兩個連邊互斥漢米爾頓迴圈的網路傳輸應用在折疊超立方體中

Network transmission applications of two edge-disjoint Hamiltonian cycles in folded hypercubes

白恭瑞
明志科技大學
工業工程與管理系

Hassane MIMOUN
Innovation Technology Department ESIEE Paris, University Gustave Eiffel

陳琨太
明志科技大學
工業工程與管理系

摘要

n 維摺疊超立方體 FQ_n 是超立方體 Q_n 的一種變形結構，其頂點數與 Q_n 相同，邊數則為 Q_n 的 $(n+1)/n$ 倍，但其直徑約僅為 Q_n 的一半。在互連網路中，許多高效率的通訊演算法可建立在漢米頓迴路的基礎上；漢米頓迴路是一種恰好通過每一個頂點一次的迴路。此外，兩條邊互斥的漢米頓迴路可應用於邊故障容忍傳輸與全對全廣播等互連網路問題。本文首先提出在 FQ_3 中建構兩條邊互斥漢米頓迴路的方法。接著，透過歸納推理，進一步提出一個演算法，用以在 $n \geq 4$ 的 FQ_n 中建構兩條邊互斥的漢米頓迴路。基於上述結果，本文以所建構之兩條邊互斥漢米頓迴路作為傳輸通道，模擬 FQ_n 中的全對全廣播，並與傳統僅使用一條漢米頓迴路的方法進行比較，其中 $3 \leqslant n \leqslant 10$。實驗比較結果顯示，在所測試的 FQ_n 中，使用兩條邊互斥漢米頓迴路進行廣播，其效能優於僅使用一條漢米頓迴路的方法。

關鍵字： 邊互斥漢米頓迴路、摺疊超立方體、全對全廣播、互連網路。

Abstract

The n-dimensional folded hypercube FQ_n, a variation of the hypercube Q_n, has the same number of vertices as Q_n and $(n+1)/n$ times the number of edges, but a diameter only about half of that of Q_n. In the interconnection network, some efficient communication algorithms can be designed based on a Hamiltonian cycle, i.e., the cycle that visits each vertex exactly once. In addition, two edge-disjoint Hamiltonian cycles also provide applications such as edge-fault-tolerant transmission and all-to-all broadcasting for the interconnection network. In this paper, we first provide the construction of two edge-disjoint Hamiltonian cycles in FQ_3. By inductive reasoning, we give an algorithm to construct two edge-disjoint Hamiltonian cycles in FQ_n with $n \geqslant 4$. Based on these results, we simulate all-to-all broadcasting via the developed two edge-disjoint Hamiltonian cycles (respectively, the traditional results using only one Hamiltonian cycle) in FQ_n as the transmission channels, where $3 \leqslant n \leqslant 10$. Finally, the comparison results show that in the tested FQ_n, broadcasting with two edge-disjoint Hamiltonian cycles performs better than one Hamiltonian cycle.

Keywords: Edge-disjoint Hamiltonian Cycles, Folded Hypercubes, all-to-all broadcasting, Interconnection Networks.

1. Introduction

The design of the interconnection network is one of the important issues in parallel computing systems and data centers. The interconnect topology is usually modeled as a graph, where vertices represent processing units and edges represent communication links. Many interconnect topologies have been proposed in the past, the most popular of which are torus, mesh, hypercube, hypercube variants, etc. The hypercube [7, 14, 15], abbreviated as Q_n, is one of the earliest proposed and popular interconnection networks. It has attractive properties including regularity, vertex symmetry, link symmetry, small diameter, strong connectivity, recursive structure, partitioning capability, and low link complexity [14, 15]. Subsequently, many scholars proposed variations of the hypercube, and the folded hypercube is one of them. The n-dimensional folded hypercube FQ_n was first proposed by El-Amawy and Latifi [5] and is currently one of the most widely studied topics. It has the same number of vertices as Q_n and $(n+1)/n$ times the number of edges as Q_n, but its diameter is only about half that of Q_n.

The reliability of a network depends on its connectivity, edge connectivity, and other related properties, which are important indicators of the network. One of its nice properties is that the network has a Hamiltonian cycle. For ease of description, we will use "graph" and "network" interchangeably in this article. A Hamiltonian cycle is a cycle in a graph that visits every vertex exactly once. In other words, it is the largest cycle in the graph, and its advantages can be seen in [6-8]. Next, if k Hamiltonian cycles in a graph do not share any common edges, they are called k edge-disjoint Hamiltonian cycles. They can provide advantages for algorithms using ring structures while also providing edge-fault-tolerant Hamiltonian properties for interconnected networks. That is, when an edge in a Hamiltonian cycle fails, it can be replaced by another one for transmission.

