Hindawi
Security and Communication Networks
Volume 2024, Article ID 9798105, 1 page
https://doi.org/10.1155/2024/9798105

Retraction

Retracted: Fault-Tolerant Secure Routing of BH_n-Based Data Center Networks

Security and Communication Networks

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] X. Zhang, L. Xu, and A. Li, "Fault-Tolerant Secure Routing of BH_n-Based Data Center Networks," *Security and Communication Networks*, vol. 2021, Article ID 6723914, 10 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6723914, 10 pages
https://doi.org/10.1155/2021/6723914

Research Article

Fault-Tolerant Secure Routing of BH_n-Based Data Center Networks

Xinxin Zhang,[1,2] **Li Xu**,[1,2] **and Aihua Li**[3]

[1]*College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian, China*
[2]*Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, Fujian, China*
[3]*Department of Mathematics, Montclair State University, Montclair, NJ 07043, USA*

Correspondence should be addressed to Xinxin Zhang; 1175045320@qq.com and Li Xu; xuli@fjnu.edu.cn

Received 13 September 2021; Accepted 8 November 2021; Published 31 December 2021

Academic Editor: Xingsi Xue

As the core infrastructure of cloud computing, a large scale of the data center networks (DCNs), which consist of millions of servers with high capacity, suffer from node failure such that the reliability is deteriorated. Malicious group could inevitably compromise the quality and reliability of data; thus, how to ensure the security routing of data is an urgent practical problem. As models for large-scale DCNs, it is worth mentioning the balanced hypercube, which is well-known for its strong connectivity, regularity, and a smaller diameter. Each of which makes a balanced hypercube a trustworthy model to deal with data traffic and provides a certain degree of fault-tolerance as well. In this paper, we use the balanced hypercube as a model for the data center networks and design a reliable safety level by referring to different safety levels of related subgraph. This subgraph contains the source and destination nodes, and the shortest feasible paths are located so that the reliable transmission is achieved. Then, we get that the length of fault-tolerant safety routing of data center networks based on balanced hypercube is always no greater than the Hamming distance plus two. Experiment shows that our fault-tolerant security routing scheme is more effective in the same reliable network environment of DCNs.

1. Introduction

With the advent of 5G mobile communication technology, demands for network information and user are increasing. Internet giants at home and abroad continue to expand their own data centers. Many large data centers are being built to provide increasingly popular online services, such as gaming, e-mail, cloud storage, and infrastructure services. For example, Microsoft implied that Bing, Hotmail, and a handful of other smaller services would be backed by a million servers [1]. The popularity of IoT is increasing with artificial intelligence technology, which covers a wide range of applications. As crucial components of IoT, sensor networks include connected sensors and associated devices, which deploy plenty of heterogeneous sensing nodes for capturing environmental data [2–4]. The secure routing is a key focus since the industrial environment is extremely vulnerable and many potential threats (e.g., distributed denial of service and selfish attacks) are introduced due to the connection to the Internet [5]. Load balancing is a high-end solution for large-scale networks to solve high-load access and a large number of concurrent requests. The goal of a more feasible approach for architests is to achieve flexible scaling and high reliability. With the expansion of the data center network scale, distributed equilibrium has more and more advantages. DCNs fall into two main categories: switch-centric and server-centric. Server-centric DCNs have the advantage of reducing the cost and providing a higher degree of programmability. However, servers usually have a much larger processing delay than switches and by using servers for forwarding packets [6]. As the number of servers in data center networks (DCNs) increases, server failures may become the common phenomenon rather than exceptions. Zhou et al. [7] proposed an adaptive authenticated model for big data stream SAVI in SDN-based data center networks. Usual secure data transmission in a DCN is based on the condition of the set of arbitrary faulty servers.

Fault-tolerant routing means that efficient transmission of information routing can still be found when there is a certain number of fault nodes in the network systems. It is an important application in different networks and has been studied extensively in the literature. Chiu and Chen [8] proposed an algorithm to facilitate efficient fault-tolerant routing of messages in hypercube multicomputers which routes a message in an attempt to minimize rerouting. Derek et al. [9] discussed the fault-tolerant message routing using local neighborhood information, while Xiao et al. [10] studied it with load balancing support. Duong and Kaneko [11] developed two new fault-tolerant routing algorithms for hypercubes based on approximate directed routable probabilities and conducted a computer experiment to verify the effectiveness of their algorithms. Similarly, Park et al. [12] proposed a fault-tolerant routing algorithm in a dual-cube with the same idea and presented results of a computer experiment showing better performance of their algorithm. The edge fault-tolerant properties of hypercubes and folded hypercubes were proposed by Xu et al. [13], and these conclusions provide a solid theoretical basis for fault-tolerant routing algorithms. As the network environment becomes increasingly complex, more and more researches projects on the fault-tolerant routing in these network environments are conducted. For example, Mohammad [14] put forward an adaptive fault-tolerant routing algorithm for an injured tree-hypercube which requires that each node only knows the condition of its link. Penaranda et al. [15] presented a fault-tolerant routing methodology for the KNS topology that degrades performance gracefully in the presence of faults and tolerates a reasonably large number of faults without disabling any healthy node. Xiang et al. [16] used limited global safety information to study fault-tolerant routing in 2D tori or meshes, and then they [17] studied fault-tolerant efficient routing in dragonfly networks which have been widely used in the current high-performance computers or high-end servers. Robert et al. [18] investigated a fault-tolerant routing strategy for k-ary n-direct s-indirect topologies based on intermediate nodes.

At present, research on the data center networks is mostly based on DCell which is proposed by Guo et al. [19] in 2008. Two years later, Markus et al. [20] proposed a generalized DCell structure for load-balanced data center networks. Marc et al. [21] developed the connectivity of data center networks modeled by DCell. Wang et al. [22] proposed an efficient algorithm for finding a disjoint paths in the node-to-set routing of DCell networks. Subsequently, they studied the Hamiltonian properties of DCell networks and extend it to all generalized DCell Hamiltonian-connected [23], the restricted h-connectivity [24], and vertex-disjoint paths in DCell networks successively [25]. Li et al. [6] studied the diagnosability and g-good-neighbor conditional diagnosability of the data center network DCell. Gu et al. [26] discussed the pessimistic diagnosability of data center networks, and Lv et al. [27] studied the reliability evaluation of DCell networks.

