

Hamilton Connectivity Analysis of Data Center Network Based on Meta-structure Network

Hui Li, Linyu Yang, Xiaoyu Du

Abstract—Hamilton-connected is essential property of data center network. In this paper, we propose $X(n, k)$ and research under what conditions $X(n, k)$ is Hamiltonian-connected, which is small specific structures to extend large data center network architectures. Secondly, We study the Hamilton connectivity of $\text{BCube}(n, k)$ in all cases and some other DCN (Data Center Network) structures by using $X(n, k)$ theorem and inference. Finally, we propose $O(\log k * n^k)$ time complexity algorithm to find the Hamiltonian path between any two distinct servers in $\text{BCube}(n, k)$ ($n \geq 3$ and $k \geq 0$).

Index Terms—Data center network, $\text{BCube}(n, k)$, Hamiltonian-connected, Meta-structure network

I. INTRODUCTION

CLOUD computing has seen extensive adoption across both industrial and academic sectors due to its innovative model for organizing computing resources, which offers significant advantages[1]. As the cornerstone of cloud computing, data centers, play a key role in determining the performance of cloud computing and furthering the economical ecosystem. Therefore, it has become a hot topics in this focused areas[2]. Driven by rapid technological progress, major online service providers—including Google, Amazon, and Microsoft—have constructed massive data centers comprising tens of thousands of servers to serve as their core infrastructure. Within these facilities, the performance of the data center network is particularly vital. These large-scale systems enable the dynamic allocation of vast hardware, software, and database resources to millions of simultaneous Internet users. As the amount of data grows, we need more storage capacity to store the data[3].

The Hamiltonian is one of the most basic requirements when designing the data center (DCN) network. For decades, the Hamiltonian proof of network structure has been widely studied. The network with the Hamiltonian path or the Hamiltonian circle can efficiently carry out linear communication[4]. However, there is no necessary and sufficient condition to judge whether a graph is a Hamiltonian. In fact, the problem of finding Hamiltonian paths or cycles is NP-complete[5]. Therefore, many problems

in this field still need mathematical proof by professional researchers.

In recent years, there have been many Hamiltonian analyses of specific networks. Fan et al. studied the Hamiltonian-connected of HSDC and proposed that when $n \geq 3$, H_n is Hamiltonian-connected[6]. When $n \geq 3$, Möbius cubes is Hamiltonian-connected[7]. On the other hand, network failure needs also to be considered, because it is unavoidable. As to k -ary n -cube. Fan et al. studied Hamiltonian cycles and path embedding in k -ary n -cubes fault structures ($H = k_1, k_{1,1}, k_{1,2}, k_{1,3}$)[8]. If the number of fault elements does not exceed $n - 2$, and each fault element is isomorphic to any connected subgraph of $K_{1,3}$, then there is a fault free Hamiltonian path connecting the two nodes[9].

In fact, Hamiltonian path or Hamiltonian cycle has important applications in network information communication, which is used in two-path and multi-path multicast algorithms to reduce or avoid congestion and deadlock. For example, the multicast routing algorithm based on Hamiltonian model without deadlock and other additional resource requirements is more efficient than the traditional multicast routing algorithm based on multicast tree[10]. In addition, Hamiltonian properties can also be applied to reduce the number of fault diagnosis and improve the efficiency of fault diagnosis. Fujita and Araki et al. completed fault diagnosis of the network in up to three rounds of tests by using Hamiltonian cycles and paths in binary n -cubes [11]. Ye and Liang et al. proposed a five-round adaptive diagnosis scheme for hamitonian cycle networks[12]. Fan et al. propose a network structure BC, and analyze its Hamiltonian and diagnosability[13].

Hamiltonian cycles are also used in practical problems, such as the online optimization of complex flexible manufacturing systems[14] and wormhole routing[15]. These applications have also stimulated scholars' interest in studying Hamiltonian, and some results have been obtained. For example, Wang et al. studied the Hamiltonian connectivity of DCell[16]. Liu et al. found HCN-Cubes are Hamiltonian-connected[17]. Yang et al. proved the balanced hypercubes BH_n is $(2n - 2)$ -fault-prescribed hamiltonian laceable[18]. Abdallah and Cheng look at the case when the 3-cycles form a “tree-like structure”, and analyze the fault-Hamiltonian connectivity of such graphs. They prove that these graphs are $(2n - 7)$ -fault-tolerant Hamiltonian-connected[19]. Dong et al. research the fault-tolerant hamiltonian connectivity of RHLNs under the conditional fault model[20].

In this paper, we use the idea of mathematical recursion to make the Hamilton connectivity of a specific DCN, propose and prove the meta-structure network $X(n, k)$ and its Hamilton connectivity, which is a switch structure connecting multiple servers.

Manuscript received June 9, 2025; revised October 19, 2025.

This work was supported in part by the Special Project for Key R&D and the Promotion of Science, Technology Department of Henan Province (232102210071, 232102211009, 242102210202, 242102210196), Kaifeng Science and Technology Development Plan (2201010).

Hui Li is an associate professor at the School of Information Engineering, Henan Kaifeng College of Science Technology and Communication, Kaifeng 475001, China (e-mail: jwc@hmc.edu.cn).

Linyu Yang is a teaching assistant at the School of Information Engineering, Henan Kaifeng College of Science Technology and Communication, Kaifeng 475001, China (corresponding author to provide e-mail: yanglinyu@henu.edu.cn).

Xiaoyu Du is a professor at the School of Computer and Information Engineering, Henan University, Kaifeng 475001, China (e-mail: dxy@henu.edu.cn).

The contributions of this paper are as follows:

- 1) We propose and prove that the network is Hamiltonian-connected if the network structure based on meta-structure network $X(n, k)$ type conforms to a few of practical restrictions (Section 3 will describe in detail).
- 2) We analyze $X(n, k)$ Hamiltonian-connected in condition (1), and propose the theorems to judge $X(n, k)$ that it is no Hamiltonian-connected graph: if $X(n, k)$ degree is 2, and $X(n, k)$ contains m $X(n, k - 1)$ ($m > 3$), then $X(n, k)$ does not Hamiltonian graph. In addition, if the minimum edge of the edge cut set in $X(n, k)$ is less than 3, then $X(n, k)$ also does not Hamiltonian-connected graph.
- 3) We use the above properties of $X(n, k)$ to study the Hamiltonian connectivity of a particular network. In this paper, we study the Hamiltonian connectivity in any case in FiConn(4, 1), EH(1, 2) and BCube(n, k), where n and k in BCube(n, k) can take any value.
- 4) We design three specific Hamiltonian algorithm for finding any two different nodes Hamiltonian path in BCube(n, k) ($n \geq 3, k \geq 0$).

The rest of this paper is organized as follows. Section 2 introduces the preliminaries. And then Section 3 proves Theorems and Inferences. In Section 4, according to the theorem and inference of $X(n, k)$, we prove the Hamiltonian connectivity of BCube(n, k) under different conditions, and propose a Hamiltonian path algorithm to find any two nodes. Finally, we make a conclusion in Section 5.

II. PRELIMINARIES

The topology of a data center network (DCN) is naturally represented by a graph, where vertices denote servers and edges represent communication links between them. In this way, the study of data center network topology can be regarded as the problem of graph structure. Graph is a mathematical model of data center network topology structure, therefore the properties and parameters of graph can be used to measure the performances of data center network topology structure. In different literatures, there are differences between the description of graph's concepts and the representation of symbols. To distinguish, there are some basic concepts and symbolic representations of graph theory related to data center networks.

The data center network can be represented as a simple graph $G = (V(G), E(G))$. We use $V(G)$ and $E(G)$ to represent vertex sets and edge sets in graph G . Vertices and edges to represent servers and links in data center networks respectively, while switches are considered transparent network devices.

Graph G is composed a nonempty finite set V of vertices and the combination of binary subset E is defined by V , where V is called vertex set, each element in E is called edge of G , V and E are called vertex set and edge set of graph G respectively, if $V(G) \times V(G)$ is a set of ordered vertex pairs, graph G is called a digraph, else $V(G) \times V(G)$ is a set of disordered vertex pairs, the graph G is called an undirected graph, and the graphs are undirected graphs in this paper.

The following are the basic concepts of graph theory used in this paper:

Hamiltonian path: A path that passes through each vertex only once is called a Hamiltonian path.

Hamiltonian cycle: A closed Hamiltonian path is called a Hamiltonian circuit.

Hamiltonian graph: A Hamiltonian Graph is a connected graph that contains a Hamiltonian cycle.

Hamiltonian-connected graph: The graph G is Hamiltonian-connected if there is a Hamiltonian path between any two distinct vertices in the graph G .

Node degree: The number of edges associated with this node.

Cut vertex: Let G be a connected graph. A vertex $V \subseteq G$ is called a cut vertex of G , if $G - V$ (Delete V from G) results in a disconnected graph. Removing a cut vertex from a graph breaks it in to two or more graphs.

Cut edge: Let G be a connected graph. An edge $E \subseteq G$ is called a cut edge if $G - E$ results in a disconnected graph.

