Edge Congestion and Topological Properties of Crossed Cubes

Chien-Ping Chang, Ting-Yi Sung, *Member, IEEE*, and Lih-Hsing Hsu

Abstract—An n-dimensional crossed cube, CQ_n, is a variation of hypercubes. In this paper, we give a new shortest path routing algorithm based on a new distance measure defined herein. In comparison with Efe's algorithm, which generates one shortest path in $O(n^2)$ time, our algorithm can generate more shortest paths in $O(n)$ time. Based on a given shortest path routing algorithm, we consider a new performance measure of interconnection networks called edge congestion. Using our shortest path routing algorithm and assuming that message exchange between all pairs of vertices is equally probable, we show that the edge congestion of crossed cubes is the same as that of hypercubes. Using the result of edge congestion, we can show that the bisection width of crossed cubes is 2^{n-1}. We also prove that wide diameter and fault diameter are $\lceil \frac{n}{2} \rceil + 2$. Furthermore, we study embedding of cycles in cross cubes and construct more types than previous work of cycles of length at least four.

Index Terms—Crossed cubes, hypercubes, shortest path routing, wide diameter, fault diameter, edge congestion, bisection width, embedding.

1 Introduction

Network topology is a crucial factor for interconnection networks since it determines the performance of a network. Many interconnection network topologies have been proposed in the literature for connecting hundreds or thousands of processing elements. Network topology is always represented by a graph in which vertices represent processors and edges represent links between processors. Among these topologies, the binary n-cube (abbreviated as hypercube), denoted by Q_n, is one of the popular topologies. However, a hypercube does not make the best use of its hardware, since it is possible to fashion networks with lower diameters than that of Q_n. One such topology is the *crossed cube*, which was first proposed by Efe [1]. An n-dimensional crossed cube, denoted by CQ_n, is derived from Q_n by changing the connection of some hypercube links. It has a diameter of $\lceil \frac{n+1}{2} \rceil$, an improvement of approximately a factor of 2, in a trade-off of reducing high degree of symmetry in Q_n.

Though embedding and some topological properties of crossed cubes have been studied in the literature [1], [2], [3], [6], [7], [9], we study some different topological properties and provide different schemes from previous work. To be specific, we consider a new performance measure called *edge congestion*. In addition, we consider the following performance measures of crossed cubes: shortest path routing complexity, diameter, wide diameter, fault diameter, bisection width, and embedding of cycles.

Efe presented a shortest path routing algorithm of crossed cubes in [1], which generated one shortest path for any pair of vertices in $O(n^2)$ time. In this paper, we define a new distance measure which enables us to find more shortest paths for any pair of vertices in $O(n)$ time.

When transmitting or broadcasting messages, heavily congested edges will delay communication time. A network having a relatively balanced communication load of edges under the specified routing algorithm is preferred. Motivated by this observation, we introduce the notion of *edge congestion*, independent of Fiduccia and Hedrick's work [4]. Assuming that message exchange between all pairs of vertices is equally probable, we thus consider all-pair shortest path routing for calculating edge congestion. For each edge, we measure the number of pairs of vertices that will route through this edge given a specific routing algorithm. Edge congestion of a network under a specified routing algorithm is the maximum of the congestion of all edges. We define edge congestion of a network by taking the minimum over all routing algorithms. Smaller edge congestion is preferred. In this paper, we first specify our routing strategy based on our shortest path routing algorithm and show that the edge congestion of the crossed cube CQ_n is equal to that of the hypercube Q_n. Using the result of edge congestion, we can calculate the bisection width of crossed cubes.

Disjoint paths between a pair of vertices contribute to multipath communication between these two vertices and provide alternative routes in the case of vertex or link failures. The notion of connectivity, wide diameter, and fault diameter is defined based on multiple disjoint paths. In [6], the author showed the existence of n disjoint paths for any pair of vertices in a crossed cube CQ_n to prove its connectivity, without obtaining the length of these paths. In

- *C.-P. Chang is with the Chung Shan Institue of Science and Technology, P.O. Box 90008-15-14 Lung Tan, Taoyuan, Taiwan 325, Republic of China. E-mail: cpchang@ccit.edu.tw.*
- *T.-Y. Sung is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan 115, Republic of China. E-mail: tsung@iis.sinicia.edu.tw.*
- *L.-H. Hsu is with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 300, Republic of China. E-mail: lhhsu@cc.nctu.edu.tw.*

Manuscript received 11 Dec. 1997; accepted 26 Feb. 1999.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 106027.

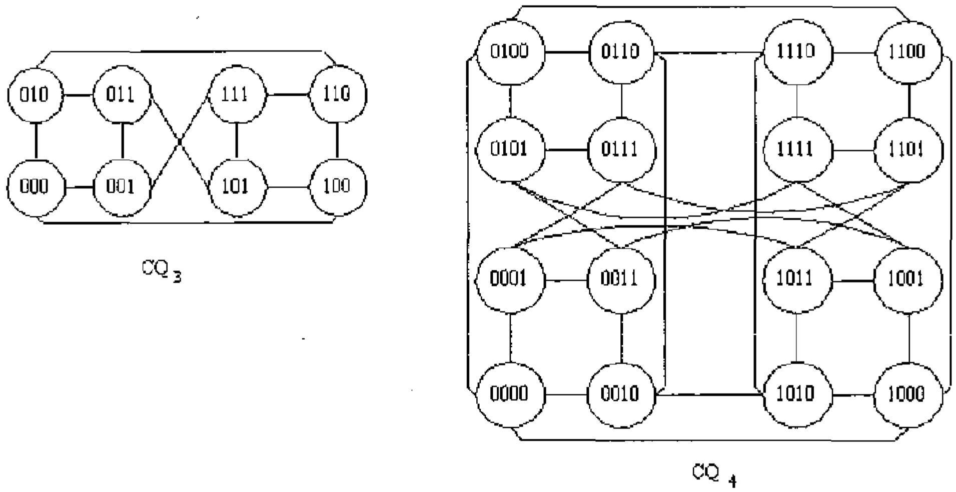


Fig. 1. CQ_3 and CQ_4.

other words, one cannot get wide diameter from this construction. Wide diameter and fault diameter of a crossed cube CQ_n are studied in this paper.

The problem of simulating one network by another can be modeled as a graph embedding problem. Embeddings of complete binary trees and cycles into crossed cubes were presented in [3], [7], [9]. In comparison with [3], we give a concrete construction of cycles of arbitrary length. In [9], the authors constructed one type of cycles for an arbitrary length, whereas we construct various types of cycles in this paper.

The rest of this paper is organized as follows: Section 2 summarizes some known results on crossed cubes and introduces notation used in this paper. In Section 3, we define a distance measure. Based on this measure, we give a new shortest path routing algorithm which runs in $O(n)$ time. Wide diameter and fault diameter are studied in Section 4. In Section 5, we define the notion of edge congestion and compare edge congestion of hypercubes and crossed cubes. In addition, we calculate bisection width of crossed cubes. Embedding of cycles into crossed cubes by constructing various types of cycles is presented in Section 6. Finally, we make concluding remarks in Section 7.

2 Preliminaries and Notation

Let G be a graph. We use $V(G)$ and $E(G)$ to denote the vertex set and the edge set of G, respectively. Let x and y be two vertices. We use $d_G(x, y)$ to denote the distance between x and y in G. To define crossed cubes, we first introduce the notation of *"pair related."* Let

$$R = \{(00, 00), (10, 10), (01, 11), (11, 01)\}.$$

Two binary strings $x = x_1x_0$ and $y = y_1y_0$ are *pair related* if and only if $(x, y) \in R$.

Definition 1. *An n-dimensional crossed cube CQ_n is recursively constructed as follows: CQ_1 is a complete graph with two vertices labeled by 0 and 1, respectively. CQ_n consists of two identical $(n-1)$-dimensional crossed cubes CQ^0_{n-1} and CQ^1_{n-1}. The vertex $u = 0u_{n-2}\cdots u_0 \in V(CQ^0_{n-1})$ and the vertex $v = 1v_{n-2}\cdots v_0 \in V(CQ^1_{n-1})$ is an edge in CQ_n if and only if*

1. *$u_{n-2} = v_{n-2}$ if n is even, and*
2. *$(u_{2i+1}u_{2i}, v_{2i+1}v_{2i}) \in R$ for all $0 \le i < \lfloor \frac{n-1}{2} \rfloor$.*

Examples of CQ_3 and CQ_4 are illustrated in Fig. 1.

Throughout the paper, each vertex in CQ_n is represented by an n-dimensional binary vector, e.g., $u = u_{n-1}u_{n-2}\cdots u_0$ and $v = v_{n-1}v_{n-2}\cdots v_0$. For $k < n$, the *k-prefix* of u, $p_k(u)$, is defined as $u_{n-1}u_{n-2}\cdots u_{n-k}$. We can thus write

$$u = p_k(u)u_{n-k-1}\cdots u_0.$$

Let x be an l-bit string with $l \le n$. We use $CQ_n(x)$ to denote the subgraph of CQ_n induced by the set of vertices with the prefix x. It is shown in [7] that $CQ_n(x)$ is isomorphic to $CQ_{n-|x|}$.

Let x and y be two distinct l-bit strings with $l < n$. If $CQ_n(x)$ and $CQ_n(y)$ can be joined by an edge in CQ_n, then $CQ_n(x)$ and $CQ_n(y)$ are called *adjacent subgraphs* of CQ_n. It can be easily verified that if $CQ_n(x)$ and $CQ_n(y)$ are adjacent subgraphs of CQ_n, (x, y) is an edge in CQ_l. However, $(x, y) \in E(CQ_l)$ does not necessarily imply that $CQ_n(x)$ and $CQ_n(y)$ are adjacent subgraphs of CQ_n. For example, when $n = 5$, $x = 01$, and $y = 11$ (i.e., $n - l = 3$), 01 and 11 are adjacent in CQ_2, but $CQ_5(01)$ and $CQ_5(11)$ are not adjacent subgraphs of CQ_5. Note that if (x, y) is an edge in CQ_l and $n - l$ is even, then $CQ_n(x)$ and $CQ_n(y)$ are adjacent subgraphs of CQ_n.

Let $CQ_n(x, y)$ denote the subgraph of CQ_n induced by $CQ_n(x) \cup CQ_n(y)$. It is proven in [1] that $CQ_n(x, y)$ is isomorphic to $CQ_{n-|x|+1}$ if $CQ_n(x)$ and $CQ_n(y)$ are adjacent subgraphs of CQ_n. In particular, let $|x| = |y| = 2$ and (x, y) be an edge in CQ_2. Obviously, $CQ_{2k}(x)$ and $CQ_{2k}(y)$ are isomorphic to CQ_{2k-2}. It follows from the above discussion that $CQ_{2k}(x)$ and $CQ_{2k}(y)$ are adjacent subgraphs and that $CQ_{2k}(x, y)$ is isomorphic to CQ_{2k-1}. Thus, we can contract those vertices in CQ_{2k} having the same prefix of length two into a vertex and obtain a graph with four vertices. It is easy to see that this four-vertex graph is isomorphic to CQ_2, as shown in Fig. 2a. For any two vertices u, v in CQ_n with

n even and $n \geq 4$, the following statements can be easily observed [1]:

1. If $p_2(u) = p_2(v)$, then u and v are in a subgraph isomorphic to CQ_{n-2}.
2. If $d_{CQ_2}(p_2(u), p_2(v)) = 1$, then u and v are in a subgraph isomorphic to CQ_{n-1}.
3. If $d_{CQ_2}(p_2(u), p_2(v)) = 2$, then a neighbor u' of u and v are in a subgraph isomorphic to CQ_{n-1}.

Similarly, we can contract those vertices in CQ_{2k+1} with the same prefix of length three into a vertex and obtain a graph with eight vertices. Again, this eight-vertex graph is isomorphic to CQ_3, as illustrated in Fig. 2b. We can also obtain the following observations for any two vertices u, v in CQ_n with n odd and $n \geq 3$ [1]:

1. If $p_3(u) = p_3(v)$, then u and v are in a subgraph isomorphic to CQ_{n-3}.
2. If $d_{CQ_3}(p_3(u), p_3(v)) = 1$, then u and v are in a subgraph isomorphic to CQ_{n-2}.
3. If $d_{CQ_3}(p_3(u), p_3(v)) = 2$, then a neighbor u' of u and v are in a subgraph isomorphic to CQ_{n-2}.

Definition 2. *Let (u, v) be an edge of CQ_n. When vertices u and v have a leftmost differing bit at position d, we say that v is the d-neighbor of u and that the edge (u, v) is an edge of dimension d or $dim(u, v) = d$. We call (u, v) a dim-d edge.*

For example, let $u = 10111$. The 4-, 3-, 2-, 1-, and 0-neighbors of u are given by 01101, 11101, 10001, 10101, and 10110, respectively. We use $N(u)$ to denote the set of neighbors of u.