Another application of k edge-disjoint Hamiltonian cycles is all-to-all broadcast transmission [6]. Consider an all-to-all communication algorithm, where each vertex sends a package to all other vertices. Assume that the number of vertices in the network is m. In the single I/O port model, each vertex can send and receive one package at a time. First, a Hamiltonian cycle is formed in the network, and then at the i^{th} step, $i \in \{1, 2, 3, \cdots, m-1\}$, each vertex can send the package received from the previous vertex at the $(i-1)^{\text{th}}$ step to the next vertex while keeping a copy of the package. This all-to-all communication algorithm requires $m-1$ steps. Now, assume that each vertex has d I/O ports, which means that each vertex can send and receive d packages per unit time. In this case, if we can construct k ($\leqslant d/2$) edge-disjoint Hamiltonian cycles, then each package of size q can be split into packages of size q/k, and each package can be sent along the k edge-disjoint Hamiltonian cycles. Therefore, the time efficiency of the algorithm can be improved by a factor of k.

Finding k edge-disjoint Hamiltonian cycles has attracted the attention of scholars, and some previous works related to this problem in specific networks are described below. Barth and Raspaud [4] showed that there are two edge-disjoint Hamiltonian cycles in butterfly networks. Then, Barden et al. [3] constructed the maximum number of edge-disjoint spanning trees in a hypercube. Bae et al. [2] studied edge-disjoint Hamiltonian cycles in k-ary n-cubes and hypercubes. Hung presented how to construct two edge-disjoint Hamiltonian cycles in locally twisted cubes [9], augmented cubes [10], twisted cubes [12], transposition networks, and hypercube-like networks [11], respectively. Albader and Bose [1] show how to obtain two edge-disjoint Hamiltonian cycles in Gaussian networks for any generator $\alpha = a + bi$. Hussain et al. [13] described that the hexagonal network is a special case of the EJ network that can be obtained by $\alpha = a+(a+1)\rho$, and showed how to generate three edge-disjoint Hamiltonian cycles in the EJ network with generator $\alpha = a + b\rho$ for $\gcd(a, b) = 1$. Yang et al. [16] proved that there exist two edge-disjoint Hamiltonian cycles in the spined cube SQ_n when $n \geqslant 4$.

The rest of the paper is organized as follows: In Section 2, the structures of hypercubes and folded hypercubes are introduced, and some notations are given. Section 3 shows how to build two edge-disjoint Hamiltonian cycles in FQ_n while $n \geqslant 3$, and presents the construction algorithm and its proof. Section 4 simulates data broadcasting in FQ_n for $3 \leqslant n \leqslant 10$ and compares the experimental performance with traditional data broadcasting using one Hamiltonian cycle. Finally, Section 5 is the concluding remarks of this paper.

2. Preliminaries

An interconnected network is usually modeled as an undirected simple graph $G = (V, E)$, where the vertex set V (= $V(G)$) and the edge set E (= $E(G)$) represent the set of processing units and the set of communication links between units, respectively. Next, let G be a labeled graph whose vertices are associated with different binary strings. A vertex of the n-dimensional hypercube Q_n is represented by a binary string of length n. A binary string b of length n is denoted by $b_{n-1}b_{n-2} \cdots b_1b_0$, where b_{n-1} is the most significant bit. Then, we provide the definition of the hypercube as follows:

Definition 1. (Leighton [7]) *The n-dimensional hypercube, denoted by Q_n, is a graph with 2^n vertices such that each vertex corresponds to an n-tuple $(b_n, b_{n-1}, \cdots b_2, b_1)$ on the set $\{0, 1\}^n$ and two vertices are linked by an edge if and only if they differ in exactly one coordinate.*

For example, Figure 1(a) shows the 3-dimensional hypercube Q_3. Next we present the definition of the folded hypercube as follows:

Definition 2. (El-Amawy and Latifi [5]) *The n-dimensional folded hypercube, denoted by FQ_n, is a graph with 2^n vertices such that each vertex corresponds to an n-tuple $(b_n, b_{n-1}, \cdots b_2, b_1)$ on the set $\{0, 1\}^n$. Its edge set consists of two types of edges:*

- *Two vertices are linked by a hypercube edge if and only if they differ in exactly one coordinate.*
- *Two vertices are linked by a complementary edge if and only if they differ in each coordinate.*

