


Article

Three Edge-Disjoint Hamiltonian Cycles in Folded Locally Twisted Cubes and Folded Crossed Cubes with Applications to All-to-All Broadcasting

Kung-Jui Pai 

Department of Industrial Engineering and Management, Ming Chi University of Technology,
New Taipei City 24301, Taiwan; poter@mail.mcut.edu.tw

Abstract: All-to-all broadcasting means to distribute the exclusive message of each node in the network to all other nodes. It can be handled by rings, and a Hamiltonian cycle is a ring that visits each vertex exactly once. Multiple edge-disjoint Hamiltonian cycles, abbreviated as EDHCs, have two application advantages: (1) parallel data broadcast and (2) edge fault-tolerance in network communications. There are three edge-disjoint Hamiltonian cycles on n -dimensional locally twisted cubes and n -dimensional crossed cubes while $n \geq 6$, respectively. Locally twisted cubes, crossed cubes, folded locally twisted cubes (denoted as $FLTQ_n$), and folded crossed cubes (denoted as FCQ_n) are among the hypercube-variant network. The topology of hypercube-variant network has more wealth than normal hypercubes in network properties. Then, the following results are presented in this paper: (1) Using the technique of edge exchange, three EDHCs are constructed in $FLTQ_5$ and FCQ_5 , respectively. (2) According to the recursive structure of $FLTQ_n$ and FCQ_n , there are three EDHCs in $FLTQ_n$ and FCQ_n while $n \geq 6$. (3) Considering that multiple faulty edges will occur randomly, the data broadcast performance of three EDHCs in $FLTQ_n$ and FCQ_n is evaluated by simulation when $5 \leq n \leq 9$.

Keywords: interconnection networks; edge-disjoint Hamiltonian cycles; folded locally twisted cubes; folded crossed cubes; fault-tolerant data broadcasting

MSC: 05C90; 68R10



Citation: Pai, K.-J. Three Edge-Disjoint Hamiltonian Cycles in Folded Locally Twisted Cubes and Folded Crossed Cubes with Applications to All-to-All Broadcasting. *Mathematics* **2023**, *11*, 3384. <https://doi.org/10.3390/math11153384>

Academic Editor: Ripon Kumar Chakraborty

Received: 29 June 2023

Revised: 31 July 2023

Accepted: 1 August 2023

Published: 2 August 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The design of interconnection networks is one of the important issues in parallel computing systems and data centers. An interconnection network is often modeled by a graph, where vertices represent processing units and edges represent communication links. Hypercubes [1,2] have become one of the most popular interconnection networks due to their attractive features, including regularity, vertex symmetric, link symmetric, small diameter, strong connectivity, recursive construction, partition capability, and small link complexity. Locally twisted cubes (denoted as LTQ_n), crossed cubes (denoted as CQ_n), folded locally twisted cubes (denoted as $FLTQ_n$), and folded crossed cubes (denoted as FCQ_n) are among the hypercube-variant network. They are similar to hypercubes in that the vertices can be one-to-one labeled with 0–1 binary strings of length n , and their definition is presented in Section 2. The topology of a hypercube-variant network has more wealth than a normal hypercube in network properties, e.g., its diameter is about half that of the same-dimensional hypercube.

Ring structures are essential to high-performance computing architectures and are often used as a baseband for data transmission in interconnect networks and control flow in parallel and distributed environments. Many efficient algorithms with low communication costs have been developed based on the ring structure [3,4]. A Hamiltonian cycle in a graph is a cycle (or ring) that visits each vertex exactly once. Hamiltonian cycles in the

graph are said to be edge-disjoint if they do not share any common edges. In addition, k (≥ 2) edge-disjoint Hamiltonian cycles, abbreviated as EDHCs, also provide the edge-fault tolerant Hamiltonicity for interconnection networks.

All-to-all broadcast means to distribute the exclusive message of each node in the network to all other nodes. This is an important issue for high-performance computing and communication networks, including data center operations. All-to-all broadcasts have been dealt with previously for several network topologies, such as linear arrays, meshes, toruses, hypercube networks, rings [5], and more. The ring is an important network topology because of its simple structure, easy deployment, and strong fault tolerance. Consider all-to-all communication in a network system with n nodes, where each node needs to send a distinct message to all other nodes. Applying the ring structure, once a message starts sending, a node can receive a new message from the previous node at each step, keep a copy of the new message for itself, and send the received message to the next node. Therefore, the process can be accomplished in $n - 1$ steps. Furthermore, one way to achieve fault-tolerant inter-processor communication is to efficiently utilize disjoint paths between pairs of source and destination nodes. Especially when link fault tolerance is considered, the technique of using edge-disjoint paths is a common strategy. Therefore, if a fault occurs on one edge of a Hamiltonian ring, messages can be transmitted along the other Hamiltonian ring. Finally, construct multiple EDHCs applications with enhanced edge fault-tolerance in data transmission [6,7].

The following describes the previous related work on EDHCs. Rowley and Bose [7] presented that a slightly modified degree $2r$ de Bruijn networks can be decomposed into r Hamiltonian cycles when r is a power of a prime. Barth and Raspaud [8] provided two EDHCs on the butterfly networks. Lee and Shin [9] achieved reliable all-to-all broadcasting on meshes and hypercubes using EDHCs. Bae and Bose [6] studied EDHCs in k -ary n -cubes and hypercubes. Petrovic and Thomassen [10] characterized the number of EDHCs in hypertournaments. Hung et al. constructed two or multiple EDHCs in LTQ_n [11], augmented cubes [12], twisted cubes [13], CQ_n , transposition networks, and hypercube-like networks [14], respectively. Wang et al. Ref. [15] presented that two EDHCs can be embedded into parity cubes. Hussain et al. Ref. [16] gave a construction of three EDHCs in Eisenstein–Jacobi networks. Albader and Bose showed how to obtain two EDHCs in Gaussian networks [17]. In recent years, Chen obtained two edge-disjoint Hamiltonian cycles of bubble-sort star graphs BS_n when $n \geq 4$ [18]. Yang proved that there exist two edge-disjoint Hamiltonian cycles in spined cube SQ_n when $n \geq 4$ [19]. Pai [20] provided a parallel algorithm for constructing two EDHCs in CQ_n . Li et al. Ref. [21] construct two EDHCs in LTQ_n by using a parallel algorithm. Then, Pai et al. presented that three EDHCs can be embedded in LTQ_n [22] and CQ_n [23].

There are five papers [11,14,20,22,23] that consider the construction of two or three EDHCs in LTQ_n and CQ_n . Ref. [11] provided two EDHCs in LTQ_n while $n \geq 4$. Ref. [22] presents three EDHCs in LTQ_n for $n \geq 6$. Similarly, [14] provided two EDHCs in CQ_n while $n \geq 4$. Ref. [23] presents three EDHCs in CQ_n for $n \geq 6$. Then, [20] proposed a parallel algorithm for the construction of EDHCs to improve the sequential construction of [14]. However, there is no article that studied three EDHCs on $FLTQ_n$ (respectively, FCQ_n), which is to add folded edges on LTQ_n (respectively, CQ_n). In this paper, the following results are presented: (1) Using the technique of edge exchange, three EDHCs are constructed in $FLTQ_5$ and FCQ_5 , respectively. (2) According to the recursive structure of $FLTQ_n$ and FCQ_n , there are three EDHCs in $FLTQ_n$ and FCQ_n while $n \geq 6$. (3) Considering that multiple faulty edges will occur randomly, the data broadcast performance of three EDHCs in $FLTQ_n$ and FCQ_n is evaluated by simulation when $5 \leq n \leq 9$. The rest of the paper is organized as follows: Section 2 introduces the necessary definitions and theorems for LTQ_n , CQ_n , $FLTQ_n$, FCQ_n , and EDHCs. In Section 3, three EDHCs are presented on $FLTQ_n$ and FCQ_n , respectively. Next, in Section 4, the performance assessment of data broadcasting by using three EDHCs on $FLTQ_n$ and FCQ_n is presented. Section 5 discusses the results of the simulations. The conclusion of this paper will be presented placed in Section 6.