Complete graph: For undirected graph in which each pair of distinct vertices is connected by exactly one edge. $X(n, k)$ is also complete graph, which is a basic unit of study in this article. $X(n, k)$ has one switch connects to n servers.

In this paper, we first proposed a meta-structured network called $X(n, k)$, where n represents the number of servers and k represents the number of $X(n, k)$ recursive layers. We find that a data center network has Hamiltonian-connected if: 1) its architecture conforms to a combination of $X(n, k)$ and its recursive units. 2) Its sub-structure $X_i(n, k - 1)$ is connected with at least one edge of $X_j(n, k - 1)$ ($i \neq j$) and the minimum cut set of $X(n, k)$ is greater than or equal to 4.

The method is to think recursively in mathematics. The properties of Hamilton connectivity when $k = 0$, $X(n, 0)$ are analyzed first, then the properties of Hamilton connectivity when $k = 1$, $X(n, 1)$ are analyzed. Finally, the cases of $k > 1$, $n > 2$ are expanded. Each progressive proof process is analyzed with reference to the previous conclusions. This part is in Theorem 1 of Chapter 3 and its derivation.

Then we proved, If $X(n, k)$ contains a node with degree 2 and the amount of nodes in $X(n, k)$ is more than 3, then $X(n, k)$ does not Hamiltonian structure network. And lemma 1 is derived: If $\min_{F \in \mathcal{F}} |F| \leq 2$, there is no Hamiltonian connectivity in structure $X(n, k)$ (F_i is a cut of graph E and $\mathcal{F}$ is the set of all cuts. $|F|$ is the number of element in the cut). We also use these methods on specific networks. FiConn(4, 1), EH(1, 2) and ExCCC(1, 2) use the above methods to prove that they do not Hamilton-connected.

We focus on studying the Hamilton connectivity of the structure of BCube(n, k) network, and generalize the theorem and inference obtained by $X(n, k)$ to BCube(n, k), so that we know that 1) For any integer n with $n \geq 2$ and $k = 0$, BCube($n, 0$) is Hamilton-connected. 2) For any integer n and k with $n < 3$ and $k > 0$, BCube(n, k) is not Hamilton-connected. 3) For any integer n and k with $n \geq 3$ and $k > 0$, BCube(n, k) is Hamiltonian-connected.

Finally, we designed an algorithm to find the Hamiltonian path between any two points of BCube(n, k), using the idea of recursion. This part is described in detail in section 4.3.

III. HAMILTONIAN-CONNECTED ANALYSIS

Theorem 1. The network structure which formed from meta-structure network $X(n, k)$ is Hamiltonian-connected,

if it has the following three properties:

- 1) Recursively, meta-structure network $X(n, k)$ is built from m copies of $X(n, k-1)$ ($m > 2$), and $X(n, 0)$ is mini Cube(n). Figure 1 shows the structure of $X(4, 0)$.
- 2) $\forall X(n, k)$, there is at least one edge between $X^i(n, k-1)$ and $X^j(n, k-1)$. $X^i(n, k-1)$, $X^j(n, k-1)$ have disjoint edges. ($X^i(n, k-1)$, $X^j(n, k-1) \in X(n, k)$).
- 3) $\mathcal{F}$ is the set of all $X(n, k)$'s cut, that is to say $\mathcal{F} = \{F_1, F_2, \dots, F_\lambda\}$, $|\bullet|$ is the number of element in the set and the following inequality is true.

$$\min_{F_i \in \mathcal{F}} |F_i| \geq 4 \quad (1)$$

Proof. First we derive and prove the Hamiltonian-connected with $n = 1$ or 2 and $k = 0$ or 1. Second we prove an induction step which holds for $X(n, k)$.

Let $n = 1$, then only one switch connects to one server. Obviously $X(1, k)$ is not Hamiltonian-connected and which does not make sense in the data center network structure.

So the case of $n > 1$ is discussed below:

(1) For $k = 0$, $X(n, 0)$ is the basic element for $X(n, k)$, suppose $X(n, 0)$ nodes labeled $(0, 0)$, $(0, 1) \dots (0, n-1)$, according to the nature of the network, any two nodes are connected by edges.

$S_i = (i, 0)$ and $S_j = (j, 0)$ are two arbitrary nodes in $X(n, 0)$, where $i \neq j$ & $i, j \in 0, 1, \dots, n-1$. Path sequence $\pi = [\pi_0, \pi_1, \dots, \pi_{n-3}]$ is an random permutations of array $[0, \dots, i-1, i+1, \dots, j-1, j+1, \dots, n-1]$, $\pi_k \neq i$, $\pi_k \neq j$, and $\pi_k \in \pi$. π_k represent the address of the node, that is, the elements in π can represent the order of the nodes that the network path passes through. For example, the π is any arrangement of set $\{0, 1, 2, 3\}$, then $\pi = \{1, 3, 2, 0\}$ is one of the permutations, that is, $\pi_0 = 1$, $\pi_1 = 3$, $\pi_2 = 2$, $\pi_3 = 0$.

Thus based on the analysis process described above, we have the Hamiltonian path from S_i to S_j is $S_i \rightarrow S_{\pi_0} \rightarrow S_{\pi_1} \rightarrow \dots \rightarrow S_{\pi_{n-3}} \rightarrow S_j$.

So $X(n, 0)$ is Hamiltonian-connected. Figure 1 is the structure and corresponding graph of $X(4, 0)$, Table 1 is the Hamiltonian path for $X(4, 0)$.

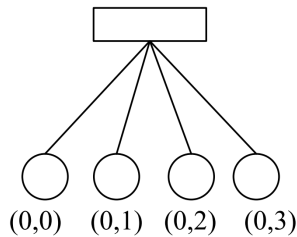


Fig. 1. Structure and corresponding graph of $X(4, 0)$.

TABLE 1
HAMILTONIAN PATH OF $X(4, 0)$

Node	Path
$(0,0) \rightarrow (0,1)$	$(0,0) \rightarrow (0,3) \rightarrow (0,2) \rightarrow (0,1)$
$(0,0) \rightarrow (0,2)$	$(0,0) \rightarrow (0,1) \rightarrow (0,3) \rightarrow (0,2)$
$(0,0) \rightarrow (0,3)$	$(0,0) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (0,3)$
$(0,1) \rightarrow (0,2)$	$(0,1) \rightarrow (0,0) \rightarrow (0,2) \rightarrow (0,3)$
$(0,2) \rightarrow (0,3)$	$(0,2) \rightarrow (0,1) \rightarrow (0,0) \rightarrow (0,3)$

Our goal is to prove that there is a Hamiltonian path between any pair of distinct vertices, $u, v \in V$, and we

consider two cases: either u and v are in the same copy of $X(n, k-1)$, or else they are in distinct copy of $X(n, k-1)$. It is obvious that $X(n, k)$ uses a recursively defined structure to interconnect servers according to Theorem 1, so high-level $X(n, k)$ recursively forms many low-level ones. Let $X^i(n, k-1)$ denote a level $k-1$ structure and i denotes the i th copy.

(2) For $k = 1$, $X(n, 1)$ is built from m copies of $X(n, 0)$ and it achieves three conditions in the Theorem 1. The second condition of Theorem 1 tells us, there is at least one edge between $X^i(n, 0)$ and $X^j(n, 0)$. If $X(n, 0)$ is regarded as a node, $X(n, 1)$ will be equivalent to m interconnected nodes, that is $X(m, 0)$. So according to (1), the network structure after the equivalence has Hamiltonian. Let u and v be any two distinct nodes in $X(n, 1)$, and prove the Hamiltonian of $X(n, 1)$ in two cases.

Case 1. $u, v \in X^{i_0}(n, 0)$, where $i_0 \in (0, 1, \dots, m-1)$, that is to say, u and v are in the same copy of $X(n, 0)$. For each pair $X(n, 0)$, $(X^i(n, 0), X^j(n, 0))$, they are connected to each other, so a new network structure graph G_1 is formed after $X^{i_0}(n, 0)$ is equivalent to a node $X_0^{i_0}$. Then the new network G_1 is equivalent to the network $X(m, 0)$, in which a $(X_0^{i_0}, X_0^{i_{m-1}})$ -Hamiltonian path $\langle X_0^{i_0}, X_0^{i_1}, \dots, X_0^{i_{m-2}}, X_0^{i_{m-1}} \rangle$ exists, where $\langle i_0, i_1, \dots, i_{m-1} \rangle$ is permutation of array $[m-1, m-2, \dots, 1, 0]$.

