulty node along with the healthy links that are connected to it. Therefore, additional spare links are needed to connect the node that replaces the faulty node to the neighboring nodes of the faulty node. In our approach, we only need to replace the faulty component of the node and therefore can communicate with the neighboring nodes of the faulty node using the existing links. As a result, our scheme minimizes the required number of spare links, which is an important factor in the design of massive parallel machines [26]. To facilitate communication, we utilize the circuit-switched capabilities of various spare nodes' communication modules to construct dedicated paths between multiple faulty regular nodes and multiple spare nodes. We use this property to show that our scheme can globally assign faulty nodes to spare nodes and has a significant higher fault-tolerant capability than other approaches. Our approach is both practical and flexible such that, depending on the desired reliability of the system, one can design a fault-tolerant system with different numbers of spare nodes.

The rest of the paper is organized as follows: In the next section, notation and definitions that are used throughout the paper are given. An overview of our approach is presented in Section 3. Theoretical and simulation results are presented in Sections 4 and 5, respectively. Finally, concluding remarks are presented in Section 6.

2 Notation and Definitions

Each node of a k-ary n-cube is identified, in radix k, by *n-tuple* $(a_{n-1} \cdots a_i \cdots a_0)_k$, where a_i is a radix k digit and represents the node's position in the ith dimension. Each node is connected to its neighboring nodes $(a_{n-1} \cdots (a_i \pm 1 \; mod \; k) \cdots a_0)_k$, along the dimension i. Each spare node, in addition to n digits, is labeled with a prefix S i.e. $(Sa_{n-1} \cdots a_i \cdots a_0)_k$. The link connecting any two nodes P and Q is represented by $P \rightarrow Q$. A cluster whose local spare is labeled $Sa_{n-1} \cdots a_i \cdots a_0$ is called *cluster* $a_{n-1} \cdots a_i \cdots a_0$. The topology that interconnects the spare nodes is called the spare network. We define the *Connection Requirement* (C_R) of a spare node in a cluster with multiple faulty nodes as the number of edge-disjoint paths that must be constructed, within the spare network from that spare node to other spare nodes in the fault-free clusters, so that faulty nodes can be tolerated. Each of these edge-disjoint paths is referred to as a dedicated path from a faulty node to its assigned spare node. The graph theoretical notations used throughout the paper follow standard graph theory definitions as outlined in [27].

3 Overview of the Enhanced Cluster Approach

An enhanced cluster k-ary n-cube (ECKN), consisting of k^n regular nodes, is constructed by assigning one spare node to each cluster of j^m regular nodes; each spare node is connected to each of the j^m regular nodes of its cluster via an intracluster spare link. Hence, there exist $\frac{k^n}{j^m}$ spare nodes assigned to $\frac{k^n}{j^m}$ clusters. Moreover, the spare nodes are interconnected using intercluster spare links. To keep the interconnecting topology of the spare nodes similar to the one that interconnects the regular nodes, one can either design for $k = j$ $(0 \leq m \leq n)$ or $n = m$ $(1 \leq j \leq k$ and $k = c \times j$, where c is a constant); the resulting topology among the spare nodes would then either be a k-ary $(n-m)$-cube or a $\frac{k}{j}$-ary n-cube, respectively. For example, the topology among the spare nodes of an enhanced cluster 6-ary 4-cube with $j = k = 6$ and $m = 2$ is a 6-ary 2-cube, and the spare nodes of an enhanced cluster 6-ary 4-cube with $m = n = 4$ and $j = 2$ are interconnected as a 3-ary 4-cube. In Section 4, we show that, in our approach, a higher connectivity among spare nodes results in a higher fault tolerance. Therefore, we choose the $\frac{k}{j}$-ary n-cube as our spare network. As a result, in our scheme, one spare node is connected to each of the j^n regular nodes of every cluster via intracluster spare links. Moreover, the $(\frac{k}{j})^n$ spare nodes are interconnected as a $\frac{k}{j}$-ary n-cube using intercluster spare links. We next present two designs for the ECKN. In the first scheme, $j \neq \frac{k}{2}$ and, in the second one, $j = \frac{k}{2}$; the second case is a special case of the first where the spare network is a 2-ary n-cube (hypercube).

3.1 The ECKN with $j \neq \frac{k}{2}$

The resultant structure consists of k^n regular nodes and $(\frac{k}{j})^n$ spare nodes. The degree of each regular node and spare node is $2n + 1$ and $2n + j^n$, respectively. Naturally, a smaller j yields a smaller cluster, a smaller spare node degree, and a higher number of spare nodes, resulting in a higher fault tolerance. Fig. 1 depicts an enhanced cluster 6-ary 2-cube with $j = 2$. In the figure, each spare node is assigned to a cluster of 2^2 regular nodes and the spare nodes are connected as a 3-ary 2-cube.

We next describe how the ECKN tolerates faulty nodes. It is common to assume that faulty nodes retain their ability to communicate since, normally, the hardware complexity of the communication module is much lower than the computational module. Therefore, the probability of failure in the communication module is much lower than the computation module. However, depending on the value of j and n, the spare node degree could be large enough that such an assumption would not be appropriate. In such a case, the communication module of each node should be duplicated. This provides similar protection to the schemes where a spare node replaces the computation and communication modules

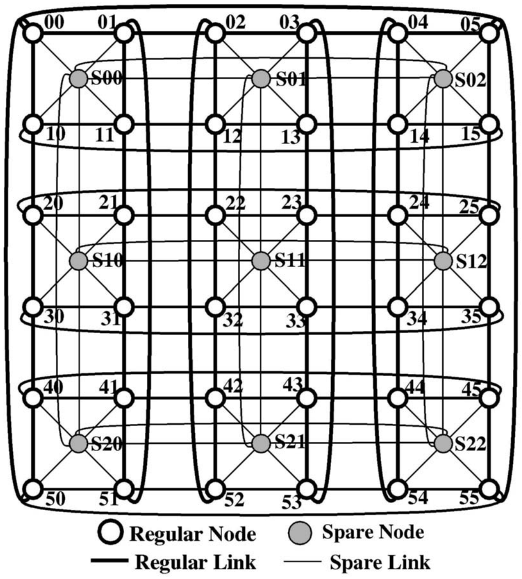

Fig. 1. An enhanced cluster 6-ary 2-cube with $j = 2$.

of the faulty node. The difference is that, in our scheme, the replacement computation and communication modules would be physically in different nodes. The downside of duplicating the communication module, aside from increased VLSI area, is the higher probability of stuck-at faults in the related links.

To tolerate a faulty node, the computation module of the spare node logically replaces the computation module of the faulty node. In addition, if the spare node resides in the cluster of the faulty node, the new communication module consists of the functional communication module of the faulty node logically merged with the appropriate routing channel of the local spare node. If the assigned spare node and the faulty node belong to different clusters, a dedicated path is constructed by linking the appropriate routing channels of the intermediate spare nodes.

Fig. 2 illustrates reconfiguration of an enhanced cluster 6-ary 2-cube with the spare network of a 3-ary 2-cube in the presence of indicated faulty nodes. For the sake of clarity, only active spare links are shown in the figure and dedicated paths are drawn in a variety of line styles to distinguish them. In the figure, faulty nodes 22, 23, 32, 33, 34, 35, 40, 43, and 50 are replaced by spare nodes $S02$, $S01$, $S10$, $S11$, $S22$, $S12$, $S20$, $S21$, and $S00$, respectively. Note that, by utilizing the intermediate spare nodes, in effect, four logical spare nodes are present in cluster 11. Fig. 3 illustrates how spare nodes S01 and S11 replace faulty nodes 23 and 33, respectively, by merging their communication modules. The dashed and solid light lines in Fig. 3 pertain to similar lines in Fig. 2 and represent effective permanent circuit-switched dedicated paths after the reconfiguration. Once such paths are established, due to the circuit-switched capability of the wave-switching communication modules, the physical location of the faulty nodes and their assigned spare nodes become irrelevant. Therefore, the logical topology is preserved with similar congestion and higher dilation. Moreover, no modification of the available computation or communication algorithm is required.