Nevertheless, compared with the DCell, the balanced hypercube proposed by Wu and Huang can handle a reasonable amount of message traffic and also provide certain degree of fault tolerance since the balanced hypercube is a variation of the standard hypercube with desirable properties of strong connectivity, regularity, and small node multidiameter and it is a special type of load-balanced graph designed to tolerate processor failure [28]. The balanced hypercube originates from the hypercube problem that relates to the construction of networks of parallel processors. The diameter of a graph is an important measure of communication delay. With the deployment of high connectivity data center network, the network provides multiple available transmission paths and a large amount of available bandwidth for data transmission. Normally, the shorter the diameter is, the lower the communication delays. The balanced hypercube meets the topological requirements of the data center network. In [28], Wu proposed the balanced hypercube for fault-tolerant applications. Cheng [29] studied the fault-tolerant cycle embedding in a balanced hypercube. Zhang et al. [30] studied the feedback numbers of a balanced hypercube. All of these provide a theoretical basis for our research. We study the (t, k)-diagnosability of balanced hypercube under the PMC model [31]. The data center network based on the balanced hypercube model is very suitable for increasingly complex network structure.

The main motivation of this study is to propose an efficient fault-tolerant security routing of BH_n-based data center network, and our ideas come from [6, 32]. Our contributions to this paper are as follows.

We design the safety level to characterize the reliability of a balanced hypercube.

According to the different safety levels of the spanning subgraph, we determine the shortest feasible paths, which are located such that reliable transmission is achieved.

Experiment shows that our fault-tolerant security routing scheme is more effective in the same reliable network environment of DCNs.

We organize our paper as follows. In Section 2, we outline terminologies and notations used throughout this paper and then propose the definition and propositions of data center networks based on the balanced hypercube model. In Section 3, we introduce the definition of local safety level. In Section 4, we give and prove the minimum feasible path with local safety information. In Section 5, we propose fault-tolerant routing with local safety levels. Section 6 is the numerical value analysis. Section 6 is a summary of this paper.

2. Preliminaries

In this section, we first give some important terminologies and notations throughout the paper, and then we introduce the definition and propositions of data center networks based on the balanced hypercube models.

2.1. Terminologies and Notations. In DCNs, the topology of which is often represented by a graph $G = (V, E)$, where each node $u \in V$ denotes a processor the edge $(u, v) \in E$ denotes a link between nodes u and v. If $V' \subseteq V$ and $E' \subseteq E$, we call $G' = (V', E')$ a subgraph of G. The number of nodes in a set U is defined as $|U|$, and the number of edges in the edge set of U is defined as $|E(U)|$. If X and Y are subsets of G, we define the set of edges between X and Y as $|E(X, Y)|$ and denote the neighbor set of v as $N(v)$, where $N(v) - \{u\}$ indicates removing node u from $N(v)$ for $u \in N(v)$. $P(u, v)$ represents the number of hops routing from u to v, and $P_m(u, v)$ is the number of hops of the minimum feasible path from u to v.

2.2. The Balanced Hypercube. In the following, a special type of load-balanced graph called *balanced hypercube* (BH_n), where the positive integer n is called the dimension of the balanced hypercube. The DCN-based BH_n, which interconnect servers, is defined in a recursive style. That is, an n-dimensional BH_n can be recursively constructed from four $(n-1)$-dimensional ones. It is load balanced because for every node in BH_n, there exists another node matching it and these two nodes have the same adjacent nodes. Due to this structure, BH_n can effectively alleviate the problem of congestion and deadlock in DCNs.

Definition 1. (see [28]). An n-dimensional balanced hypercube BH_n consists of 2^{2n} nodes $(a_0, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1})$, where $a_0 \in \{0, 1, 2, 3\}$ and $a_1 \in \{0, 1, 2, 3\}\ (1 \le i \le n-1)$. Every node $(a_0, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1})$ connects the following $2n$ nodes

$(x, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1})$, where

$$x = (a_0 \pm 1) \bmod 4. \tag{1}$$

$(x, a_1, \ldots, a_{i-1}, y_i, a_{i+1}, \ldots, a_{n-1})$, where

$$\begin{aligned} x &= (a_0 \pm 1) \bmod 4, \\ y_i &= a_i + (-1)^{a_0}. \end{aligned} \tag{2}$$

For convenience, we assume that all the arithmetic operations on the indexes of nodes in BH_n are four modulated. Figure 1 shows the structures of two balanced hypercubes BH_1 and BH_2, where BH_2 consists of four BH_1 s. In BH_n, the first element a_0 of node $(a_0, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1})$ is named *inner indices*, while the other elements $a_i\ (1 \le i \le n-1)$ are *outer indices*. Any node in BH_n has two types of adjacent nodes: inner and outer. Every node in BH_n has two inner adjacent nodes and $2n-2$ outer adjacent nodes.

The n-dimensional balanced hypercube is very similar to a $2n$-dimensional hypercube Q_{2n} in the sense of topological properties. These two structures share similar features in terms of symmetry and connectivity, which will be discussed in the following section. Note that a $2n$-dimensional hypercube has a diameter of $2n$, BH_n has a smaller diameter than Q_{2n}, when $n \ge 1$ is odd. Thus, the balanced hypercube is more suitable for routing. Based on Definition 1, it is clear that BH_n can be derived from four BH_{n-1}s by adding a new dimension as the nth outer index of every node in BH_n.

Proposition 1 (see [28, 33]). *The* BH_n *has some important properties as follows:*

(1) BH_n *is a load balanced graph, and nodes in* BH_n *can be partitioned into a set of matching pairs,* $v = (a_0, a_1, \ldots, a_{n-1})$ *and* $v' = (a_0 + 2, a_1, \ldots, a_{n-1})$.

(2) BH_n *has* 2^{2n} *nodes, each of which has* $2n$ *adjacent nodes.*

(3) BH_n *is* $2n$*-connected, for any pair of nodes in* BH_n, *there exist* $2n$ *disjoint paths between them.*

(4) *Let* X *and* Y *be two distinct partite sets of* BH_n *which is a bipartite graph. Assume* u *and* x *are two different nodes in* X, v, *and* y *are two different nodes in* Y. *Then there exist two node-disjoint paths* $P[x, y]$ *and* $P[u, v]$, *and* $V(P[x, y])V(P[u, v]) = V(\mathrm{BH}_n)$ *for* $n \ge 1$, *where* $P[x, y]$ *and* $P[u, v]$ *represent paths from* x *to* y *and from* u *to* v, *respectively.*

3. Local Safety Level

Two processors are connected by a bidirectional link if and only if the binary representations of the two processors differ in exactly one bit or two bits according to Definition 1. The Hamming distance between two nodes s and d, denoted by $H(s, d)$, is the number of hops between s and d. In the BH_n, there are many subgraphs, among them $\mathrm{BH}_{n-1}^{(0)}, \mathrm{BH}_{n-1}^{(1)}, \mathrm{BH}_{n-1}^{(2)}, \mathrm{BH}_{n-1}^{(3)}$ are the $(n-1)$-dimension subcubes. The spanning balanced hypercube $\mathrm{SBH}(s, d)$ between source s and destination d is the smallest spanning balanced hypercube that contains s and d.