We can know from the condition 3 of Theorem 1, the degree of the node is more than or equal to 4, and there are at least 4 edges between any $X(n, 0)$ and other copies. α and β are two nodes, and $\alpha, \beta \in X^{i_0}(n, 0)$. There is a edge between α and the substructure $X^{i_1}(n, 0)$ and other edges between β and $X^{i_{m-1}}(n, 0)$ too. $X_0^{i_0}(n, 0)$ is the basic element in the network and any two nodes are interconnected, so there is a Hamiltonian path $\langle u, S_{\pi_0}, S_{\pi_1}, \dots, \alpha, \beta, \dots, S_{\pi_{n-5}}, v \rangle$ in substructure $X^{i_0}(n, 0)$, where S_{π_k} is the node of network $X^{i_0}(n, 0)$. Thus there must be a Hamiltonian path in $X(n, 1)$ which is $\langle (u, S_{\pi_0}, S_{\pi_1}, \dots, \alpha), p_{i_1}, \dots, p_{i_{m-1}}, (\beta, \dots, S_{\pi_{n-5}}, v) \rangle$, where p_i is the Hamiltonian path in $X^i(n, 0)$. (As shown in Figure 2)

Case 2. u and v are in distinct copies of $X(n, 0)$, $u \in X^{i_0}(n, 0)$, $v \in X^{i_{m-1}}(n, 0)$. According to the previous analysis, the equivalent network structure graph has a Hamiltonian path $\langle X_0^{i_0}, X_0^{i_1}, \dots, X_0^{i_{m-1}} \rangle$. It can be known from the condition 2 of Theorem 1, there is at least one edge between $X^i(n, 0)$ and $X^j(n, 0)$. Let α and β be two nodes which are connected by edges to vertices in $X^{i_1}(n, 0)$ and $X^{i_{m-2}}(n, 0)$ respectively, where $\alpha \in X^{i_0}(n, 0)$ and $\beta \in X^{i_{m-1}}(n, 0)$. We have proved that $X^i(n, 0)$ has Hamiltonian path between any two nodes, so there is a Hamiltonian path that include both paths $\langle u, S_{\pi_0}, S_{\pi_1}, \dots, S_{\pi_{n-3}}, \alpha \rangle$ and $\langle \beta, S_{\pi_0}, S_{\pi_1}, \dots, S_{\pi_{n-3}}, v \rangle$ in $X^{i_0}(n, 0)$ and $X^{i_{m-1}}(n, 0)$ respectively. Thus there is a Hamiltonian path between u and v , $\langle (u, p_{i_0}, \alpha), p_{i_1}, \dots, p_{i_{m-2}}, (\beta, p_{i_{m-1}}, v) \rangle$. (As shown in Figure 3)

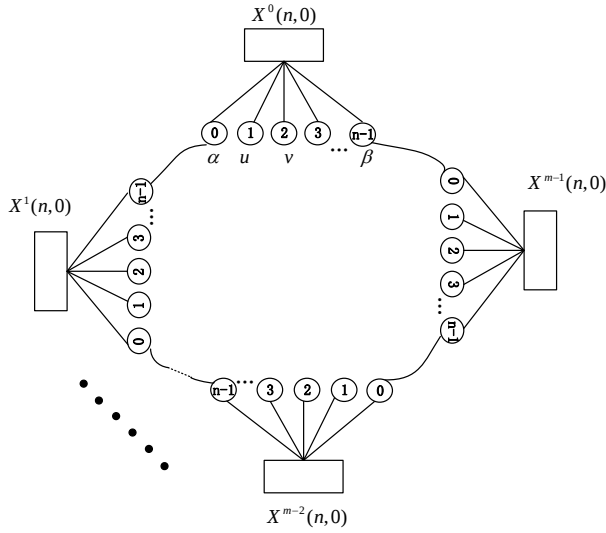


Fig. 2. For $k = 1$, both u and v in same substructure.

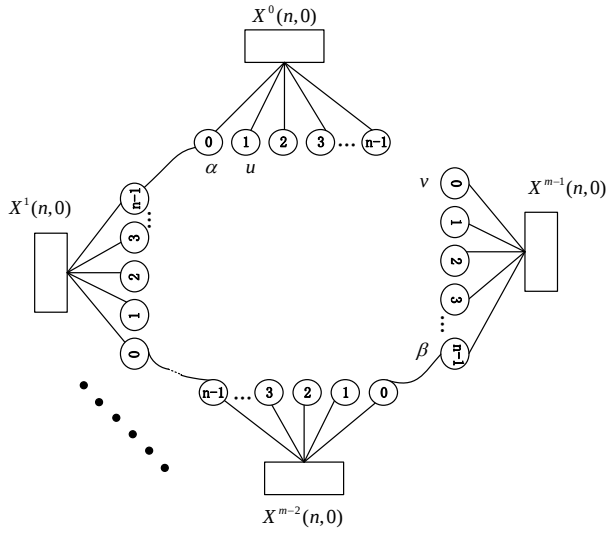


Fig. 3. For $k = 1$, both u and v in distinct substructure.

(3) For $k > 1$, $n > 2$, the Hamiltonian connectivity of $X(n, k)$ is proved by mathematical induction. From the above analysis, it can be seen that for $k = 0$ or 1 , $n > 1$, $X(n, k)$ is Hamiltonian-connected. Assuming that $k = \tau - 1$ ($\tau \geq 2$), the Theorem 1 proposition is established, then the Theorem 1 will be proved for $k = \tau$. we can know from the condition 1 of Theorem 1 that $X(n, \tau)$ is built from m copies of $X(n, \tau - 1)$, which are $X^0(n, \tau - 1), X^1(n, \tau - 1), \dots, X^{m-1}(n, \tau - 1)$ ($m > 2$). Similarly, $X(n, \tau - 1)$ is built from m copies of $X(n, \tau - 2)$, which are $X^0(n, \tau - 2), X^1(n, \tau - 2), \dots, X^{m-1}(n, \tau - 2)$ ($m > 2$). Because when $k = \tau - 1$, the Theorem 1 proposition is established, for each $X^0(n, \tau - 1), X^1(n, \tau - 1), X^2(n, \tau - 1), \dots, X^{m-1}(n, \tau - 1)$, there is a Hamiltonian path between any two nodes. Then if $X(n, \tau - 1)$ is regarded as a node, $X(n, \tau)$ will be equivalent to m interconnected nodes. Similarly, if $X(n, \tau - 2)$ is regarded as a node, $X(n, \tau - 1)$ will be equivalent to m interconnected nodes. Let u and v be any two distinct nodes in $X(n, \tau)$, and prove the Hamiltonian of $X(n, \tau)$ in two cases.

Case 1. $u, v \in X^{i_0}(n, \tau - 1)$, and $u \in X^{l_0}(n, \tau - 2)$, $v \in X^{l_{m-1}}(n, \tau - 2)$, where $i_0, l_0 \in (0, 1, \dots, m - 1)$, and

$X^{l_0}(n, \tau - 2), X^{l_{m-1}}(n, \tau - 2) \in X^{i_0}(n, \tau - 1)$, that is to say, u and v are in the same copy of $X(n, \tau - 1)$, but they are in distinct copies of $X(n, \tau - 2)$. We can know from the condition 3 of Theorem 1, the degree of the node is bigger than or equal to 4, and there are at least 4 edges between any $X(n, \tau - 2)$ and other copies. α and β are two nodes, and $\alpha \in X^{l_i}(n, \tau - 2), \beta \in X^{l_{i+1}}(n, \tau - 2)$. There is an edge between α and the structure $X^{i_1}(n, \tau - 1)$ and other edges between β and $X^{i_{m-1}}(n, \tau - 1)$ too.

From the previous analysis, $X^{i_0}(n, \tau - 1)$ and $X^{l_0}(n, \tau - 2)$ can be equivalent to nodes $X_{\tau-1}^{i_0}$ and $X_{\tau-2}^{l_0}$. And we assume that when $k = \tau - 1$, Theorem 1 is true, so there is a $(X_{\tau-2}^{l_0}, X_{\tau-2}^{l_{m-1}})$ -Hamiltonian path $\langle X_{\tau-2}^{l_0}, X_{\tau-2}^{l_1}, \dots, X_{\tau-2}^{l_i}, X_{\tau-2}^{l_{i+1}}, \dots, X_{\tau-2}^{l_{m-1}} \rangle$ in structure $X^{i_0}(n, \tau - 1)$. Furthermore, this Hamiltonian path can also be expressed as $\langle p_{l_0}, p_{l_1}, \dots, p_{l_i}, p_{l_{i+1}}, \dots, p_{l_{m-1}} \rangle$, where p_{l_0} is a Hamiltonian path starting with u in $X^{l_0}(n, \tau - 2)$, p_{l_i} is a Hamiltonian path in $X^{l_i}(n, \tau - 2)$, and $p_{l_{m-1}}$ is the Hamiltonian path with the end of v in $X^{l_{m-1}}(n, \tau - 2)$. Then in the equivalent network there is a $(X_{\tau-1}^{i_0}, X_{\tau-1}^{i_{m-1}})$ -Hamiltonian path $\langle X_{\tau-1}^{i_0}, X_{\tau-1}^{i_1}, \dots, X_{\tau-1}^{i_{m-1}} \rangle$ in structure $X(n, \tau)$, it also can be expressed as $\langle p_{i_0}, p_{i_1}, \dots, p_{i_{m-1}} \rangle$, where p_i is the Hamiltonian path in $X^i(n, \tau - 1)$. Thus we can find there is a Hamiltonian path $\langle p_{l_0}, p_{l_1}, \dots, p_{l_i}, p_{i_1}, p_{i_2}, \dots, p_{i_{m-1}}, p_{l_{i+1}}, \dots, p_{l_{m-1}} \rangle$ with u to v in $X(n, \tau)$. So we can find the Hamiltonian path that exists in $X(n, \tau)$. (As shown in Figure 4)