For example, Figure 1(b) shows the 3-dimensional folded hypercube FQ_3. In addition, a path P_k of length k in G, denoted by v_0 - v_1 - v_2 - … - v_{k-2} - v_{k-1} - v_k, is a sequence $(v_0, v_1, v_2, \ldots, v_{k-1}, v_k)$ of vertices such that $(v_{k-1}, v_0) \in E$ and $(v_i, v_{i+1}) \in E$ for $0 \leq i \leq k-2$. A cycle C_k of length k in G, denoted by v_0 - v_1 - v_2 - … - v_{k-2} - v_{k-1} - v_0, is a sequence $(v_0, v_1, v_2, \ldots, v_{k-1}, v_0)$ of vertices such that $(v_{k-1}, v_0) \in E$ and $(v_i, v_{i+1}) \in E$ for $0 \leq i \leq k-2$. Next, the neighborhood of a vertex v in a graph G, denoted by $N(v)$, is the set of vertices adjacent to v in G. In FQ_n, two vertices u and v are called the j-neighbors to each other, and are denoted as $N_j(u) = v$ or $N_j(v) = u$ if they differ in b_j and $j \in \{1, 2, \ldots, n\}$. In addition, $N_c(u) = v$ or $N_c(v) = u$ means that u and v are adjacent by a complementary edge. Let e_j be an edge $(u, N_j(u))$ in G while $j \in \{c, 1, 2, 3, \ldots, n\}$. Furthermore, we more clearly describe a path or cycle in FQ_n by $v_0 \overset{e_a}{-} v_1 \overset{e_b}{-} \ldots v_k$, where $a, b \in \{c, 1, 2, 3, \ldots, n\}$. Take Figure 1 as an example.

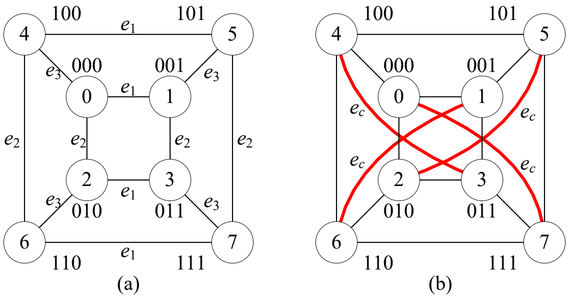


Figure 1. (a) The 3-dimensional hypercube Q_3 and (b) the 3-dimensional folded hypercube FQ_3, where the thick red lines indicate complementary edges and vertex labels are expressed in both binary and decimal.

3. Main results

We will show how to construct two edge-disjoint Hamiltonian cycles in FQ_n while $n \geqslant 3$, and start from FQ_3. Since FQ_3 has only 8 vertices and 16 edges, these edges can be exactly partitioned into 2 edge-disjoint Hamiltonian cycles. Then, we get the following lemma.

Lemma 1. *There exist two edge-disjoint Hamiltonian cycles on FQ_3.*

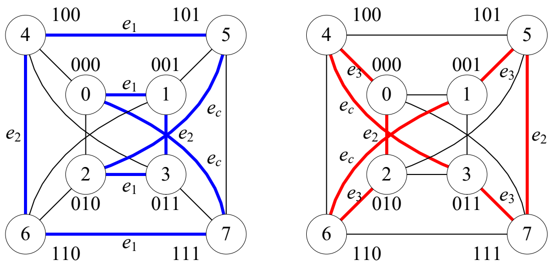
Proof. Let $C_8^A = 0 \overset{e_1}{-} 1 \overset{e_2}{-} 3 \overset{e_1}{-} 2 \overset{e_c}{-} 5 \overset{e_1}{-} 4 \overset{e_2}{-} 6 \overset{e_1}{-} 7 \overset{e_c}{-} 0$

and $C_8^B = 4 \overset{e_3}{-} 0 \overset{e_2}{-} 2 \overset{e_3}{-} 6 \overset{e_c}{-} 1 \overset{e_3}{-} 5 \overset{e_2}{-} 7 \overset{e_3}{-} 3 \overset{e_c}{-} 4$. Both C_8^A and C_8^B are cycles that visit each vertex exactly once, and their edge-disjoint properties can be visually observed in Figure 2. Therefore, C_8^A and C_8^B form two edge-disjoint Hamiltonian cycles in FQ_3. □

When visiting vertices, C_8^A and C_8^B adopt specific edge sequences $Eset_1$ = {1, 2, 1, -1, 1, 2, 1, -1} and $Eset_2$ = {3, 2, 3, -1, 3, 2, 3, -1}, respectively. For each x in $Eset_1$ and $Eset_2$, $x = -1$ indicates a complementary edge e_c and $x \geq 1$ indicates a hypercube edge e_x. Therefore, we can use the following algorithm to generate C_8^A and C_8^B.