2. Preliminaries

The topology of a network is usually modeled as an undirected graph $G = (V(G), E(G))$. The neighborhood of a vertex v in a graph G , denoted by $N_G(v)$, is the set of vertices adjacent to v in G . A cycle C_m of length m in G , denoted by $v_1 - v_2 - v_3 - \dots - v_{m-1} - v_m - v_1$, is a sequence $(v_1, v_2, v_3, \dots, v_{m-1}, v_m, v_1)$ of vertices such that $(v_m, v_1) \in E(G)$ and $(v_i, v_{i+1}) \in E(G)$ for $1 \leq i \leq m - 1$. For convenience, replace $e \in E(C_m)$ with $e \in C_m$, and the terms “networks” and “graphs”, “nodes” and “vertices”, “links” and “edges” are often used interchangeably in this paper. Yang et al. gave the following definitions of locally twisted cubes [24]:

Definition 1 ([24]). The n -dimensional locally twisted cube LTQ_n is the labeled graph with the following recursive fashion:

- (1) LTQ_1 is the complete graph on two vertices labeled by 0 and 1. LTQ_2 is a graph consisting of four vertices with labels 00, 01, 10, and 11 together with four edges (00, 01), (00, 10), (01, 11), and (10, 11).
- (2) For $n \geq 3$, LTQ_n is composed of two subcubes LTQ_{n-1} , denoted as LTQ_{n-1}^0 and LTQ_{n-1}^1 , such that each vertex $x = 0x_{n-1}x_{n-2} \dots x_2x_1 \in V(LTQ_{n-1}^0)$ is connected with the vertex $y = 1(x_{n-1} \oplus x_1)x_{n-2} \dots x_2x_1 \in V(LTQ_{n-1}^1)$ by an edge, where x and y are called the n -neighbors to each other.

Definition 2 ([25]). For $n \geq 2$, an n -dimensional folded locally twisted cube, denoted by $FLTQ_n$, is defined based on the definition of LTQ_n as follows: $FLTQ_n$ is a graph obtained from LTQ_n by adding all complementary edges, which join a vertex $u = u_nu_{n-1} \dots u_2u_1$ to another vertex $\bar{u} = \bar{u}_n\bar{u}_{n-1} \dots \bar{u}_2\bar{u}_1$ for every $u \in V(LTQ_n)$, where $\bar{u}_i = 1 - u_i$.

For conciseness of representation, sometimes the labels of nodes are changed to the use of decimals. For example, Figure 1 shows LTQ_4 and $FLTQ_4$, where each node is labeled by the binary code and its corresponding decimal (inside the circle).

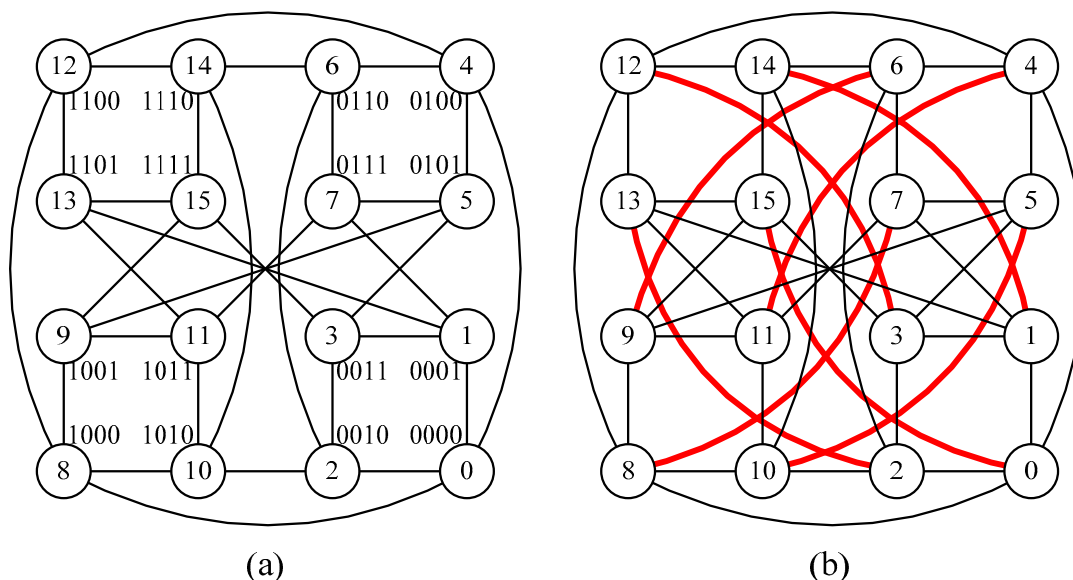


Figure 1. (a) LTQ_4 and (b) $FLTQ_4$ where thick lines indicate complementary edges.

Efe [26] defined two binary strings $x = x_2x_1$ and $y = y_2y_1$ to be pair-related, denoted $x \sim y$, if and only if $(x, y) \in \{(00, 00), (10, 10), (01, 11), (11, 01)\}$.

Definition 3 ([26]). The vertex set of CQ_n is $\{u_n u_{n-1} \cdots u_2 u_1 \mid u_i \in \{0, 1\}, 1 \leq i \leq n\}$. For any two vertices $u = u_n u_{n-1} \cdots u_2 u_1$ and $v = v_n v_{n-1} \cdots v_2 v_1$ of CQ_n , u is adjacent to v if and only if the following conditions are established:

- (1) $u_n u_{n-1} \cdots u_{i+1} = v_n v_{n-1} \cdots v_{i+1}$;
- (2) $u_i \neq v_i$;
- (3) $u_{i-1} = v_{i-1}$ if i is even;
- (4) $u_{2k} u_{2k-1} \sim v_{2k} v_{2k-1}$ for $1 \leq k \leq \lceil (i-1) / 2 \rceil$.

Definition 4. For $n \geq 2$, an n -dimensional folded crossed cube, denoted by FCQ_n , is constructed from CQ_n by adding all complementary edges, which join a vertex $u = u_n u_{n-1} \cdots u_2 u_1$ to another vertex $\bar{u} = \bar{u}_n \bar{u}_{n-1} \cdots \bar{u}_2 \bar{u}_1$ for every $u \in V(LTQ_n)$, where $\bar{u}_i = 1 - u_i$.

For example, Figure 2 shows CQ_4 and FCQ_4 . According to Definition 2 (respectively, 4), $FLTQ_n$ (respectively, FCQ_n) is constructed from LTQ_n (respectively, FCQ_n) by adding all complementary edges. For $FLTQ_n$ and FCQ_n , when $n \geq 6$, the main two theorems in this paper will use the following two theorems, respectively:

Theorem 1 ([22]). For $n \geq 6$, there exist three edge-disjoint Hamiltonian cycles in LTQ_n .

Theorem 2 ([23]). For $n \geq 6$, there exist three edge-disjoint Hamiltonian cycles in CQ_n .

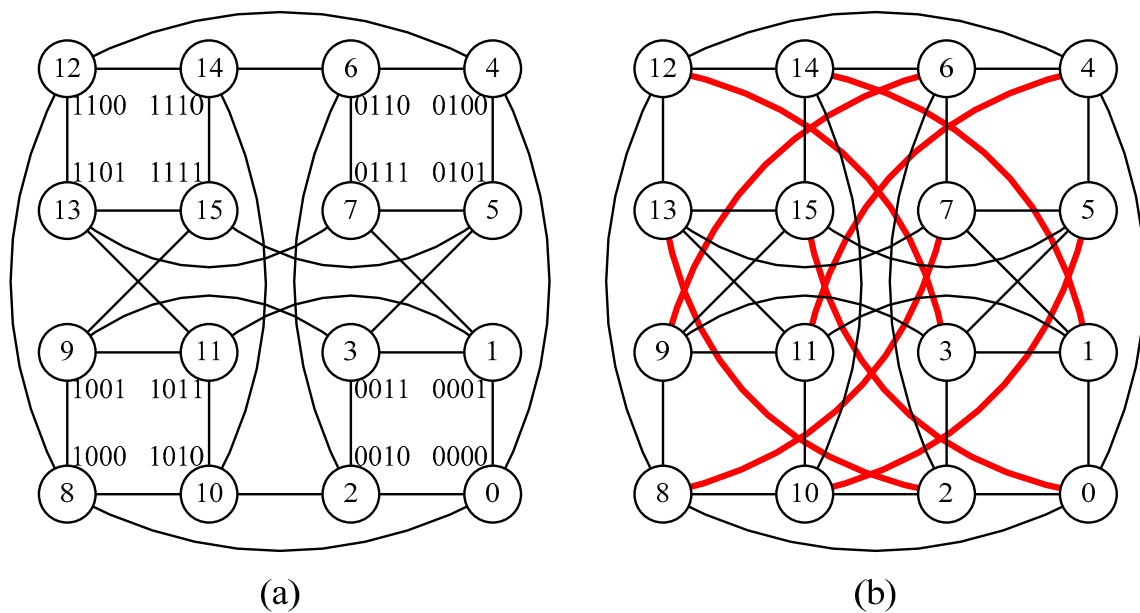


Figure 2. (a) CQ_4 and (b) FCQ_4 where thick lines indicate complementary edges.