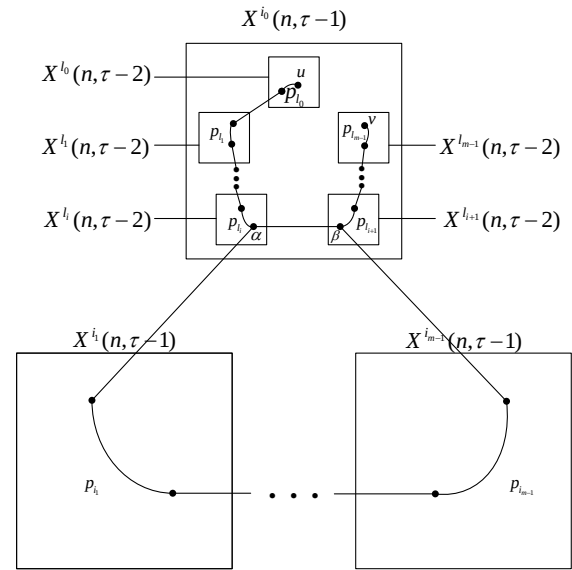


Fig. 4. u and v are in the same copy of $X(n, \tau - 1)$, but they are in distinct copies of $X(n, \tau - 2)$.

Case 2. $u, v \in X^{i_0}(n, \tau - 1)$, and $u, v \in X^{l_0}(n, \tau - 2)$, $X^{l_0}(n, \tau - 2) \in X^{i_0}(n, \tau - 1)$, until $u, v \in X(n, 0)$, that is to say, u and v are in the same copy of $X(n, 0)$. If $u, v \notin X(n, 0)$, we can obtain the Hamiltonian path between any two distinct nodes in $X(n, \tau)$ through a proof similar to Case 1 in (3). Then, we prove the Hamiltonian connectivity of $X(n, \tau)$ for $u, v \in X(n, 0)$.

We can know from the condition 3 of Theorem 1, the degree of the node is bigger than or equal to 4, and there are at least 4 edges between any $X(n, \tau - 2)$ and other

copies. α and β are two nodes, and $\alpha \in X^{l_i}(n, \tau - 2), \beta \in X^{l_{i+1}}(n, \tau - 2)$. There is an edge between α and the structure $X^{i_1}(n, \tau - 1)$ and other edges between β and $X^{i_{m-1}}(n, \tau - 1)$ too. Then let γ and δ be two nodes which are connected by the Hamiltonian path to vertices in $X^{l_1}(n, \tau - 2)$ and $X^{l_{m-1}}(n, \tau - 2)$ respectively, where $\gamma, \delta \in X(n, 0)$.

From the above analysis, there is a Hamiltonian path $\langle u, S_{\pi_0}, S_{\pi_1}, \dots, \gamma, \delta, \dots, S_{\pi_{n-5}}, v \rangle$ in structure $X(n, 0)$, thus there is a Hamiltonian path $\langle u, S_{\pi_0}, S_{\pi_1}, \dots, \gamma, \dots, p_{l_1}, \dots, p_{l_i}, p_{l_{i+1}}, \dots, p_{l_{m-1}}, \dots, (\delta, \dots, S_{\pi_{n-5}}, v) \rangle$ in structure $X^{i_0}(n, \tau - 1)$. And because there is a Hamiltonian path $\langle p_{i_0}, p_{i_1}, \dots, p_{i_{m-1}} \rangle$, where p_i is the Hamiltonian path in $X^i(n, \tau - 1)$. Thus there is a Hamiltonian path between u and v $\langle u, S_{\pi_0}, S_{\pi_1}, \dots, \gamma, \dots, p_{l_1}, \dots, p_{l_i}, p_{i_1}, p_{i_2}, \dots, p_{i_{m-1}}, p_{l_{i+1}}, \dots, p_{l_{m-1}}, \dots, (\delta, \dots, S_{\pi_{n-5}}, v) \rangle$ in structure $X(n, \tau)$. (As shown in Figure 5)

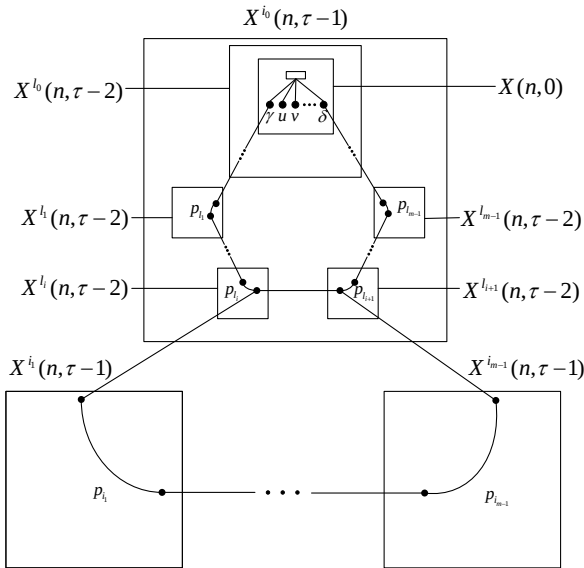


Fig. 5. u and v are in the same copy of $X(n, 0)$.

Case 3. u and v are in distinct copies of $X(n, \tau)$, $u \in X^{i_0}(n, \tau - 1)$, $v \in X^{i_{m-1}}(n, \tau - 1)$. According to the previous analysis, the equivalent network structure graph has a Hamiltonian path $\langle X_{\tau-1}^{i_0}, X_{\tau-1}^{i_1}, \dots, X_{\tau-1}^{i_{m-1}} \rangle$. It can be known from the condition 2 of Theorem 1, there are at least one edge between $X^i(n, \tau - 1)$ and $X^j(n, \tau - 1)$. Let x_0 and x_{m-1} be two nodes which are connected by edges to vertices in $X^{i_1}(n, \tau - 1)$ and $X^{i_{m-2}}(n, \tau - 1)$ respectively, where $x_0 \in X^{i_0}(n, \tau - 1)$ and $x_{m-1} \in X^{i_{m-1}}(n, \tau - 1)$. We have supposed that $X^i(n, \tau - 1)$ has Hamiltonian path that include both paths $\langle u, S_{\pi_0}, S_{\pi_1}, \dots, S_{\pi_{n-3}}, x_0 \rangle$ and $\langle x_{m-1}, S_{\pi_0}, S_{\pi_1}, \dots, S_{\pi_{n-3}}, v \rangle$ in $X^{i_0}(n, \tau - 1)$ and $X^{i_{m-1}}(n, \tau - 1)$ respectively. Thus there is a Hamiltonian path between u and v , $\langle u, p_{i_0}, x_0, p_{i_1}, \dots, p_{i_{m-2}}, (x_{m-1}, p_{i_{m-1}}, v) \rangle$ in structure $X(n, \tau)$. (As shown in Figure 6)

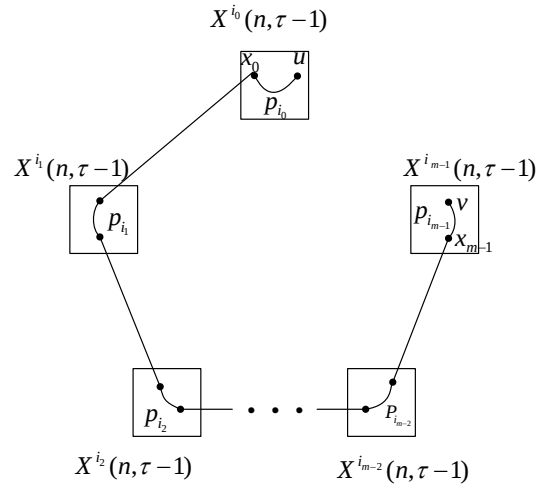


Fig. 6. Both u and v in distinct substructure.

Theorem 2. For casual structure $X(n, k)$ is Hamiltonian-connected, there must be a Hamiltonian cycle in $X(n, k)$.

Proof. According to the theorem conditions, there is a Hamiltonian path between any two nodes in $X(n, k)$, x and y are neighbor nodes, there must be a Hamiltonian path between x and y , so there is a Hamiltonian cycle in $X(n, k)$.

Theorem 3. If $X(n, k)$ contains a node with degree 2 and the amount of nodes in $X(n, k)$ is more than 3, then $X(n, k)$ does not Hamiltonian structure network.

Proof. Assume $X(n, k)$ is Hamiltonian connected and the degree of node x is 2 ($x \in X(n, k)$). Two neighbors of x are nodes x_l and x_r . Due to $X(n, k)$ is Hamiltonian-connected, it means that, there is a Hamiltonian path between any two nodes.