Algorithm HConFQ₃

```
Input: start, Eset   // start: the start vertex, Eset: the edge set
Output: HC   // vertices sequence of Hamiltonian cycle on FQ3
u = start; HC = [u]
for x in Eset do
    if x = -1 then
        u = u ^ (2^3 – 1)   // go to the next vertex through the complementary edge, ^ : xor
    else
        u = u ^ (2^(x–1))   // go to the next vertex through the hypercube edge e_x
    append u to HC
return HC
```



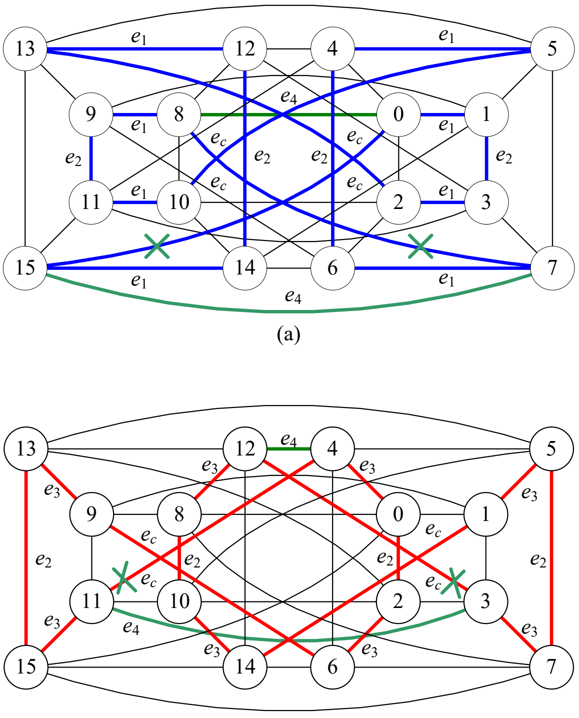
(a) (b)

Figure 2. (a) Thick blue edges and (b) thick red edges represent C_8^A and C_8^B in FQ_3, respectively.

From Definitions 1 and 2, we all know that FQ_4 consists of two Q_3s, 8 hypercube edges and 8 complementary edges. Therefore, inputting the algorithm HConFQ3 with two parameters (0 and $Eset_1$) and another two parameters (8 and $Eset_1$), two cycles C_8s can be obtained, as shown in Figure 3(a). In these two C_8s, we delete edges (0, 15), (7, 8) and add two e_4 edges (0, 8), (7, 15) to form a cycle C_{16}^A. Similarly, through the algorithm HConFQ3, we can input two parameters (4 and $Eset_2$) and another parameter of two parameters (12 and $Eset_2$) to obtain another two cycles C_8s, as shown in Figure 3(b). Delete the edges (3, 12), (4, 11) and add two e_4 edges (3, 11), (4, 12) to form cycle C_{16}^B. Then, we have the following lemma.

Lemma 2. *There exist two edge-disjoint Hamiltonian cycles on FQ_4.*

Proof. Let $C_{16}^A = 0 \overset{e_1}{-} 1 \overset{e_2}{-} 3 \overset{e_1}{-} 2 \overset{e_c}{-} 13 \overset{e_1}{-} 12 \overset{e_2}{-} 14 \overset{e_1}{-} 15 \overset{e_4}{-} 7 \overset{e_1}{-} 6 \overset{e_2}{-} 4 \overset{e_1}{-} 5 \overset{e_c}{-} 10 \overset{e_1}{-} 11 \overset{e_2}{-} 9 \overset{e_1}{-} 8 \overset{e_4}{-} 0$ and $C_{16}^B = 4 \overset{e_3}{-} 0 \overset{e_2}{-} 2 \overset{e_3}{-} 6 \overset{e_c}{-} 9 \overset{e_3}{-} 13 \overset{e_2}{-} 15 \overset{e_3}{-} 11 \overset{e_4}{-} 3 \overset{e_3}{-} 7 \overset{e_2}{-} 5 \overset{e_3}{-} 1 \overset{e_c}{-} 14 \overset{e_3}{-} 10 \overset{e_2}{-} 8 \overset{e_3}{-} 12 \overset{e_4}{-} 4$. Both C_{16}^A and C_{16}^B are cycles that visit each vertex exactly once. Since C_{16}^A is formed by deleting the edges (0, 15) and (7, 8) and adding two e_4 edges (0, 8) and (7, 15) in two cycles C_8s are constructed by Algorithm HConFQ₃. Similarly, C_{16}^B is formed by deleting the edges (3, 12) and (4, 11) and adding two e_4 edges (3, 11) and (4, 12) in two C_8s constructed by Algorithm HConFQ₃. As shown in the proof of Lemma 1, these C_8s are edge-disjoint, and the 4 newly added edges (0, 8), (7, 15), (3, 11), and (4, 12) are all different. Therefore, C_{16}^A and C_{16}^B form two edge-disjoint Hamiltonian cycles in FQ_4. □



(a)

(b)

Figure 3. (a) C_{16}^A drawn by blue and green lines in FQ_4; (b) C_{16}^B drawn by red and green lines in FQ_4

Next, C_8^A and C_{16}^A are constructed based on $Eset_1$.