Lee and Hayes [34] proposed the concept of a safe node for the first time. Routing is to avoid unsafe nodes which could lead to communication difficulties. Chiu and Wu [35] proposed a fault-free node is defined as an unsafe node if it has either two or more faulty nearest neighbors, a fault-free node that is not unsafe is called a safe node. And, in [31], we can diagnose the faulty nodes in the diagnostic process based on [35], and we redefine them as follows:

(1) A faulty node is level-0 safe

(2) An unsafe node, that has no safe neighbor is level-1 safe

(3) An unsafe node, that has at least one safe neighbor is level-2 safe

(4) A fault-free node that is not unsafe is called a level-3 safe that is a safe node

A cube is fully unsafe if it contains no safety node. Our work in this paper is presented based on the above safe node concept. We introduce the concept of local safety to facilitate fault-tolerant routing. Now, we give an example to illustrate as follows.

The 3-dimensional balanced hypercube as presented in Figure 2. Obviously, we diagnose the solid nodes are faulty called level-0 safe, then we can judge that nodes $(0, 1, 0)$ and $(2, 1, 0)$, are level-1 safe, and nodes $(0, 0, 0)$, $(2, 0, 0)$, $(0, 3, 0)$, $(2, 3, 0)$, $(1, 3, 0)$, $(1, 2, 1)$, $(3, 2, 1)$, $(1, 0, 3)$, $(3, 0, 3)$, $(1, 1, 3)$, $(2, 1, 3)$, $(1, 0, 2)$, $(3, 0, 2)$, $(1, 1, 2)$, $(2, 1, 2)$, $(0, 2, 3)$, $(2, 2, 3)$, $(0, 2, 2)$, and $(2, 2, 2)$ are level-2

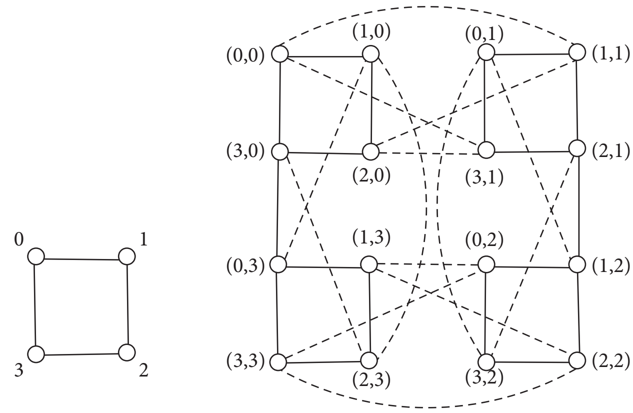

Figure 1: The illustrations of BH_1 and BH_2.

safe, and the other nodes are level-3 safe. The safety status indicates the routing capability of each node, and the safety state of each node affects the safety status of the whole system and then determines the routing efficiency of the whole network. A subgraph $BH_k^{(i)} \subseteq BH_n$, $(0 < k \le n-1$, $i = 0, 1, 2, 3)$ is fully unsafe if it has no safe node; obviously, there are no fully unsafe subgraph in Figure 2.

Next, we list the useful definitions and properties of local safety are proposed by Xiang [32, 33] in order to prove that there are feasible paths under different safety conditions:

Definition 2. (see [32, 33]). The safety level of the SC is defined as follows:

(1) A node in a hypercube is locally unsafe corresponding to a subcube SC if it has at least two faulty neighbors or at least three locally unsafe or faulty neighbors inside SC. Otherwise, it is locally safe corresponding to SC. A subcube is fully unsafe if it contains no locally safe node; otherwise, it is a safe subcube.

(2) Locally unsafe nodes corresponding to a subcube SC is classified as follows: A locally unsafe node is called locally level-2 safe if it has at least one locally safe neighbor (level-3 safe) inside SC; otherwise, it is a locally level-1 safe node.

(3) A node in a hypercube is locally unsafe inside an SC if it has at least 2 faulty neighbors or at least 3 locally unsafe or faulty neighbors inside the SC; otherwise, it is locally safe in the SC. The SC is unsafe if it contains no locally safe node (level-3 safe); otherwise, it is a safe SC. Locally unsafe nodes inside an SC are classified as a locally unsafe node is level-2 safe if it has at least one locally safe neighbor (level-3 safe) in the SC; otherwise, it is a level-2 safe node.

(4) An m-dimensional subcube is defined as a maximal safe subcube if it is safe and any k-dimensional $(k \ge m+1)$ subcube that contains it is fully unsafe.

Proposition 2 (see [32, 33]). *The following properties can be obtained from the above definition:*

(1) A k-dimensional $(k \le n)$ subcube is safe if and only if it contains a fault-free $(k-2)$-dimensional subcube, and all nodes of which have at most one faulty neighbor (level-0 safe).

(2) If a node is locally unsafe in a k-dimensional spanning subcube SC_1, it is still locally unsafe in an m-dimensional $(m > k)$ spanning subcube SC_2 if SC_2 contains SC_1.

(3) Suppose the subcube is an m-dimensional safe subcube in a faulty n-cube $(n \ge m)$. A locally level-1 safe node in SC always has a locally level-2 safe neighbor onside even though the faulty n-cube is fully unsafe.

We can use the following scheme to obtain local safety information for all fault-free nodes concurrently if the BH_n is fully unsafe. For each node $v = (a_0, a_1, \ldots, a_{n-1}) \in BH_n$ check local safety of node v in $(n-1)$-dimensional subgraphs $BH_{n-1}^{(0)}, BH_{n-1}^{(1)}, BH_{n-1}^{(2)}, BH_{n-1}^{(3)}$. If v has at least two faulty neighbors inside a subgraph, then the node is locally unsafe in the subgraph. The node stores local safety information of itself if the local safety information of the subgraph has been obtained and the subgraph is safe. The node also keeps local safety information of its neighbors inside the subgraph.