$\langle x_l, x, x_i, \dots, x_r \rangle$ is a Hamiltonian path between x_l and x_r . There exist a node x_i in the path, which is adjacent to x and $x_i \neq x_l, x_i \neq x_r$. Then the node x has three neighbors, which is contradictory to a degree of x of 2. Therefore, there is no Hamiltonian connectivity in structure $X(n, k)$. (As shown in Figure 7)

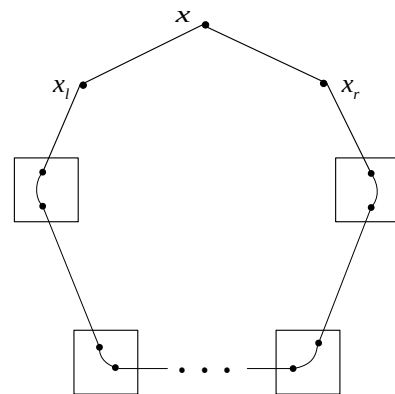


Fig. 7. An example of the conclusion of Theorem 3.

Inference1. F_i is a cut of graph E and $\mathcal{F}$ is the set of all cuts. $|F|$ is the number of element in the cut. If $\min_{F \in \mathcal{F}} |F| \leq 2$, there is no Hamiltonian connectivity in structure $X(n, k)$.

Proof. (1) For $\min_{F \in \mathcal{F}} |F| = 2$, let F' be the edge cut set of $X(n, k)$ and $|F'| = 2$, then let the point set corresponding

to edge cut set F' be (V_1, V_2) , and assume the sub-graph corresponding to V_1 be E_1 and the sub-graph corresponding to V_2 be E_2 . Thus it can be proved in two cases:

Case 1. If there are only two points in V_1 , as shown in Figure 8 and Figure 9, the degree of y in Figure 8 is 1, and the degree of x and y in Figure 9 is 2. According to theorem 3, there is no Hamiltonian connectivity Figure 8 and Figure 9. (In Figure 8, for the convenience of discussion, the degree of y is temporarily allowed to be 1, although this does not satisfy the minimum cut set of 2).

Case 2. If $|V_1| > 2$ and $|V_2| > 2$, because $|F'| = 2$, which is E_1 to E_2 , have two sides, and these two sides connect to x and y . Let $E'_1 = E_1 - x - y$ be the point in E_1 except for x and y . Assume there is a Hamiltonian path between x and y that is $x \rightarrow E_2 \rightarrow E'_1 \rightarrow y$. Then E_2 has three edges connected to E_1 , which are E_2 and x , E_2 and E'_1 , E_2 and y , respectively, contradicting the hypothesis. (As shown in Figure 10)

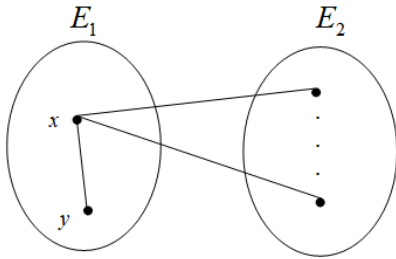


Fig. 8. Example 1 in case 1.

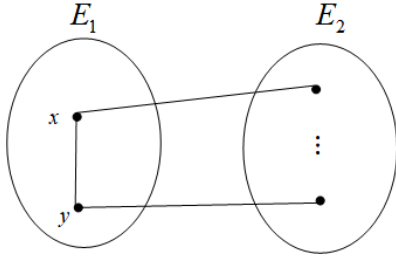


Fig. 9. Example 2 in case 1.

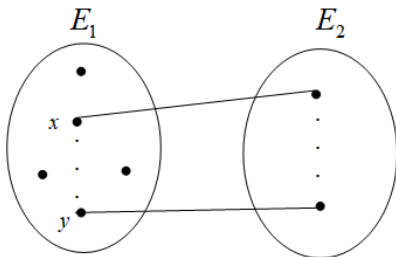


Fig. 10. Example in case 2.

Therefore, for $\min_{F \in \mathcal{F}} |F| = 2$, $X(n, k)$ does not have Hamiltonian connectivity.

(2) For $\min_{F \in \mathcal{F}} |F| < 2$, let F be the edge cut set of $X(n, k)$ and $|F| < 2$, let the point set corresponding to edge cut set F be (V_3, V_4) , the subgraph corresponding to V_3 be E_3 , and the subgraph corresponding to V_4 be E_4 . It means that V_3 is one point in E_3 , V_4 is one point in E_4 . Let u, v be the point in subgraph E_3 , assuming that there is a Hamiltonian path between u and v that's $u \rightarrow E_4 \rightarrow (E_3 - u - v) \rightarrow v$, then E_4 has two edges connected to E_3 , because $|F| < 2$, so the assumption is not true. Therefore, for $\min_{F \in \mathcal{F}} |F| < 2$,

$X(n, k)$ does not have Hamiltonian connectivity. (As shown in Figure 11 and Figure 12)

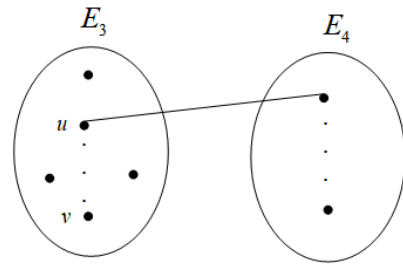


Fig. 11. Sample graph with $|F|$ equal to 1.

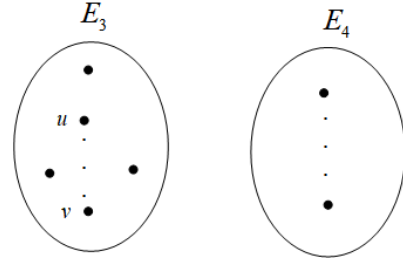


Fig. 12. Sample graph with $|F|$ equal to 0.

Illustration 1. FiConn(4, 1) does not have Hamiltonian connectivity.

Proof. Figure 13 shows the structure of FiConn(4, 1), there is no Hamiltonian path between points (0, 0) and (0, 2). In addition, according to the example 2 in case 1 of Inference1, FiConn(4, 1) does not have Hamiltonian connectivity.

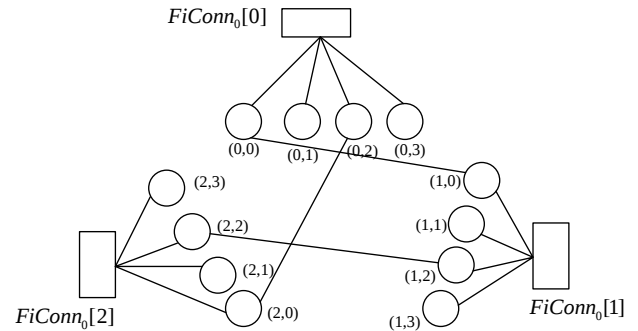


Fig. 13. FiConn(4, 1) structure diagram.

A. Basic Properties of the ExCCC Structure

ExCCC, the Exchanged Cube-Connected Cycles[21], is a new network structure by replacing each node in Exchanged Hypercube (EH) with a cycle. EH network is created by removing some edges from Hypercube network.

EH network can be represented by an undirected graph $EH(s, t) = \{(V, E) | s \geq 1, t \geq 1\}$, the set of nodes $V = \{a_{s-1} \dots a_0 b_{t-1} \dots b_0 c | a_i, b_j, c \in \{0, 1\}, i \in [0, s), j \in [0, t)\}$. This set contains three types of edges, which are

$$E_1 : v_1 \oplus v_2 = 1$$

$$E_2 : v_1[0] = v_2[0] = 1, v_1[s+t:t+1] = v_2[s+t:t+1]$$

$$H(v_1[t:1], v_2[t:1]) = 1$$

$$E_3 : v_1[0] = v_2[0] = 0, v_1[t:1] = v_2[t:1]$$

$$H(v_1[s+t:t+1], v_2[s+t:t+1]) = 1$$

Here $\oplus$ denotes XOR operations, $v[x : y]$ represents the address from x to y in the address string, for example, for node 0111, $v[2, 3] = 01$ represents the Hamming distance between node x and y .

There are two types of Hypercube in $EH(s, t)$ networks, one is the s -dimensional hypercube $H(s : y_{t-1} \dots y_0 c)$ and the other is the t -dimensional hypercube $H(t : x_{s-1} \dots x_0 c)$, in which the number of s -dimensional hypercube is 2^t and the number of t -dimensional hypercube is 2^s .

Illustration 2. $EH(1, 2)$ does not have Hamiltonian connectivity.

Proof. Because node 1000 and 0000 are neighbors in the graph, and their degrees are all 2, $EH(1, 2)$ is not Hamiltonian-connected. Similarly, $ExCCC(1, 2)$ is not Hamiltonian-connected. (As shown in Figure 14)

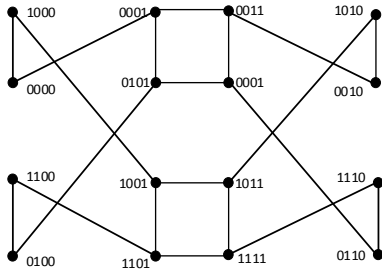


Fig. 14. Structure of $EH(1, 2)$.

IV. HAMILTONIAN ANALYSIS OF BCUBE NETWORK STRUCTURE

A. Design of Hamiltonian Path Algorithm

Depth-first search is a case of traversal of graphs. There are two main types of graph traversal, depth-first search traversal and breadth-first search traversal. The design of the Hamiltonian path algorithm mainly uses depth-first search traversal.