In the research direction of EDHCs, briefly summarize the solution methods in [7–23]. Most of the articles use the method of recursive construction. Firstly, k (≥ 2) EDHCs are proposed on the small-dimensional graph class as the basis. Then, based on this, a recursive algorithm was proposed to construct EDHCs. These methods usually remove one or more edges in the original cycles and then connect the endpoint of the cycle to the endpoint of another cycle to form a Hamiltonian cycle on the next dimensional graph. In addition, some articles will go out of the cycle along the edge of a specific direction or a specific dimension. Then, by exchanging some edges, multiple cycles are connected to form EDHCs. In this paper, we use these two approaches at the same time, and the details are presented in Section 3.

3. Methods and Results

Using previous results in [20,21] and the technique of edge exchange, three EDHCs in $FLTQ_5$ and FCQ_5 were constructed, respectively. Then, according to the recursive structure of $FLTQ_n$ and FCQ_n , there are three EDHCs in $FLTQ_n$ and FCQ_n while $n \geq 6$.

3.1. Three EDHCs in $FLTQ_n$

In this subsection, there are two EDHCs of $FLTQ_5$ that were built from [21]. Then, by fine-tuning some edges in the second Hamiltonian cycle so that the remaining edges are sufficient to configure the third Hamiltonian cycle.

Let HC_1 and HC_2 be two EDHCs of LTQ_4 , as shown in Figure 3. According to Definitions 1 and 2, $FLTQ_5$ can be decomposed into two copies of LTQ_4 , and each copy has two Hamiltonian cycles mentioned above. Divide $FLTQ_5$ into two disjoint subcubes LTQ_4^i for $i \in \{0, 1\}$, and let HC_k^i for $k \in \{1, 2\}$ be the corresponding k -th Hamiltonian cycle in the subcube LTQ_4^i such that each cycle maps to HC_k in LTQ_4 . Then, the two Hamiltonian cycles of $FLTQ_5$, namely $HC_{1'}$ and $HC_{2'}$, can be constructed by merging of HC_k^0 and HC_k^1 for $k = 1, 2$ by adjusting two edges in each cycle, which are described as follows:

$$E(HC_{1'}) = E(HC_1^0) \cup E(HC_1^1) \cup \{(0, 16), (4, 20)\} - \{(0, 4), (16, 20)\} \tag{1}$$

and

$$E(HC_{2'}) = E(HC_2^0) \cup E(HC_2^1) \cup \{(2, 18), (6, 22)\} - \{(2, 6), (18, 22)\} \tag{2}$$

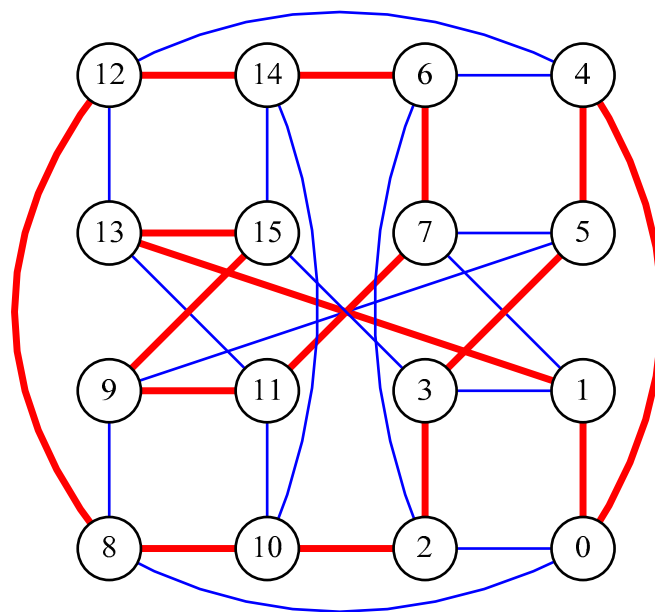


Figure 3. Two EDHCs in LTQ_4 . Thick red lines (respectively, thin blue line) indicate the first (respectively, second) Hamiltonian cycle.

For example, Figure 4 depicts the Hamiltonian cycles $HC_{1'}$ and $HC_{2'}$ of $FLTQ_5$ constructed from Equation (1) and Equation (2), respectively. From the drawing, readers may imagine that the dashed line divides the entire $FLTQ_5$ into two subcubes of equal size, and nodes in the left and right parts are mirrored, and their labels have a difference of ± 16 .

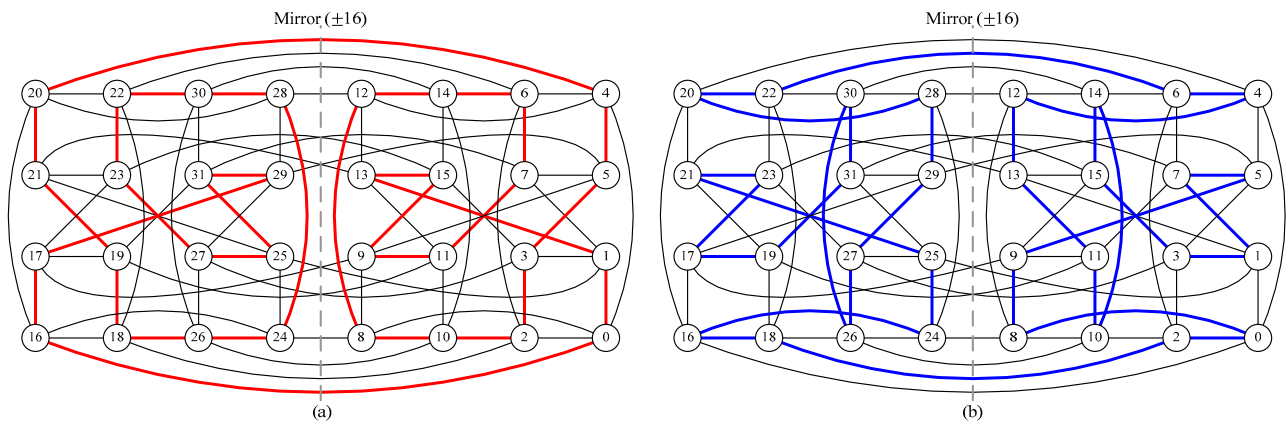


Figure 4. The Hamiltonian cycles (a) $HC_{1'}$ and (b) $HC_{2'}$ of $FLTQ_5$ were constructed from Equation (1) and Equation (2), respectively, where thick lines indicate the edges of the cycle. Complementary edges are omitted because it is easy to judge visually and the two Hamiltonian cycles do not use them.

That $E(FLTQ_5) - \{E(HC_{1'}) \cup E(HC_{2'})\}$ includes two C_{16} cycles: $C_{16}^A = 0 - 4 - 27 - 3 - 28 - 12 - 19 - 11 - 20 - 16 - 15 - 23 - 8 - 24 - 7 - 31 - 0$ and $C_{16}^B = 1 - 25 - 6 - 2 - 29 - 5 - 26 - 10 - 21 - 13 - 18 - 22 - 9 - 17 - 14 - 30 - 1$. The third Hamiltonian cycle of $FLTQ_5$, $HC_{3'}$, can be constructed as follows:

$$E(HC_{3'}) = E(C_{16}^A) \cup E(C_{16}^B) \cup \{(1, 3), (5, 7), (27, 29), (30, 31)\} - \{(1, 30), (3, 27), (5, 29), (7, 31)\} \quad (3)$$

Figure 5a illustrates the construction of $HC_{3'}$, where the bold lines indicate the edges that will be added to $HC_{3'}$ in Equation (3). Then, lines with cross marks are the edges that will be removed from $HC_{3'}$ in Equation (3). In fact, the construction of the third Hamiltonian cycle mentioned above is accomplished using the edge-swapping technique. Therefore, with the above adjustments, it is necessary to modify the edge set of $HC_{2'}$ obtained from Equation (2) by swapping two edge sets, $\{(1, 3), (5, 7), (27, 29), (30, 31)\}$ and $\{(1, 30), (3, 27), (5, 29), (7, 31)\}$, as follows:

$$E(HC_{2'}) = E(HC_{2'}) \cup \{(1, 30), (3, 27), (5, 29), (7, 31)\} - \{(1, 3), (5, 7), (27, 29), (30, 31)\} \quad (4)$$

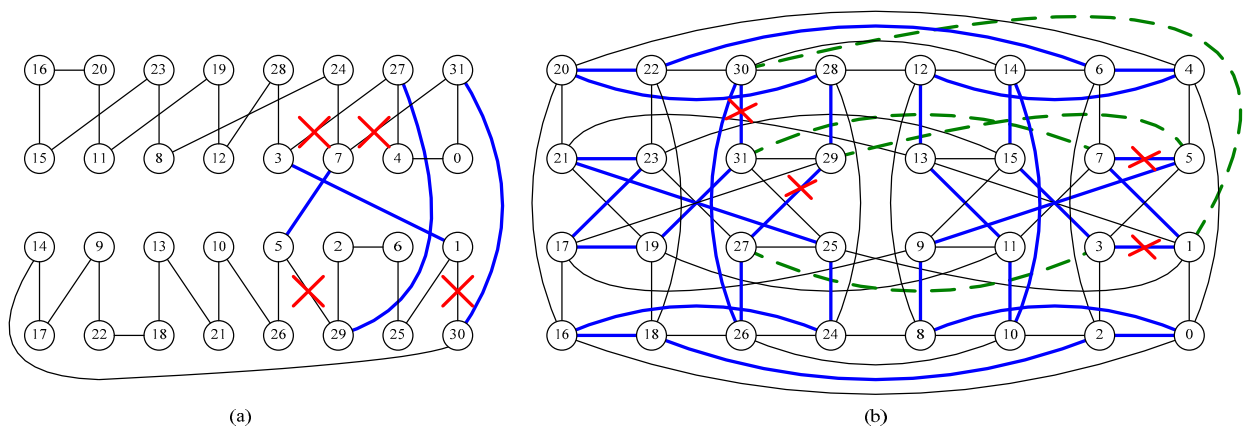


Figure 5. The Hamiltonian cycles (a) $HC_{3'}$ and (b) new $HC_{2'}$ of $FLTQ_5$ were constructed from Equations (3) and (4), respectively.