Let k be a positive integer satisfying $k < n - 1$ and $n - k$ even. Let $x = p_k(u)$ and $y = p_k(v)$. Assume that

$$(x, y) \in E(CQ_k).$$

It follows from the definition of CQ_n that $(u, v) \in E(CQ_n)$ if and only if $(u_{2i+1}u_{2i}, v_{2i+1}v_{2i}) \in R$ for $0 \leq i < \frac{n-k}{2}$. Since the set R induces a one-to-one correspondence for all of the 2-bit strings, a vertex v satsifying $(u, v) \in E(CQ_n)$ can be uniquely determined by (x, y) and u. We can thus denote v by $f(u; x, y)$. For example, given $(x, y) = (101, 111) \in E(CQ_3)$ and $u = 10111$, the neighbor of u in CQ_5 with prefix 111 can be uniquely identified and is given by $v = 11101$.

Let $Q = \langle x = z_0, z_1, \cdots, z_m = q \rangle$ be a path in CQ_k. The length of Q is denoted by $|Q|$. Two terminal vertices x and q of Q can be denoted by $t(x; Q) = q$ and $t(q; Q) = x$. In this paper, paths can be considered in a directed sense, e.g., the aforementioned path Q is directed from x to q. Given path Q, z_i is called the *immediate predecessor* of z_{i+1}. We use $P(u; Q)$ to denote the path in CQ_n induced from Q, which is given by $P(u; Q) = \langle u = w_0, w_1, \cdots, w_m \rangle$, where

$$w_i = f(w_{i-1}; z_{i-1}, z_i)$$

for all $1 \leq i \leq m$. The path $P(u; Q)$ preserves the length of Q, i.e., $|P(u; Q)| = |Q|$. For example, let

$$Q = \langle 10100, 01100, 01101 \rangle$$

and $v = 1010011$. Then,

$$P(v; Q) = \langle 1010011, 0110001, 0110111 \rangle.$$

Let $P_1 = \langle u, u^1, u^2, \cdots, u^m = v \rangle$ be a (u, v)-path, and let P_2 be a (v, w)-path. We define $P_1 \cup P_2$ as the concatenation of P_1 and P_2 which yields a (u, w)-path, and $P_1 \cup P_2$ can also be written as $\langle u, u^1, u^2, \cdots, u^m, P_2 \rangle$. The path obtained from P_1 by removing the subpath $\langle u^k, u^{k+1}, \cdots, u^m \rangle$ with $k < m$ is denoted by $P_1 - \langle u^k, u^{k+1}, \cdots, u^m \rangle$.

3 Shortest Path Routing

Let u and v be two distinct vertices in CQ_n. The ith *double bit* of vertex u is defined as a 2-bit string $u_{2i+1}u_{2i}$ for $0 \leq i \leq \lfloor \frac{n}{2} \rfloor - 1$, and as simply a single bit u_{2i} for $i = \lfloor \frac{n}{2} \rfloor$ and n odd. Bit l is called *the most significant differing bit* between u and v if $p_{n-l-1}(u) = p_{n-l-1}(v)$ and $u_l \neq v_l$. Let

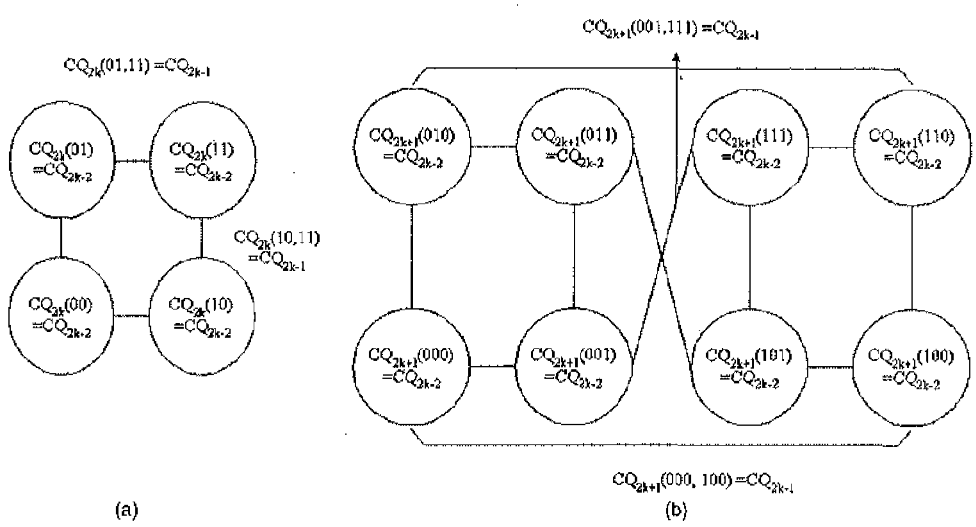


Fig. 2. Subgraphs of CQ_{2k} and CQ_{2k+1}.

$i^* = \lfloor \frac{i}{2} \rfloor$, called the *most significant differing double bit*. We define a function ρ on u, v as follows:

$$\rho_j(u,v) = 0 \text{ for all } j \geq i^* + 1,$$
$$\rho_{i^*}(u,v) = \begin{cases} 2 & \text{if } u_{2i^*+1}u_{2i^*} = \bar{v}_{2i^*+1}\bar{v}_{2i^*}, \\ 1 & \text{otherwise.} \end{cases}$$

Subsequently, we recursively define $\rho_j(u,v)$ for $j \leq i^* - 1$ using the notion of *distance-preserving pair related* (abbreviated as *d.-p. pair related*) which is motivated from the concept of pair related.

Definition 3. *$u_{2j+1}u_{2j}$ and $v_{2j+1}v_{2j}$, for $j \leq i^* - 1$, are distance-preserving pair related if one of the following conditions holds:*

1. $(u_{2j+1}u_{2j}, v_{2j+1}v_{2j}) \in \{(01,01),(11,11)\}$ *and*
$$\sum_{k=j+1}^{\lfloor \frac{n-1}{2} \rfloor} \rho_k(u,v) \text{ is even},$$
2. $(u_{2j+1}u_{2j}, v_{2j+1}v_{2j}) \in \{(01,11),(11,01)\}$ *and*
$$\sum_{k=j+1}^{\lfloor \frac{n-1}{2} \rfloor} \rho_k(u,v) \text{ is odd},$$
3. $(u_{2j+1}u_{2j}, v_{2j+1}v_{2j}) \in \{(00,00),(10,10)\}$.

We write $u_{2j+1}u_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}$ if $u_{2j+1}u_{2j}$ and $v_{2j+1}v_{2j}$ are d.-p. pair related, and $u_{2j+1}u_{2j} \overset{\text{d.-p.}}{\not\sim} v_{2j+1}v_{2j}$ otherwise.

Then $\rho_j(u,v)$ for $j \leq i^* - 1$ is recursively defined as follows:

$$\rho_j(u,v) = \begin{cases} 0 & \text{if } u_{2j+1}u_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}, \\ 1 & \text{otherwise.} \end{cases}$$

The *pair related distance* between u and v is defined, denoted by $\rho(u,v)$, as

$$\rho(u,v) = \sum_{j=0}^{\lfloor \frac{n-1}{2} \rfloor} \rho_j(u,v).$$

For example, let

$$u = 01101101100001$$

and $v = 01010011100101$ be vertices in CQ_{14}. We recursively find ρ_i as follows: $\rho_6(u,v) = 0$, $\rho_5(u,v) = 2$, $\rho_4(u,v) = 1$, $\rho_3(u,v) = 0$, $\rho_2(u,v) = 0$, $\rho_1(u,v) = 1$, $\rho_0(u,v) = 0$, and $\rho(u,v) = 4$.

Now consider the relationship between $d_{CQ_n}(u,v)$ and $\rho(u,v)$. Let P be a shortest path from u to v. If $u_{2i^*+1}u_{2i^*} = \bar{v}_{2i^*+1}\bar{v}_{2i^*}$, it requires two steps in P. Otherwise, by the definition of i^* it requires a single edge in P. If $u_{2j+1}u_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}$, after identifying bits at positions $p \geq 2j+2$ with those at v using $\sum_{k=j+1}^{\lfloor \frac{n-1}{2} \rfloor} \rho_k(u,v)$ steps, we can reach a vertex such that bits at positions $2j+1$ and $2j$ are identical to v_{2j+1} and v_{2j}. If $u_{2j+1}u_{2j} \overset{\text{d.-p.}}{\not\sim} v_{2j+1}v_{2j}$, at least one step is required in P to change the $(2j+1)$th and the $2j$th bits to v_{2j+1} and v_{2j}. With the above observation, we have the following remark.

Remark 1. $\rho(u,v)$ is a lower bound for $d_{CQ_n}(u,v)$, i.e., $d_{CQ_n}(u,v) \geq \rho(u,v)$.

We now present a shortest path routing algorithm which can generate multiple shortest paths and is different from the one proposed in [1].

Algorithm: Shortest_Path_Routing

Input: Vertices u and v.

Output: Shortest paths from u to v.

Initialization: Determine $\rho_i(u,v)$ for all $0 \leq i \leq \lfloor \frac{n}{2} \rfloor - 1$.
Define $Q_1 = Q_2 = \{j \mid \rho_j(u,v) \neq 0\}$,
$T = \{j \mid \rho_j(u,v) \neq 0, j < i^*,$ and either
$\bar{u}_{2j+1}u_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}$ or $u_{2j+1}\bar{u}_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}\}$.

Step 1: **If** $T = \emptyset$
then go to Step 2;
else find a $j \in T$ and call ONE_STEP_ROUTE(j, Q_1).

Step 2: **If** $Q_1 = \emptyset$
then output the sequence S which yields shortest paths, set S to be empty, and go to Step 4.

Step 3: **If** either $\bar{u}_{2i+1}u_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ or $u_{2i+1}\bar{u}_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ holds for some $i \in Q_1$
then choose such smallest i;
else choose $i = max\{j \mid j \in Q_1\}$.
Call ONE_STEP_ROUTE(i, Q_1) and go to Step 2.

Step 4: **If** $Q_2 = \emptyset$
then output the sequence S which yields shortest paths and STOP.

Step 5: Choose $l = max\{j \mid j \in Q_2\}$ and $m = max\{j \mid j \in Q_2$ and $j < l\}$.
If $\rho_l(u,v) = 1$ and $u_{2m+1}\bar{u}_{2m} \overset{\text{d.-p.}}{\sim} v_{2m+1}v_{2m}$
then call ONE_STEP_ROUTE(m, Q_2);
else
  if $\rho_l(u,v) = 1$ and $\bar{u}_{2m+1}u_{2m} \overset{\text{d.-p.}}{\sim} v_{2m+1}v_{2m}$
  then call ONE_STEP_ROUTE(m, Q_2) or call ONE_STEP_ROUTE(l, Q_2);
  else call ONE_STEP_ROUTE(l, Q_2).
Go to Step 4.

ONE_STEP_ROUTE(j, Q)
If $\rho_j(u,v) = 2$
then route to u', i.e., the $2j$th or $(2j+1)$th neighbor of u,
  $\rho_j(u,v) = \rho_j(u,v) - 1$;

else route to u', the $(2j+1)$th neighbor of u if $\bar{u}_{2j+1}u_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}$ and the $2j$th neighbor of u if $u_{2j+1}\bar{u}_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}$,
set $Q = Q - \{j\}$,
$\rho_j(u,v) = \rho_j(u,v) - 1$.
Put u' in the sequence S, and set $u = u'$.
end

To illustrate this algorithm, we still use

$$u = 01101101100001$$

and $v = 01010011100101$ in CQ_{14} as an example. At initial stage, we have $\rho_5(u,v) = 2$, $\rho_4(u,v) = 1$, $\rho_1(u,v) = 1$, $Q_1 = Q_2 = \{5,4,1\}$ and $T = \emptyset$. Since $T = \emptyset$ and we cannot find any $i \in Q_1$ satisfying $\bar{u}_{2i+1}u_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ or

$$u_{2i+1}\bar{u}_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i},$$

in Step 3 we choose $i = 5$ reaching 01000111100011 or 01110111100011, and reduce $\rho_5(u,v)$ by one. Now, the set Q_1 is still $\{5,4,1\}$. Repeating Step 3, we find $i = 1$ in this iteration and route to 01000111100101 or 01110111100101. It reduces Q_1 to $\{5,4\}$. Subsequently, $i = 4$ and $i = 5$ are found in the following iterations of Step 3, which render Q_1 to be empty. In summary, per-forming Step 1 to Step 3, we can obtain the following two shortest paths:

$$P_1 = \langle\ 01101101100001, 01000111100011, 01000111100101, 01000001101111, 01010011100101\ \rangle,$$

$$P_2 = \langle\ 01101101100001, 01110111100011, 01110111100101, 01110001101111, 01010011100101\ \rangle.$$

Performing Step 4 to Step 5, we can obtain two shortest paths P_3 and P_4 as follows:

$$P_3 = \langle\ 01101101100001, 01000111100011, 01000001100001, 01010011100011, 01010011100101\ \rangle,$$

$$P_4 = \langle\ 01101101100001, 01110111100011, 01110001100001, 01010011100011, 01010011100101\ \rangle.$$

The paths P_3 and P_4 are obtained by changing the fifth, fourth, fifth and first double bits sequentially.