Similarly, C_8^B and C_{16}^B are formed according to $Eset_2$. Therefore, we can modify Algorithm HConFQ$_3$ as follows. It can build a Hamiltonian cycle based on $Eset_1$ or $Eset_2$ for any integer $n \geq 4$ in FQ_n.

Algorithm HConFQ$_n$

Input: *start, Eset, n // start: the start vertex, Eset: the edge set, n: dimensions of* FQ_n

Output: *HC // vertices sequence of Hamiltonian cycle on* FQ_3

```
1.  u = start; HC = [u]; p = 2^n   // p: numbers of vertices
2.  for i = 0 to p – 1 do
3.      if i % 8 < 7 then   // %: modular arithmetic
4.          j = Eset[i % 8]
5.          if j = -1 then
6.              u = u ^ (2^n – 1)   // go to the next vertex through the complementary edge, ^ : xor
7.          else
8.              u = u ^ (2^(j–1))   // go to the next vertex through the hypercube edge e_j
9.      else
10.         t = int(i / 8)   // int( ): keep integer
11.         for j = 4 to n do
12.             if t % 2 = 0 then break
13.             t = int(t / 2)
14.         u = u ^ (2^(j–1))   // go to the next vertex through the hypercube edge e_j
15.     append u to HC
16. return HC
```

Steps 2 to 15 of the algorithm HConFQn form a for loop, repeated p (the number of vertices in FQ_n) times, each time the algorithm determines the next vertex of the Hamiltonian cycle. Since FQ_n can be decomposed into 2^{n-3} Q_3 subgraphs, steps 4 to 8 are similar to the algorithm HConFQ3, describing the local paths in each Q_3. Each time the algorithm completes a local path in a Q_3, it must decide which Q_3 to connect to. The decision is made by steps 10 to 13, which is based on the binary carry, choosing from the edge e_4 to e_n. Finally, according to this algorithm, inputting 3 parameters (0, $Eset_1$ and n) and another 3 parameters (4, $Eset_2$ and n) we get two Hamiltonian cycles. Then we have the following lemma.

Lemma 3. *There exist two edge-disjoint Hamiltonian cycles on* FQ_n for any integer $n \geq 4$.

Proof. Let $p = |V(FQ_n)|$, C_p^A is obtained by inputting the algorithm HConFQn with three parameters vertex (0, $Eset_1$ and n) and C_p^B is obtained by inputting the algorithm HConFQn with three parameters vertex (4, $Eset_2$ and n). Since C_p^A and C_p^B are Hamiltonian cycles, we only need to prove that they are edge-disjoint. First, according to Lemma 1, the local paths of C_p^A and the local paths of C_p^B are edge-disjoint in each Q_3. Therefore, the only question is whether the edges connecting the local paths are edge-disjoint. The end vertices of local paths of C_p^A (respectively, C_p^B) are $8i$ and $8i + 7$ (respectively, $8i + 4$ and $8i + 3$) while $i \in \{0, 1, 2, \ldots, 2^{n-3} - 1\}$. Since the endpoints of the edges are all different, the connected edges are edge-disjoint. Finally, C_p^A and C_p^B are two edge-disjoint Hamiltonian cycles on FQ_n. □

For example, in FQ_5, inputting the algorithm HConFQn with three parameters (0, $Eset_1$ and 5), we will get C_{32}^A = (0 - 1 - 3 - 2 - 29 - 28 - 30 - 31) - (23 - 22 - 20 - 21 - 10 - 11 - 9 - 8) - (24 - 25 - 27 - 26 - 5 - 4 - 6 - 7) - (15 - 14 - 12 - 13 - 18 - 19 - 17 - 16) - 0. We note that the path inside the parentheses is the local path. Similarly, inputting the algorithm HConFQn with three parameters (4, $Eset_2$ and 5), we will get C_{32}^B = (4 - 0 - 2 - 6 - 25 - 29 - 31 - 27) - (19 - 23 - 21 - 17 - 14 - 10 - 8 - 12) - (28 - 24 - 26 - 30 - 1 - 5 - 7 - 3) - (11 - 15 - 13 - 9 - 22 - 18 - 16 - 20) - 4. Both C_{32}^A and C_{32}^B visited all vertices of FQ_5. The local paths and the edges connecting the local paths are edge-disjoint. Therefore, as shown in Figure 4, C_{32}^A and C_{32}^B are two edge-disjoint Hamiltonian cycles on FQ_5. Finally, according to Lemmas 3, 4, we have the following theorem.