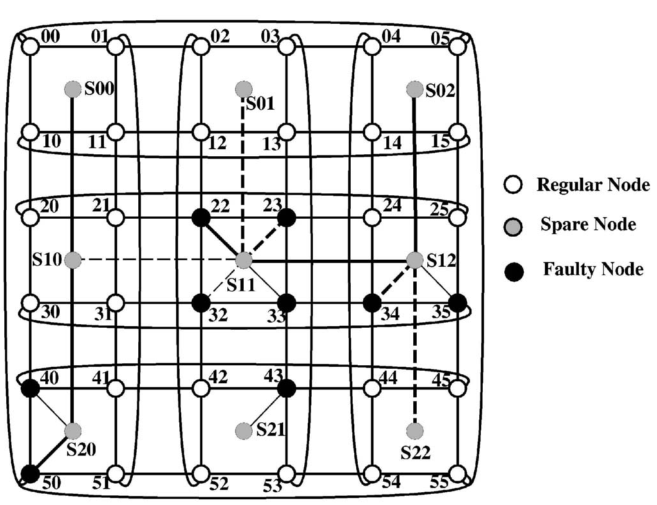

Fig. 2. Reconfiguration of an ECKN in presence of faulty nodes.

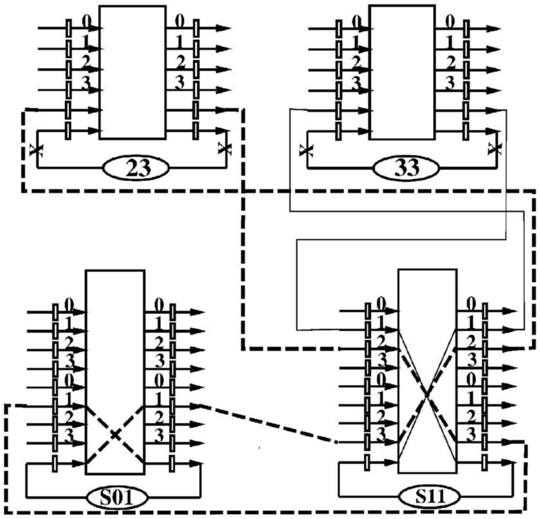

Fig. 3. Replacing faulty nodes 23 and 33 with spare nodes S01 and S11, respectively.

The node degree of the regular node and spare node in Fig. 2 is five and eight, respectively. In contrast, to tolerate the same number of faulty nodes, the scheme in [15], which has one of the least node degrees among graph-based schemes, requires a node degree of 22. Also, the non-node-minimal scheme in [19] would require a node degree of 13 at the expense of 7,128 spare nodes.

3.2 The ECKN With $j = \frac{k}{2}$

If $j = \frac{k}{2}$, the network interconnecting the spare nodes becomes a 2-ary (binary) n-cube. The resultant structure consists of k^n regular nodes and 2^n spare nodes. The degree of each regular node and spare node is $2n + 1$ and $n + (\frac{k}{2})^n$, respectively. Fig. 4 depicts a 6-ary 2-cube with $j = 3$. Hence, each spare node is assigned to a cluster of 3^2 regular nodes and the spare nodes are interconnected as a two-dimensional hypercube. Faulty nodes are tolerated, as discussed before.

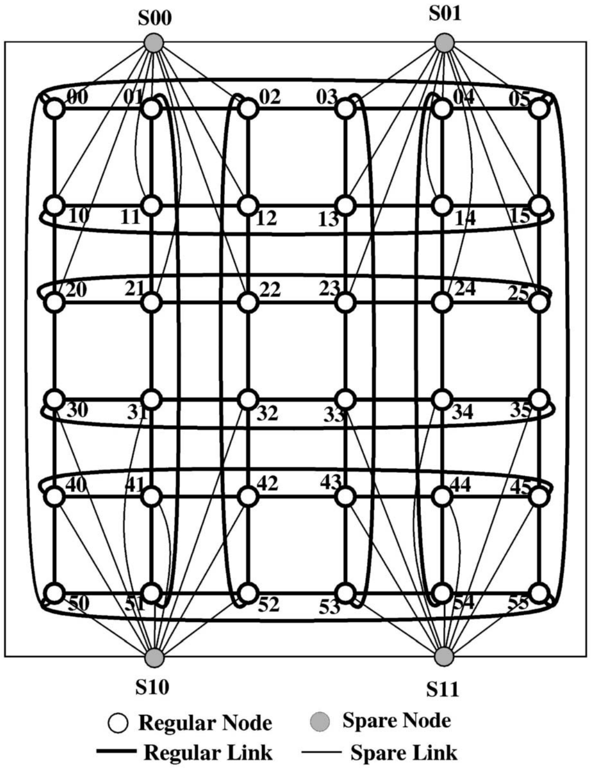

Fig. 4. An enhanced cluster 6-ary 2-cube with spare nodes connected as a hypercube.

4 Theoretical Results

4.1 The ECKN with $j \neq \frac{k}{2}$

Let us define a cluster with one or more faulty nodes as a *faulty cluster*. Since, within a cluster, the local spare node is directly connected to every regular node, every dedicated path between a faulty node of a cluster and its assigned spare node must pass through the local spare node of the cluster. Therefore, the number of edge-disjoint paths between the faulty nodes of a cluster and the unassigned spare nodes in other clusters is the same as the number of edge-disjoint paths (within the spare network) between the local spare node of the faulty cluster and the unassigned spare nodes. The reconfigurability of the ECKN is then a function of the number of dedicated and edge-disjoint paths that can be established, within the spare network, between the local spare node of a cluster with multiple faulty nodes and the available spare nodes in the fault-free clusters. The Connection Requirement (C_R) of the spare node of a faulty cluster then represents the required number of such paths. For example, in Fig. 2, since three out of four logical spare nodes of cluster 11 physically belong to other clusters, the C_R of the spare node $S11$ is three. Note that the C_R of a spare node is equal to the number of faulty nodes in its cluster minus one. We next set the upper bound on the number of faulty nodes that can be tolerated in a cluster and the lower bound on the number of the faulty nodes that an ECKN can tolerate regardless of their distribution.

Lemma 1. *The upper bound on the number of faulty nodes that an enhanced cluster k-ary n-cube, with $j \neq \frac{k}{2}$, can tolerate in a cluster is $2n+1$.*

From Lemma 1, it follows that the maximum C_R of a spare node is $2n$.

Theorem 1. *In an enhanced cluster k-ary n-cube, with $j \neq \frac{k}{2}$, $2n+1$ faulty nodes can be tolerated regardless of fault distribution.*

Even though, compared with the published results in the literature [6], [7], [8], [9], [10], [13], [14], [15], [16], [17], [19], [18], [20], [24], the ECKN under Theorem 1 has an attractive fault-tolerant capability, only a small fraction of spare nodes and spare links are utilized to tolerate $2n+1$ faulty nodes. Hence, with the majority of the spare nodes and spare links left intact, one should expect that the actual fault tolerance of the ECKN would be much higher.

We next examine whether the ECKN can tolerate $(\frac{k}{j})^n$ faulty nodes under specific fault patterns. Certainly, if each faulty cluster is limited to one faulty node, the ECKN can tolerate all $(\frac{k}{j})^n$ faulty nodes by assigning each one to its local spare node. To examine the case with multiple faulty nodes per faulty clusters, let us group the spare nodes into three sets: S_S (set of source nodes), S_U (set of used nodes), and S_T (set of target nodes). A source node is a spare node in a cluster with multiple faulty nodes. The set S_S then represents the spare nodes with a C_R greater than 0. S_T is the set of unassigned spare nodes, and S_U consists of spare nodes that have been assigned to faulty nodes and have a C_R of 0. For example, considering only the faulty nodes in Fig. 2, after assigning the local spare node to a local faulty node in each faulty cluster, $\mathrm{S}_S = \{S11, S12, S20\}$, $\mathrm{S}_U = \{S21\}$, and $\mathrm{S}_T = \{S00, S01, S02, S10, S22\}$. During our reconfiguration algorithm, which is discussed in Section 5, the spare nodes are dynamically assigned to the various sets. To illustrate this, suppose the C_R of a spare node $\alpha \in \mathrm{S}_S$ is greater than 0 and there exists a dedicated path from α to $\beta \in \mathrm{S}_T$. Consequently, β replaces a faulty node in the cluster of α via a dedicated path. β is then called *used* and is assigned to S_U. Also, the C_R is reduced by one. If the C_R of α becomes zero, it is also marked as *used* and is assigned to S_U. The ECKN is called reconfigured when S_S becomes an empty set.