4. Minimum Feasible Path with a Local Safety Level

The safety status of the spanning balanced hypercube $SBH(s, d)$ has an important effect on routing between s and d and a path is called a feasible path if it travels through no faulty nodes. A feasible path is called minimum if it has the shortest length among all feasible paths between the source and destination nodes. In this section, we will prove some theorems about minimum feasible path with a local safety level of the balanced hypercubes, and these theorems will be used in the routing algorithm in Section 5.

Theorem 1. *There always exists a minimum feasible path from s to d if either s or d is locally safe in $SBH(s, d)$ even though the faulty BH_n is fully unsafe.*

Proof. Without loss of generality, we only consider the case when source s is locally safe. The case when the destination d is locally safe is similar. We use the inductive hypothesis. When $H(s, d) = 1$, messages from s can be routed to d directly, so $P_m(s, d) = 1$. when $H(s, d) = 2$, according to Definition 1 (1) there exists at least one safe neighbor s' in $SBH(s, d)$ because source s is locally safe corresponding to $SBH(s, d)$. Messages in s can be transmitted to destination d via s' along a minimum feasible path and $P_m(s, d) = 2$. Assume the theorem holds when $H(s, d) = k$ and $P_m(s, d) = 1$. We would like to prove the theorem also holds when $H(s, d) = k+1$. Assume the source $s \in BH_{n-1}^{(0)}$, we know that s has $2k$ neighbors (among them there are two extra neighbors) in $2k$ distinct minimum feasible paths between s and d and there exists more than one locally safe neighbor s' of s. In addition, s' is locally safe in $SBH(s', d)$ according to Proposition 1 (1). Whether $s' \in BH_{n-1}^{(0)}$ (see in Figure 3(a)) or $s' \in BH_{n-1}^{(i)}$ $(i = 1, 2, 3)$ (see in Figure 3(b)),

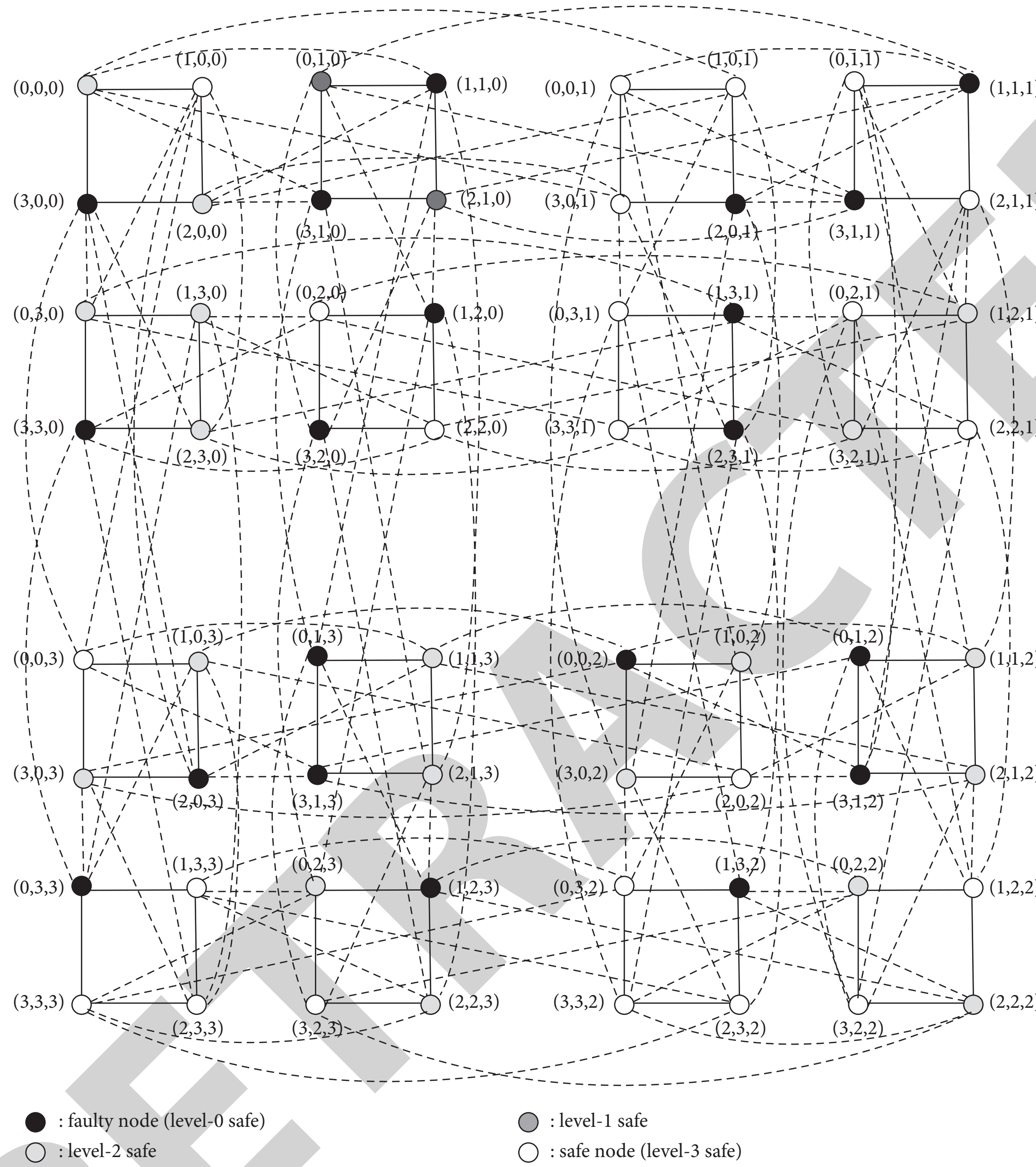

Figure 2: Global safety information of BH_3.

indicating $H(s', s) = 1$. Thus, when $H(s,d) = k+1$ and $P_m(s,d) = H(s,s') + H(s',d) = k+1$, and there exists one minimum feasible path. □

Theorem 2. *Suppose the spanning balanced hypercube* $\mathrm{SBH}(s,d)$ *is safe, where both of s and d are locally unsafe in* $\mathrm{SBH}(s,d)$. *There always exists a minimum feasible path between s and d even if the faulty* BH_n *is fully unsafe.*

Proof. There are two cases based on the different safety levels of unsafe nodes according to Definition 1 (3). □

Case 1. Either s or d is level-2 safe. It is sufficient to only consider the source s is level-2 safe corresponding to $\mathrm{SBH}(s,d)$. The situation is the same when the destination d is level-2 unsafe inside $\mathrm{SBH}(s,d)$. Assume source s is a level-2 safe node; it has a locally safe neighbor s' inside $\mathrm{SBH}(s,d)$ according to Definition 1 (3). Node s' is in one of the minimum feasible paths from s to d according to the premise, which should still be a locally safe node inside $\mathrm{SBH}(s,d)$ according to Proposition 1 (2). The spanning balanced hypercube $\mathrm{SBH}(s,d)$ is safe according to Definition 1 (1) because it contains at least one locally safe node s'. Now, this case of conversion to Theorem 1, and it is easy to find a minimum path from s' to d which length $P_m(s,d) = H(s,d)$, and $P_m(s,d) = H(s,d)$.