We summarize in Table 1 the verification of the statement whether $\bar{u}_{2i+1}u_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ or

$$u_{2i+1}\bar{u}_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$$

for $i \neq i^*$ is satisfied in Steps 3 and 5. In Table 1, "even" and "odd" represent $\sum_{k=i+1}^{\lfloor\frac{n-1}{2}\rfloor} \rho_k(u,v)$ being even and odd, respectively. We use "yes" and "no" to indicate that the statement holds and does not hold, respectively, and "—" to indicate that $u_{2i+1}u_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$. In other words, ONE_STEP_ROUTE can be performed on those cases that are marked with "yes". Furthermore, observing from Table 1, we can obtain the following remark.

Remark 2. For those cases marked with "no" in Table 1, reducing a $\rho_{i'}(u,v)$, $i' > i$ by one enables us to perform ONE_STEP-_ROUTE on the ith double bit. Furthermore, let $k = i^*$ or let it be the smallest index in Q_1 satisfying $\bar{u}_{2i+1}u_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ or $u_{2i+1}\bar{u}_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$. After performing ONE_STEP_ROUTE(k, Q_1) and reaching u', it can be observed that $\bar{u}'_{2j+1}u'_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}$ or

$$u'_{2j+1}\bar{u}'_{2j} \overset{\text{d.-p.}}{\sim} v_{2j+1}v_{2j}$$

is satisfied for all $j < k$ and $j \in Q_1$.

Theorem 1.

1. *The Shortest_Path_Routing algorithm correctly finds shortest paths from the source to the destination.*
2. $d_{CQ_n}(u,v) = \rho(u,v)$.

Proof. Let u be the source and v be the destination. Let u^c denote a current vertex reached by the algorithm, and Q_1^c and Q_2^c be the subsets of Q_1 and Q_2, respectively, obtained by the algorithm. We note that ONE-_STEP_ROUTE is performed at the ith double bit if $u^c_{2i+1}\bar{u}^c_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ or $\overline{u^c_{2i+1}}u^c_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$. Therefore, at the end of $\rho(u,v)$ steps of performing ONE_STEP-_ROUTE, the ith double bit becomes identical to that of v. Observing from Steps 1, 3, and 5, we note that

TABLE 1
Applicability of ONE_STEP_ROUTE

		$u_{2i+1}u_{2i}$			
		00	10	11	01
$v_{2i+1}v_{2i}$	00	–	yes	no	yes
	10	yes	–	yes	no
	11	(even, no)	(even, yes)	(even, –)	(even, yes)
		(odd, yes)	(odd, no)	(odd, yes)	(odd, –)
	01	(even, yes)	(even, no)	(even, yes)	(even, –)
		(odd, no)	(odd, yes)	(odd , –)	(odd, yes)

ONE_STEP_ROUTE is performed at those double bits with $\rho_i(u,v) \neq 0$. Furthermore, ONE_STEP_ROUTE is performed $|Q_1| = |Q_2|$ times if $\rho_{i^*}(u,v) = 1$ and $|Q_1|+1$ times if $\rho_{i^*}(u,v) = 2$, i.e., exactly $\rho(u,v)$ times. Therefore,

$$d_{CQ_n}(u,v) \leq \rho(u,v).$$

On the other hand, we have $d_{CQ_n}(u,v) \geq \rho(u,v)$ by Remark 1. Consequently, we have $d_{CQ_n}(u,v) = \rho(u,v)$, and moreover, the algorithm generates shortest paths from u to v.

If initially $T = \emptyset$ and no $i \in Q_1$ satisfies

$$u_{2i+1}\bar{u}_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$$

or $\bar{u}_{2i+1}u_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$, let $i' = max\{j \mid j \in Q_1\}$ as found in Step 3. It follows that we can choose $i' = i^*$ first and then continue the procedure ONE_STEP_ROUTE(i, Q_1) for all $i \in Q_1$ until $Q_1 = \emptyset$. Otherwise, we can always perform ONE_STEP_ROUTE on any $i \in T$ as specified in Step 1 and on the smallest $i \in Q_1$ (actually, arbitrary $i \in Q_1$ is allowed), satisfying $\bar{u}_{2i+1}u_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ or

$$u_{2i+1}\bar{u}_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i},$$

as specified in Step 3. In subsequent iterations of repeating Step 3, it follows from Remark 2 that an $i \in Q_1$ satisfying $\bar{u}^c_{2i+1}u^c_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ or $u^c_{2i+1}\bar{u}^c_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ can always be found, until $Q_1 = \emptyset$. Thus, repeating Steps 1 to 3, we can obtain shortest paths.

Similarly, in Steps 4 and 5, the algorithm checks each double bit i with $\rho_i(u,v) \neq 0$ from left to right order and generates other shortest paths. Hence, the theorem follows. ☐

Step 1 takes $O(1)$ time. In Step 3, if either

$$\bar{u}_{2i+1}u_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$$

or $u_{2i+1}\bar{u}_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ is satisfied, we perform ONE-STEP-_ROUTE. Otherwise, we perform ONE_STEP_ROUTE on the jth double bit with $j \in Q_1$ and $j > i$ reaching u^c, and then we obtain $u^c_{2i+1}\overline{u^c}_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$ or $\overline{u^c_{2i+1}}u^c_{2i} \overset{\text{d.-p.}}{\sim} v_{2i+1}v_{2i}$. Thus, we examine each double bit i with $\rho_i(u,v) \neq 0$ at most twice in increasing order of i. Steps 2 and 3 are executed at most $2\rho(u,v)$ times. Steps 4 and 5 are executed exactly $\rho(u,v)$ times, each taking $O(1)$ time. Hence, the time complexity of the algorithm is $O(n)$. The algorithm proposed in [1] can generate one shortest path in $O(n^2)$ time, while ours can generate more shortest paths in $O(n)$ time. Furthermore, our algorithm can be modified to generate all shortest paths in $O(n^2)$ time.

Theorem 2. *The diameter of CQ_n, denoted by $D(CQ_n)$, is $\lceil \frac{n+1}{2} \rceil$.*

Proof. Let u, v be two distinct vertices. Since there are $\lceil \frac{n}{2} \rceil$ double bits, it follows that $d_{CQ_n}(u,v) = \rho(u,v) \leq \lceil \frac{n+1}{2} \rceil$. In particular, we choose $u^* = 00\cdots0$ and $v^* = 11\cdots1$ such that $\rho(u^*,v^*) = \lceil \frac{n+1}{2} \rceil$. Hence, the theorem follows. ☐

4 Wide Diameter and Fault Diameter

The connectivity of a network $G = (V,E)$, denoted by $\kappa(G)$ or κ, is the minimum number of vertices whose removal leaves the remaining graph disconnected or trivial. It follows from Menger's theorem that there always exist κ *internally vertex-disjoint* (abbreviated as *disjoint*) paths between any two vertices.

Let α and β be two positive integers such that $\alpha \leq \kappa$ and $\beta \leq \kappa - 1$. Given any two distinct vertices u and v of G, let $C(u,v)$ denote the set of all α disjoint paths between u and v. Each element of $C(u,v)$ consists of α disjoint paths. The number of elements in $C(u,v)$ is denoted by $|C(u,v)|$. Let $l_i(u,v)$ denote the longest length among these α paths of the ith element of $C(u,v)$. We define $d_\alpha(u,v)$ and $d^f_\beta(u,v)$ as follows:

$$d_\alpha(u,v) = \min_{1 \leq i \leq |C(u,v)|} l_i(u,v),$$

$$d^f_\beta(u,v) = \min_{\substack{F \subset V \\ |F| = \beta}} \{d_{G-F}(u,v) \mid u,v \notin F\},$$

where $G - F$ denotes the subgraph of G induced by $V - F$. In other words, $d^f_\beta(u,v)$ denotes the shortest distance between u and v when any β-vertex fault occurs.

Definition 4. *The α-wide diameter of G, denoted by $D_\alpha(G)$, is defined as*

$$D_\alpha(G) = \max_{u,v \in V}\{d_\alpha(u,v)\}.$$

In particular, we call $D_\kappa(G)$ simply wide diameter of G.

Note that $D_1(G)$ is simply the diameter of G.

Definition 5. *The β-fault diameter, denoted by $D^f_\beta(G)$, is given by*

$$D^f_\beta(G) = \max_{u,v \in V}\{d^f_\beta(u,v)\}.$$

$D^f_{\kappa-1}(G)$ is, in particular, called fault diameter of G.

Fault diameter estimates the impact on diameter when faults occur, i.e., the removal of vertices from G. Small $(\kappa-1)$-fault diameter is also desirable to obtain smaller communication delay when vertex faults occur. Obviously, we have $D(G) \leq D^f_{\kappa-1}(G) \leq D_\kappa(G)$.

For hypercubes, it is known that $\kappa(Q_n) = n$ and

$$D_n(Q_n) = D^f_{n-1}(Q_n) = n+1.$$

In this section, we compare wide diameter and fault diameter of crossed cubes with those of hypercubes and prove that $D_n(CQ_n) = D^f_{n-1}(CQ_n) = \lceil \frac{n}{2} \rceil + 2$ for all $n \geq 2$.

Lemma 1. $D^f_{n-1}(CQ_n) \geq \lceil \frac{n}{2} \rceil + 2$.

Proof. Let u, v, u' be vertices of CQ_n, and let F be an $(n-1)$-fault set given by $F = N(u) - \{u'\}$. Consider $n = 2k, k \geq 1$. We choose $u = 00\cdots0$, $v = 011\cdots1$, and $u' = 100\cdots0$. The shortest path from u to v in $CQ_n - F$

has the form $\langle u, u', P^*(u', v)\rangle$, where $P^*(u', v)$ is a shortest path from u' to v without traversing any vertex in F. Since $\rho(u', v) = \lceil\frac{n}{2}\rceil + 1$, it follows that

$$d_{CQ_n-F}(u,v) = 1 + \rho(u', v) = \left\lceil\frac{n}{2}\right\rceil + 2.$$

Therefore, $D^f_{n-1}(CQ_n) \geq \lceil\frac{n}{2}\rceil + 2$.

Next, let us consider n odd. Let $u = 000\,00\cdots0$ and $u' = 001\,00\cdots0$. We choose $v = 100\ 1101\,1101\cdots1101\,11$ for $n = 4k+1$, and

$$v = 100\,1101\,1101\cdots1101$$

for $n = 4k+3$. The shortest path from u to v in $CQ_n - F$ has the form $\langle u, u', u'', P^*(u'', v)\rangle$, where u'' is a neighbor of u' and $P^*(u'', v)$ is a shortest path from u'' to v without traversing vertices in F and u. Note that u'' can be arbitrarily chosen from the set

$$W = \{011\,00\cdots0,\ 111\,00\cdots0\} \bigcup \{001\,00\cdots01\overbrace{00\cdots0}^{j} \mid 0 \leq j \leq n-4\}.$$

Since any vertex w in W satisfies $\rho(w, v) = \lceil\frac{n}{2}\rceil$, it follows that $d_{CQ_n-F}(u,v) = 2 + \rho(u'', v) = \lceil\frac{n}{2}\rceil + 2$. Therefore,

$$D^f_{n-1}(CQ_n) \geq \left\lceil\frac{n}{2}\right\rceil + 2.$$

Hence the lemma follows. □

Lemma 2. *Let u and v be two vertices of CQ_n for $n \geq 2$. Then, there are n disjoint paths $P_1, P_2, \cdots, P_n$ joining u to v with nondecreasing length, i.e., $|P_1| \leq |P_2| \leq \cdots \leq |P_n|$, such that*

1. $|P_i| \leq \lceil\frac{n}{2}\rceil + 2$ *for all i,*
2. $|P_1| \leq D(CQ_n) = \lceil\frac{n+1}{2}\rceil$,
3. $|P_2| = \lceil\frac{n+1}{2}\rceil$ *if $|P_1| = \lceil\frac{n+1}{2}\rceil$ and n is even.*

We leave the proof to Appendix A. Note that Lemma 1 and Lemma 2 immediately imply $\kappa(CQ_n) = n$ since each vertex has a degree of n. Since Lemma 2 implies that $d_n(u,v) \leq \lceil\frac{n}{2}\rceil + 2$, we then have the following corollary.