Similarly, Figure 5b illustrates the construction of the new $HC_{2'}$, where the bold dash lines indicate the edges that will be added to $HC_{2'}$ in Equation (4). Then, lines with cross marks are edges that will be removed from $HC_{2'}$ in Equation (4). Starting from any starting

node, visit the paths formed by the edges in Figure 5a,b in turn, and finally obtain two EDHCs. The results are shown below.

Lemma 1. *The three Hamiltonian cycles HC_i' for $i = 1, 2, 3$ constructed from Equations (1)–(4) are edge-disjoint in $FLTQ_5$.*

Clearly, $|E(FLTQ_5)| = (6 \times 32) / 2 = 96$. Since each Hamiltonian cycle has 32 edges in $FLTQ_5$, all edges of $FLTQ_5$ are exhausted when the above three EDHCs are constructed. Finally, the construction algorithm was provided as follows (Algorithm 1):

Algorithm 1: Constructing 3 EDHCs in $FLTQ_5$

Input: $FLTQ_5$

Output: 3 EDHCs $HC_{1'}, HC_{2'}, HC_{3'}$

Step 1 $HC_1 = 0 - 1 - 13 - 15 - 9 - 11 - 7 - 6 - 14 - 12 - 8 - 10 - 2 - 3 - 5 - 4 - 0$;

Step 2 $HC_2 = 0 - 2 - 6 - 4 - 12 - 13 - 11 - 10 - 14 - 15 - 3 - 1 - 7 - 5 - 9 - 8 - 0$;

Step 3 $E(HC_{1'}) = E(HC_1^0) \cup E(HC_1^1) \cup \{(0, 16), (4, 20)\} - \{(0, 4), (16, 20)\}$;

Step 4 $E(HC_{2'}) = E(HC_2^0) \cup E(HC_2^1) \cup \{(2, 18), (6, 22)\} - \{(2, 6), (18, 22)\}$;

Step 5 $C_{16}^A = 0 - 4 - 27 - 3 - 28 - 12 - 19 - 11 - 20 - 16 - 15 - 23 - 8 - 24 - 7 - 31 - 0$;

Step 6 $C_{16}^B = 1 - 25 - 6 - 2 - 29 - 5 - 26 - 10 - 21 - 13 - 18 - 22 - 9 - 17 - 14 - 30 - 1$;

Step 7 $E(HC_{3'}) = E(C_{16}^A) \cup E(C_{16}^B) \cup \{(1, 3), (5, 7), (27, 29), (30, 31)\} - \{(1, 30), (3, 27), (5, 29), (7, 31)\}$;

Step 8 $E(HC_{2'}) = E(HC_2) \cup \{(1, 30), (3, 27), (5, 29), (7, 31)\} - \{(1, 3), (5, 7), (27, 29), (30, 31)\}$;

Step 9 **Return** $HC_{1'}, HC_{2'}, HC_{3'}$;

Theorem 3. *For $n \geq 5$, there exist three edge-disjoint Hamiltonian cycles in $FLTQ_n$.*

Proof of Theorem 3. First, according to Lemma 1, this theorem holds when $n = 5$. By Definition 2, $FLTQ_n$ is a graph obtained from LTQ_n by adding all complementary edges. According to Theorem 1, there exist three EDHCs in LTQ_n while $n \geq 6$. Therefore, this theorem is proved. \square

3.2. Three EDHCs in FCQ_n

There are two EDHCs of FCQ_5 that were built from [20]. Then, we adjust some of the edges in the second Hamiltonian cycle so that the remaining edges can be used to construct the third Hamiltonian cycle.

Let HC_1 and HC_2 be two EDHCs of CQ_4 , as shown in Figure 6. According to Definitions 3 and 4, FCQ_5 can be decomposed into two copies of CQ_4 , and each with two of the aforementioned Hamiltonian cycles. Partition FCQ_5 into two disjoint subcubes CQ_4^i for $i \in \{0, 1\}$, and let HC_k^i for $k \in \{1, 2\}$ be the corresponding k -th Hamiltonian cycle in the subcube CQ_4^i such that each cycle maps to HC_k in CQ_4 . Next, the two Hamiltonian cycles of FCQ_5 , $HC_{1'}$ and $HC_{2'}$, can be constructed by merging HC_k^0 and HC_k^1 for $k = 1, 2$ by adjusting two edges in each cycle, which are described as follows:

$$E(HC_{1'}) = E(HC_1^0) \cup E(HC_1^1) \cup \{(0, 16), (2, 18)\} - \{(0, 2), (16, 18)\} \tag{5}$$

and

$$E(HC_{2'}) = E(HC_2^0) \cup E(HC_2^1) \cup \{(8, 24), (10, 26)\} - \{(8, 10), (24, 26)\} \tag{6}$$

For example, Figure 7 depicts the Hamiltonian cycles $HC_{1'}$ and $HC_{2'}$ of FCQ_5 according to Equations (5) and (6), respectively. From the drawing, readers may imagine that the dotted line divides the whole FCQ_5 into two subcubes of equal size, and nodes in the left and right parts are mirrored, and their labels differ by ± 16 .

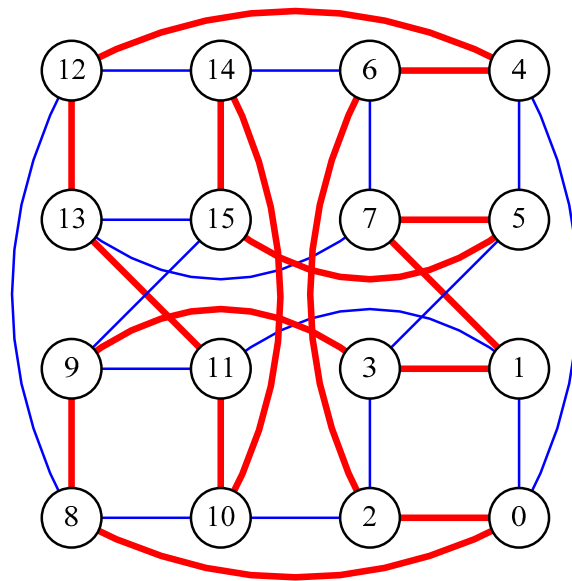


Figure 6. Two EDHCs in CQ_4 . Thick red lines and thin blue lines indicate the first and second Hamiltonian cycles, respectively.

That $E(FCQ_5) - \{E(HC_{1'}) \cup E(HC_{2'})\}$ includes four C_8 cycles: $C_8^A = 0 - 2 - 29 - 7 - 24 - 26 - 5 - 31 - 0$, $C_8^B = 1 - 19 - 12 - 20 - 11 - 25 - 6 - 30 - 1$, $C_8^C = 3 - 17 - 14 - 22 - 9 - 27 - 4 - 28 - 3$, and $C_{16}^D = 8 - 10 - 21 - 15 - 16 - 18 - 13 - 23 - 8$. The third Hamiltonian cycle of FCQ_5 , $HC_{3'}$, can be constructed as follows:

$$E(HC_{3'}) = E(C_8^A) \cup E(C_8^B) \cup E(C_8^C) \cup E(C_8^D) \cup \{(0, 1), (2, 3), (16, 17), (18, 19)\} - \{(0, 2), (1, 19), (3, 17), (16, 18)\} \quad (7)$$