Case 2. When both of source s and destination d are locally level-1 safe. The Hamming distance $H(s,d) \geq 3$, when both s and d are locally level-1 safe in their safe spanning balanced

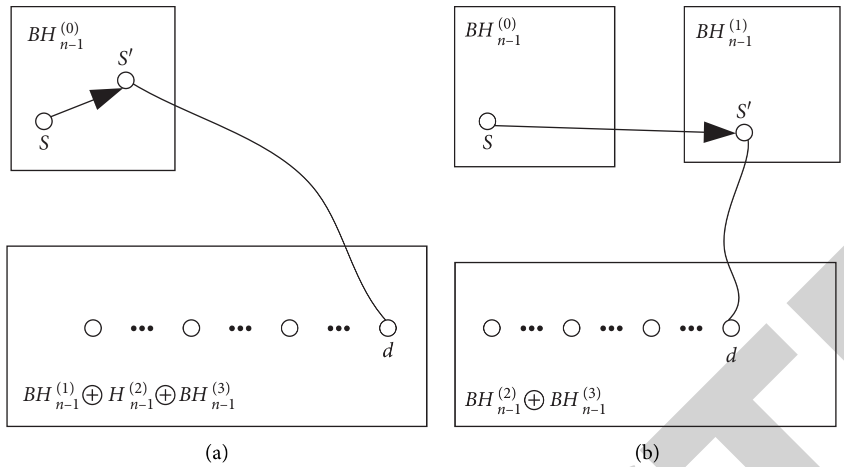

FIGURE 3: The illustration of Theorem 1.

hypercube SBH(s, d). The locally level-1 safe source s has at least one locally level-2 safe neighbor s' inside SBH(s, d) according to Proposition 1 (3) because SBH(s, d) is a safe subgraph. All $H(s, d)$ neighbors of s inside which SBH(s, d) is in one of the minimum paths from s to d. Certainly, s' is in one of the minimum feasible paths from s to d. The node s' has at least one locally level-3 safe neighbor s'' corresponding to SBH(s, d) according to Definition 1 (1). Among the $H(s, d)$ neighbors of s', one is s and the other is $H(s, d)$ -1 distinct. Neighbors are $H(s, d) - 1$ distinct all in minimum paths from s' to d. Certainly, the node s'' is in one of the minimum paths of length $H(s, d) - 2$ from s' to d. It is clear that s'' is also a locally safe node in the spanning balanced hypercube SBH(s'', d) according to Proposition 1 (2) because SBH(s, d) contains SBH(s'', d). Now, in the case of conversion to Theorem 1, it is easy to find a minimum path from s'' to d.

In summary, there always exists a minimum feasible path whose length is $P_m(s, d) = H(s, d)$ from a source s to a destination d if the spanning balanced hypercube SBH(s, d) is safe even though the faulty BH_n is fully unsafe.

Theorem 3. *There always exists a minimum feasible path from a source s to a destination d if the spanning balanced hypercube* SBH(s, d) *is unsafe even though the faulty* BH_n *is fully unsafe.*

Proof. When the source s and destination d are fault-free in an unsafe spanning balanced hypercube SBH(s, d), we only consider the source s. Since, all the neighbors in the path from s to d are not level-3 safe in the SBH(s, d), we need to only find a safe neighbor $s' \in N(s)$ outside of the subcube SBH(s, d) (see in Figure 4) and then find a $s'' \in N(s) - \{s'\}$ in the subgraph. Consequently, the length of the feasible path from s to d is $P_m = H(s, d) + 1$. □

Theorem 4. *There exists a minimum feasible path from source s to destination d if there exists a fault-free node s' along with minimum feasible path from s to d such that both* SBH(s, s') *and* SBH(s', d) *are safe.*

Proof. Let $D(s, d)$ be the set of dimensions that source s and destination d differ and $\backslash\{D_1, D_2\backslash\}$ be a partition of $D(s, d)$. That is to say, $D_1 \cap D_2 = \varnothing$ and $D_1 \cup D_2 = D(s, d)$. A spanning balanced hypercube SBH(s, s') can be generated corresponding to D_1 from source s and another spanning balanced hypercube SBH(s', d) can be established corresponding to D_2. There exists a minimum feasible path from source s to node s' according to Theorem 1 because the spanning balanced hypercube SBH(s, s') is safe. Similarly, there exists a minimum feasible path from s' to destination d because SBH(s', d) is also a safe subgraph. Therefore, there exists a minimum feasible path from source s to destination d. □

5. Fault-Tolerant Routing with Local Information

In this section, we use the above theorems about minimum feasible path to give a corresponding fault-tolerant routing algorithm in balanced hypercube DCNs.

A fault-tolerant routing algorithm is presented based on local safety information. The main idea of this routing scheme is as follows.

Assume information about maximal safe subgraphs of the balanced hypercube is available at each node. When the spanning subgraph of a source and a destination is found to be contained in a maximal safe subgraph, the following algorithm can always pass the message from the source s to the destination d.

However, when a maximal safe subcube that contains the spanning balanced hypercube is not available, the message is sent to a fault-free neighbor s', taking the preference to the one in a minimum feasible path from the source s to the destination d. We first check whether the spanning balanced hypercube SBH(s', d) is contained in a maximal safe subgraph or not. If the spanning balanced hypercube SBH(s', d) is contained in a maximal safe subgraph, a feasible path has been found. The procedure message-pass(s, d, S) passes the message from s to d when it is certain that the message can be routed successfully inside the maximal safe subgraph S

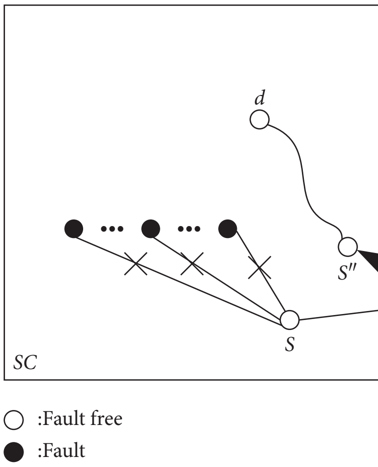

Figure 4: The illustration of Theorem 3.

that contains the spanning subgraph $SBH(s,d)$. Then, according to the current node s and the destination d, we propose the following cases the message delivery:

If $H(s,d)=1$, send the message through a different dimension.