Corollary 1. $D_n(CQ_n) \leq \lceil\frac{n}{2}\rceil + 2$ *for all $n \geq 2$.*

Since $D^f_{n-1}(CQ_n) \leq D_n(CQ_n)$, we can easily obtain the following theorem from Corollary 1 and Lemma 1.

Theorem 3. $D^f_{n-1}(CQ_n) = D_n(CQ_n) = \lceil\frac{n}{2}\rceil + 2$.

5 Edge Congestion and Bisection Width

In this section, we treat a path as a directed routing from a source to a destination. To distinguish different orientation of an edge (u, v), we write $[u, v]$ and $[v, u]$ as traversing from u to v and from v to u, respectively. A path $P = \langle x = x^0, x^1, \ldots, x^m = y\rangle$ is treated as a directed path from x to y consisting of $[x^i, x^{i+1}]$ for $0 \leq i \leq m-1$. For convenience, we also treat a path as a set of edges and write $[x^i, x^{i+1}] \in P$ to mean that $[x^i, x^{i+1}]$ is in P. We say that an edge $e = (u, v)$ is *incident* on P if $[u, v] \in P$ or $[v, u] \in P$, i.e., $u = x^i, v = x^{i+1}$ or $v = x^i, u = x^{i+1}$ for some i. When considering edge congestion, we treat any routing algorithm A for a network $G = (V, E)$ as a function assigning each $(x, y) \in V \times V$ to only one path from x to y, denoted by $P_A(x, y)$. We consider only the shortest path routing algorithms.

Definition 6. *The edge congestion of an edge $e \in E$ under the routing algorithm A, denoted by $c_A(e)$, is defined as the number of (x, y) pairs such that e is incident on $P_A(x, y)$, i.e.,*

$$c_A(e) = |\{(x, y) \mid x, y \in V,\ e \text{ is incident on } P_A(x, y)\}|.$$

The edge congestion of the network G under the routing algorithm A and the edge congestion of the network G, denoted by $c_A(G)$ and $c(G)$, respectively, are defined as follows:

$$c_A(G) = \max\{c_A(e) \mid \textit{for all } e \in E\},$$
$$c(G) = \min\{c_A(G) \mid \text{for all routing algorithms } A \textit{ for } G\}.$$

Edge congestion can also provide lower bound on the area and the longest wire length required by VLSI layout of networks. A routing algorithm achieving $c(G)$ is called an *optimal* routing algorithm.

Definition 7. *The bisection width of G, denoted by $\omega(G)$, is the minimum number of edges to be removed to disconnect the graph into two (not necessarily connected) subgraphs with $\lceil\frac{|V|}{2}\rceil$ and $\lfloor\frac{|V|}{2}\rfloor$ vertices, respectively.*

The problem of finding the bisection width of a graph is NP-hard. The bisection width of an interconnection network is a critical factor in determining the speed with which the network can perform a computation and the area needed to layout this network [8].

In this section, we will prove that $c(CQ_n) = c(Q_n) = 2^n$ and $\omega(CQ_n) = 2^{n-1}$.

Theorem 4. $c(Q_n) = 2^n$.

Proof. For any vertex u in Q_n, it is known that exactly $\binom{n}{i}$ vertices at distance i from u. Thus for any routing algorithm A, we have

$$\sum_{e \in E(Q_n)} c_A(e) = \sum_{(x,y) \subset V \times V} |P_A(x,y)|$$
$$\geq \sum_{x \in V} \sum_{i=0}^{n} i\binom{n}{i} = 2^n \sum_{i=0}^{n} i\binom{n}{i} = 2^n n 2^{n-1}.$$

Since there are exactly $n\,2^{n-1}$ edges in $E(Q_n)$,

$$c_A(Q_n) = max\{c_A(e) \mid e \in E(Q_n)\} \geq 2^n.$$

Thus, $c(Q_n) \geq 2^n$.

On the other hand, let C be the routing strategy which routes x to y by changing the rightmost differing bit iteratively. To be precise, assume that x differs from y at k bits, say, the l_1-, l_2-, ..., l_kth bits with

$$0 \leq l_1 < l_2 < \ldots < l_k \leq n-1.$$

Then, $P_C(x, y)$ is given by $\langle x = x^0, x^1, \ldots, x^k = y\rangle$, where x^d differs from x^{d+1} at the l_{d+1}th bit. Let $e = (u, v)$ be an edge of Q_n with u and v differing at the jth bit such that (u, v) is incident on $P_C(x, y)$. That is, either $[u, v] \in P_C(x, y)$ or $[v, u] \in P_C(x, y)$. Consider $[u, v] \in P_C(x, y)$. It follows from Algorithm C that $x_{n-1}\cdots x_j = u_{n-1}\cdots u_j$ and $y_j\cdots y_1y_0 = v_j\cdots v_1v_0$. Therefore,

$$|\{(x,y) \mid x,y \in V(Q_n), [u,v] \in P_C(x,y)\}| = 2^{n-j-1}2^j = 2^{n-1}.$$

Similarly,

$$|\{(x,y) \mid x,y \in V(Q_n), [v,u] \in P_C(x,y)\}| = 2^{n-j-1}2^j = 2^{n-1}.$$

Hence, $c(Q_n) = 2^{n-1} + 2^{n-1} = 2^n$. □

The proposed routing algorithm for the proof of Theorem 4 is an optimal routing algorithm for hypercubes and generates uniform congestion for all edges.

To calculate $c(CQ_n)$, we restrict our routing algorithm to executing only Steps 2 and 3. Furthermore, in ONE_STEP_ROUTE when $\rho_j(u,v) = 2$, we route to the $(2j+1)$th neighbor of u. This algorithm is denoted by B. In this way, Algorithm B generates only one shortest path from one vertex to another. For example, the shortest paths between all pairs of vertices in CQ_2 generated by Algorithm B are given as follows:

$$\begin{aligned}&\langle 00,10\rangle, \langle 00,10,11\rangle, \langle 00,01\rangle,\\ &\langle 01,00\rangle, \langle 01,11,10\rangle, \langle 01,11\rangle,\\ &\langle 10,00\rangle, \langle 10,00,01\rangle, \langle 10,11\rangle,\\ &\langle 11,10\rangle, \langle 11,01,00\rangle, \langle 11,01\rangle.\end{aligned}$$

Using this algorithm, we obtain $c_B(e)$ for each edge e in CQ_2 and CQ_3, as illustrated in Figs. 3a and 3b.

Lemma 3. $c_B(e) = 2^{k-1} + 2$ *for any dim-1 edge in* CQ_k.

Proof. Let $e = (u,v) = (u_{k-1}\cdots u_1u_0, v_{k-1}\cdots v_1v_0)$ be any dim-1 edge in CQ_k. Obviously, $p_{k-2}(u) = p_{k-2}(v)$ and $\bar{u}_1u_0 = v_1v_0$. We assume without loss of generality that $u_1 = 0$. Let $x = x_{k-1}x_{k-2}\cdots x_1x_0$ and $y = y_{k-1}y_{k-2}\cdots y_1y_0$ be any two vertices such that e is incident on $P_B(x,y)$, where $P_B(x,y)$ can be written as $\langle x = x^0, x^1, \ldots, x^m = y\rangle$. Obviously, we have either $[u,v] \in P_B(x,y)$ or

$$[v,u] \in P_B(x,y).$$

In order to calculate $c_B(e)$, we need to identify all possible (x,y)-pairs that route through e.

We first consider $u_0 = 0$. That is, $u_1u_0 = 00$ and $v_1v_0 = 10$. First, assume that $[u,v] \in P_B(x,y)$. It follows from Step 3 of Algorithm B that the change of the zeroth double bit is first executed, i.e., $x = x^0 = u$. When y_1y_0 is either 00 or 01, it follows from Algorithm B that $x^1 \neq v$. Thus,

$$|\{(x,y) \mid [u,v] \in P_B(x,y),\ u_1u_0 = 00,\ y_1y_0 = 00 \text{ or } 01\}| = 0.$$

When $y_1y_0 = 10$, $[u,v] \in P_B(x,y)$ always holds. Thus,

$$|\{(x,y) \mid [u,v] \in P_B(x,y),\ u_1u_0 = 00,\ y_1y_0 = 10\}| = 2^{k-2}.$$

When $y_1y_0 = 11$, it follows from Algorithm B that y is exactly the vertex $u_{k-1}u_{k-2}\cdots u_2 11$. Therefore, there are $2^{k-2} + 1$ (x,y)-pairs routing through $[u,v]$.

Next, consider $[v,u] \in P_B(x,y)$. Similarly, we first change the 0th double bit and thus, $x = x^0 = v$. When $y_1y_0 =$ 10 or 11, it follows from Algorithm B that x^1 cannot be u. Thus,

$$|\{(x,y) \mid [v,u] \in P_B(x,y),\ v_1v_0 = 10,\ y_1y_0 = 10 \text{ or } 11\}| = 0.$$

When $y_1y_0 = 00$, $P_B(x,y)$ always routes through $[v,u]$. Thus,

$$|\{(x,y) \mid [v,u] \in P_B(x,y),\ v_1v_0 = 10,\ y_1y_0 = 00\}| = 2^{k-2}.$$

When $y_1y_0 = 01$, it follows from Algorithm B that y is exactly the vertex $v_{k-1}v_{k-2}\cdots v_2 01$. Therefore, there are $2^{k-2} + 1$ (x,y)-pairs routing through $[v,u]$. From the above discussion, $c_B(e) = 2^{k-2} + 1 + 2^{k-2} + 1 = 2^{k-1} + 2$.

Next, we consider $u_0 = 1$, i.e., $u_1u_0 = 01$ and $v_1v_0 = 11$. Again, we have either $[u,v] \in P_B(x,y)$ or $[v,u] \in P_B(x,y)$. First, assume that $[u,v] \in P_B(x,y)$. Suppose $u = x^1$. It follows that $x_1x_0 = 11$, $x_1x_0 = v_1v_0$, and equivalently $\bar{x}_1x_0 \overset{d.-p.}{\sim} v_1v_0$, which contradicts $[u,v] \in P_B(x,y)$. Therefore, we have $u = x^0 = x$. When $y_1y_0 = 01$, $[u,v] \in P_B(x,y)$ implies that $\rho(p_{k-2}(u), p_{k-2}(y))$ is odd. When $y_1y_0 = 11$, $[u,v] \in P_B(x,y)$ implies that $\rho(p_{k-2}(u), p_{k-2}(y))$ is even. Therefore,

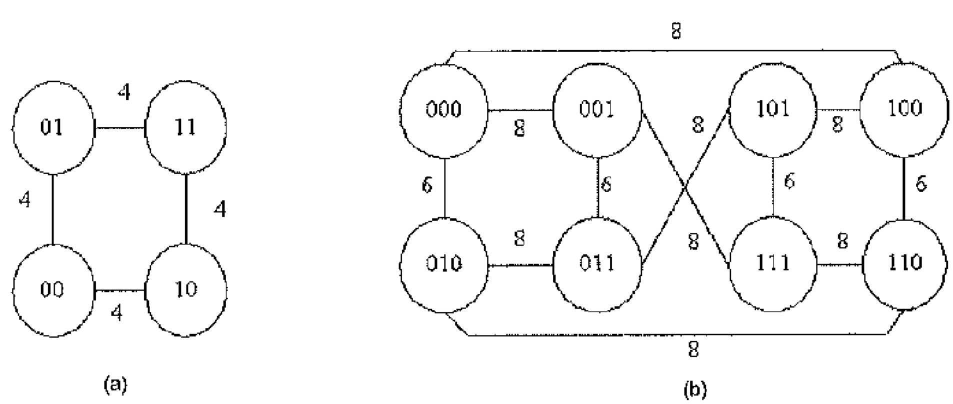


Fig. 3. Edge congestion of CQ_2 and CQ_3.

$$|\{(x,y) \mid [u,v] \in P_B(x,y),\ u_1u_0 = 01,\ y_1y_0 = 01 \text{ or } 11\}| = 2^{k-2}.$$

Since $[u,v] \in P_B(x,y)$, it follows that $y_1y_0 \neq 00$. When $y_1y_0 = 10$, y can be only $u_{k-1}u_{k-2}\cdots u_2 10$ and, thus,

$$|\{(x,y) \mid [u,v] \in P_B(x,y),\ u_1u_0 = 01\}| = 2^{k-2} + 1.$$

Furthermore, using similar arguments we can also obtain

$$|\{(x,y) \mid [v,u] \in P_B(x,y),\ v_1v_0 = 11\}| = 2^{k-2} + 1.$$

Thus, $c_B(e) = 2^{k-2} + 2^{k-2} + 2 = 2^{k-1} + 2$. The lemma now follows. □

Lemma 4. *$c_B(e) = 2^k$ for any dim-0 edge in CQ_k.*

Proof. Let $e = (u,v) = (u_{k-1}\cdots u_1u_0, v_{k-1}\cdots v_1v_0)$ be any dim-0 edge in CQ_k. Obviously, $p_{k-2}(u) = p_{k-2}(v)$ and $u_1\bar{u}_0 = v_1v_0$. Without loss of generality, we can assume that $u_0 = 0$. Let $x = x_{k-1}x_{k-2}\cdots x_1x_0$ and

$$y = y_{k-1}y_{k-2}\cdots y_1y_0$$

be any two vertices such that e is incident on $P_B(x,y)$, where $P_B(x,y) = \langle x = x^0, x^1, \ldots, x^m = y\rangle$.