As mentioned before, the reconfigurability of the ECKN is a function of the number of dedicated and edge-disjoint paths, within the spare network, that can be established between the local spare nodes (nodes in S_S) of the clusters with multiple faulty nodes and the available spare nodes (nodes in S_T) of the fault-free clusters. Obviously, if the spare nodes are interconnected as a complete graph, the ECKN can tolerate all $(\frac{k}{j})^n$ faulty nodes regardless of their distribution. Hence, the reconfigurability of the ECKN is a direct consequence of the connectivity of the topology of the spare network. Let us denote the topology of the graph connecting the spare nodes by $G = (V, E)$, where $V = \mathrm{S}_S \bigcup \mathrm{S}_U \bigcup \mathrm{S}_T$ and E consists of the appropriate spare links. Let the C_R of a node $n \in \mathrm{S}_S$ be represented by $\mathrm{C}_R(n)$ and let us denote the sum of the C_Rs of all nodes in a set P as $\sum_{n \in P} \mathrm{C}_R(n)$. Since the number of faulty nodes cannot exceed the number of spare nodes, $|S_T| \geq \sum_{n \in S_S} \mathrm{C}_R(n)$. The following theorem examines the connectivity of G as it pertains to the reconfigurability of the ECKN.

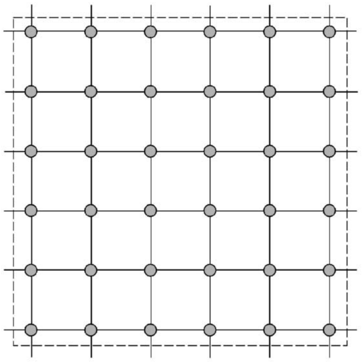

Fig. 5. Spare nodes of a *k*-ary 2-cube.

Theorem 2. *Consider a graph $G(V, E)$, where $V = \mathrm{S}_S \bigcup \mathrm{S}_U \bigcup \mathrm{S}_T$. The necessary and sufficient condition for every node $n \in \mathrm{S}_S$ to have C_R edge-disjoint paths to C_R nodes in S_T is that the minimum number of edges leaving any subset of nodes $P \subseteq V$ be greater than or equal to $\sum_{n\in(P\bigcap S_S)} \mathrm{C}_R(n) - |P \bigcap \mathrm{S}_T|$.*

Based on Theorem 2, the ECKN can tolerate a given distribution of faulty nodes provided the sum of the C_Rs of any set of spare nodes (with nonzero C_R) is smaller than the number of edges leaving the set. To examine reconfiguration under multiple faulty nodes per faulty cluster (spare node's $\mathrm{C}_R \geq 1$), consider the example depicted by Fig. 5, representing a segment of a spare network. If every spare node in the figure has a $\mathrm{C}_R = 1$ (two faulty nodes per faulty cluster), the sum of the C_R of the entire set would be 36 (72 faulty nodes within the set). However, since the number of edges leaving the set is only 24, reconfiguration would fail according to Theorem 2. The result can be obtained for any dimension of the ECKN since, for a given dimension, one can find a radix that violates Theorem 2. Therefore, under the maximum number of faulty nodes, no theoretical lower bound on the number of faulty nodes per cluster can be established.

4.2 Theoretical Results under $j = \frac{k}{2}$

The spare network under $j = \frac{k}{2}$ is a hypercube of dimension n (spare cube). The theorems given below examine the connectivity of the spare cube and, therefore, the reconfigurability of the ECKN. We first set the upper bound on the number of faulty nodes that can be tolerated in a cluster. We then examine the lower bound on the number of faulty nodes that an ECKN can tolerate for any fault distribution. Finally, we examine the conditions under which the maximum number of faulty nodes can be tolerated.

Lemma 2. *The upper bound on the number of faulty nodes that an enhanced cluster k-ary n-cube, with $j = \frac{k}{2}$, can tolerate in a cluster is $n + 1$.*

Theorem 3. *In an enhanced cluster k-ary n-cube, with $j = \frac{k}{2}$, $n + 1$ faulty nodes can be tolerated regardless of fault distribution.*

We next apply Theorem 2 to the spare cube and examine the reconfigurability of the ECKN. Theorems 4 and 5 set the bounds on the number of faulty nodes per cluster, under the maximum number of faulty nodes (2^n), that can be tolerated regardless of the fault distribution.

Theorem 4. *Clusters with four or more faulty nodes can cause the reconfiguration of an ECKN, with $j = \frac{k}{2}$ and in the presence of 2^n faulty nodes, to fail.*

Theorem 5. *The ECKN, with $j = \frac{k}{2}$, can tolerate 2^n faulty nodes with up to three faulty nodes per cluster regardless of the fault distribution.*

Hence, the ECKN has a higher theoretical fault tolerance if $j = \frac{k}{2}$.

5 Simulation Results

Our theoretical results indicated that some patterns of faulty nodes can cause the reconfiguration of the ECKN to fail. However, the probability that the faulty nodes can form such patterns is very low. Therefore, a more realistic measure of the reconfigurability of the ECKN would be under random fault distributions.

An optimal reconfiguration algorithm can be developed by utilizing the maxflow algorithm. Here, optimality is measured as the ability to assign a spare node to every faulty node whenever such an assignment is feasible vis-a-vis Theorem 2. The main drawback to reconfiguration using the above algorithm is that a digraph representation of the spare network has to be constructed [28] and the spare node assignment has to be done by the host processor. To overcome these deficiencies, we next present a near-optimal reconfiguration algorithm, which is called *Alloc-Spare*. The algorithm consists of three parts, as specified below:

1. **Early Abort:** The following solvability checks are performed to determine whether the reconfiguration is feasible. If the total number of faulty nodes is greater than the number of spare nodes ($(\frac{k}{j})^n$), the reconfiguration fails. If the C_R of a spare node is greater than $2n$ for $j \neq \frac{k}{2}$ or n for $j = \frac{k}{2}$, the reconfiguration fails due to Lemmas 1 or 2, respectively. The reconfiguration also fails if the sum of the C_R of any two neighboring spare nodes in the spare network is greater than $4n - 2$ for $j \neq \frac{k}{2}$ or $2n - 2$ for $j = \frac{k}{2}$, based on Theorem 2.
2. **Local Assignment:** The local spare node of every faulty cluster is assigned to a faulty node within the cluster. If all faulty nodes are covered, the ECKN is reconfigured.
3. **Nonlocal Assignment:** To find a set of candidate spare nodes that can be assigned to a faulty node, we utilize Lee's path-finding algorithm [29]. The algorithm begins by constructing a breadth-first search of minimum depth d $(1 \leq d \leq (\frac{k}{j})^n - 1)$ in the spare network from the local spare node of a faulty cluster with a nonzero C_R. If a free spare node is found, a path to the source node is formed. The algorithm guarantees that a path to a spare node will be found if one exits and that the path will be the shortest possible [29]. Therefore, all faulty nodes that are one link away from available spare nodes (at depth 1) are assigned first. Once a path is formed, the links

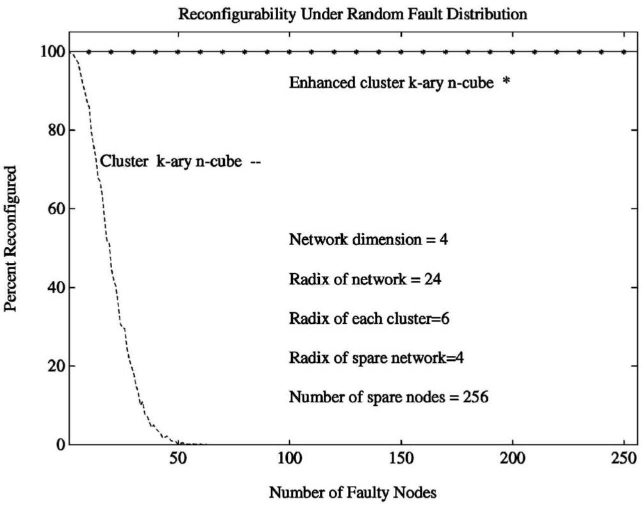

Fig. 6. The ECKN, with $j \neq \frac{k}{2}$, under random fault distribution.