Theorem 4. *There exist two edge-disjoint Hamiltonian cycles on* FQ_n for any integer $n \geq 3$.

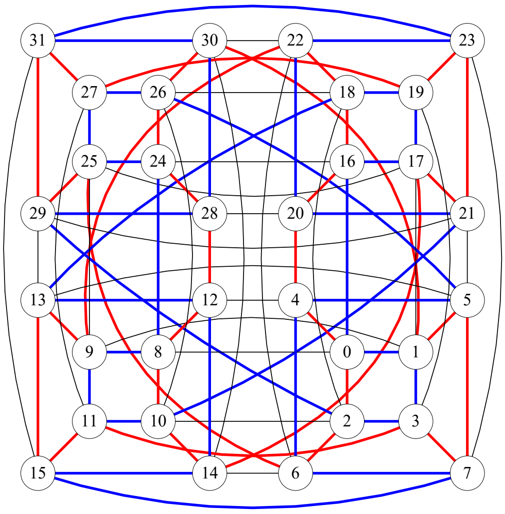


Figure 4. (a) C_{32}^{A} drawn by blue lines in FQ_5; (b) C_{32}^{B} drawn by red lines in FQ_5

4. Performance evaluation

This section aims to simulate data broadcasting using two edge-disjoint Hamiltonian cycles as transmission channels in folded hypercubes. Through performance analysis, we compare our results with the traditional results using only one Hamiltonian cycle. Particularly, we also evaluate the efficiency through two metrics related to the broadcast transmission time (or broadcast latency) in our experiments. The network processed by the simulation is FQn, where the integers $3 \leqslant n \leqslant 10$. Technically, the algorithms for constructing edge-disjoint Hamiltonian cycles and simulating data broadcasting are implemented using C programs. We conducted experiments using a 13th-generation Intel(R) Core(TM) i9-13900F CPU and 16 GB RAM under the Linux operating system.

We simulate the scenario where there is a message with a size not exceeding 6 MB, coming from a random source node in FQ_n, broadcast via one or two edge-disjoint Hamiltonian cycles. Taking the most common Ethernet frame as an example, its frame length can carry about 1500 bytes of data (excluding the initial preamble, frame delimiter, and the frame check sequence at the end). Therefore, the message can be divided into a maximum of 6MB / 1500 bytes, which is approximately 4196 data packets. To understand the difference in broadcast efficiency between one Hamiltonian cycle and two edge-disjoint Hamiltonian cycles, we performed 1,000,000 simulation instances for each case. To be fair, the same message scripts were used in both cases.

The source node x of each broadcast is randomly generated over the entire network FQ_n, while $3 \leqslant n \leqslant 10$. Then, the broadcast is transmitted to all nodes in the entire network in one direction along the Hamiltonian cycle. Since we assume that the message size does not exceed 6 MB, let m be the number of broadcast packets, chosen randomly in the range $1 \leqslant m \leqslant 4196$. For the case of one Hamiltonian cycle, the source node x has two adjacent nodes in the Hamiltonian cycle, and the broadcasting randomly chooses a direction to propagate. In each time slot, a data packet is transmitted to the next node sequentially along the Hamiltonian cycle until it finally returns to the node x, indicating that this round of transmission is successful. For the case of two edge-disjoint Hamiltonian cycles, we take two cycles as transmission channels for data broadcasting. To balance the load of all transmission channels, we use a round-robin strategy to invoke these channels in sequence for packet transmission. That is, the first channel bears odd-numbered data packets, and the second channel carries even-numbered data packets.

For each vertex $v \in V(FQ_n) - \{x\}$ and every i with $1 \leqslant i \leqslant m$, let $t_v(i)$ be the transmission time of the i-th packet received by vertex v, which is measured from the beginning. Then the time required for vertex v to receive the entire message can be expressed as $t(v) = \mathrm{Max}_{i=1}^{m}\{t_v(i)\}$. Since we simulated 1,000,000 broadcast message instances, let msg_j contain m packets and $1 \leqslant j \leqslant 1{,}000{,}000$. Next, two specific metrics called the average transmission time $avg(msg_j) = \sum_{v \in V(FQ_n)-\{x\}} \mathrm{Max}_{i=1}^{m} t_v(i) \,/\, (V(FQ_n) - 1)$ and the maximum transmission time $mx(msg_j) = \mathrm{Max}_{v \in V(FQ_n)}\{\mathrm{Max}_{i=1}^{m}\{t_v(i)\}\}$ are defined.