If $H(s,d)=2$, send the message to a fault-free neighbor along with one of the minimum feasible paths from s to d if possible. Otherwise, if a fault-free neighbor in a minimum feasible path from the source to the destination is not available, deroute the message to a proper neighbor.

If $H(s,d)\geq 3$, according to the safety level of the node passes the message as follows:

(1) When the current node s is locally level-1 safe in the subgraph S, send the message to one of its locally level-2 safe neighbors in a minimum feasible path from s to d if possible. Otherwise, send the message to a locally level-1 safe neighbor of s in a minimum path from s to d if possible. When both of the above conditions are not met, according to Theorem 3, send the message to a local level-2 safe neighbor in a nonminimum path from s to d, which is always available.

(2) When s is locally level-2 safe in the subgraph S, send the message to a locally safe neighbor in a minimum feasible path from s to d if possible. Otherwise, send the message to a locally level-2 safe neighbor of s in a minimum feasible path from s to d if possible. When both conditions are not met, according to Theorem 3, send the message to a locally safe neighbor of s in a nonminimum path from s to d.

(3) When s is local safe in the subgraph S, send the message to a local safe neighbor of s in a minimum feasible path from s to d if $H(s,d)\geq 3$; this path is in a minimum feasible path from s to d, and the length of the path is $P_m=H(s,d)$.

We give an example to illustrate that, as shown in Figure 5. It is a fault BH_3 with several faulty nodes. According to the structure of balanced hypercube, the source node s and destination d are both level-2 safe and locally safety in $SBH(s,d)$, and $SBH(s,d)$ is contained in a maximal safety subgraph according to Theorem 4. So, the message can be sent by minimum feasible paths from s to d. They are path 1: $(0,0,0)\longrightarrow(1,0,0)\longrightarrow(0,3,0)\longrightarrow(3,3,1)\longrightarrow(2,2,1)\longrightarrow(1,2,2)\longrightarrow(2,2,2)$; path 2: $(0,0,0)\longrightarrow(1,0,0)\longrightarrow(0,0,3)\longrightarrow(1,1,3)\longrightarrow(2,1,2)\longrightarrow(1,2,2)\longrightarrow(2,2,2)$; path 3: $(0,0,0)\longrightarrow(1,0,0)\longrightarrow(2,3,0)\longrightarrow(3,3,1)\longrightarrow(2,2,1)\longrightarrow(1,2,2)\longrightarrow(2,2,2)$; path 4: $(0,0,0)\longrightarrow(3,0,1)\longrightarrow(0,3,1)\longrightarrow(3,3,1)\longrightarrow(2,2,1)\longrightarrow(1,2,2)\longrightarrow(2,2,2)$; path 5: $(0,0,0)\longrightarrow(1,0,1)\longrightarrow(0,3,1)\longrightarrow(3,3,1)\longrightarrow(2,2,1)\longrightarrow(1,2,2)\longrightarrow(2,2,2)$; and path 6: $(0,0,0)\longrightarrow(1,0,1)\longrightarrow(0,3,1)\longrightarrow(3,3,2)\longrightarrow(2,2,2)$. The length of all paths are no greater than 6. We know that the Hamming distance of the balanced hypercube is 4, so the length of fault tolerant routing is balanced no more than $H(s,d)+2$. In summary, the hops of fault tolerant routing from s to d is always no more than $H(s,d)+2$ by local information.

By the above analysis, we propose Algorithm 1 for fault-tolerant routing with local safety level (see Algorithm 1).

Xiang [32] proposed that the length of fault-tolerant routing is no more than $H(s,d)+4$ in a faulty hypercube, where $H(s,d)$ represents the Hamming distance from the source node s to the destination node d. In this paper, we use the local safety level to propose and prove that, in a faulty balanced hypercube, if there is a neighbor node of s which satisfies the conditions in Section 5, and it appears on the shortest feasible path, we send messages in a minimum feasible path whose length is $H(s,d)$. On the contrary, if neither condition is satisfied, we choose an appropriate nonminimum path to send the messages form s to d and the length of this fault-tolerant routing is at most $H(s,d)+2$ by calculating. Thus, we can know that the length of all paths for sending a message is no more than $H(s,d)+2$. Specifically, when $H(s,d)\leq 2$, the length of fault-tolerant routing is equal to $H(s,d)$ whether it is in hypercube networks or balanced hypercube networks, however, when $H(s,d)\geq 3$, the length of fault-tolerant routing in balanced hypercube networks is two hops less than that in hypercube networks. It shows that the message routing in an injured balanced hypercube network is more efficient than that in an injured hypercube. This indicates the balanced hypercube is more suitable for increasingly complex network fault-tolerant routing.

In order to make it more intuitive, we give a comparison diagram with the conclusion of [32] (see Figure 6). The contrast diagram expresses that, when $H(s,d)\leq 2$, the length of fault tolerant routing is equal to $H(s,d)$ whether it is in hypercube networks or in balanced hypercube networks;