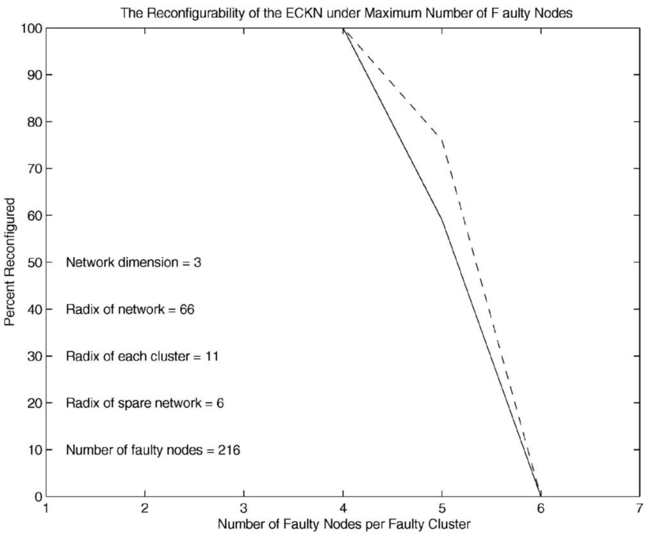

Fig. 7. The ECKN, with $j \neq \frac{k}{2}$, under maximum number of faulty nodes.

associated with that path are deleted from the spare network, resulting in a new structure. If there still remain some uncovered faulty nodes, a solvability test to check the C_R of neighboring spare nodes, similar to Early Abort, is performed on the new structure and Step 3 is repeated for a higher depth d. Reconfiguration fails if $d > (\frac{k}{j})^n - 1$, which is the longest acyclic path in the spare network.

In an ECKN, at most, there would be $\frac{1}{2}(\frac{k}{j})^n$ source nodes. Since the longest path from a source node to a target node is $(\frac{k}{j})^n - 1$, the time complexity of the algorithm is $O((\frac{k}{j})^{2n})$, provided the host processor implements it. The algorithm can also be applied in a distributed manner in $O((\frac{k}{j})^n)$ by making each source node do its own breadth-first search [30].

We implemented the algorithm Alloc-Spare for an ECKN with $k = 24$, $n = 4$, and $j = 6$ $(j \neq \frac{k}{2})$; spare nodes are interconnected as a 4-ary 4-cube. Step 3 of Alloc-Spare was implemented by first assigning spare nodes to the faulty nodes in the clusters with the lowest indices, i.e., the breadth-first search was first formed from the spare node $S0000_4$ (provided its C_R is nonzero), followed by the spare node $S0001_4$, and so on. The simulation result for up to 256 randomly placed faulty nodes is shown in Fig. 6. One thousand simulation runs were performed for each given number of faulty nodes. The result indicates 100 percent reconfigurability for the ECKN under up to 256 randomly placed faulty nodes. The other plot in the figure pertains to the result of our local reconfiguration scheme, called the cluster scheme [28], whose performance is similar to the Interstitial scheme proposed by Singh [7].

To simulate the case under $j = \frac{k}{2}$, we implemented algorithm Alloc-Spare for an ECKN with $k = 6$, $n = 10$, and $j = 3$, resulting in a spare cube of dimension 10. Our simulation result for up to 1,024 randomly placed faulty nodes indicates 100 percent reconfiguration.

Perhaps a better measure of the fault-tolerant capability of the ECKN is the connectivity of the spare network after the reconfiguration. As a measure of its connectivity, we calculated the average number of spare links per spare node that were left unused after the reconfiguration. Our result reveals that, under random distribution of faults, on the average, about one spare link per spare node is used to reconfigure the ECKN. Hence, the spare network of the ECKN is a well-connected graph even after the reconfiguration.

To examine the limitation of the ECKN under random fault distribution, we next assumed that the number of faulty nodes in the ECKN is the maximum $((\frac{k}{j})^n)$. Moreover, we assumed that each faulty cluster contains a fixed number of faulty nodes. We first present our simulation results for $j \neq \frac{k}{2}$. Since, by Lemma 1, a faulty cluster may have up to $2n + 1$ faulty nodes, simulation runs for 1 to $2n + 1$ faulty nodes per cluster were carried out. Note that, under the maximum number of faulty nodes, the number of faulty clusters is equal to the number of faulty nodes divided by the given number of faulty nodes per cluster. If the division results in a remainder, an additional cluster with the number of faulty nodes equal to the remainder needs to be allocated as well. The faulty clusters were then randomly allocated in an ECKN with $k = 66$, $j = 11$, and $n = 3$; 216 spare nodes connected as a 6-ary 3-cube. The simulation result is shown by the solid line in Fig. 7, where each point in the graph represents 1,000 simulation runs. Fig. 7 indicates the percent number of cases where the ECKN was able to reconfigure. The figure shows that nearly 100 percent reconfiguration is achieved (all 216 faulty nodes are tolerated) for up to four faulty nodes per cluster. Moreover, our other simulations reveal that, on the average, less than half of the six spare links per spare node were used to make the reconfiguration feasible. Therefore, the spare network remains a well-connected graph via the unused spare links, even after the reconfiguration.

During Step 3 of the reconfiguration algorithm, once a breadth-first search is made, there may be a number of candidate spare nodes that can be assigned to a faulty node. So far, the selection among the candidate spare nodes has been random. However, it may be more appropriate to select a free spare node, which is least likely to be picked by the faulty nodes in other clusters. To do so, we added a preprocess step to Step 3 of the algorithm. During the preprocess, weights are assigned to each free spare node based on the proximity of the free spare node to other faulty

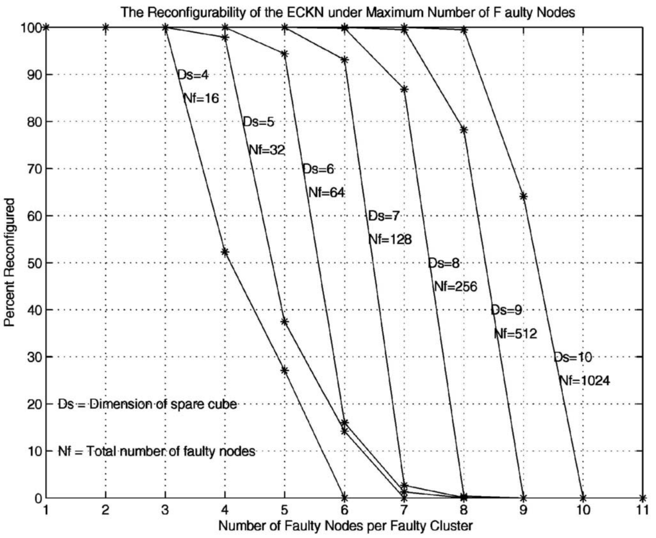

Fig. 8. The ECKN, with $j = \frac{k}{2}$, under maximum number of faulty nodes.

nodes; the higher the weight, the higher the number of faulty nodes that are in the vicinity of the free spare. Therefore, during Step 3, the candidate free spare node with the lowest weight is assigned to the faulty node first. The pertaining simulation results are shown in Fig. 7 by the dashed line. The plot indicates some performance improvement. However, the effectiveness of the modified algorithm is less vivid in ECKN's with higher dimensions.