Next, for a certain dimension of folded hypercubes FQ_n and $3 \leqslant n \leqslant 10$, we use the following two measures to evaluate broadcasting efficiency, one is called the average broadcasting latency $ABL = \sum_{j=1}^{1000000} avg(msg_j) \,/\, 1000000$, and the other is the maximum broadcasting latency $MBL = \mathrm{Max}_{j=1}^{1000000}\{mx(msg_j)\}$.

All experimental results showing ABL and MBL are listed in Tables 1 and 2 for both cases of one Hamiltonian cycle and two edge-disjoint Hamiltonian cycles.

Table 1. *ABL* and *MBL* using a Hamiltonian cycle in folded hypercubes FQ_n while $3 \leqslant n \leqslant 10$

	FQ_3	FQ_4	FQ_5	FQ_6
ABL	2100.92	2104.9	2113.42	2129.28
MBL	4201	4209	4225	4257
	FQ_7	FQ_8	FQ_9	FQ_{10}
ABL	2160.29	2224.49	2352.76	2608.94
MBL	4321	4449	4705	5217

We combine the data from Tables 1 and 2 to plot Figure 5 to make it easier to compare ABL and MBL in two conditions: using one Hamiltonian cycle and using two edge-disjoint Hamiltonian cycles as the broadcasting channels.

Table 2. *ABL* and *MBL* using two edge-disjoint Hamiltonian cycles in FQ_n while $3 \leqslant n \leqslant 10$

	FQ_3	FQ_4	FQ_5	FQ_6
ABL	1057.93	1079.45	1068.85	1089.52
MBL	2103	2111	2127	2159
	FQ_7	FQ_8	FQ_9	FQ_{10}
ABL	1131.24	1215.79	1385.22	1723.19
MBL	2223	2351	2607	3119

From Tables 1, 2, and Figure 5, we are aware of the following three phenomena:

- For both measures, *ABL* and *MBL*, the broadcast delay performance of using two edge-disjoint Hamiltonian cycles is better than the traditional method of using only one. For example, the ratio of the *ABL* of two edge-disjoint Hamiltonian cycles to the *ABL* of one Hamiltonian cycle is close to 50.36% in FQ_3 and close to 66.05% in FQ_{10}. Then, the ratio of the *MBL* of two edge-disjoint Hamiltonian cycles to the *MBL* of one Hamiltonian cycle in FQ_3 reached 50.05%, and was close to 59.79% in FQ_{10}.

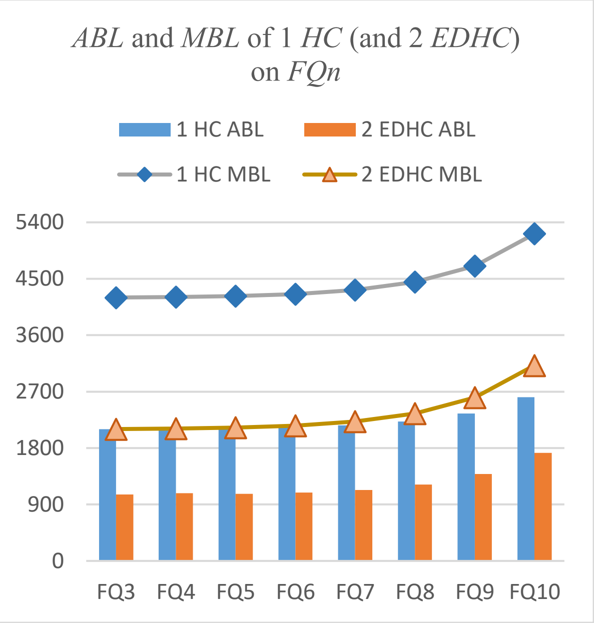


Figure 5. The comparisons of *ABL* and *MBL* using one Hamiltonian cycle and two edge-disjoint Hamiltonian cycles as the broadcasting channels on FQ_n, respectively.

- As the network scale increases, i. e. n increases, *ABL* and *MBL* will also increase. In other words, both *ABL* and *MBL* should be positively correlated with the network scale.
- In the range of network dimensions in our experiments, the value of *MBL* is approximately twice that of *ABL*. For example, for FQ_n with n ranging from 3 to 10, when broadcasting using a single Hamiltonian cycle, the ratio remains constant at 2.00. When broadcasting using two edge-disjoint Hamiltonian cycles, the ratios are 1.99, 1.96, 1.99, 1.98, 1.97, 1.93, 1.88, and 1.81, respectively. Obviously, this ratio deviates further and further from 2.