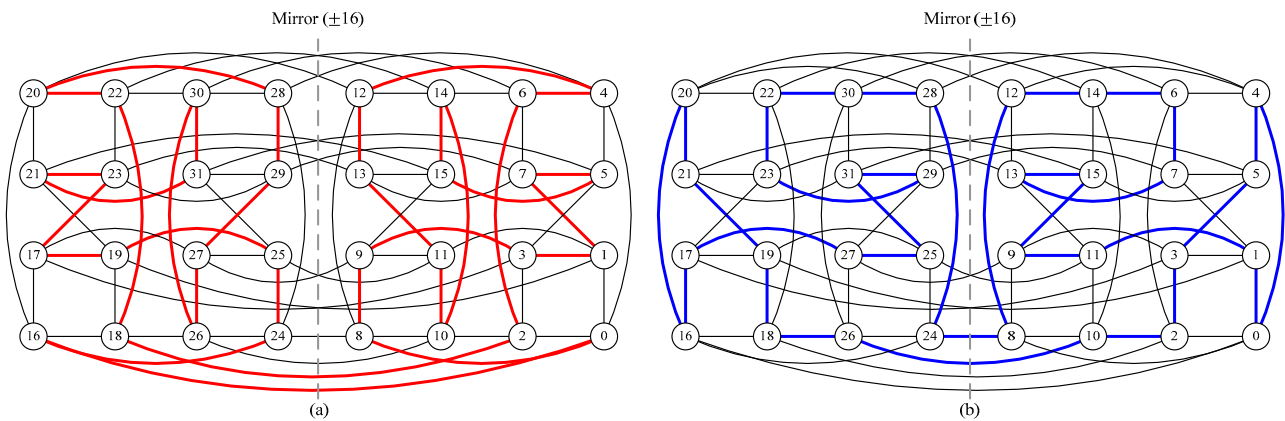


Figure 7. The Hamiltonian cycles (a) $HC_{1'}$ and (b) $HC_{2'}$ of FCQ_5 according to Equation (5) and Equation (6), respectively, where thick lines indicate the edges of the cycle. Complementary edges are omitted because it is convenient for intuitive judgment and the two Hamiltonian cycles do not use them.

Figure 8a depicts the construction of $HC_{3'}$, where the bold lines indicate the edges that will be added to $HC_{3'}$ by Equation (7). Then, lines marked with crosses are the edges that will be removed from $HC_{3'}$ by Equation (7). Again, the construction of the third Hamiltonian cycle mentioned above is obtained using the edge-swapping technique. Therefore, with the above fine-tuning, it is necessary to adjust the edge set of $HC_{2'}$, obtained

from Equation (6) by swapping two edge sets, $\{(0, 1), (2, 3), (16, 17), (18, 19)\}$ and $\{(0, 2), (1, 19), (3, 17), (16, 18)\}$, as follows:

$$E(HC_{2'}) = E(HC_2) \cup \{(0, 2), (1, 19), (3, 17), (16, 18)\} - \{(0, 1), (2, 3), (16, 17), (18, 19)\} \quad (8)$$

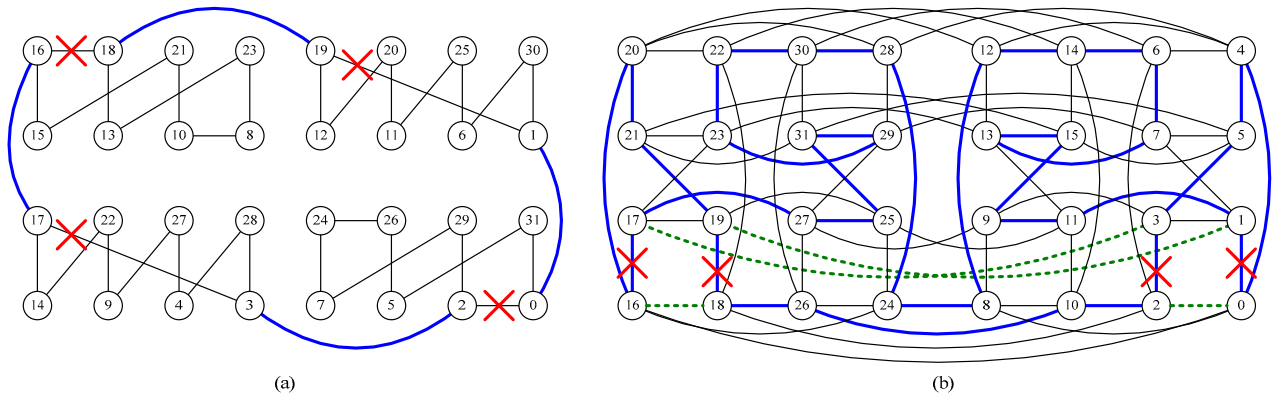


Figure 8. The Hamiltonian cycles (a) $HC_{3'}$ and (b) new $HC_{2'}$ of FCQ_5 were constructed from Equation (7) and Equation (8), respectively.

Similarly, Figure 8b illustrates the construction of the new $HC_{2'}$, where the bold dash lines indicate the edges that will be added to $HC_{2'}$ in Equation (8). Then, lines with cross marks are edges that will be removed from $HC_{2'}$ in Equation (8). Starting from any node, visit the paths formed by the edges in Figure 8a,b in sequence, and finally obtain two EDHCs, and hence the result is shown below.

Lemma 2. The three Hamiltonian cycles $HC_{i'}$ for $i = 1, 2, 3$ constructed from Equations (5)–(8) are edge-disjoint in FCQ_5 .

Clearly, $|E(FCQ_5)| = (6 \times 32) / 2 = 96$. Since each Hamiltonian cycle of FCQ_5 has 32 edges, all edges of FCQ_5 are exhausted when the above three EDHCs are constructed. Finally, the construction algorithm was provided as follows (Algorithm 2):

Algorithm 2: Constructing 3 EDHCs in FCQ_5

Input: FCQ_5

Output: 3 EDHCs $HC_{1'}, HC_{2'}, HC_{3'}$

Step 1 $HC_1 = 0 - 2 - 6 - 4 - 12 - 13 - 11 - 10 - 14 - 15 - 5 - 7 - 1 - 3 - 9 - 8 - 0$;

Step 2 $HC_2 = 0 - 1 - 11 - 9 - 15 - 13 - 7 - 6 - 14 - 12 - 8 - 10 - 2 - 3 - 5 - 4 - 0$;

Step 3 $E(HC_{1'}) = E(HC_1^0) \cup E(HC_1^1) \cup \{(0, 16), (2, 18)\} - \{(0, 2), (16, 18)\}$;

Step 4 $E(HC_{2'}) = E(HC_2^0) \cup E(HC_2^1) \cup \{(8, 24), (10, 26)\} - \{(8, 10), (24, 26)\}$;

Step 5 $C_8^A = 0 - 2 - 29 - 7 - 24 - 26 - 5 - 31 - 0$;

Step 6 $C_8^B = 1 - 19 - 12 - 20 - 11 - 25 - 6 - 30 - 1$;

Step 7 $C_8^C = 3 - 17 - 14 - 22 - 9 - 27 - 4 - 28 - 3$;

Step 8 $C_{16}^D = 8 - 10 - 21 - 15 - 16 - 18 - 13 - 23 - 8$;

Step 9 $E(HC_{3'}) = E(C_8^A) \cup E(C_8^B) \cup E(C_8^C) \cup E(C_8^D) \cup \{(0, 1), (2, 3), (16, 17), (18, 19)\} - \{(0, 2), (1, 19), (3, 17), (16, 18)\}$;

Step 10 $E(HC_{2'}) = E(HC_2) \cup \{(0, 2), (1, 19), (3, 17), (16, 18)\} - \{(0, 1), (2, 3), (16, 17), (18, 19)\}$;

Step 11 **Return** $HC_{1'}, HC_{2'}, HC_{3'}$;

Theorem 4. For $n \geq 5$, there exist three edge-disjoint Hamiltonian cycles in FCQ_n .

Proof of Theorem 4. First, when $n = 5$, this theorem holds by Lemma 2. Then, according to Definition 4, FCQ_n is constructed by adding all complementary edges to CQ_n . By Theorem 2, there exist three EDHCs in CQ_n while $n \geq 6$. Therefore, this theorem is proved. \square

4. Performance Evaluation

In this section, considering that multiple faulty edges will occur randomly, the performance of data broadcasting is simulated and evaluated by two and three EDHCs in $FLTQ_n$ and FCQ_n when $5 \leq n \leq 9$. In this paper, three EDHCs of $FLTQ_n$ (respectively, FCQ_n) are improved from the two EDHCs of LTQ_n [11] (respectively, CQ_n [14]). Therefore, the construction of three EDHCs has adopted the method in Section 3, and the construction of two EDHCs has adopted the method of [11] and [14]. For two kinds of networks, some C programs are used to implement data broadcasting according to two and three EDHCs, respectively. To speed up the evaluation, the simulation was carried out by using a 5.10 GHz Intel® Core™ i9–12900 CPU and 32 GB RAM under the Linux operating system.