Under $j = \frac{k}{2}$, additional simulations were carried out to examine the fault-tolerant limitation of the ECKN. Note that, by Lemma 2, a faulty cluster may have up to $(n+1)$ faulty nodes. The simulation results for an ECKN of different dimensions are shown in Fig. 8. The figure shows that, under the maximum number of faulty nodes and random fault distribution, an ECKN can nearly achieve 100 percent reconfiguration for up to $(n-2)$ faulty nodes per cluster, which is higher than the theoretical lower bound of three faulty nodes per cluster for $n > 5$. Comparing Figs. 7 and 8 reveals that, for the same number of faulty nodes, the ECKN under $j = \frac{k}{2}$ has a higher fault tolerance than the ECKN under $j \neq \frac{k}{2}$. Our simulations also show that the ECKN, with $j = \frac{k}{2}$ and under the maximum number of faulty nodes and the highest number of reconfigurable faulty nodes per cluster, uses on average less than three spare links per spare node to reconfigure.

To measure the effect of the size of the radix and the dimension of the network on its fault tolerance, additional simulations were carried out. For $j \neq \frac{k}{2}$, our results indicate that, for a given dimension, the radix of the spare network of the ECKN is inversely proportional to its reconfigurability. Moreover, for all values of j, a higher dimension network uses a lower percent of its spare links to reconfigure and therefore is more fault-tolerant.

6 Conclusion

In this paper, we have presented a strongly fault-tolerant design for the k-ary n-cube multiprocessor and examined its reconfigurability. Our theoretical results indicate that our scheme can always tolerate $(2n+1)$ faulty nodes for $j \neq \frac{k}{2}$ and $(n+1)$ for $j = \frac{k}{2}$, regardless of their distribution. For $j = \frac{k}{2}$, our theoretical results further indicate that the ECKN can tolerate 2^n faulty nodes for up to three faulty nodes per faulty cluster. Our experimental results suggest that, under random fault distribution, $(\frac{k}{j})^n$ faulty nodes (the maximum) is tolerated with a very high probability. For faster reconfiguration, our near-optimal reconfiguration algorithm is more appropriate than the optimal one, with negligible difference in the end result. Our scheme has a regular node degree of $(2n+1)$, which is optimal. The node degree of the spare node is $2n + j^n$ if $j \neq \frac{k}{2}$ and $n + (\frac{k}{2})^n$ if $j = \frac{k}{2}$. Hence, depending on the value of j, k, and n, the required spare node degree could become large. However, they are significantly lower than the published schemes for the same number of tolerated faulty nodes; those schemes can theoretically guarantee the maximum number of faulty nodes, where our scheme can only probabilistically guarantee it.

The ECKN has the advantage of having the flexibility of utilizing the rich interconnection properties of the hypercube to attain a higher fault tolerance by setting $j = \frac{k}{2}$. However, this is achieved at the expense of having a fixed number of spare nodes (2^n). With $j \neq \frac{k}{2}$, one has the flexibility to incorporate more spare nodes to attain a higher reliability or reduce cost by using a smaller number of spare nodes, at the expense of lower fault tolerance and higher spare node degree. Compared with other proposed schemes, the ECKN can tolerate significantly more faulty nodes for the same overhead.

Appendix

Proof of Lemma 1. Given a cluster with multiple faulty nodes, the local spare node can replace one of them. Since the local spare node has a degree of $2n$ within the spare network, at most $2n$ edge-disjoint paths may be constructed from the local spare node to free spare nodes. Therefore, the maximum number of tolerated faulty nodes in a cluster is $2n + 1$. □

Proof of Theorem 1. The proof is by induction. The base case is shown for $n = 1$. k regular nodes form a ring, as depicted in Fig. 9a for $k = 12$ and $j = 3$. Similarly, $\frac{k}{j}$ spare nodes form a ring. Hence, each spare node has a direct spare link connecting it to each of its two immediate neighboring spare nodes. By inspection, even if all $2 \times 1 + 1 = 3$ faulty nodes reside in the same cluster, there exists three edge-disjoint paths connecting the faulty nodes to the local spare node and its immediate neighboring spare nodes.

Next, let us consider an ECKN of dimension $n+1$. By construction, an $(n+1)\text{-dimensional)}$ CKN consists of k n-dimensional ECKN modules such that each spare node, in addition to its $2n$ spare links within its module, is connected to two spare nodes in other modules (Fig. 9b). By the induction hypothesis, each ECKN of dimension n can tolerate $2n + 1$ faulty nodes. Suppose there exist $2(n+1) + 1 = 2n + 3$ faulty nodes. If the distribution of faulty nodes is such that at most $2n + 1$ faulty nodes reside in one ECKN of dimension n, the system can tolerate them by the induction hypothesis. Consider the case where all faulty nodes reside in the

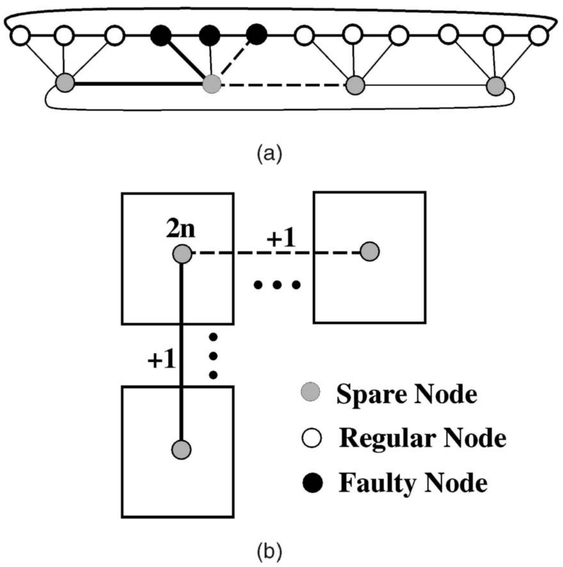

Fig. 9. An enhanced cluster (a) *k*-ary 1-cube. (b) *k*-ary $(n+1)$-cube.

same n-dimensional ECKN module. $2n+1$ of the faulty nodes can be tolerated locally by the induction hypothesis. Furthermore, since every faulty node has an unused spare link to its local spare node and the local spare node has two unused spare links connecting it to two free spare nodes in other modules (unassigned spare nodes within two fault-free ECKN of dimension n), two dedicated and edge-disjoint paths between the last two faulty nodes and the unassigned spare nodes can be established. The system can therefore tolerate $2(n+1)+1$ faulty nodes and the theorem follows by induction. □

Proof of Theorem 2. We first prove the necessary condition: If, from every node $n \in \mathrm{S}_S$, there exists C_R edge-disjoint paths to C_R nodes in S_T, then the minimum number of edges leaving any subset of nodes $P \subseteq V$ must be greater than or equal to $\sum_{n\in(P\bigcap \mathrm{S}_S)} \mathrm{C}_R(n) - |P\bigcap \mathrm{S}_T|$, which is the sum of the C_Rs of S_S nodes within P minus the number of S_T nodes in P. Let us consider a subset $P_1 \subseteq \mathrm{S}_S$. Each of the edge-disjoint paths from a node in S_S to a node in S_T must be carried over at least one edge in the cutset $(P_1, V-P_1)$. Therefore, the sum of the C_Rs of the nodes in P_1, which represents the total required number of edge-disjoint paths from the nodes in P_1 to the nodes in S_T, must be smaller than or equal to the number of edges in the cutset $(P_1, V-P_1)$. Now, let us consider a subset $P \subseteq V$ and denote the graph interconnecting the nodes of P as g. Obviously, g is a subgraph of G. Within g, there exists only $|P\bigcap \mathrm{S}_T|$ target nodes. Therefore, at most $|P\bigcap \mathrm{S}_T|$ of the edge-disjoint paths may exist in g. The rest of the paths must then be carried over the cutset $(P, V-P)$. Therefore, the necessary condition follows.