The main idea of depth-first search is to assume that in the initial state none of the vertices in the graph have been accessed. Depth-first search can start from a vertex v in the graph, first visit the node, and then start depth-first traversal of the graph successively from the unvisited nodes adjacent to v , until all vertices connected to v edges in the graph are accessed. If there are any unaccessed vertices in the graph, select another unaccessed vertex as the starting point and repeat the process until all vertices in the graph are accessed.

The steps for depth-first search are roughly as follows:

- 1) First access vertex v .
- 2) Then access the first adjacency point w of v .
- 3) If w exists and w is not accessed, the algorithm is recursively executed from vertex w . Then access the next adjacency of vertex v until all vertices have been accessed.

As shown in Figure 15, it is assumed that the search is started from the node v_1 , the node v_1 is accessed first, and then the node v_2 adjacent to the v_1 node is accessed. Since the node v_2 is not accessed, the search is started from the node v_2 . And so on, starting from the nodes v_4, v_8, v_5 in order to search. After accessing node v_5 , because the neighboring points of node v_5 have been accessed, the access search returns to node v_8 . Also continue back to nodes v_4

and v_2 until it returns to node v_1 . And since the node v_1 has an adjacent point that is not accessed, the search starts again from the node v_1 to the node v_3 . Then continues the search. Therefore, the resulting vertex access sequence is $v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_8 \rightarrow v_5 \rightarrow v_3 \rightarrow v_6 \rightarrow v_7$.

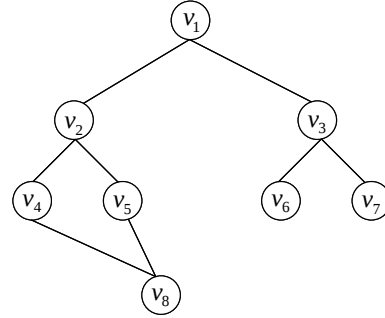


Fig. 15. An example of a tree structure.

B. Hamilton Connectivity of BCube Network Structure

Our goal is to prove that there is a Hamiltonian path between two arbitrary nodes. The Hamiltonian connectivity of $BCube(n, k)$ is analyzed sequentially from the bottom to the top. For $n = 1$, only one switch is connected to a server. Obviously, $BCube(1, k)$ is not Hamiltonian-connected, which makes no sense for the research of data center network structure. So we only discuss the case of $n \geq 2$.

Lemma 1. For any integer n with $n \geq 2$ and $k = 0$, $BCube(n, 0)$ is Hamiltonian-connected.

Proof. For $BCube(n, 0)$ ($n \geq 2$), $BCube(n, 0)$ is built from n servers and an n -port switch. According to the nature of the network, any two nodes are connected by edges in $BCube(n, 0)$. Thus it can be known from (1) in the proof of Theorem 1, that $BCube(n, 0)$ is Hamiltonian-connected.

Lemma 2. For any integer n and k with $n < 3$ and $k > 0$, $BCube(n, k)$ is not Hamiltonian-connected.

Proof. For $BCube(n, k)$ ($n < 3, k > 0$), suppose $k = 1$, $BCube(2, 1)$ consists of two $BCube(2, 0)$. As shown in Figure 16 and Figure 17, the degree of each server is 2 and the number of servers is bigger than or equal to 3. According to the inference 1, $BCube(2, 1)$ does not have Hamiltonian connectivity.

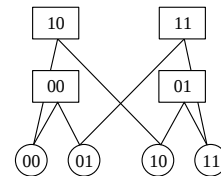


Fig. 16. Structure of $BCube(2, 1)$ which is built from 2 copies of $BCube(2, 0)$.

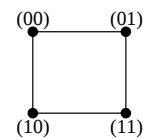


Fig. 17. $BCube(2, 1)$ corresponding diagram.

For $k = 2$, $BCube(2, 2)$ is built from two copies of $BCube(2, 1)$. As shown in Figure 18, the Hamiltonian path from the point $(0, 0, 0)$ to the point $(0, 1, 1)$ is $(0, 0, 0) \rightarrow (1, 0, 0) \rightarrow (1, 1, 0) \rightarrow (1, 1, 1) \rightarrow (1, 0, 1) \rightarrow (0, 0, 1) \rightarrow (0, 1, 1)$, and this path obviously does not contain all the points in the graph. Therefore, $BCube(2, 2)$ is not Hamiltonian-connected.

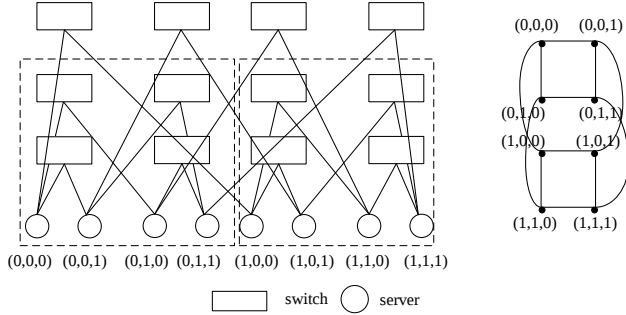


Fig. 18. $BCube(2, 2)$ corresponding structure and diagram.

For $k > 2$, $BCube(2, k)$ is built from two copies of $BCube(2, k - 1)$, $BCube(2, k - 1)$ consists of two $BCube(2, k - 2)$, $BCube(2, k - 2)$ is equivalent to a node, equivalently, $BCube(2, k)$ contains nodes of degree 2, and the number of nodes is more than 3. According to theorem 2, there is no Hamiltonian connectivity for $BCube(2, k)$.

Theorem 4. For any integer n and k with $n \geq 3$ and $k > 0$, $BCube(n, k)$ is Hamiltonian-connected.

Proof. For $BCube(n, k)$ ($n \geq 3, k > 0$), $BCube(4, 1)$ is built from four copies of $BCube(4, 0)$ and four 4-ports switches which is shown in Figure 19. More generically, a $BCube(n, k)$ is constructed from n $BCube(n, k - 1)$ and n^k n -ports switches. Each server in a $BCube(n, k)$ has $k + 1$ ports, which are numbered from level-0 to level- k . So, a $BCube_k$ has n^k servers and $k + 1$ level of switches, with each level having n^k n -port switches.

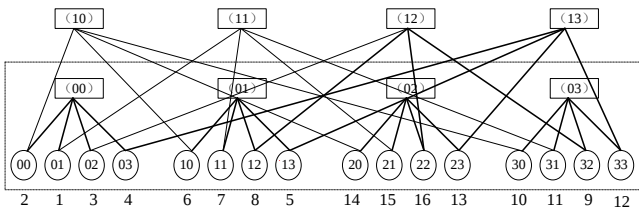


Fig. 19. Structure of $BCube(4, 1)$.

The Hamming distance between two servers A and B is represented by $H(A, B)$. If two servers are neighbors, the Hamming distance between them is 1. More generically, if two servers are connected to the same level- i switch, it is the same about the i -th digit in their address arrays.

In the Hamiltonian path of $BCube$, $S = [a_{n-1}, a_{n-2}, \dots, a_0]$ is the source server, $D = [d_{n-1}, d_{n-2}, \dots, d_0]$ is the destination server, for any unit of $BCube(n, k - l)$ $X^i(n, k - l)$ ($l = 0, 1, 2, \dots, k$). In Figure 20, assuming that the source server S is $(0, 1)$ and the target server D is $(2, 2)$, a Hamiltonian path can be found from the source server to the destination server as $(0, 1) \rightarrow (0, 0) \rightarrow (0, 2) \rightarrow (0, 3) \rightarrow (1, 3) \rightarrow (1, 0) \rightarrow (1, 1) \rightarrow (1, 2) \rightarrow (3, 2) \rightarrow (3, 0) \rightarrow (3, 1) \rightarrow (3, 3) \rightarrow$

$(2, 3) \rightarrow (2, 0) \rightarrow (2, 1) \rightarrow (2, 2)$. In the figure, the square represents the switch, the circle represents the server, the black line represents the way of the Hamiltonian path of the source server $(0, 1)$ and the target server $(2, 2)$, and the bottom digit represents the order of the servers in the path from the source server to the target server.

1) If the source server S_l and destination server D are in different units, the highest bit is different, according to the depth-first principle, S traversal all servers of other modules first and finally returned to the module traversal of the target server D .

2) If the source server S_l and destination server D are in same units, the highest bit is same, we need to go through the rest $X^j(n, k - l)$, then to $X^i(n, k - l)$ ($i \neq j$).

According to the depth-first principle, we can look for the Hamiltonian path between any pair of servers of $BCube(n, k)$. And the servers are numbered $v_0, v_1, \dots, v_n^{k+1} - 1$. Then we find the Hamiltonian paths of these four structures, which are $BCube(4, 0)$, $BCube(4, 1)$, $BCube(5, 0)$, and $BCube(5, 1)$. Then, the results are shown in Table 2.

TABLE II
THE HAMILTONIAN PATH OF THE BCUBE STRUCTURE.