Assume that $[u,v] \in P_B(x,y)$. It follows from Step 3 of Algorithm B that there are two types of (x,y)-pairs routing through $[u,v]$. The shortest path of the first type of (x,y)-pairs is first to change the zeroth double bit, while the second type is first to change some tth double bit, $1 \le t \le \lfloor\frac{k-1}{2}\rfloor$, and then the zeroth double bit. In other words, we have $[u,v] = [x^0,x^1]$ in the former and $[u,v] = [x^1,x^2]$ in the latter.

Consider that $[u,v] = [x^0,x^1]$, i.e., $x^0 = u$ and $x^1 = v$. If $y_1y_0 = u_1u_0$ or $\bar{u}_1u_0$, x^1 cannot be v, given routing under Algorithm B, since $u_0 = 0$. When $y_1y_0 = u_1\bar{u}_0 = u_11$, it follows that $[u,v] \in P_B(x,y)$ iff $\rho(p_{k-2}(u), p_{k-2}(y))$ is even. When $y_1y_0 = \bar{u}_1\bar{u}_0$, it therefore follows that $[u,v] \in P(x,y)$ iff $\rho(p_{k-2}(u), p_{k-2}(y))$ is odd. Thus,

$$|\{(x,y) \mid [u,v] \in P_B(x,y) \text{ and } x = u\}| = 2^{k-2}.$$

Next, consider that $[u,v] = [x^1,x^2]$. Clearly, $x \in N(u)$ and x is not the 0-neighbor of u. When x is the 1-neighbor of u, it follows from Algorithm B that y is the vertex v. Now let x be the l-neighbor of u with $l > 1$. It follows that $p_{n-l-1}(x) = p_{n-l-1}(u)$, $(x_{2j+1}x_{2j}, u_{2j+1}u_{2j}) \in R$ for all $0 \le j < \lfloor\frac{l}{2}\rfloor$, and $x_l = \bar{u}_l$ for l even and $x_lx_{l-1} = \bar{u}_lu_{l-1}$ for l odd. It follows from Algorithm B that $\rho_0(x,y) = 1$ and $\rho_0(v,y) = 0$, i.e., $u_1\bar{u}_0 \overset{\text{d.-p.}}{\sim} y_1y_0$. Furthermore, using Remark 2, if $\rho_j(x,y) = 1$ and $1 \le j < \lfloor\frac{l}{2}\rfloor$, it follows that $\bar{x}_{2j+1}x_{2j} \overset{\text{d.-p.}}{\not\sim} y_{2j+1}y_{2j}$ and

$$x_{2j+1}\bar{x}_{2j} \overset{\text{d.-p.}}{\not\sim} y_{2j+1}y_{2j}$$

by Algorithm B. For $\rho_j(x,y) = 0$ and $1 \le j < \lfloor\frac{l}{2}\rfloor$, we have $x_{2j+1}x_{2j} \overset{\text{d.-p.}}{\sim} y_{2j+1}y_{2j}$. Therefore, given $y_{k-1}y_{k-2}\cdots y_{l+1}$, there are $2^{\lfloor\frac{l}{2}\rfloor-1}$ choices of y satisfying $[u,v] \in P_B(x,y)$. Note that in ONE_STEP-_ROUTE when $\rho_j(u,v) = 2$, we route to the $(2j+1)$th neighbor of u. When l is even, it implies $\rho_{\lfloor\frac{l}{2}\rfloor}(x,y) = 1$. It follows that when l is even, we have

$$|\{(x,y) \mid x \text{ is the } l\text{-neighbor of } u \text{ and } [u,v] \in P_B(x,y)\}| = \begin{cases} 2^{k-l-2}2^{\lfloor\frac{l}{2}\rfloor-1} = 2^{k-\lfloor\frac{l}{2}\rfloor-3} & \text{if } 1 < l < k-1, \\ 2^{\lfloor\frac{l}{2}\rfloor-1} & \text{if } l = k-1. \end{cases}$$

When $l > 1$ is odd, then

$$|\{(x,y) \mid x \text{ is the } l\text{-neighbor of } u, \text{ and } [u,v] \in P_B(x,y)\}| = \begin{cases} 2^{\lfloor\frac{l}{2}\rfloor-1} & \text{if } \rho_{\lfloor\frac{l}{2}\rfloor}(x,y) = 2, \\ 2^{k-\lfloor\frac{l}{2}\rfloor-3} & \text{if } \rho_{\lfloor\frac{l}{2}\rfloor}(x,y) = 1. \end{cases}$$

Therefore, for even k, we have

$$|\{(x,y) \mid [u,v] \in P_B(x,y), \text{and } [u,v] = [x^1,x^2]\}| = 1 + \sum_{i=1}^{\lfloor\frac{k-1}{2}\rfloor}(2^{k-i-2} + 2^{i-1}) = 1 + \sum_{i=0}^{k-3} 2^i = 2^{k-2}.$$

Similarly, for odd k, we have

$$\begin{aligned} &|\{(x,y) \mid [u,v] \in P_B(x,y) \text{ and } [u,v] = [x^1,x^2]\}| \\ &= 1 + \sum_{i=1}^{\lfloor\frac{k-3}{2}\rfloor}(2^{k-i-2} + 2^{i-1}) + 2^{\lfloor\frac{k}{2}\rfloor-1} \\ &= 1 + \sum_{i=1}^{\lfloor\frac{k}{2}\rfloor-1}(2^{k-i-2} + 2^{i-1}) + 2^{\lfloor\frac{k}{2}\rfloor-1} = 1 + \sum_{i=0}^{k-3} 2^i = 2^{k-2}. \end{aligned}$$

Assume that $[v,u] \in P_B(x,y)$. Again, $[v,u]$ is either $[x^0,x^1]$ or $[x^1,x^2]$. If $x^0 = v$ and y_1y_0 is v_1v_0 or $\bar{v}_1v_0$ or $\bar{v}_1\bar{v}_0$, it follows from Algorithm B that $x^1 \neq u$, i.e., $[v,u] \notin P_B(x,y)$. Furthermore, $[v,u] = [x^0,x^1] \in P_B(x,y)$ if and only if $y_1y_0 = v_1\bar{v}_0$. Thus,

$$|\{(x,y) \mid [v,u] \in P_B(x,y), x = v\}| = 2^{k-2}.$$

Suppose that $[v,u] = [x^1,x^2]$. Using similar arguments for the case $[u,v] = [x^1,x^2]$, we have

$$|\{(x,y) \mid [v,u] \in P_B(x,y), x \text{ is a neighbor of } v\}| = 2^{k-2}.$$

Thus, $c_B(e) = 2^{k-2} + 2^{k-2} + 2^{k-2} + 2^{k-2} = 2^k$. □

Let $Q(x,y) = \langle x = x^0, x^1, \ldots, x^m = y\rangle$ be a path in CQ_k, $k > 2$, that traverses from x to y. Based on $Q(x,y)$, we define a new path $\tilde{Q}(p_{k-2}(x), p_{k-2}(y))$ in CQ_{k-2} by replacing all x^i in Q with $p_{k-2}(x^i)$ and deleting $p_{k-2}(x^{i+1})$ if

$$p_{k-2}(x^{i+1}) = p_{k-2}(x^i).$$

Definition 8. *We say that (x,y) and (x',y') are in an equivalence class, denoted by $(x,y) \cong (x',y')$, if and only if $\tilde{P}_B(x,y) = \tilde{P}_B(x',y')$.*

Let (u,v) be a dim-l edge in CQ_k where $l \geq 2$. Given an edge $(u,v) \in E(CQ_k)$, let

$$\chi(u,v) = \{(x,y) \mid [u,v] \in P_B(x,y) \text{ or } [v,u] \in P_B(x,y)\}.$$

We can partition each set $\chi[u,v]$ into different equivalence classes. It is obvious that each equivalence class contains at most 16 elements since $x'_1, x'_0, y'_1, y'_0 \in \{0,1\}$. However, in the following lemma we show that each equivalence class contains exactly four elements.

Lemma 5. *Let (u,v) be a dim-d, $d \geq 2$, edge of CQ_k. Each equivalence class in $\chi[u,v]$ contains exactly four elements.*

We leave the proof of Lemma 5 to Appendix B.

Lemma 6. *Let e be an edge of dimension d in CQ_n for $n \geq 2$. Then,*

$$c_B(e) = \begin{cases} 2^{n-1} + 2^d & \text{if } d \text{ is odd,} \\ 2^n & \text{if } d \text{ is even.} \end{cases}$$

Proof. The proof is by induction on n. The theorem is true for $n = 2, 3$, as illustrated in Fig. 3. Suppose that this theorem holds for CQ_{n-1}. It follows from Lemmas 3 and 4 that the statement is also true for $d = 0, 1$. Let $e = (u,v)$ be a dim-d, $d \geq 2$, edge in CQ_n where $n \geq 2$. Obviously, $(p_{n-2}(u), p_{n-2}(v))$ is an edge of dimension $(d-2)$ in CQ_{n-2}. By induction,

$$c_B(p_{n-2}(u), p_{n-2}(v)) = \begin{cases} 2^{n-3} + 2^{d-2} & \text{if } d \text{ is odd,} \\ 2^{n-2} & \text{if } d \text{ is even.} \end{cases}$$

Since $\tilde{P}_B(x,y) = P_B(p_{n-2}(x), p_{n-2}(y))$, it follows from Lemma 5 that we have

$$c_B(u,v) = \begin{cases} 2^{n-1} + 2^d & \text{if } d \text{ is odd,} \\ 2^n & \text{if } d \text{ is even.} \end{cases}$$

Hence, the lemma follows. □

Lemma 6 immediately yields the following corollary.

Corollary 2. $c(CQ_n) \leq 2^n$.

We use a similar proof technique of finding $\omega(Q_n)$ to find $\omega(CQ_n)$, which is stated in the following theorem.

Theorem 5. $\omega(CQ_n) = 2^{n-1}$.

Proof. Note that CQ_n is constructed from two identical $(n-1)$-dimensional crossed cubes CQ^0_{n-1} and CQ^1_{n-1}, which are connected by *dim*-$(n-1)$ edges of CQ_n. Since these dim-$(n-1)$ edges form a perfect matching and removal of these edges disconnects CQ_n, it follows that $\omega(CQ_n) \leq 2^{n-1}$.

We define an embedding of a directed complete graph of 2^n vertices, denoted by K, into CQ_n where each edge from u to v in K is embedded by $P_B(u,v)$ in CQ_n.

Suppose $\omega(CQ_n) = w < 2^{n-1}$. It follows that CQ_n can be partitioned into two subgraphs of equal size by removing a cut of w edges. This cut of CQ_n also induces a bisection of K. Since each edge of CQ_n is contained in at most 2^n shortest paths following from Lemma 6, it follows that $\omega(K) \leq w2^n < 2^{2n-1}$, which is contradictory to the known fact that $\omega(K) \geq 2^{2n-1}$. □

We will use the following lemma (originally stated in [4]) to find a lower bound for $c(G)$.

Lemma 7. *Let G be an n-vertex graph such that the removal of a cut of k edges partitions G into two, not necessarily connected, subgraphs of t vertices and $n-t$ vertices, respectively. Then,*

$$c(G) \geq \lceil \frac{2t(n-t)}{k} \rceil. \quad (1)$$

Moreover, this bound is tight.

In particular, we can choose a cut that bisects G in (1), which can be restated as follows:

$$c(G) \geq \begin{cases} \lceil n^2/2\omega(G) \rceil & \text{for } n \text{ even,} \\ \lceil (n^2-1)/2\omega(G) \rceil & \text{for } n \text{ odd.} \end{cases} \quad (2)$$

Theorem 6. $c(CQ_n) = 2^n$.