We next prove the sufficient condition: If the minimum number of edges leaving any subset of nodes $P \subseteq V$ is greater than or equal to $\sum_{n\in(P\bigcap \mathrm{S}_S)} \mathrm{C}_R(n) - |P\bigcap \mathrm{S}_T|$, every node $n \in \mathrm{S}_S$ would have $\mathrm{C}_R(n)$ edge-disjoint paths to $\mathrm{C}_R(n)$ nodes in S_T. Let us create a new graph $G' = (V', E')$ by adding two nodes s and t to G, as specified below and depicted by Fig. 10. Each node in S_T is connected to t via a single edge. Each node $n \in \mathrm{S}_S$ is connected to s via $\mathrm{C}_R(n)$ parallel edges. Let the sum of the C_R of all nodes in S_S be L. The number of edge-disjoint paths between s and t in G', according to Menger's theorem [27], is equal to the size of the mincut in G'. We will show that there always exists an (s,t) mincut in G' whose size is equal to L. The mincut in G' may exist at s, t, G, or some combination of them. By construction, the size of the cut at s equals L. Similarly, the cutsize at t is greater than or equal to L since $|S_T| \geq \sum_{n\in S_S} \mathrm{C}_R(n) = \mathrm{L}$. Per stated condition, for $P = s\bigcup \mathrm{S}_S$ or $P = s\bigcup \mathrm{S}_S \bigcup \mathrm{S}_U$, the cut $(P, V'-P)$ must have a cutsize greater than or equal to L. Consider a general cut in G' crossing L_1 of the edges connecting s to S_S nodes, L_2 edges of G, and L_3 of the edges connecting S_T nodes to t (Fig. 10). The number of S_T nodes on the unshaded side of the cut is L_3. The sum of the C_Rs of S_S nodes within the same side of the cut is $\mathrm{L}-\mathrm{L}_1$. Therefore, the stated condition can be formulated as $\mathrm{L}_2 \geq (\mathrm{L}-\mathrm{L}_1) - \mathrm{L}_3$ or $\mathrm{L}_1 + \mathrm{L}_2 + \mathrm{L}_3 \geq \mathrm{L}$. From this inequality, it follows that any cut in G' has a cutsize greater than or equal to L. Therefore, L is the size of the mincut. Hence, there exist L edge-disjoint paths between nodes s and t. Each of these *s-t* edge-disjoint paths must pass through a unique node in S_T because each node in S_T is connected to t via a single edge. Since there only exist L edges from s (one per path), the number of edge-disjoint paths from s that passes through each node $n \in \mathrm{S}_S$ is equal to $\mathrm{C}_R(n)$. Therefore, each node $n \in \mathrm{S}_S$ can make $\mathrm{C}_R(n)$ edge-disjoint paths to $\mathrm{C}'_R(n)$ distinct nodes in S_T. □

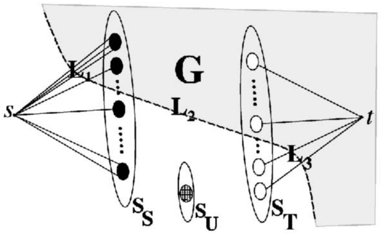

Fig. 10. A cut in graph G'.

Proof of Lemma 2. The local spare node can tolerate one faulty node. Since, within the spare cube, the local spare node has a degree of n, at most n edge-disjoint paths may be constructed from it to free spare nodes. Hence, at most $(n+1)$ faulty nodes may be tolerated in a cluster. □

Proof of Theorem 3. The proof is by induction. The base case is shown for $n=0$. There exists one spare node, which is connected to one regular node. Upon failure of the regular node, the spare node can replace it directly. Therefore, it can tolerate $n+1=1$ faulty node.

Next, let us consider an ECKN of dimension $n+1$. By construction, an $(n+1)$-dimensional ECKN consists of k n-dimensional ECKN modules such that each spare node, in addition to its n spare links within its half of the ECKN, is connected to a spare node in the other half of the ECKN. By the induction hypothesis, each half of the ECKN can tolerate $n+1$ faulty nodes. Suppose there exist $(n+1)+1 = n+2$ faulty nodes. If the distribution of faulty nodes is such that at most $n+1$ faulty nodes reside in one half of the ECKN, the system can tolerate them by the induction hypothesis. Consider the case where all faulty nodes reside in the same half of the ECKN. $n+1$ of the faulty nodes can be tolerated locally by the induction hypothesis. Since every faulty node has an unused spare link to its local spare node and the local spare node has an unused spare link in the dimension $n+1$ to an unassigned spare node within the fault-free half of the ECKN, a dedicated path within the spare cube between the $(n+2)\mathrm{th}$ faulty node and the unassigned spare node can be established. The system can therefore tolerate $(n+2)$ faulty nodes and the theorem follows by induction. □

Proof of Theorem 4. We prove the theorem by showing that Theorem 2 does not hold for a distribution of four faulty nodes per cluster. Let S_S contain k_1 spare nodes with a C_R of 1, k_2 spare nodes with a C_R of 2, and k_3 spare nodes with a C_R of 3. Also, let S_T and S_U contain k_0 and k_u spare nodes, respectively. Next, consider a subset $P \subseteq V$ with $j_1 \leq k_1$ spare nodes with a C_R of 1, $j_2 \leq k_2$ spare nodes with a C_R of 2, $j_3 \leq k_3$ spare nodes with a C_R of 3, $j_0 \leq k_0$ unassigned spare nodes, and $j_u \leq k_u$ used spare nodes. Since the degree of each spare node within the spare cube is n, the number of links crossing P is $nj_3 + nj_2 + nj_1 + nj_0 + nj_u - l$, where l represents the number of internal links among the nodes in P. If an ECKN is to tolerate up to four faulty nodes per cluster, by Theorem 2, it is necessary to show that

$$n(j_3 + j_2 + j_1 + j_0 + j_u) - l \geq 3j_3 + 2j_2 + j_1 - j_0. \quad (1)$$

Consider the case where $P = \mathrm{S}_S$ and every spare node has a C_R of 3, ($j_3 = k_3 = 2^{(n-2)}$, $j_2 = j_1 = j_0 = 0$). Inequality (1) can then be rewritten as $n \times 2^{(n-2)} - l \geq 3 \times 2^{(n-2)}$. The left side of the inequality represents the total number of links crossing P. Since $2^{(n-2)}$ nodes can form a subcube of dimension $(n-2)$, the least number of links crossing P is $2 \times 2^{(n-2)}$. Therefore, $2 \times 2^{(n-2)} \ngeq 3 \times 2^{(n-2)}$ and, hence, by Theorem 2, the ECKN may not be able to tolerate 2^n faulty nodes under four faulty nodes per cluster. □

Proof of Theorem 5. We prove the theorem by showing that Theorem 2 holds for the ECKN with up to three faulty nodes per cluster. Let S_S contain k_1 spare nodes with a C_R of 1 and k_2 spare nodes with a C_R of 2. Also, let S_T and S_U contain k_0 and k_u spare nodes, respectively. Since the total number of faulty nodes cannot exceed the total number of spare nodes,

$$k_0 \geq 2k_2 + k_1. \quad (2)$$

Adding $k_2 + k_1 + k_u$ to both sides results in

$$k_2 + k_1 + k_0 + k_u \geq 3k_2 + 2k_1 + k_u. \quad (3)$$

Since the left side of (3) represents the total number of spare nodes in the spare cube,

$$2^n \geq 3k_2 + 2k_1 + k_u \quad (4)$$

or

$$n \geq \lceil \log_2(3k_2 + 2k_1 + k_u) \rceil. \quad (5)$$

Now, consider a subset $P \subseteq V$ with $j_1 \leq k_1$ spare nodes having a C_R of 1, $j_2 \leq k_2$ spare nodes having a C_R of 2, $j_0 \leq k_0$ unassigned spare nodes, and $j_u \leq k_u$ used spare nodes. Since the degree of each spare node within the spare cube is n, the number of links crossing P (cutsize of P) is $nj_2 + nj_1 + nj_0 + nj_u - l$, where l represents the number of internal links among the nodes in P. We will show that

$$nj_2 + nj_1 + nj_0 + nj_u - l \geq 2j_2 + j_1 - j_0. \quad (6)$$

Hence, by Theorem 2, an appropriate number of edge-disjoint paths within the spare cube between the spare nodes in S_S and the free spare nodes in S_T must exist. Note that