Values	Node	Path
$n = 4, k = 0$	$v_0 \rightarrow v_3$	$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3$
$n = 4, k = 1$	$v_0 \rightarrow v_{12}$	$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_7 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_{10} \rightarrow v_8 \rightarrow v_9 \rightarrow v_{11} \rightarrow v_{15} \rightarrow v_{13} \rightarrow v_{14} \rightarrow v_{12}$
$n = 5, k = 0$	$v_0 \rightarrow v_4$	$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$
$n = 5, k = 1$	$v_0 \rightarrow v_{20}$	$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_9 \rightarrow v_5 \rightarrow v_6 \rightarrow v_7 \rightarrow v_8 \rightarrow v_{13} \rightarrow v_{10} \rightarrow v_{11} \rightarrow v_{12} \rightarrow v_{14} \rightarrow v_{19} \rightarrow v_{15} \rightarrow v_{16} \rightarrow v_{17} \rightarrow v_{18} \rightarrow v_{23} \rightarrow v_{21} \rightarrow v_{22} \rightarrow v_{24} \rightarrow v_{20}$

C. The Algorithm for Finding BCube Hamiltonian Paths

In this part, we design a Hamilton path algorithm between any two different nodes in $BCube$. According to the method of finding Hamiltonian path in Theorem 4, we designed the algorithm in this section. But before we introduce our algorithm, we should explain some solutions in the algorithm.

As shown in Figure 20, when constructing the algorithm shown in Theorem 4 with 02 as the initial node and 13 as the target node in $BCube(4, 1)$, we find that traversing through node 33 will not continue to traverse back to (01). We summarize this problem into four cases:

- 1) When the starting node S and the destination node D are the last nodes in a $BCube(n, 0)$, the algorithm cannot be used if the remainder of n^{k+1} divided by 2 is 1 for all servers.
- 2) When the initial node S is the last node in a $BCube(n, 0)$, the target node D is the last-to-last node in a $BCube(n, 0)$. This algorithm cannot be used if the remainder of n^{k+1} divided by 2 is 0 for all servers.
- 3) When the initial node S is not the last node in a $BCube(n, 0)$, the target node D is the last-to-last node in a $BCube(n, 0)$. The algorithm cannot be used if the remainder of n^{k+1} divided by 2 is 1 for all servers.
- 4) When the initial node S is not the last node in a $BCube(n, 0)$, the target node D is the last-to-last node

in a $\text{BCube}(n, 0)$. This algorithm cannot be used if the remainder of n^{k+1} divided by 2 is 0 for all servers.

It is worth noting that this problem can be easily improved. When traversing the penultimate $\text{BCube}(n, 0)$ node, let the last node in the $\text{BCube}(n, 0)$ equal to D_{n-1} be placed in another location, we also considered this when designing the algorithm.

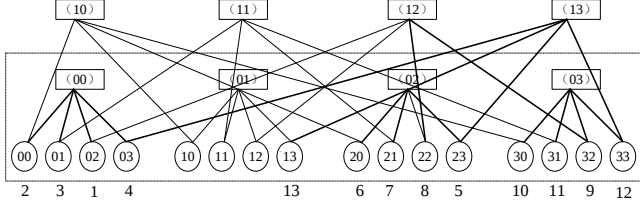


Fig. 20. (02) to (13) cannot use this method directly.

Our proposed algorithm can be described by the following processes:

Step1: First, judge the same bits of the source node $S = [s_0, s_1, \dots, s_{n-1}]$ and the target node $D = [d_0, d_1, \dots, d_{n-1}]$, execute the string matching match() function, and get the number of *level*, indicating that the S is different from the $level = D$. This indicates that the initial node and the target node are in the same $\text{BCube}(n, k - level)$ node.

Step 2: Use the algorithm 2 Traveling function to traverse the $\text{BCube}(n, k - level - 1)$ which contain S to get the Hamiltonian path to all nodes.

Step 3: Then traverse all $\text{BCube}(n, k - level - 1)$ from 0 to $n^{k-level}$ (except the $\text{BCube}(n, k - level - 1)$ node where the target node D is located).

Step 4: At this time, all the nodes except the $\text{BCube}(n, k - level - 1)$ where the target node D is located have been traversed. Algorithm 3 will continue to traverse the remaining $\text{BCube}(n, n - level - 1)$ until all the nodes are traversed to get the Hamiltonian path.

We designed algorithm 1,2,3 to implement the above process.

This algorithm 1 is an algorithm for traversing $\text{BCube}(n, 0)$. This algorithm is designed to achieve: to give an initial node S , it can return Hamiltonian path L and the Destination server S . The time complexity of the algorithm is $O(n)$. This algorithm can be called by the latter two algorithms to realize the traversal of any $\text{BCube}(n, k)$ node meeting certain restrictions ($n > 2$).

The first solution to the problem in this section is to judge by the length of Node set L . If $L = n^k + 1 - 2n + 1$, it means that the second-to-last $\text{BCube}(n, 0)$ is traversed and the server number corresponding to the last D_{n-1} of target node D in the second-to-last $\text{BCube}(n, 0)$ is placed in L to avoid the above problem.

The function of algorithm 2 is that given source sever S and the *level*, which is same as k in the corresponding $\text{BCube}(n, k)$. The X_{level}^i stored in Π is the $\text{BCube}(n, level)$ node to be traversed. As the Traveling() function is executed continuously, the recursive function will be executed to achieve traversal. The *level* will change constantly, so that all the nodes in T are traversed, and the Total L (Hamiltonian path sequence) will be output. The time complexity of the algorithm is $O(n^{level})$.

Algorithm 1 BCube Meta-structure Hamiltonian Path

Input: Source server $S = [a_0, a_1, \dots, a_{n-1}]$
Output: Node set L , Destination server D

```

1: function BASICCELLPATH( $S$ )
2:   if ( $TotalL$ ).len ==  $n^{k+1} - 2n + 1$  then
3:      $L[n] = [S]$ ;
4:   else
5:      $L[n] = [S, [a_0, a_1, \dots, D_{n-1}]]$ ;
6:   end if
7:    $i = 0$ ;
8:   while  $L[n].len < n$  do
9:     if  $i \neq a_n$  then
10:       $L.add([a_0, a_1, \dots, i])$ ;
11:    end if
12:     $i++$ ;
13:  end while
14:   $D = L[n - 1]$ ;
15: end function

```

Algorithm 2 BCube Recursive Search for Hamiltonian Path

Input: Currently traversed $\text{BCube}(n, k)$, Source server S , Currently traversed *level*
Output: Hamiltonian paths $TotalL$, Target server D

```

1: function  $TotalL, D = \text{TRAVERSING}(\Pi, S, level)$ 
2:   if  $level = 0$  then
3:      $L, D = \text{BasicCellPath}(S)$ ;
4:      $level = level + 1$ ;
5:      $TotalL = TotalL + L$ ;
6:   else
7:      $\Pi = [X_{level}^0, X_{level}^1, \dots, X_{level}^{n-2}, X_{level}^{n-1}]$ ;
8:     for  $i = 0$  to  $n - 1$  do
9:       ( $TotalL, D$ ) =  $\text{Traversing}(\Pi[i], S, level - 1)$ ;
10:       $S = D$ ;
11:    end for
12:  end if
13: end function

```

In algorithm 3, the initial and target nodes are returned to the same bit level, indicating that the two nodes are in the same $\text{BCube}(n, level)$. The algorithm first traverses the $\text{BCube}(n, level - 1)$ where the initial node is located, then traverses the other $\text{BCube}(n, level - 1)$ except where the target node D is located, and finally returns the $\text{BCube}(n, level - 1)$ where D is located. The algorithm continues until $S = D$, returns the Hamilton path sequence Total L .

The complexity of single execution of algorithm 3 is $O(n^k)$, because in fact, the complexity of algorithm 2 is $O(n^{level})$, while the number of times that algorithm 3 executes Traversing() is $n^{k-level}$. The number of recursion of algorithm 3 is $\log_{level} k$ according to level and k , so the time complexity of the whole algorithm 3 is $O(\log k * n^k)$.

Algorithm 3 BCube Hamiltonian Path Any Different Unit

Input: Source server S , Destination server D
Output: Hamiltonian paths $TotalL$

```

1: function  $L = \text{FINDBCUBE}(S, D)$ 
2:   for  $i = 0$  to  $n - 1$  do
3:     if  $\text{match}(S! = D)$  then
4:        $level = i$ ;
5:     else
6:       return  $TotalL$ ;
7:     end if
8:   end for
9:    $(L, D_1) = \text{Travsesing}([X_S], S, level - 1)$ ;
10:   $S = D_1$ ;
11:   $TotalL = TotalL + L$ ;
12:   $T[] = [X^0, X^1, \dots, X^{D-1}, X^{D+1}, \dots, X^{n^{k-level}-1}]$ ;
13:  for  $i = 0$  to  $n^{k-level} - 1$  do
14:     $(L, D_1) = \text{Travsesing}(T[i], S, level - 1)$ ;
15:     $S = D_1$ ;
16:     $TotalL = TotalL + L$ ;
17:  end for
18:  FindBCube( $S, D$ )
19: end function

```

D. Hamilton Path of DCell Network Structure

About the Hamiltonian of DCell network structure, Wang Xi has already proved in the paper “Research on Path Coverage and Restricted Connectivity on DCell Network” [22]. The main idea is that enumeration method is used to prove Hamiltonian connectivity for low-level DCell(n, k). In addition, Mathematical Induction is used for high-level DCell(n, k).