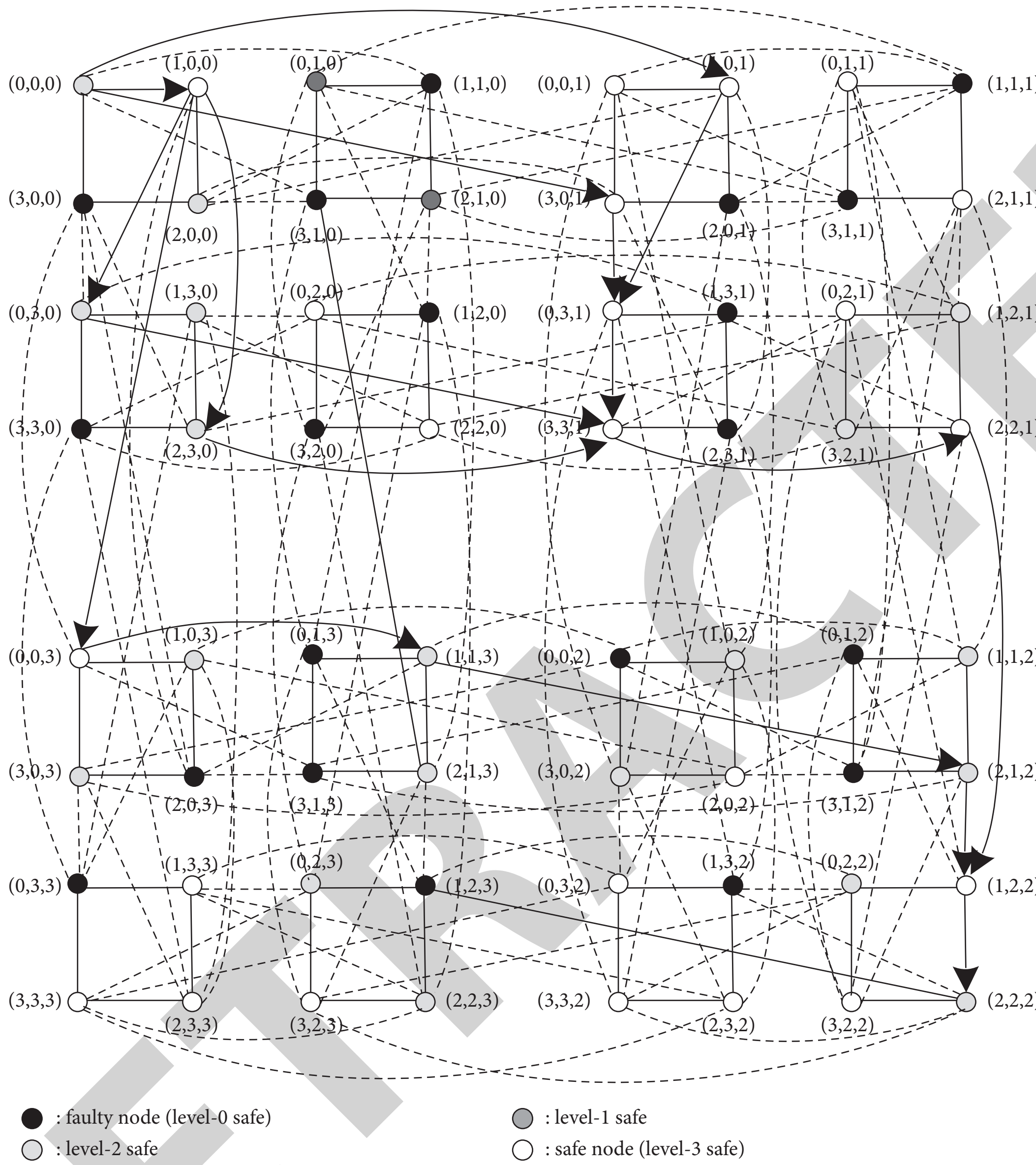

Figure 5: The illustration of nonminimum path routing.

Input: send the message from the source node s to destination node d in a fully unsafe BH_n.
Output: a fault-tolerant routing call message-pass(s, d, msg)-.
for source s and destination d are locally safe in SBH(s, d) **do**
  if the SBH(s, d) is contained in a maximal safe subgraph msg; **then**
    The message is sent by the minimum feasible path from s to d;
  else
    Select a neighbor s' along with a minimum feasible path from s to d where SBH(s', d) is contained in a maximal safe subgraph msg, pass the message to s', and pass the message;
  end if
end for
return message-pass(s, d, msg);

Algorithm 1: Fault-tolerant routing with local safety level.

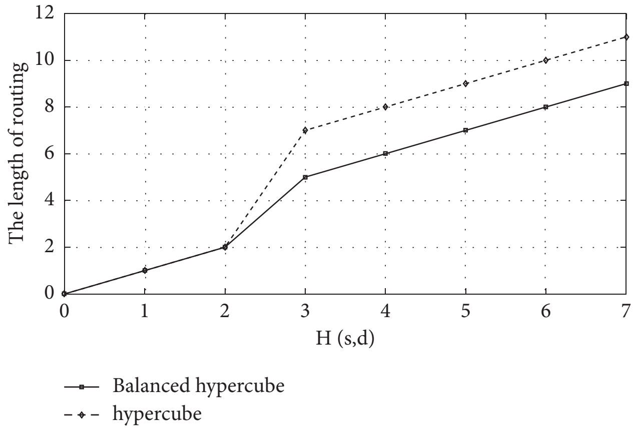

Figure 6: Fault-tolerant routing length contrast diagram.

however, when $H(s,d) \geq 3$, we can see that the longest length of fault tolerant routing in balanced hypercube networks is two hops less than that in hypercube networks; it means the message routing in a hurt balanced hypercube network is more efficiency than that in an injured hypercube. This indicates that the balanced hypercube is more suitable for increasingly complex network fault-tolerant routing.

6. Conclusion

When considering a model for a large-scale data center network, it is worth mentioning that as a deformation of a hypercube; the balanced hypercube is a better fit for its strong connectivity, regularity, and a smaller diameter. We propose the balanced hypercube as a model for the DCN and design an efficient fault-tolerant security routing of BH_n-based DCNs, in which the shortest feasible paths are located such that the reliable routing is achieved. Then, we claim that the length of fault-tolerant security routing of data center networks based on balanced hypercube is always no more than $H(s,d) + 2$; in other words, our fault-tolerant secure routing scheme is more effective in the same reliable network environment of DCNs.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the National Natural Science Foundation of China (nos. U1905211, 62 171 132, and 62 102 088).

References

[1] S. Anthony, "Microsoft now has one million servers-less than Google, but more than Amazon, says Ballmer, ExtremeTech.com," 2013.

[2] X. Xue and J. Zhang, "Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm," *Applied Soft Computing*, vol. 106, pp. 1–11, 2021.

[3] X. Xue, C. Yang, C. Jiang, P.-W. Tsai, G. Mao, and H. Zhu, "Optimizing ontology alignment through linkage learning on entity correspondences," *Complexity*, vol. 2021, Article ID 5574732, 12 pages, 2021.

[4] X. Xue, X. Wu, C. Jiang, G. Mao, and H. Zhu, "Integrating sensor ontologies with global and local alignment extractions," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6625184, 10 pages, 2021.

[5] J. Cao, X. Wang, M. Huang, B. Yi, and Q. He, "A security-driven network architecture for routing in industrial Internet of Things," *Transactions on Emerging Telecommunications Technologies*, vol. 32, pp. 1–22, 2021.