$$nj_u \geq 0 \quad (7)$$

and

$$nj_0 \geq -j_0. \quad (8)$$

Also, from (5), it follows that

$$nj_1 \geq j_1. \quad (9)$$

Subtracting (7), (8), and (9) from (6) results in

$$nj_2 - l \geq 2j_2. \quad (10)$$

Inequality (10) pertains to a case where every spare node in P has a C_R of 2 ($P \subseteq \mathrm{S}_S$). Obviously, if (10) holds, so must (6). The left side of (10) would be the minimum when l is the maximum. l is the maximum when the nodes within P have the highest connectivity among themselves. In a hypercube, a set of nodes have the highest connectivity when they form a subcube; j_2 nodes form a subcube of dimension $(n-r)$, $j_2 = 2^{(n-r)}$. Of the n links of each node of this subcube, $(n-r)$ are used by the subcube itself and the remaining links $(n-(n-r) = r)$ are used to connect the node to the other neighboring nodes of the spare cube, outside of the subcube. Consequently, for $j_2 = 2^{(n-r)}$, (10) can be formulated as

$$r \times 2^{(n-r)} \geq 2 \times 2^{(n-r)}. \quad (11)$$

To examine whether (11) can ever be violated, it is necessary to find the minimum r. r is the minimum when the dimension of $(n-r)$ is the maximum. Given

$$j_2 \leq k_2 = \left\lfloor \frac{2^n}{3} \right\rfloor \quad (12)$$

and

$$\frac{2^n}{3} = \frac{2^n}{4} + \frac{2^n}{12}, \quad (13)$$

the largest number of nodes with a C_R of 2 that can form a subcube is

$$j_2 = \frac{2^n}{4} = 2^{(n-2)}. \quad (14)$$

Case (1) ($0 < j_2 \leq 2^{(n-2)}$): From the foregoing discussion, the smallest value of r is 2. Therefore, (11) and, hence, (10) are satisfied for the specified range of this case.

Case (2) ($2^{(n-2)} < j_2 \leq 2^{(n-2)} + 2^{(n-4)}$): Let us consider $j_2 = 2^{(n-2)} + j_2'$. Then,

$$j_2' \leq \frac{2^n}{3} - \frac{2^n}{4} = \frac{2^n}{12}. \qquad (15)$$

For this case, (10) can then be formulated as

$$n(2^{(n-2)} + j_2') - l \geq 2(2^{(n-2)} + j_2'). \qquad (16)$$

Since $2^{(n-2)}$ nodes have the highest connectivity when they form a $(n-2)$ subcube, the above inequality can be stated as

$$2 \times 2^{(n-2)} + nj_2' - l' \geq 2 \times 2^{(n-2)} + 2 \times j_2', \qquad (17)$$

where l' represents the number of internal links among the j_2' nodes as well as the number of links connecting j_2' nodes to the $(n-2)$ subcube. Subtracting $2 \times 2^{(n-2)}$ from both sides of (17), results in

$$nj_2' - l' \geq 2 \times j_2'. \qquad (18)$$

The least number of links that cross P exists when j_2' nodes form a subcube of the highest possible dimension and the subcube is concatenated with the $(n-2)$ subcube. Since

$$j_2' \leq \frac{2^n}{12} = \frac{2^n}{16} + \frac{2^n}{48}, \qquad (19)$$

the largest subcube that can be formed within j_2' nodes is of dimension $(n-4)$. Then, $(n-4)$ links of each node of this subcube are used within the subcube and the remaining four links cross the subcube. The total number of links crossing the $(n-4)$-dimensional subcube is then $4 \times 2^{(n-4)}$. When both $(n-2)$ and $(n-4)$ subcubes are concatenated (the highest connectivity between them), the links that connect these two subcubes together would not be in the cutset. The number of such links could be as high as $2^{(n-4)}$. Therefore, to evaluate the minimum cutsize, $2^{(n-4)}$ needs to be subtracted from the total count of each subcube. The minimum cutsize of $(P, V-P)$ is then

$$\begin{aligned}&(2 \times 2^{(n-2)} - 2^{(n-4)}) + (4 \times 2^{(n-4)} - 2^{(n-4)}) = \\ &2(2^{(n-2)} + 2^{(n-4)}).\end{aligned} \qquad (20)$$

Therefore, (17) and, hence, (10) are always satisfied for the specified range of this case.

Case (3) ($2^{(n-2)} + 2^{(n-4)} < j_2 \leq 2^{(n-2)} + 2^{(n-4)} + 2^{(n-6)}$): Let us consider $j_2 = 2^{(n-2)} + 2^{(n-4)} + j_2'$. Then,

$$j_2' \leq \frac{2^n}{3} - \frac{2^n}{4} - \frac{2^n}{16} = \frac{2^n}{48}. \qquad (21)$$

Inequality (10) for this case can then be formulated as

$$n(2^{(n-2)} + 2^{(n-4)} + j_2') - l \geq 2(2^{(n-2)} + 2^{(n-4)} + j_2'). \qquad (22)$$

Since $2^{(n-2)} + 2^{(n-4)}$ nodes have the highest connectivity when they form two concatenated subcubes, the above inequality can be stated as

$$2(2^{(n-2)} + 2^{(n-4)}) + nj_2' - l' \geq 2(2^{(n-2)} + 2^{(n-4)}) + 2 \times j_2', \qquad (23)$$

where l' represents the number of internal links among the j_2' nodes as well as the number of links connecting j_2' nodes to the $(n-2)$ and $(n-4)$ subcubes. Subtracting $2(2^{(n-2)} + 2^{(n-4)})$ from both sides of (23) results in

$$nj_2' - l' \geq 2 \times j_2'. \qquad (24)$$

The least number of links that cross P exists when j_2' nodes form a subcube of highest possible dimension and the subcube is concatenated with both $(n-2)$ and $(n-4)$ subcubes. Since

$$j_2' \leq \frac{2^n}{48} = \frac{2^n}{64} + \frac{2^n}{192}, \qquad (25)$$

the largest subcube that can be formed within j_2' nodes is of dimension $(n-6)$. Then, $(n-6)$ links of each node of this subcube are used within the subcube and the remaining six links cross the subcube. The total number of links crossing the $(n-6)$-dimensional subcube is then $6 \times 2^{(n-6)}$. When the subcubes $(n-2)$, $(n-4)$, and $(n-6)$ are concatenated (the highest connectivity between them), the links that connect them would not be in the cutset. The number of such links could be as high as $2^{(n-6)}$. Therefore, to evaluate the minimum cutsize, $2^{(n-6)}$ needs to be subtracted from the total count of each subcube. The minimum cutsize of $(P, V-P)$ is then

$$\begin{aligned}&(2 \times 2^{(n-2)} - 2^{(n-6)}) + (2 \times 2^{(n-4)} - 2^{(n-6)}) \\ &+ (6 \times 2^{(n-6)} - 2 \times 2^{(n-6)}),\end{aligned} \qquad (26)$$

which is equal to $2(2^{(n-2)} + 2^{(n-4)} + 2^{(n-6)})$. Therefore, (17) and, hence, (10) are always satisfied for the specified range of this case.

By repeating the process established in (13), (19), and (25), the following equality can be written

$$\left\lfloor \frac{2^n}{3} \right\rfloor = 2^{(n-2)} + 2^{(n-4)} + 2^{(n-6)} + 2^{(n-8)} + \cdots + 2^{(n-2h)}, \qquad (27)$$

where $(n-2h)$ is equal to 0 or 1 depending if the dimension of the spare cube is even or odd, respectively.

Similar to the above cases, subcubes $(n-8)$ through $(n-2h)$ can be added to P. At each step, $(P, V-P)$ would have the smallest cutsize when the newly added nodes form a subcube and the added subcube would be neighbor to all previously established subcubes. We next show that (10) holds after the last set of nodes are included in P.

Case (*h*): Similar to prior cases, the smallest cutsize would result if the last set of node(s) (two nodes or one node if n is odd or even, respectively) form a subcube of dimension $(n-2h)$ and this subcube is adjacent to all other $(h-1)$ subcubes. The total number of links that are not used within the $(n-2h)$ subcube is $2h \times 2^{(n-2h)}$. The minimum cutsize of $(P, V-P)$ is then

$$\begin{aligned}&(2 \times 2^{(n-2)} - 2^{(n-2h)}) + (2 \times 2^{(n-4)} - 2^{(n-2h)}) + \cdots \\ &+ (2 \times 2^{(n-2h)} - (h-1) \times 2^{(n-2h)}),\end{aligned}$$

which is equal to $2 \times (2^{(n-2)} + 2^{(n-4)} + \cdots + 2^{(n-2h)})$. Therefore, (10) and, hence, (6) holds for any distribution of three faulty nodes per cluster. Therefore, by Theorem 2, up to three faulty nodes per cluster can be tolerated by the ECKN with $j = \frac{k}{2}$. □

REFERENCES

[1] J. Brandenburg, "Technology Advances in the Intel Paragon System," *Proc. Ann. ACM Symp. Parallel Algorithms and Architectures,* pp. 182-182, June 1993.