For each dimension n and number of faulty edges m while $6 \leq n \leq 9$ and $1 \leq m \leq 10$, the program randomly generates 1,000,000 instances of number-list $(s, f_1, f_2, \dots, f_m)$ with $f_1 \neq f_2 \neq \dots \neq f_m$ for $FLTQ_n$ and FCQ_n , where s and f_i are the source node and faulty edge label, respectively. In general, when the source node s needs to send a distinct message to all other nodes. Applying the ring structure, once a message starts sending, a node can receive a new message from the previous node at each step, keep a copy of the new message for itself, and send the received message to the next node. Considering that m faulty edges will appear randomly and increase the probability of successful broadcast to all nodes, the source node s send the messages to the next nodes simultaneously in two directions through three Hamiltonian cycles.

Firstly, this study is interested in evaluating the broadcast success rate, abbreviated as BSR, which is the ratio of the number of successful data broadcasts over generated instances. Then, when the broadcast fails, the program computes three statistical quantities related to the number of unreachable nodes: (a) mean, (b) standard deviation, and (c) maximum number. More descriptions of the simulation process are available on the website [27] as Supplementary Materials.

Table 1 (respectively, Table 2) shows the simulation results of BSR for data broadcasting in $FLTQ_n$ adopting two EDHCs (respectively, three EDHCs) as the broadcasting channels in two directions, respectively. When the number of faulty edges $m \geq 4$ in Table 1 and $m \geq 6$ in Table 2, sometimes the broadcast fails. Then, Tables 3 and 4 show two quantities mentioned above that are calculated by the usual way in statistics.

Table 1. BSR of fault-tolerant data broadcasting in $FLTQ_n$ using two EDHCs while $1 \leq m \leq 10$.

| m | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $FLTQ_5$ | 1.000 | 1.000 | 1.000 | 0.943 | 0.829 | 0.686 | 0.540 | 0.410 | 0.303 | 0.216 |
| $FLTQ_6$ | 1.000 | 1.000 | 1.000 | 0.970 | 0.903 | 0.807 | 0.696 | 0.583 | 0.476 | 0.382 |
| $FLTQ_7$ | 1.000 | 1.000 | 1.000 | 0.984 | 0.942 | 0.877 | 0.795 | 0.704 | 0.612 | 0.522 |
| $FLTQ_8$ | 1.000 | 1.000 | 1.000 | 0.990 | 0.963 | 0.918 | 0.858 | 0.786 | 0.709 | 0.629 |
| $FLTQ_9$ | 1.000 | 1.000 | 1.000 | 0.993 | 0.975 | 0.943 | 0.898 | 0.842 | 0.779 | 0.711 |

Table 2. BSR of fault-tolerant data broadcasting in $FLTQ_n$ using three EDHCs while $1 \leq m \leq 10$.

| m | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $FLTQ_5$ | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.930 | 0.796 | 0.636 | 0.481 | 0.348 |
| $FLTQ_6$ | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.970 | 0.897 | 0.790 | 0.668 | 0.545 |
| $FLTQ_7$ | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.989 | 0.955 | 0.897 | 0.820 | 0.730 |
| $FLTQ_8$ | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.994 | 0.976 | 0.940 | 0.887 | 0.822 |
| $FLTQ_9$ | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.997 | 0.987 | 0.967 | 0.934 | 0.889 |

According to Tables 1–4, Figure 9a,b are drawn, respectively. Obviously, BSR decreases as the number of faulty edges increases, and the mean of the number of unreachable nodes increases as the dimension of $FLTQ_n$ increases.

Table 3. In $FLTQ_n$, while $5 \leq n \leq 9$ and $6 \leq m \leq 10$, the mean and standard deviation of the number of unreachable nodes by using two EDHCs.

| m | 6 | 7 | 8 | 9 | 10 |
|----------|--------------|--------------|--------------|--------------|--------------|
| $FLTQ_5$ | 6.7 (5.8) | 7.5 (6.1) | 8.3 (6.4) | 9.2 (6.7) | 10.1 (6.9) |
| $FLTQ_6$ | 13.8 (11.7) | 15.1 (12.2) | 16.4 (12.6) | 17.8 (13.1) | 19.3 (13.4) |
| $FLTQ_7$ | 28.0 (23.1) | 30.0 (23.9) | 32.1 (24.6) | 34.5 (25.5) | 36.8 (26.1) |
| $FLTQ_8$ | 56.2 (45.6) | 59.5 (46.8) | 63.2 (48.3) | 66.9 (49.5) | 70.6 (50.8) |
| $FLTQ_9$ | 112.1 (89.8) | 118.1 (92.1) | 123.9 (94.7) | 130.0 (96.8) | 136.1 (98.9) |

Numbers in parentheses are standard deviations.

Table 4. In $FLTQ_n$, while $5 \leq n \leq 9$ and $6 \leq m \leq 10$, the mean and standard deviation of the number of unreachable nodes by using three EDHCs.

| m | 6 | 7 | 8 | 9 | 10 |
|----------|-------------|-------------|-------------|-------------|-------------|
| $FLTQ_5$ | 2.5 (2.7) | 3.0 (3.2) | 3.6 (3.5) | 4.3 (4.0) | 5.0 (4.4) |
| $FLTQ_6$ | 4.6 (5.3) | 5.4 (5.9) | 6.3 (6.6) | 7.3 (7.3) | 8.4 (8.0) |
| $FLTQ_7$ | 9.1 (10.4) | 10.2 (11.3) | 11.4 (12.1) | 12.8 (13.1) | 14.2 (14.0) |
| $FLTQ_8$ | 25.3 (27.4) | 28.2 (29.4) | 31.1 (31.5) | 34.1 (33.3) | 37.4 (35.1) |
| $FLTQ_9$ | 58.9 (61.4) | 65.9 (65.7) | 70.1 (67.8) | 74.8 (70.5) | 81.1 (73.7) |

Numbers in parentheses are standard deviations.

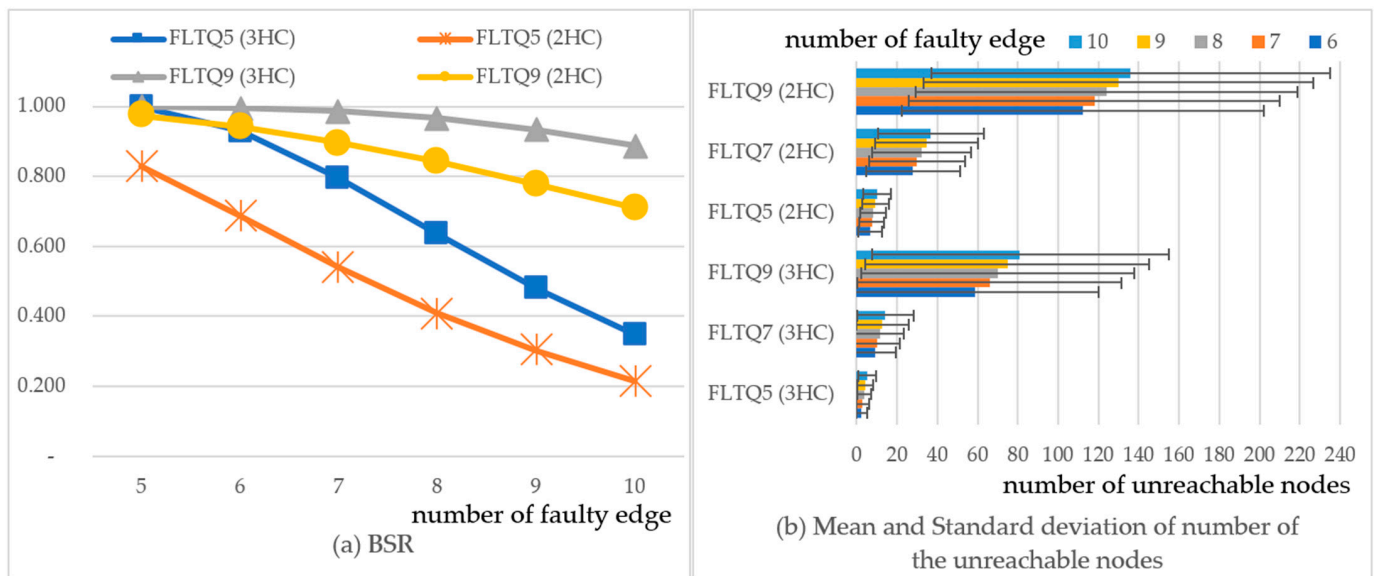


Figure 9. (a) BSR and (b) the mean and standard deviation of the number of unreachable nodes in $FLTQ_n$ while $5 \leq n \leq 9$ and $6 \leq m \leq 10$.

Table 5 (respectively, Table 6) shows the simulation results of BSR for data broadcasting in FCQ_n adopting two EDHCs (respectively, three EDHCs) as the broadcasting channels in two directions, respectively. Then, Tables 7 and 8 show two quantities mentioned above that are calculated by the usual way in statistics.