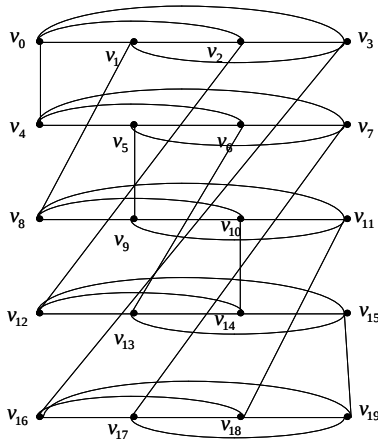


Fig. 21. Structure corresponding graph of DCell(4, 1).

Similar to BCube(n, k), we can also find servers between any two points in DCell(n, k) according to the depth-first principle. The structure corresponding graph of DCell(4, 1) is shown in Figure 21. Then we find the Hamiltonian paths of these four structures, which are DCell(4, 0), DCell(4, 1), DCell(5, 0), and DCell(5, 1). And the results are shown in Table 3.

TABLE III
THE HAMILTONIAN PATH OF THE DCELL STRUCTURE.

Values	Node	Path
$n = 4, k = 0$	$v_0 \rightarrow v_2$	$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_2$
$n = 4, k = 1$	$v_0 \rightarrow v_4$	$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_{16} \rightarrow v_{17} \rightarrow v_7 \rightarrow v_5 \rightarrow v_9 \rightarrow v_8 \rightarrow v_{10} \rightarrow v_{11} \rightarrow v_{18} \rightarrow v_{19} \rightarrow v_{15} \rightarrow v_{12} \rightarrow v_{14} \rightarrow v_{13} \rightarrow v_6 \rightarrow v_4$
$n = 5, k = 0$	$v_0 \rightarrow v_4$	$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$
$n = 5, k = 1$	$v_0 \rightarrow v_{14}$	$v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_{25} \rightarrow v_{26} \rightarrow v_{27} \rightarrow v_{28} \rightarrow v_{29} \rightarrow v_{24} \rightarrow v_{22} \rightarrow v_{21} \rightarrow v_{20} \rightarrow v_{23} \rightarrow v_{18} \rightarrow v_{17} \rightarrow v_{15} \rightarrow v_{16} \rightarrow v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_5 \rightarrow v_6 \rightarrow v_{11} \rightarrow v_{10} \rightarrow v_{12} \rightarrow v_{13} \rightarrow v_{14}$

V. CONCLUSION

In this paper, by studying the Hamiltonian and structural characteristics of $X(n, k)$, which is the meta-structure of the large data center network, we have proposed and proved if a network structure is Hamilton-connected, it will follow three conditions.

We also propose the theorems to judge a network structure that it is no Hamiltonian graph. In addition, we also proved that for BCube(n, k) ($n \geq 3, k \geq 0$), any two servers A and B can find a Hamiltonian path from A to B through the unique coding algorithm of BCube. That is, the BCube(n, k) ($n \geq 3, k \geq 0$) is Hamiltonian-connected.

The paper mainly proves the Hamiltonian connectivity to the proposed theorem and what situation there is no Hamiltonian connectivity properties, the Hamiltonian connectivity properties of BCube are discussed. But the disjoint paths and restricted connectivity of BCube networks are still open issues, which we will continue to discuss later. Furthermore, we have a preliminary understanding of another structure FiConn, which can simply prove the Hamiltonian connectivity of the low-level FiConn structure, and will also discuss and prove the Hamiltonian connectivity of the high-level FiConn structure in the next period of time.

REFERENCES

- [1] Ye PG, Li Z, Yang Z, Chen P, Zhang Z, Li N, Zheng J. Periodic watermarking for copyright protection of large language models in cloud computing security. *Computer Standards & Interfaces*. 2025 Aug 1;94:103983.
- [2] Zhang Y, Tang H, Li H, Wang S. Unlocking the flexibilities of data centers for smart grid services: Optimal dispatch and design of energy storage systems under progressive loading. *Energy*. 2025 Feb 1;316:134511.
- [3] He W, Xu Q, Liu S, Wang T, Wang F, Wu X, Wang Y, Li H. Analysis on data center power supply system based on multiple renewable power configurations and multi-objective optimization. *Renewable Energy*. 2024 Feb 1;222:119865.
- [4] Katsanikas M, Wiggins S. 3D generating surfaces in Hamiltonian systems with three degrees of freedom-I. *International Journal of Bifurcation and Chaos*. 2024 Feb;34(02):2430004.
- [5] Chakraborti D, Kim J, Lee H, Seo J. Hamilton transversals in tournaments. *Combinatorica*. 2024 Aug 15:1-20.
- [6] Dong H, Fan J, Cheng B, Wang Y, Xu L. Hamiltonian properties of the data center network HSDC with faulty elements. *The Computer Journal*. 2023 Aug;66(8):1965-81.
- [7] Fan J. Hamilton-connectivity and cycle-embedding of the Möbius cubes. *Information Processing Letters*. 2002 Apr 30;82(2):113-7.
- [8] Lv Y, Lin CK, Fan J. Hamiltonian cycle and path embeddings in k -ary n -cubes based on structure faults. *The Computer Journal*. 2017 Feb 7;60(2):159-79.
- [9] Lv Y, Lin CK, Fan J, Jia X. Hamiltonian cycle and path embeddings in 3-ary n -cubes based on K1, 3-structure faults. *Journal of Parallel and Distributed Computing*. 2018 Oct 1;120:148-58.

- [10] Lin X, McKinley PK, Ni LM. Deadlock-free multicast wormhole routing in 2-D mesh multicomputers. *IEEE transactions on Parallel and Distributed Systems*. 1994 Aug 31;5(8):793-804.
- [11] Fujita S, Araki T. Three-round adaptive diagnosis in binary n-cubes. In *Algorithms and Computation: 15th International Symposium, ISAAC 2004, Hong Kong, China, December 20-22, 2004. Proceedings 15 2005* (pp. 442-451). Springer Berlin Heidelberg.
- [12] Ye LC, Liang JR. Five-round adaptive diagnosis in Hamiltonian networks. *IEEE Transactions on Parallel and Distributed Systems*. 2014 Aug 21;26(9):2459-64.
- [13] Zhu Q, Wang XK, Cheng G. Reliability evaluation of BC networks. *IEEE Transactions on Computers*. 2012 May 29;62(11):2337-40.
- [14] Bradshaw P, Halasz K, Stacho L. From one to many rainbow Hamiltonian cycles. *Graphs and Combinatorics*. 2022 Dec;38(6):188.
- [15] Antoniuk S, Dudek A, Reiher C, Ruciński A, Schacht M. High powers of Hamiltonian cycles in randomly augmented graphs. *Journal of Graph Theory*. 2021 Sep;98(2):255-84.
- [16] Wang X, Fan J, Zhou J, Lin CK. The restricted h-connectivity of the data center network DCell. *Discrete Applied Mathematics*. 2016 Apr 20;203:144-57.
- [17] Liu Xin, Fan Jianxi, Zong Xufeng, Xu Cuixia. HCH - Hamiltonian connectivity of the cube. *Computer Engineering and Applications*. 2005;41(32):83-6.
- [18] Yang Y, Song N. Fault-free Hamiltonian paths passing through prescribed linear forests in balanced hypercubes with faulty links. *Theoretical Computer Science*. 2023 Jan 4;939:161-9.
- [19] Abdallah M, Cheng E. Fault-tolerant Hamiltonian connectivity of 2-tree-generated networks. *Theoretical Computer Science*. 2022 Mar 12;907:62-81.
- [20] Dong Q, Zhou J, Fu Y, Gao H. Hamiltonian connectivity of restricted hypercube-like networks under the conditional fault model. *Theoretical Computer Science*. 2013 Feb 11;472:46-59.
- [21] Zhang Z, Deng Y, Min G, **e J, Huang S. ExCCC-DCN: A highly scalable, cost-effective and energy-efficient data center structure. *IEEE Transactions on Parallel and Distributed Systems*. 2016 Sep 14;28(4):1046-60.
- [22] Liu X, Meng J. The k-restricted edge-connectivity of the data center network DCell. *Applied Mathematics and Computation*. 2021 May 1;396:125941.

Linyu Yang received her MS degree from the School of Computer and Information Engineering, Henan University, Kaifeng, China, in 2025. Currently, she is a teaching assistant at the School of Information Engineering, Henan Kaifeng College of Science Technology and Communication, Kaifeng, China. Additionally, she is working at the Henan Engineering Research Center for AI Technology of Converged Media. Her research interests include data center networks, wireless networks, and fault-tolerant routing algorithms.