Proof. It follows from (2) and Theorem 5 that $c(CQ_n) \geq 2^n$. On the other hand, Corollary 2 states $c(CQ_n) \leq 2^n$. Thus the theorem follows. □

Since our proposed shortest routing Algorithm B achieves $c(CQ_n)$, Algorithm B is an optimal routing algorithm. Theorem 6 implies that $c(CQ_n) = c(Q_n)$. Furthermore, since hypercubes have an optimal routing algorithm which generates uniform congestion of all edges, Theorem 6 implies that each edge of crossed cubes has smaller congestion than or equal congestion to that of hypercubes.

6 Embedding of Cycles

A cycle is often used as a connection structure for local area networks, and can also be used as a control/data flow structure for distributed computation in arbitrary networks. In this section, we present embedding of cycles into CQ_n that our embedding of cycles is different from the one proposed in [9]. A cycle of length k is denoted by C_k. Let u, v be binary strings of length $n-2$ satisfying $(u,v) \in E(CQ_{n-2})$. In order to construct large cycles in CQ_n, we define two types of *primitive paths* from $u00$ or $u10$ to $v00$ as follows:

Type 1: Four primitive paths from $u00$ to $v00$ given by

$$P_{1,1} = \langle u00, u01, u11, v01, v00 \rangle,$$
$$P_{1,2} = \langle u00, u10, v10, v11, v01, v00 \rangle,$$
$$P_{1,3} = \langle u00, u10, u11, u01, v11, v01, v00 \rangle,$$
$$P_{1,4} = \langle u00, u01, u11, u10, v10, v11, v01, v00 \rangle,$$

as shown in Fig. 4a;

Type 2: Four primitive paths from $u10$ to $v00$ given by

$$P_{2,1} = \langle u10, u11, u01, u00, v00 \rangle,$$
$$P_{2,2} = \langle u10, v10, v11, u01, u00, v00 \rangle,$$
$$P_{2,3} = \langle u10, u11, v01, v11, u01, u00, v00 \rangle,$$
$$P_{2,4} = \langle u10, v10, v11, v01, u11, u01, u00, v00 \rangle,$$

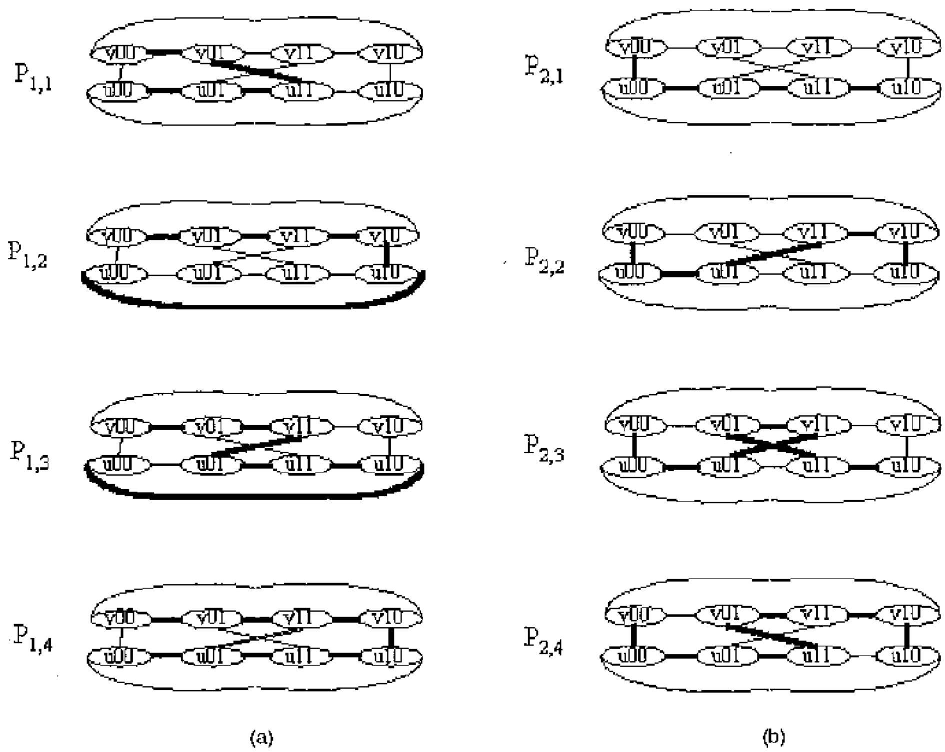


Fig. 4. Primitive paths of (a) type 1 and (b) type 2.

as shown in Fig. 4b. Each $P_{i,j}$ has the length of $j+3$ for $i = 1, 2$.

Theorem 7. *CQ_n is a pancyclic network for $n \geq 2$, i.e., CQ_n contains C_l for all $4 \leq l \leq 2^n$ as subgraphs.*

Proof. The theorem can be easily proven for $n = 2, 3$ by using type-1 primitive paths. For $n = 4$, since CQ_3 is a subgraph of CQ_4, it suffices to construct cycles C_l for $9 \leq l \leq 16$ as follows:

$$C_9 = \langle 0000, 0001, 0011, 0010, 0110, 1110, 1111, 0101, 0100, 0000\rangle,$$
$$C_{10} = \langle 0000, 0001, 0011, 0010, 0110, 1110, 1111, 1101, 1100, 0100, 0000\rangle,$$
$$C_{11} = \langle 0000, 0001, 0011, 0010, 0110, 0111, 0101, 1111, 1101, 1100, 0100, 0000\rangle,$$
$$C_{12} = \langle 0000, 0001, 0011, 0010, 0110, 1110, 1111, 0101, 0111, 1101, 1100, 0100, 0000\rangle,$$
$$C_{13} = \langle 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, P_{1,1}, 1000, 0000\rangle,$$
$$C_{14} = \langle 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, P_{1,2}, 1000, 0000\rangle,$$
$$C_{15} = \langle 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, P_{1,3}, 1000, 0000\rangle,$$
$$C_{16} = \langle 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, P_{1,4}, 1000, 0000\rangle.$$

We next prove the theorem for $n \geq 5$ by induction. Since CQ_{n-1} is a subgraph of CQ_n, it suffices to construct cycles of length $2^{n-1} + 1 \leq l \leq 2^n$. Let s be an arbitrary integer satisfying $2^{n-3} + 1 \leq s \leq 2^{n-2}$. By the induction hypothesis, there is a cycle C_s in CQ_{n-2} given by

$$C_s = \langle u^1, \cdots, u^s, u^1\rangle.$$

When s is even, construct C_i based on C_s as follows:

$$C_i = \langle u^1 00, u^1 01, u^1 11, u^1 10, u^2 10, u^2 11, u^2 01, u^2 00, u^3 00, u^3 01, u^3 11, u^3 10, u^4 10, \cdots, u^{s-2} 00, u^{s-1} 00, P', u^s 00, u^1 00\rangle,$$

where P' is a type-1 primitive path from $u^{s-1}00$ to u^s00. It follows that $i = 4(s-2) + |P'| + 1$ and to be precise, $i = 4s-3, 4s-2, 4s-1, 4s$.

When s is odd, we use C_s to construct C_j similar to C_i as follows:

$$C_j = \langle u^1 00, u^1 01, u^1 11, u^1 10, u^2 10, u^2 11, u^2 01, u^2 00, u^3 00, u^3 01, u^3 11, u^3 10, \cdots, u^{s-2} 10, u^{s-1} 10, P'', u^s 00, u^1 00\rangle,$$

where P'' is a type-2 primitive path from $u^{s-1}10$ to u^s00. It follows that j can be $4s-3, 4s-2, 4s-1, 4s$. Therefore, we have obtained cycles C_l such that

$$4(2^{n-3}+1) - 3 \leq l \leq 4(2^{n-2}),$$

i.e., $2^{n-1} + 1 \leq l \leq 2^n$. Hence, the theorem follows. □

Based on the above proof idea, we can easily construct a cycle of arbitrary length. We here illustrate an example of constructing C_{19} in CQ_5. First, we find a cycle of length $s = \lceil \frac{19}{4} \rceil = 5$ in CQ_3, which is given by

$$C_5 = \langle 000, 001, 011, 101, 100, 000 \rangle.$$

Since s is odd and $19 = 4s - 1$, we can use C_5 and a $P_{2,3}$ from 10110 to 10000 to construct a C_{19} as follows:

$$C_{19} = \langle 00000, 00001, 00011, 00010, 00110, 00111, 00101, 00100, 01100, 01101, 01111, 01110, 10110, 10111, 10001, 10011, 10101, 10100, 10000, 00000 \rangle.$$

Embedding cycles presented in [9] can construct one type of cycles for an arbitrary length, while ours can construct various types of cycles since we can start with any vertex and make modification of type-1 and type-2 primitive paths to construct cycles. For example, to construct C'_{19} in CQ_5, we define a different cycle of length 5, which is given as follows:

$$C'_5 = \langle 011, 010, 000, 100, 101, 011 \rangle.$$

We can use C'_5 and a path P' from 10011 to 10101 of length six to construct C'_{19} as follows:

$$C'_{19} = \langle 01111, 01110, 01100, 01101, 01011, 01010, 01000, 01001, 00011, 00010, 00000, 00001, 10011, 10001, 10000, 10010, 10110, 10100, 10101, 01111 \rangle.$$

7 Concluding Remarks

Though topological properties of crossed cubes have been studied in the literature, we introduce a new measure in this paper called pair related distance. Using this measure, we can easily find the shortest distance between any pair of vertices. Furthermore, we use this measure to give a shortest path routing algorithm in $O(n)$ time rather than $O(n^2)$ time in comparison with previous work. In this paper, we also define a new performance measure called edge congestion. Given the shortest path routing algorithm presented in this paper, we show that the edge congestion of crossed cubes is 2^n, equal to that of hypercubes. Bisection width of crossed cubes is 2^{n-1}. We also prove that the wide diameter and the fault diameter of crossed cubes are approximately half of those of hypercubes. Furthermore, crossed cubes are shown to be pancyclic networks with more types of cycles constructed.

Appendix A

Proof of Lemma 3. We prove this lemma by induction on n. Obviously, the lemma is true for $n = 2$. Assume that such disjoint paths exist for any two distinct vertices in CQ_l for $l < n$ and $n \geq 3$. Let $u = u_{n-1}u_{n-2}\cdots u_1u_0$ and $v = v_{n-1}v_{n-2}\cdots v_1v_0$ be two vertices in CQ_n.

We first consider n odd and distinguish the following cases.

Case 1.1. $d_{CQ_3}(p_3(u), p_3(v)) \leq 1$ or $u_{n-1} = v_{n-1}$.

It follows that u, v belong to a subgraph G_1 which is isomorphic to CQ_{n-1}. By the induction hypothesis, there are $n-1$ disjoint paths $P_1, P_2, \cdots, P_{n-1}$ joining u to v such that $|P_i| \leq \lceil \frac{n}{2} \rceil + 1$ for all $i \leq n-1$ and $|P_1| \leq \lceil \frac{n}{2} \rceil$. There exist two binary strings $x^1 = x^1_{n-1}x^1_{n-2}x^1_{n-3}$ and $y^1 = y^1_{n-1}y^1_{n-2}y^1_{n-3}$ such that

1. $(p_3(u), x^1) \in E(CQ_3)$ and $(p_3(v), y^1) \in E(CQ_3)$,
2. $x^1 \neq p_3(v)$, $y^1 \neq p_3(u)$,
3. $x^1_{n-1} \neq u_{n-1}$, $y^1_{n-1} \neq v_{n-1}$ if $u_{n-1} = v_{n-1}$,
4. $x^1_{n-3} \neq u_{n-3}$, $y^1_{n-3} \neq v_{n-3}$ if $d_{CQ_3}(p_3(u), p_3(v)) \leq 1$,
5. $x^1 = y^1$ if $p_3(u) = p_3(v)$.

Fig. 5a illustrates an example of choice for x^1 and y^1. Those figures in Fig. 5 illustrate an example of choice of specified vertices whose positions are not unique.

Let u^1 and v^1 be neighbors of u and v, given by $u^1 = f(u; p_3(u), x^1)$ and $v^1 = f(v; p_3(v), y^1)$, where in particular $u^1 = x^1$ and $v^1 = y^1$ for $n = 3$. Furthermore, u^1 and v^1 are chosen from $G_2 = CQ_n - G_1$, which is also isomorphic to CQ_{n-1}. Let S_1 be a shortest path in G_2 joining u^1 and v^1 and satisfying $|S_1| \leq \lceil \frac{n}{2} \rceil$. Construct P_n as

$$P_n = \langle u, u^1, S_1, v^1, v \rangle,$$

which is disjoint from $P_1, P_2, \cdots, P_{n-1}$. Thus,

$$P_1, P_2, \cdots, P_n$$

satisfy Conditions 1, 2, and 3.