Table 5. BSR of fault-tolerant data broadcasting in FCQ_n using two EDHCs while $1 \leq m \leq 10$.

| m | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| FCQ_5 | 1.000 | 1.000 | 1.000 | 0.943 | 0.830 | 0.688 | 0.542 | 0.412 | 0.304 | 0.218 |
| FCQ_6 | 1.000 | 1.000 | 1.000 | 0.970 | 0.901 | 0.804 | 0.692 | 0.578 | 0.471 | 0.375 |
| FCQ_7 | 1.000 | 1.000 | 1.000 | 0.983 | 0.941 | 0.875 | 0.793 | 0.701 | 0.607 | 0.516 |
| FCQ_8 | 1.000 | 1.000 | 1.000 | 0.990 | 0.963 | 0.917 | 0.857 | 0.785 | 0.707 | 0.627 |
| FCQ_9 | 1.000 | 1.000 | 1.000 | 0.993 | 0.975 | 0.942 | 0.897 | 0.842 | 0.778 | 0.709 |

Table 6. BSR of fault-tolerant data broadcasting in FCQ_n using three EDHCs while $1 \leq m \leq 10$.

| m | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| FCQ_5 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.933 | 0.802 | 0.645 | 0.492 | 0.360 |
| FCQ_6 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.970 | 0.898 | 0.792 | 0.670 | 0.547 |
| FCQ_7 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.989 | 0.955 | 0.896 | 0.818 | 0.727 |
| FCQ_8 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.994 | 0.940 | 0.940 | 0.888 | 0.821 |
| FCQ_9 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.997 | 0.987 | 0.967 | 0.934 | 0.889 |

Table 7. In FCQ_n , while $5 \leq n \leq 9$ and $6 \leq m \leq 10$, the mean and standard deviation of the number of unreachable nodes by using two EDHCs.

| M | 6 | 7 | 8 | 9 | 10 |
|---------|--------------|--------------|--------------|--------------|--------------|
| FCQ_5 | 6.3 (5.5) | 7.0 (5.8) | 7.8 (6.1) | 8.6 (6.4) | 9.5 (6.7) |
| FCQ_6 | 13.4 (11.4) | 14.6 (11.9) | 15.9 (12.4) | 17.3 (12.8) | 18.7 (13.2) |
| FCQ_7 | 27.4 (23.0) | 29.5 (23.9) | 31.6 (24.6) | 33.9 (25.4) | 36.2 (26.1) |
| FCQ_8 | 55.3 (45.4) | 59.0 (46.9) | 62.6 (48.4) | 66.3 (49.6) | 70.2 (50.8) |
| FCQ_9 | 111.9 (90.4) | 116.7 (92.0) | 123.2 (94.6) | 129.7 (96.9) | 136.0 (99.2) |

Numbers in parentheses are standard deviations.

Table 8. In FCQ_n , while $5 \leq n \leq 9$ and $6 \leq m \leq 10$, the mean and standard deviation of the number of unreachable nodes by using three EDHCs.

| m | 6 | 7 | 8 | 9 | 10 |
|---------|-------------|-------------|-------------|-------------|-------------|
| FCQ_5 | 2.6 (2.7) | 3.1 (3.1) | 3.6 (3.6) | 4.3 (3.9) | 5.0 (4.3) |
| FCQ_6 | 4.4 (5.0) | 5.2 (5.7) | 6.0 (6.4) | 7.0 (7.1) | 8.1 (7.7) |
| FCQ_7 | 9.3 (10.6) | 10.3 (11.4) | 11.6 (12.4) | 12.9 (13.3) | 14.4 (14.3) |
| FCQ_8 | 25.3 (27.6) | 31.0 (31.5) | 31.0 (31.5) | 33.9 (33.3) | 37.3 (35.2) |
| FCQ_9 | 60.0 (62.6) | 64.9 (64.3) | 70.2 (67.5) | 75.5 (70.8) | 80.8 (73.8) |

Numbers in parentheses are standard deviations.

According to Tables 5–8, Figure 10a,b are drawn, respectively. Clearly, BSR decreases as the number of faulty edges increases, and the mean of unreachable nodes increases as the dimension of FCQ_n increases.

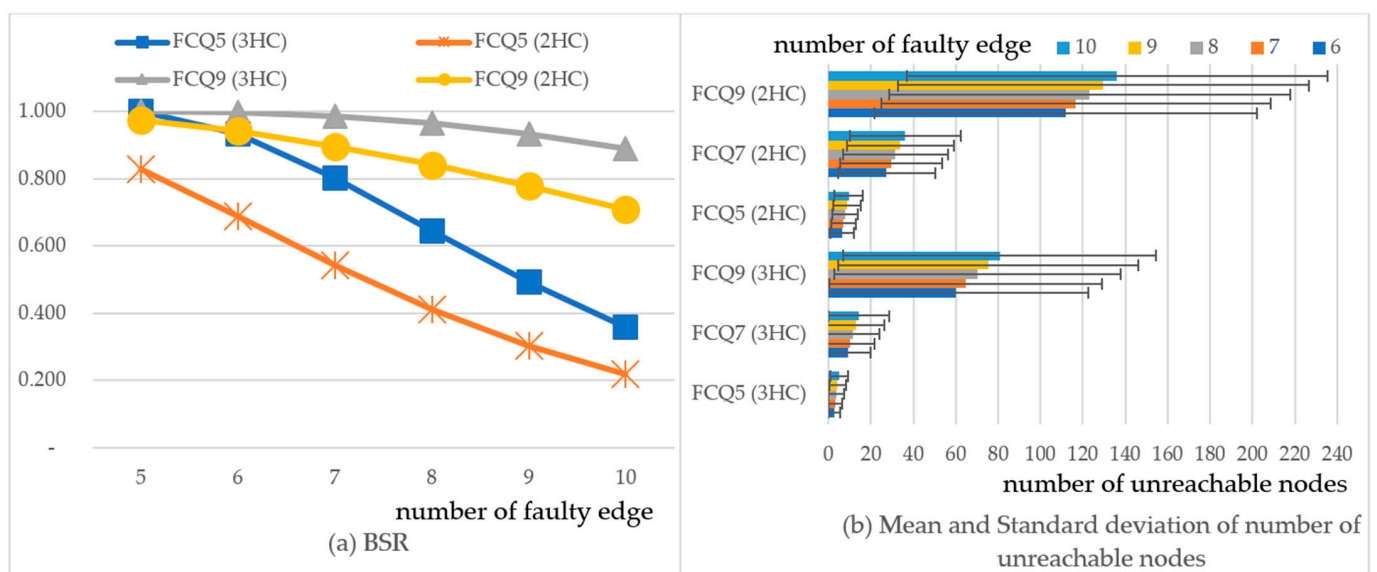


Figure 10. (a) BSR and (b) the mean and standard deviation of the number of unreachable nodes in FCQ_n while $5 \leq n \leq 9$ and $6 \leq m \leq 10$.

5. Discussion

From Tables 1–8 and Figures 9 and 10, the simulation results of $FLTQ_n$ and FCQ_n are very similar, and there are four phenomena as follows:

- According to the main Theorems 3 and 4, there exist three edge-disjoint Hamiltonian cycles in $FLTQ_n$ and FCQ_n while $n \geq 5$. The research results provide three EDHCs as broadcasting channels and transmission in two directions, no matter what scale of $FLTQ_n$ and FCQ_n for $5 \leq n \leq 9$, its transmission can reach 100% success when the number of faulty edges $m \leq 5$. The intuitive prediction is that BSR should present a decreasing function to the number of faulty edges. As expected, all simulations are consistent with this phenomenon. For example, take FCQ_5 in Table 5 as an example for illustration. If $BSR \geq 80\%$ is required, then it only allows seven faulty edges. It can tolerate eight faulty edges while allowing no more than half of the broadcasts to fail. Accordingly, this means that the least number of faulty edges, the larger the corresponding BSR.
- For $m \geq 6$, BSR increases with expanding the scale of $FLTQ_n$ and FCQ_n . The reason is obvious since when m is fixed, the edge failures that occur in Hamiltonian cycles will reduce their probability as the network expands, thus leading to an increase in the success rate. For example, take $FLTQ_5$ and $FLTQ_6$ in Table 1 as an example for illustration. All edges of $FLTQ_5$ are used in 3 EDHCs, but 6/7 edges of $FLTQ_6$ are used in 3 EDHCs. When $m = 10$, BSR increases from 0.348 in $FLTQ_5$ to 0.545 in $FLTQ_6$.
- For $m \geq 6$, the number of unreachable nodes increases with expanding the scale of $FLTQ_n$ and FCQ_n . The size of Hamiltonian cycles is equal to the size of the network. The larger the scale of networks, the larger the number of unreachable nodes. For example, take FCQ_8 and FCQ_9 in Table 8 as an example for illustration. The size of FCQ_n is 2^n , then $|V(FCQ_8)| = 256$ and $|V(FCQ_9)| = 512$. When $m = 10$, the mean of the unreachable nodes increases from 37.3 in FCQ_8 to 80.8 in FCQ_9 .
- In this paper, three EDHCs of $FLTQ_n$ (respectively, FCQ_n) are compared with the two EDHCs of LTQ_n [11] (respectively, CQ_n [14]). According to Figures 9a and 10a, the BSR of three EDHCs is better than that of two EDHCs in both $FLTQ_n$ and FCQ_n . Moreover, the smaller the size of the network, the greater the gap between the BSRs. In addition, observing Figures 9b and 10b, it can be found that the average number of unreachable nodes of the three EDHCs is 0.3~0.7 of that of the two EDHCs. In broadcast applications on $FLTQ_n$ and FCQ_n , the three EDHCs in this paper are better than the two EDHCs in [11,14].