5. Conclusion

In this paper, we study the construction of two edge-disjoint Hamiltonian cycles as broadband channels in the folded hypercubes for data broadcasting. The main contributions of this research are as follows:

1. Using the special edge sets $Eset_1$ and $Eset_2$, we provide the construction of two edge-disjoint Hamiltonian cycles in FQ_3.

2. Through inductive reasoning, we give the algorithm HConFQn to construct two edge-disjoint Hamiltonian cycles in FQ_n with n ⩾ 4.
3. Based on (1) and (2), we simulate data broadcasting via the developed two edge-disjoint Hamiltonian cycles (respectively, the traditional approach of using only one Hamiltonian cycle) in *FQn* as the transmission channels, where $n \in \{3, 4, 5, 6, 7, 8, 9, 10\}$.
4. The comparison results show that using two edge-disjoint Hamiltonian cycles for broadcasting has better performance than using one Hamiltonian cycle for broadcasting, including two measures related to broadcast delay: *ABL* and *MBL*.

Our future research will focus on whether there exist three (respectively, four) edge-disjoint Hamiltonian cycles on FQ_n when $n \geqslant 5$ (respectively, 7). To date, this remains an open problem.

References

1. Albader, B., Bose, B. (2016), “Edge Disjoint Hamiltonian Cycles in Gaussian Networks,” IEEE Trans. Comput., Vol. 65, pp. 315–321.
2. Bae, M.M., Bose, B. (2003), “Edge disjoint Hamiltonian cycles in k-ary n-cubes and hypercubes,” IEEE Trans. Comput., Vol. 52, pp.1271–1284.
3. Barden, B., Libeskind-Hadas, R., Davis, J., Williams, W. (1999), “On edge-disjoint spanning trees in hypercubes,” Inform. Process. Lett., Vol.70, pp.13–16.
4. Barth, D., Raspaud, A. (1994), “Two edge-disjoint Hamiltonian cycles in the butterfly graph,” Inform. Process. Lett., Vol. 51, pp.175–179.
5. El-Amawy A., Latifi S. (1991), “Properties and performance of folded hypercubes,” IEEE Trans. Parallel Distrib. Syst., Vol. 2, pp.31–42.
6. Lee, S., Shin, K. (1994), “Interleaved all-to-all reliable broadcast on meshes and hypercubes,” IEEE Trans. Parallel Distrib. Syst., Vol. 5, pp.449–458.
7. Leighton F.T. (1992), “Introduction to Parallel Algorithms and Architecture: Arrays, Trees, Hypercubes,” Morgan Kaufman, San Mateo, CA, USA, 1992.
8. Lin, T.J., Hsieh, S.Y., Juan, J.S.-T. (2012), “Embedding cycles and paths in product networks and their applications to multiprocessor systems,” IEEE Trans. Parallel Distrib. Syst., Vol. 23, pp.1081–1089
9. Hung, R.W. (2011), “Embedding two edge-disjoint Hamiltonian cycles into locally twisted cubes,” Theoret. Comput. Sci., Vol. 412, pp.4747–4753.
10. Hung, R.W. (2012), “Constructing two edge-disjoint Hamiltonian cycles and two-equal path cover in augmented cubes,” J. Comput. Sci., Vol. 39, 42–49.
11. Hung, R.W. (2015), “The property of edge-disjoint Hamiltonian cycles in transposition networks and hypercube-like networks,” Discrete Appl. Math., Vol. 181, pp.109–122.
12. Hung, R.W., Chan, S.J., Liao, C.C. (2012), “Embedding two edge-disjoint Hamiltonian cycles and two equal node-disjoint cycles into twisted cubes,” Lecture Notes Eng. Comput. Sci., Vol. 2195, pp.362–367.
13. Hussain, Z.A., Bose, B., Al-Dhelaan, A. (2015), “Edge disjoint Hamiltonian cycles in Eisenstein-Jacobi networks,” J. Parallel Distributed Comput. Vol. 86, pp.62–70.
14. Saad Y., Schultz, M. H. (1988), “Topological properties of hypercubes,” IEEE Trans. Comput., Vol. 37, pp.867–872.
15. Wang, D. (2001), “A low-cost fault-tolerant structure for the hypercube,” J. Supercomput., Vol. 20, pp.203–216.
16. Yang, D.W., Xu, Z., Feng, Y.Q., Lee, J. (2023), “Symmetric property and edge-disjoint Hamiltonian cycles of the spined cube,” Appl. Math. Comput., Vol. 452, pp. 128075.