Case 1.2. $d_{CQ_3}(p_3(u), p_3(v)) = 2$ and $u_{n-1} \neq v_{n-1}$.

Since $p_{n-2}(u)$ and $p_{n-2}(v)$ are two different vertices in CQ_{n-2}, by the induction hypothesis there are $n-2$ disjoint paths $P'_1, P'_2, \cdots, P'_{n-2}$ joining $p_{n-2}(u)$ and $p_{n-2}(v)$ such that $|P'_1| \leq \lceil \frac{n}{2} \rceil - 1$, $|P'_i| \leq \lceil \frac{n}{2} \rceil + 1$ for all $i \leq n-2$. We use these $n-2$ paths to construct P^*_i as follows:

$$P^*_i = P(u; P'_i) = \langle u, u^{i,1}, \cdots, u^{i,|P'_i|} = w_i \rangle. \quad (3)$$

Note that all of P^*_i are disjoint paths in CQ_n and that $|P^*_i| = |P'_i|$ for all $1 \leq i \leq n-2$. Furthermore, since

$$p_{n-2}(w_i) = p_{n-2}(v),$$

it follows that w_i and v are in a subgraph isomorphic to CQ_2. However, P^*_i are not necessarily (u, v)-paths.

We make the following modification of P^*_i for $2 \leq i \leq n-2$ in order to obtain $n-3$ (u, v)-paths P_i. If $w_i = v$, we simply let $P_i = P^*_i$. Now consider $w_i \neq v$. Let z_i be the immediate predecessor of w_i, and let z'_i be the immediate predecessor of z_i on P^*_i. Let v' be a neighbor of v in $CQ_n(p_{n-2}(z_i))$, and let v'' be a neighbor of v' in $CQ_n(p_{n-2}(z'_i))$. Since $CQ_n(p_{n-2}(z_i))$ and $CQ_n(p_{n-2}(w_i))$ are adjacent subgraphs,

$$CQ_n(p_{n-2}(z_i), p_{n-2}(w_i)) = CQ_n(p_{n-2}(z_i), p_{n-2}(v))$$

is isomorphic to CQ_3. If v' is also a neighbor of z_i (as illustrated in Fig. 5b), we construct

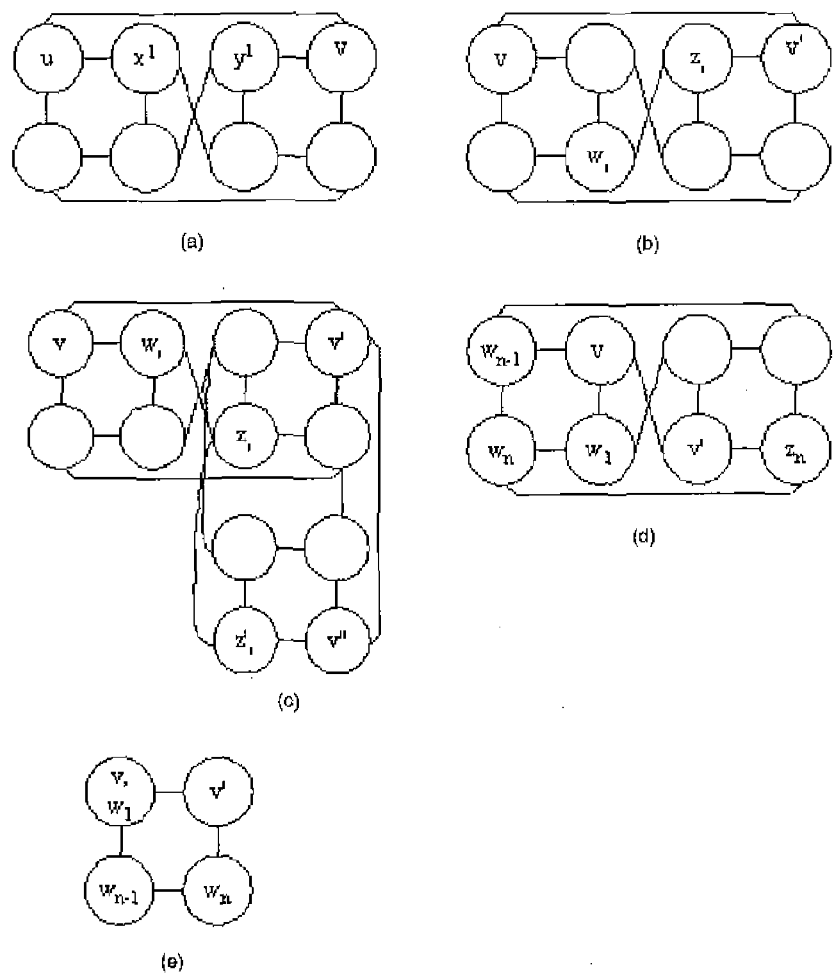


Fig. 5. Illustration in Appendix A for examples of relative positions of some specific vertices in CQ_{2k+1}.

$$P_i = (P_i^* - \langle z_i, w_i\rangle) \cup \langle z_i, v', v\rangle,$$

where $|P_i| = |P_i^*| + 1$. Otherwise, v'' is also a neighbor of z_i' (as illustrated in Fig. 5c) and we construct

$$P_i = (P_i^* - \langle z_i', z_i, w_i\rangle) \cup \langle z_i', v''v', v\rangle,$$

where $|P_i| = |P_i^*| + 1$.

We then use P_1' to construct three (u, v)-paths P_1, P_{n-1}, P_n as follows. For $i = 0, 1$, let u^i be the neighbor of u that differs from u at the ith bit. Let $S_{n-1} = P(u^1; P_1')$,

$$S_n = P(u^0; P_1'),$$

$w_{n-1} = t(u^1; S_{n-1})$, and $w_n = t(u^0; S_n)$. Note that

$$|S_{n-1}| = |S_n| = |P_1^*|.$$

Furthermore, $w_1 \neq w_{n-1} \neq w_n$ and

$$p_{n-2}(w_1) = p_{n-2}(w_{n-1}) = p_{n-2}(w_n).$$

Therefore, all of w_1, w_{n-1}, and w_n are in a subgraph isomorphic to CQ_2. Let z_n be the immediate predecessor of w_n on S_n.

Consider that all of w_1, w_{n-1}, and w_n are distinct from v, say, $d_{CQ_n}(w_1, v) = d_{CQ_n}(w_{n-1}, v) = 1$ and

$$d_{CQ_n}(w_n, v) = 2.$$

We construct

$$P_1 = \langle u, P_1^*, w_1, v\rangle, \text{ and } P_{n-1} = \langle u, u^1, S_{n-1}, w_{n-1}, v\rangle, \qquad (4)$$

where $|P_1| = |P_1^*| + 1$ and $|P_{n-1}| = |S_{n-1}| + 2$. Since

$$CQ_n(p_{n-2}(z_n), p_{n-2}(w_n))$$

is isomorphic to CQ_3, there exists a neighbor v' of z_n in $CQ_n(p_{n-2}(z_n))$ such that v' is the common neighbor of v and z_n (as illustrated in Fig. 5d). It follows that $d_{CQ_n}(z_n, v) = 2$. We construct

$$P_n = \langle u, u^0\rangle \cup (S_n - \langle z_n, w_n\rangle) \cup \langle z_n, v', v\rangle, \qquad (5)$$

where $|P_n| = |S_n| + 2$.

Now consider that one of w_1, w_{n-1}, w_n is exactly v, say, $w_1 = v$, $d_{CQ_n}(w_{n-1}, v) = 1$ and $d_{CQ_n}(w_n, v) = 2$. We simply let $P_1 = P_1^*$ and construct P_{n-1} as above. There exists a common neighbor v' of w_n and v in $CQ_n(p_{n-2}(w_n))$ with $v' \neq w_{n-1}$ (as illustrated in Fig. 5e). We construct

$$P_n = \langle u, u^0, S_n, w_n, v', v \rangle,$$

where $|P_n| = |S_n| + 3$. The other cases of relative positions of w_1, w_{n-1}, w_n, and v can be similarly treated.

Now we have constructed n (u,v)-paths satisfying Conditions 1, 2, and 3. To prove disjointness of these paths, let x be any internal vertex of P_i with $2 \le i \le n-2$ and let y be any internal vertex of P_1, P_{n-1} and P_n. Clearly,

$$p_{n-2}(x) \neq p_{n-2}(y).$$

Moreover, any internal vertex of P_1, P_{n-1} and P_n differs from each other at the last two bits. Thus, we have constructed n disjoint paths $P_1, P_2, \cdots, P_n$ which satisfy Conditions 1, 2, and 3.

Next, consider n is even. We distinguish the following cases.

Case 2.1. $d_{CQ_2}(p_2(u), p_2(v)) \le 1$.

The proof is similar to that of Case 1.1.

Case 2.2. $d_{CQ_2}(p_2(u), p_2(v)) = 2$.

Let P_i' be defined as in Case 1.2, and let P_i^* be defined as in (3) for $1 \le i \le n-2$. The first $n-4$ (u,v)-paths are constructed by P_i^* for all $3 \le i \le n-2$ as in Case 1.2, which satisfy Conditions 1, 2, and 3.

In the following discussion, we use P_1' and P_2' to construct the remaining four (u,v)-paths P_1, P_2, P_{n-1}, P_n. Let u^i be defined as in Case 1.2. If $|P_1'| \le D(CQ_{n-2}) - 1$, then P_1, P_{n-1}, P_n, and P_2 are constructed in the same way as in Case 1.2. Now, consider $|P_1'| = D(CQ_{n-2})$. It follows from Condition 3 that $|P_1'| = |P_2'| = D(CQ_{n-2})$. We define $S_{n-1} = P(u^1; P_1')$, $S_n = P(u^0; P_2')$, $w_{n-1} = t(u^1; S_{n-1})$ and $w_n = t(u^0; S_n)$. Note that $|S_{n-1}| = |S_n| = |P_1^*| = |P_2^*|$. We note that w_1 and w_2 are in CQ_2. Furthermore, P_1^* and P_2^* start from the same vertex and $|P_1^*| = |P_2^*|$. It follows that $w_1 = w_2$. Moreover, $w_{n-1} \neq w_n$ and

$$p_{n-2}(w_1) = p_{n-2}(w_2) = p_{n-2}(w_{n-1}) = p_{n-2}(w_n).$$

Therefore, all of w_1, w_2, w_{n-1}, and w_n are in a subgraph isomorphic to CQ_2. Let z_i be the immediate predecessor of w_i and z_i' be the immediate predecessor of z_i on P_i^* for $i = 1, 2$. Let z_j be the immediate predecessor of w_j and z_j' be the immediate predecessor of z_j on S_j for $j = n-1, n$. Let v' and z' be neighbors of v in $CQ_n(p_{n-2}(z_n))$ and $CQ_n(p_{n-2}(z_1))$, respectively.

Suppose that all of w_1, w_2, w_{n-1}, and w_n are distinct from v, say, $d_{CQ_n}(w_1, v) = d_{CQ_n}(w_2, v) = d_{CQ_n}(w_{n-1}, v) = 1$ and $d_{CQ_n}(w_n, v) = 2$. We construct P_{n-1} as in (4) and

$$P_2 = \langle u, P_2^*, w_2, v \rangle,$$

where $|P_2| = |P_2^*| + 1$. Since $CQ_n(p_{n-2}(z_n), p_{n-2}(w_n))$ is isomorphic to CQ_3, using the same arguments as in Case 1.2, we construct P_n as given by (5). Since $CQ_n(p_{n-2}(z_1))$ and $CQ_n(p_{n-2}(w_1))$ are adjacent subgraphs, $CQ_n(p_{n-2}(z_1), p_{n-2}(w_1))$ is isomorphic to CQ_3. If z' is also a neighbor of z_1 (as shown in Fig. 6a), we construct

$$P_1 = (P_1^* - \langle z_1, w_1 \rangle) \cup \langle z_1, z', v \rangle,$$

where $|P_1| = |P_1^*| + 1$. Otherwise, there exists a common neighbor of z_1' and z', denoted by z'', in $CQ_n(p_{n-2}(z_1'))$ (as shown in Fig. 6b) and we construct

$$P_1 = (P_1^* - \langle z_1', z_1, w_1 \rangle) \cup \langle z_1', z'', z', v \rangle,$$

where $|P_1| = |P_1^*| + 1$.