[2] M. Noakes, D. Wallach, and W. Dally, "The J-Machine Multicomputer: An Architectural Evaluation," *Proc. IEEE Int'l Symp. Computer Architecture,* pp. 224-235, May 1993.

[3] S. Scott and G. Thorson, "The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus," *Proc. HOT Interconnects IV,* Aug. 1996.

[4] N. Tzeng, "A Cube-Connected Cycles Architecture with High Reliability and Improved Performance," *IEEE Trans. Computers,* vol. 42, no. 2, pp. 246-253, 1993.

[5] N. Mahapatra and S. Dutt, "Hardware-Efficient and Highly-Reconfigurable 4- and 2-Track Fault-Tolerant Design for Mesh-Connected Multicomputers," *Proc. 26th Int'l Symp. Fault-Tolerant Computing,* pp. 272-281, 1996.

[6] M. Bae and B. Bose, "Spare Processor Allocation for Fault Tolerance in Torus-Based Multicomputers," *Proc. 26th Int'l Symp. Fault-Tolerant Computing,* pp. 282-290, 1996.

[7] A.D. Singh, "Interstitial Redundancy: An Area Efficient Fault Tolerance Scheme for Large Area VLSI Processor Arrays," *IEEE Trans. Computers,* vol. 37, no. 11, pp. 1398-1410, Nov. 1988.

[8] S. Chakravarty and S.J. Upadhyaya, "A Unified Approach to Designing Fault-Tolerant Processor Ensembles," *Proc. IEEE Int'l Conf. Parallel Processing,* pp. 339-342, 1988.

[9] M. Alam and R. Melhem, "Routing in Modular Fault-Tolerant Multiprocessor Systems," *IEEE Trans. Parallel and Distributed Systems,* vol. 6, no. 11, pp. 1206-1220, Nov. 1995.

[10] P. Banerjee and M. Peercy, "Design and Evaluation of Hardware Strategies for Reconfiguring Hypercubes and Meshes under Faults," *IEEE Trans. Computers,* vol. 43, no. 7, pp. 841-848, July 1994.

[11] P. Banerjee, "Strategies for Reconfiguring Hypercubes under Faults," *Proc. IEEE Int'l Symp. Fault-Tolerant Computing,* pp. 210-217, 1990.

[12] P. Banerjee, J. Rahmeh, C. Stunkel, V. Nair, K. Roy, V. Balasubramanian, and J. Abraham, "Algorithm-Based Fault Tolerance on a Hypercube Multiprocessor," *IEEE Trans. Computers,* vol. 39, no. 9, pp. 1132-1145, Sept. 1990.

[13] M. Alam and R. Melhem, "An Efficient Modular Spare Allocation Scheme and Its Application to Fault Tolerant Binary Hypercubes," *IEEE Trans. Parallel and Distributed Systems,* vol. 2, no. 1, pp. 117-126, Jan. 1991.

[14] M. Alam and R. Melhem, "Channel Multiplexing in Modular Fault Tolerant Multiprocessors," *Proc. IEEE Int'l Conf. Parallel Processing,* pp. I492-I496, 1991.

[15] J. Bruck, R. Cypher, and C.T. Ho, "Efficient Fault-Tolerant Mesh and Hypercube Architectures," *Proc. 22nd Ann. Int'l Symp. Fault-Tolerant Computing,* pp. 162-169, July 1992.

[16] S. Dutt, "Fast Polylog-Time Reconfiguration of Structurally Fault-Tolerant Multiprocessors," *IEEE Symp. Parallel and Distributed Processing,* pp. 161-169, 1993.

[17] N. Tsuda, "Fault-Tolerant Processor Arrays Using Additional Bypass Linking Allocated by Graph-Node Coloring," *IEEE Trans. Computers,* vol. 49, no. 5, pp. 431-442, May 2000.

[18] H. Ku and J. Hayes, "Systematic Design of Fault-Tolerant Multiprocessors with Shared Buses," *IEEE Trans. Computers,* vol. 46, no. 4, pp. 439-455, Apr. 1997.

[19] J. Bruck, R. Cypher, and C.T. Ho, "Fault-Tolerant Meshes with Small Degree," *Proc. Fifth ACM Symp. Parallel Algorithm and Architectures,* pp. 1-10, June 1993.

[20] J. Bruck, R. Cypher, and C.T. Ho, "Wildcard Dimensions, Coding Theory and Fault-Tolerant Meshes and Hypercubes," *Proc. 23rd Ann. Int'l Symp. Fault-Tolerant Computing,* pp. 260-267, July 1993.

[21] M. Chean and J.A.B. Fortes, "A Taxonomy of Reconfiguration Techniques for Fault-Tolerant Processor Arrays," *Computer,* pp. 55-69, Jan. 1990.

[22] R. Negrini, M. Sami, and R. Stefanelli, "Fault Tolerance Techniques for Array Structures Used in Supercomputing," *Computer,* pp. 78-87, Feb. 1986.

[23] T. Horita and I. Takanami, "Fault-Tolerant Processor Arrays Based on 1(1/2)-Track Switches with Flexible Spare Distribution," *IEEE Trans. Computers,* vol. 49, no. 6, pp. 542-552, June 2000.

[24] N. Tsuda and T. Shimizu, "Reconfigurable Mesh-Connected Processor Arrays Using Row-Column Bypassing and Direct Replacement," *Proc. 2000 Int'l Symp. Parallel Architectures, Algorithms and Networks,* pp. 24-29, Dec. 2000.

[25] J. Duato, P. Lopez, and S. Yalamanchili, "Deadlock- and Livelock-Free Routing Protocols for Wave Switching," *Proc. 11th Int'l Parallel Processing Symp.,* pp. 570-577, Apr. 1997.

[26] J. Dongarra and D. Walker, "The Quest for Petascale Computing," *IEEE Computing in Science and Eng.,* pp. 32-39, May 2001.

[27] F. Harary, *Graph Theory.* Addison-Wesley, 1972.

[28] B. Izadi, "Design of Fault-Tolerant Distributed Memory Multiprocessors," PhD thesis, Ohio State Univ., 1995.

[29] C.Y. Lee, "An Algorithm for Path Connection and Its Applications," *IRE Trans. Electronic Computers,* vol. 10, pp. 346-365, 1961.

[30] T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann, 1992.

Baback A. Izadi received the BS degree in electrical engineering from Oklahoma State University, Stillwater, and the MS and PhD degrees in electrical engineering from Ohio State University, Columbus. Since 1998, he has been an assistant professor in the department of electrical and computer engineering at the State University of New York-New Paltz. Prior to that, he was on the faculty of Devry University-Columbus, Ohio. His current research interests are fault-tolerant and parallel computer architectures, parallel and distributed computing, real-time parallel computing, and power aware computing. He is a member of the IEEE and the IEEE Computer Society.

Füsun Özgüner received the MS degree in electrical engineering from the Istanbul Technical University in 1972 and the PhD degree in electrical engineering from the University of Illinois, Urbana-Champaign, in 1975. She worked at the IBM T.J. Watson Research Center with the Design Automation Group for one year and joined the faculty of the Department of Electrical Engineering, Istanbul Technical University in 1976. Since January 1981, she has been with Ohio State University, where she is presently a professor of electrical engineering. Her current research interests are parallel and fault-tolerant architectures, heterogeneous computing, reconfiguration and communication in parallel architectures, wireless networking, and real-time parallel computing. She has served as an associate editor of the *IEEE Transactions on Computers* and on the program committees of several international conferences. She is a member of the IEEE and the IEEE Computer Society.