6. Conclusions

This paper first investigates the construction of three EDHCs in $FLTQ_n$. Then, the same method is also applied to FCQ_n , thus, the research results of the three EDHCs of FCQ_n are also obtained. Next, use the three EDHCs of $FLTQ_n$ and FCQ_n as transmission channels to realize fault-tolerant data broadcasting. Considering that multiple faulty edges will appear randomly, the performance of data broadcasting is evaluated by simulation with three EDHCs in $FLTQ_n$ and FCQ_n when $5 \leq n \leq 9$. The research results provide three EDHCs as broadcasting channels and transmission in two directions, and the transmission can reach 100% success when the number of faulty edges $m \leq 5$. Generally, networks with higher edge connectivity can construct more EDHCs. As future work, the study is interested in seeing if there are four EDHCs on $FLTQ_n$ or FCQ_n when $n \geq 7$. The question of whether a Hamiltonian cycle exists in a given graph is NP-complete. While the number of edges of seven-dimensional $FLTQ$ or FCQ has been increased to 128, it may not be easy to find the answer only by using the technique of edge exchange. So, this remains an open question.

Supplementary Materials: Three edge-disjoint Hamiltonian cycles in $FLTQ_5$ and FCQ_5 , simulation description, examples, detailed results, and tables can be viewed at the following website: <http://210.240.238.53/threeHC1> (accessed on 1 June 2023).

Funding: This research was funded by MOST through the Ministry of Science and Technology, Taiwan, under Grant MOST 111-2221-E-131-012.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Saad, Y.; Schultz, M.H. Topological properties of hypercubes. *IEEE Trans. Comput.* **1988**, *37*, 867–872. [CrossRef]
2. Wang, D. A low-cost fault-tolerant structure for the hypercube. *J. Supercomput.* **2001**, *20*, 203–216. [CrossRef]
3. Lin, T.-J.; Hsieh, S.-Y.; Juan, J.S.-T. Embedding cycles and paths in product networks and their applications to multiprocessor systems. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1081–1089.
4. Wang, N.-C.; Yen, C.-P.; Chu, C.-P. Multicast communication in wormhole-routed symmetric networks with Hamiltonian cycle model. *J. Syst. Arch.* **2005**, *51*, 165–183. [CrossRef]
5. Sabirgiriraj, M.; Manoharan, K. Wavelength allocation for all-to-all broadcast in bidirectional optical WDM modified ring. *Optik* **2019**, *179*, 545–556. [CrossRef]
6. Bae, M.M.; Bose, B. Edge disjoint Hamiltonian cycles in k-ary n-cubes and hypercubes. *IEEE Trans. Comput.* **2003**, *52*, 1271–1284. [CrossRef]
7. Rowley, R.; Bose, B. Edge disjoint Hamiltonian cycles in de Bruijn networks. In Proceedings of the 6th Distributed Memory Computing Conference, Portland, OR, USA, 28 April–1 May 1991; pp. 707–709.
8. Barth, D.; Raspaud, A. Two edge-disjoint Hamiltonian cycles in the butterfly graph. *Inform. Process. Lett.* **1994**, *51*, 175–179. [CrossRef]
9. Lee, S.; Shin, K. Interleaved all-to-all reliable broadcast on meshes and hypercubes. *IEEE Trans. Parallel Distrib. Syst.* **1994**, *5*, 449–458.
10. Petrovic, V.; Thomassen, C. Edge-disjoint Hamiltonian cycles in hypertournaments. *J. Graph. Theory* **2006**, *51*, 49–52.
11. Hung, R.W. Embedding two edge-disjoint Hamiltonian cycles into locally twisted cubes. *Theoret. Comput. Sci.* **2011**, *412*, 4747–4753. [CrossRef]
12. Hung, R.W. Constructing two edge-disjoint Hamiltonian cycles and two-equal path cover in augmented cubes. *J. Comput. Sci.* **2012**, *39*, 42–49.
13. Hung, R.W.; Chan, S.J.; Liao, C.C. Embedding two edge-disjoint Hamiltonian cycles and two equal node-disjoint cycles into twisted cubes. *Lect. Notes Eng. Comput. Sci.* **2012**, *2195*, 362–367.
14. Hung, R.W. The property of edge-disjoint Hamiltonian cycles in transposition networks and hypercube-like networks. *Discret. Appl. Math.* **2015**, *181*, 109–122. [CrossRef]
15. Wang, Y.; Fan, J.; Liu, W.; Wang, X. Embedding Two Edge-Disjoint Hamiltonian Cycles into Parity Cubes. *Appl. Mech. Mater.* **2013**, *336*, 2248–2251.
16. Hussain, Z.A.; Bose, B.; Al-Dhelaan, A. Edge disjoint Hamiltonian cycles in Eisenstein-Jacobi networks. *J. Parallel Distrib. Comput.* **2015**, *86*, 62–70. [CrossRef]
17. Albader, B.; Bose, B. Edge Disjoint Hamiltonian Cycles in Gaussian Networks. *IEEE Trans. Comput.* **2016**, *65*, 315–321.
18. Yang, D.W.; Xu, Z.; Feng, Y.Q.; Lee, J. Symmetric property and edge-disjoint Hamiltonian cycles of the spined cube. *Appl. Math. Comput.* **2023**, *452*, 128075. [CrossRef]
19. Cheng, D. Recursive definition and two edge-disjoint Hamiltonian cycles of bubble-sort star graphs. *Int. J. Comput. Math. Comput. Syst. Theory* **2023**, *8*, 152–159. [CrossRef]
20. Pai, K.J. A parallel algorithm for constructing two edge-disjoint Hamiltonian cycles in crossed cubes. In *AAIM 2020: Algorithmic Aspects in Information and Management, Proceedings of the 14th International Conference on Algorithmic Applications in Management, Jinhua, China, 10–12 August 2020*; Zhang, Z., Li, W., Du, D.Z., Eds.; Springer: Cham, Switzerland, 2020; Volume 12290, pp. 448–455.
21. Li, S.Y.; Chang, J.M.; Pai, K.J. A Parallel Algorithm for Constructing Two Edge-disjoint Hamiltonian Cycles in Locally Twisted Cubes. In Proceedings of the 2020 International Computer Symposium, Tainan, Taiwan, 17–19 December 2020; pp. 116–119.
22. Pai, K.J. Embedding Three Edge-Disjoint Hamiltonian Cycles into Locally Twisted Cubes. In *COCOON 2021: Computing and Combinatorics, Proceedings of the International Computing and Combinatorics Conference, Tainan, Taiwan, 24–26 October 2021*; Chen, C.Y., Hon, W.K., Hung, L.J., Lee, C.W., Eds.; Springer: Cham, Switzerland, 2021; Volume 13025, pp. 367–374.
23. Pai, K.J.; Wu, R.Y.; Peng, S.L.; Chang, J.M. Three edge-disjoint Hamiltonian cycles in crossed cubes with applications to fault-tolerant data broadcasting. *J. Supercomput.* **2023**, *79*, 4126–4145. [CrossRef]
24. Yang, X.; Evans, D.J.; Megson, G.M. The locally twisted cubes. *Int. J. Comput. Math.* **2005**, *82*, 401–413. [CrossRef]
25. Peng, S.; Guo, C.; Yang, B. Topological properties of folded locally twisted cubes. *J. Comput. Inform. Syst.* **2015**, *11*, 7667–7676.
26. Efe, K. The crossed cube architecture for parallel computation. *IEEE Trans. Parallel Distrib. Syst.* **1992**, *3*, 513–524. [CrossRef]
27. Simulation Results for Evaluating the Performance of Fault-Tolerant Data Broadcasting in FLTQn and FLTQn Using Three EDHCs. Available online: <http://210.240.238.53/threeHC1> (accessed on 1 June 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.