[6] X. Li, J. Fan, C. Lin, and X. Jia, "Diagnosability evaluation of the data center network DCell," *The Computer Journal*, vol. 61, pp. 1–15, 2018.

[7] Q. Zhou, J. Yu, and D. Li, "An adaptive authenticated model for big data stream SAVI in SDN-based data center networks," *Security and Communication Networks*, vol. 2021, Article ID 5451820, 14 pages, 2021.

[8] G. Ge-Ming Chiu and K. Kai-Shung Chen, "Use of routing capability for fault-tolerant routing in hypercube multicomputers," *IEEE Transactions on Computers*, vol. 46, no. 8, pp. 953–958, 1997.

[9] D. Pao and C. Chan, "Fault-tolerant message routing in hypercube using local neighborhood information," in *Proceedings of the Tencon 94 IEEE Region 10s Ninth International Conference Theme: Frontiers of Computer Technology*, pp. 440–444, Singapore, August 1994.

[10] X. Xiao, G. Wang, and J. Chen, *Fault-Tolerant Routing Algorithm in Hypercube Networks with Load Balancing Support*, Springer, Berlin Heidelberg, 2004.

[11] D. T. Duong and K. Kaneko, "Fault-tolerant routing based on approximate directed routable probabilities for hypercubes," *Future Generation Computer Systems*, vol. 37, pp. 88–96, 2014.

[12] J. Park, Y. Hirai, and K. Kaneko, "Fault-tolerant routing in dual-cubes based on routing probabilities," *Procedia Computer Science*, vol. 69, pp. 66–75, 2015.

[13] J. Xu, M. Ma, and Z. Du, "Edge-fault-tolerant properties of hypercubes and folded hypercubes," *Australasian Journal of Combinatorics*, vol. 25, pp. 7–16, 2006.

[14] Q. Mohammad, "Adaptive fault tolerant routing algorithm for tree-hypercube multicomputer," *Journal of Computer Science*, vol. 2, no. 2, pp. 124–126, 2006.

[15] R. Penaranda, E. Gran, T. Skeie, M. Gomez, and P. Lopez, "A new fault-tolerant routing methodology for KNS topologies," in *Proceedings of the IEEE 2nd International Workshop on High Performance Interconnection Networks Towards the Exascale and BigData Era*, Barcelona, Spain, March 2016.

[16] D. Dong Xiang and A. Ai Chen, "Fault-tolerant routing in 2D tori or meshes using limited-global-safety information," in *Proceedings of the International Conference on Parallel Processing*, Wakamatsu, Japan, September 1999.

[17] D. Xiang, B. Li, and Y. Fu, "Fault-tolerant adaptive routing in dragonfly networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 259–271, 2019.

[18] R. Pearanda, M. G. E. Gunnargran, and T. Skeie, "A fault tolerant routing strategy for k-ary n-direct s-indirect topologies based on intermediate nodes," *Concurrency and Computation: Practice and Experience*, vol. 29, pp. 1–11, 2017.

[19] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell," *ACM SIGCOMM - Computer Communication Review*, vol. 38, no. 4, pp. 75–86, 2008.

[20] M. Kliegl, J. Lee, J. Li, and X. Zhang, "Generalized DCell structure for load-balanced data center networks," in *Proceedings of the IEEE INFORCOM 2010 Proceedings*, San Diego, CA, USA, August 2010.

[21] M. Manzano, K. Bilal, E. Calle, and S. U. Khan, "On the connectivity of data center networks," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2172–2175, 2013.

[22] X. Wang and J. Fan, "A node-to-set disjoint path routing algorithm in DCell networks," in *Proceedings of the 2014 Second International Conference on Advanced Cloud and Big Data*, Huangshan, China, November 2014.

[23] X. Wang, A. Erickson, J. Fan, and X. Jia, "Hamiltonian properties of DCell networks," *The Computer Journal*, vol. 58, no. 11, pp. 2944–2955, 2015.

[24] X. Wang, J. Fan, J. Zhou, and C.-K. Lin, "The restricted h-connectivity of the data center network DCell," *Discrete Applied Mathematics*, vol. 203, pp. 144–157, 2016.

[25] X. Wang, J. Fan, C.-K. Lin, and X. Jia, "Vertex-disjoint paths in DCell networks," *Journal of Parallel and Distributed Computing*, vol. 96, pp. 38–44, 2016.

[26] M.-M. Gu, R.-X. Hao, and J.-B. Liu, "The pessimistic diagnosability of data center networks," *Information Processing Letters*, vol. 134, pp. 52–56, 2018.

[27] M. Lv, S. Zhou, X. Sun, G. Lian, and J. Liu, "Reliability evaluation of data center network DCell," *Parallel Processing Letters*, vol. 28, no. 4, 2018.

[28] J. Jie Wu and K. Ke Huang, "The balanced hypercube: a cube-based system for fault-tolerant applications," *IEEE Transactions on Computers*, vol. 46, no. 4, pp. 484–490, 1997.

[29] D. Cheng, "Cycles embedding in balanced hypercubes with faulty edges and vertices," *Discrete Applied Mathematics*, vol. 238, pp. 56–69, 2018.

[30] S. Zhang, X. Zhang, and Y. Wang, "Feedback numbers of balanced hypercubes BH_n," in *Proceedings of the 14th Internetional Conference on Computational Intelligence and Security*, Huangshan, China, November 2018.

[31] X. Zhang, L. Xu, L. Lin, Y. Huang, and X. Wang, "The (t, k) -diagnosability of balanced hypercube under the PMC model," *International Journal of Computer Mathematics: Computer Systems Theory*, vol. 3, no. 4, pp. 230–243, 2018.

[32] D. Dong Xiang, "Fault-tolerant routing in hypercube multicomputers using local safety information," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 9, pp. 942–951, 2001.

[33] D. Cheng, R.-X. Hao, and Y.-Q. Feng, "Two node-disjoint paths in balanced hypercubes," *Applied Mathematics and Computation*, vol. 242, pp. 127–142, 2014.

[34] T. C. Lee and J. P. Hayes, "A fault-tolerant communication scheme for hypercube computers," *IEEE Transactions on Computers*, vol. 41, no. 10, pp. 1242–1256, 1992.

[35] G. Ge-Ming Chiu and P. Shui-Pao Wu, "A fault-tolerant routing strategy in hypercube multicomputers," *IEEE Transactions on Computers*, vol. 45, no. 2, pp. 143–155, 1996.