Next, suppose some of w_1, w_2, w_{n-1}, w_n are exactly v, say, $w_1 = w_2 = v$, $d_{CQ_n}(w_{n-1}, v) = 1$, and $d_{CQ_n}(w_n, v) = 2$. We simply let $P_1 = P_1^*$ and construct P_{n-1} and P_n as above. In particular, we have $v' = z_2$ on P_n. Since $CQ_n(p_{n-2}(w_2))$ is isomorphic to CQ_2, there exists a neighbor w' of v in $CQ_n(p_{n-2}(w_2))$ satisfying $w' \neq w_{n-1}$. Since $CQ_n(p_{n-2}(z_2), p_{n-2}(w_2))$ and $CQ_n(p_{n-2}(z_2'), p_{n-2}(z_2))$ are isomorphic to CQ_3, there exists a neighbor w'' of w' in $CQ_n(p_{n-2}(z_2))$ with $w'' \neq z_2 \neq z_n$ and a common neighbor w''' of w'' and z_2' in $CQ_n(p_{n-2}(z_2'))$ with $w'' \neq z_n'$ (as illustrated in Fig. 6c). We construct

$$P_2 = (P_2^* - \langle z_2', z_2, w_2 = v \rangle) \cup \langle z_2', w''', w'', w', v \rangle,$$

where $|P_2| = |P_2^*| + 2$. The other cases of relative positions of w_1, w_2, w_{n-1}, w_n, and v can be similarly treated.

Thus, we have obtained n (u,v)-paths satisfying Conditions 1, 2, and 3. Let x be any internal vertex of P_i with $3 \le i \le n-2$ and y be any internal vertex of P_1, P_2, P_{n-1}, and P_n. Clearly, $p_{n-2}(x) \neq p_{n-2}(y)$. Moreover, any internal vertex of P_1 differs from those of P_{n-1} at the last two bits and differs from those of P_2, P_n at the first $n-2$ bits. Similarly, any internal vertex of P_2 differs from those of P_n at the last two bits. Thus, $P_1, P_2, \cdots, P_n$ are disjoint paths and satisfy Conditions 1, 2, and 3. Thus, the proof is completed. □

Appendix B

Proof of Lemma 5. Let $x = x_{k-1} \cdots x_1 x_0$ and $y = y_{k-1} \cdots y_1 y_0$ be two vertices in CQ_k where $(x, y) \in \chi[u, v]$. Let

$$P_B(x, y) = \langle x = x^0, x^1, \ldots, x^m = y \rangle$$

and

$$\hat{P}_B(x, y) = \langle p_{k-2}(x) = \hat{x}^0, \hat{x}^1, \ldots, \hat{x}^t = p_{k-2}(y) = \hat{y} \rangle.$$

We can assume without loss of generality that $P_B(x, y)$ is the shortest among all of $P_B(x', y')$, where each (x', y') is in the same equivalence class with (x, y), i.e., $(x', y') \in \chi[u, v]$ and $(x, y) \cong (x', y')$. It follows that $\rho_0(x, y) = 0$, since otherwise we can obtain another pair of vertices in this equivalence class having shorter path. Since (u, v) is in $P_B(x, y)$ and is also a dim-d, $d \ge 2$, edge of CQ_k, it follows that

$$(p_{k-2}(u), p_{k-2}(v)) = [\hat{x}^i, \hat{x}^{i+1}]$$

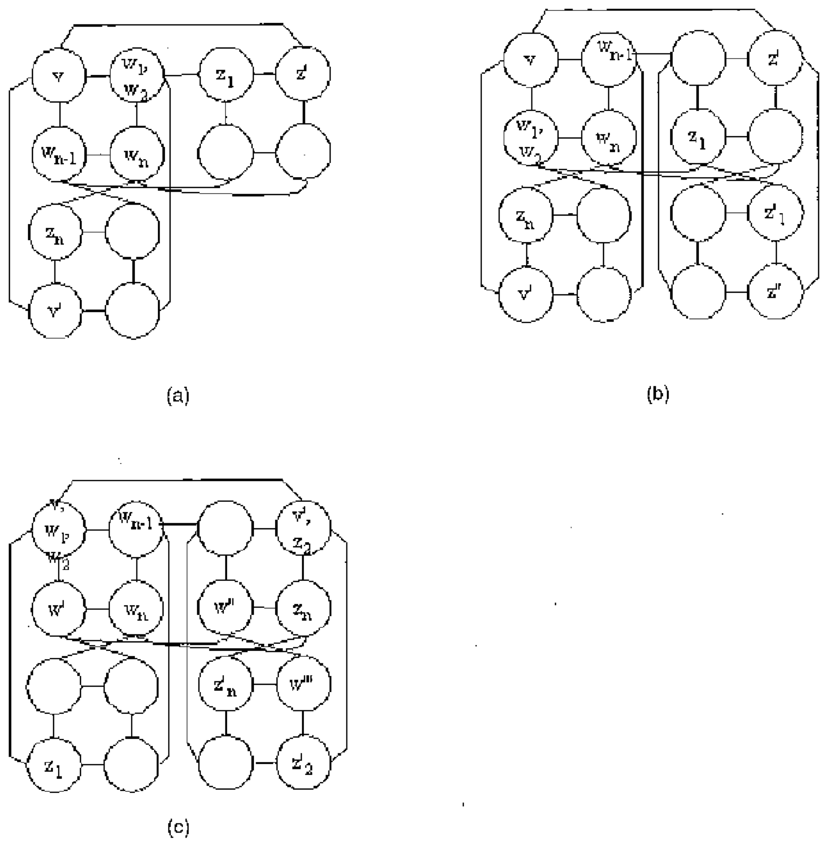


Fig. 6. Illustration in Appendix A for examples of relative positions of some specific vertices in GQ_{2k}

or $(p_{k-2}(v), p_{k-2}(u)) = [\hat{x}^i, \hat{x}^{i+1}]$ for some $i \ge 0$. We first assume that $(p_{k-2}(u), p_{k-2}(v)) = [\hat{x}^i, \hat{x}^{i+1}]$ for $i \ge 0$. In the following discussion, we abbreviate "$x'_1x'_0 \overset{d.-p.}{\sim} y'_1y'_0$ in (x', y')" by simply $x'_1x'_0 \overset{d.-p.}{\sim} y'_1y'_0$ without ambiguity from the context. Let $y' = p_{k-2}(y)y'_1y'_0$, where $y'_1y'_0 \neq y_1y_0$; and thus, $x_1x_0 \overset{d.\ p.}{\not\sim} y'_1y'_0$.

Case 1: $(p_{k-2}(u), p_{k-2}(v)) = [\hat{x}^0, \hat{x}^1]$.

Since $P_B(x, y)$ is the shortest in the equivalence class, it follows that $x = u = x^0$.

Let $x' = \hat{x}^0\bar{x}_1x_0$ or $\hat{x}^0x_1\bar{x}_0$. It follows that

$$P_B(x', y) = \langle x', P_B(x, y)\rangle.$$

Therefore, $[u, v] \in P_B(x', y)$ and $\tilde{P}_B(x', y) = \tilde{P}_B(x, y)$, i.e., $(\hat{x}^0\bar{x}_1x_0, y) \cong (x, y)$ and $(\hat{x}^0x_1\bar{x}_0, y) \cong (x, y)$. Since

$$\rho_0(\hat{x}^0\bar{x}_1\bar{x}_0, y) = 1$$

and the vertices $\hat{x}^0\bar{x}_1\bar{x}_0$ and x are not adjacent,

$$P_B(\hat{x}^0\bar{x}_1\bar{x}_0, y)$$

does not pass through (u, v), i.e.,

$$(\hat{x}^0\bar{x}_1\bar{x}_0, y) \notin \chi[u, v].$$

Thus, $(\hat{x}^0\bar{x}_1\bar{x}_0, y) \not\cong (x, y)$.

Since $\rho_0(x, y) = 0$ implies $\rho_0(x, y') = 1$, we consider the following three possibilities: First, let $x_1x_0 \overset{d.-p.}{\sim} y'_1y'_0$. Then $P_B(x, y')$ and $P_B(\hat{x}^0\bar{x}_1\bar{x}_0, y')$ will first change the zeroth double bit. It follows that $[x, v] \notin P_B(x, y')$ and

$$[x, v] \notin P_B(\hat{x}^0\bar{x}_1\bar{x}_0, y'),$$

i.e., $(x, y') \notin \chi[u, v]$ and $(\hat{x}^0\bar{x}_1\bar{x}_0, y') \notin \chi[u, v]$. Furthermore,

$$P_B(\hat{x}_0\bar{x}_1x_0, y')$$

will not route through $[u, v]$ since $x_1x_0 \overset{d.\ p.}{\not\sim} v_1v_0$. Therefore, $(\hat{x}^0\bar{x}_1x_0, y') \notin \chi[u, v]$. Since $x_1x_0 \overset{d.-p.}{\sim} y'_1y'_0$ implies $x_1\bar{x}_0 \overset{d.-p.}{\not\sim} y'_1y'_0$ and $\bar{x}_1\bar{x}_0 \overset{d.-p.}{\not\sim} y'_1y'_0$, then

$$P_B(\hat{x}_0x_1\bar{x}_0, y')4$$

will first route to a d-neighbor of $\hat{x}^0x_1\bar{x}_0$ and then change the zeroth double bit. It follows that

$$(u, v) \notin P_B(\hat{x}_0x_1\bar{x}_0, y'),$$

i.e., $(\hat{x}_0x_1\bar{x}_0, y') \notin \chi[u, v]$. Thus when $x_1x_0 \overset{d.-p.}{\sim} y'_1y'_0$, none of (x, y'), $(\hat{x}^0x_1\bar{x}_0, y')$, $(\hat{x}^0\bar{x}_1x_0, y')$, and $(\hat{x}^0x_1\bar{x}_0, y')$ is in $\chi[u, v]$.

Next, let $x_1\bar{x}_0 \overset{d.-p.}{\sim} y'_1y'_0$. Similarly, we can also show that none of (x, y'), $(\hat{x}^0x_1\bar{x}_0, y')$, $(\hat{x}^0\bar{x}_1x_0, y')$, and $(\hat{x}^0\bar{x}_1\bar{x}_0, y')$ is in $\chi[u, v]$.

Finally, let $\bar{x}_1\bar{x}_0 \overset{d.-p.}{\sim} y'_1y'_0$. We can also show that

$$(\hat{x}^0x_1\bar{x}_0, y') \notin \chi[u, v],$$

$(\hat{x}^0\bar{x}_1x_0, y') \notin \chi[u, v]$, and $(\hat{x}^0\bar{x}_1\bar{x}_0, y') \notin \chi[u, v]$. Since $\bar{x}_1\bar{x}_0 \overset{d.-p.}{\sim} y'_1y'_0$, then $P_B(x, y')$ will route through (x, v) and then change the zeroth double bit. Let L be a path in CQ_{k-2} given by $L = \tilde{P}_B(v, y)$. To be specific,

$$P_B(x, y') = \langle x = u, v, p_{k-2}(v)v_1\bar{v}_0, P(p_{k-2}(v)v_1\bar{v}_0; L)\rangle.$$

It follows that $(x, y') \in \chi[u, v]$ and $\tilde{P}_B(x, y') = \tilde{P}_B(x, y)$. Therefore, $(x, y') \cong (x, y)$ given $y'_1y'_0 \overset{d.-p.}{\sim} \bar{x}_1\bar{x}_0$. Thus, we obtain exactly four elements in the same equivalence class, i.e., (x, y), $(\hat{x}^0x_1\bar{x}_0, y)$, $(\hat{x}^0\bar{x}_1x_0, y)$, and (x, y'), where $y'_1y'_0 \overset{d.-p.}{\sim} \bar{x}_1\bar{x}_0$.

Case 2: $(p_{k-2}(u), p_{k-2}(v)) = [\hat{x}^i, \hat{x}^{i+1}]$ for $i \geq 1$.

Since $P_B(x, y)$ is the shortest in the equivalence class, it follows that $x^j = u$ for some j, $\rho_0(x, u) = 0$ and $\rho_0(u, y) = 0$.

Let $x' = \hat{x}^0\bar{x}_1x_0$ or $\hat{x}^0x_1\bar{x}_0$. It follows that

$$P_B(x', y) = \langle x', P_B(x, y)\rangle.$$

Therefore, $[u, v] \in P_B(x', y)$ and $\tilde{P}_B(x', y) = \tilde{P}_B(x, y)$, i.e., $(\hat{x}^0\bar{x}_1x_0, y) \cong (x, y)$ and $(\hat{x}^0x_1\bar{x}_0, y) \cong (x, y)$. Let

$$x' = \hat{x}^0\bar{x}_1\bar{x}_0$$

and let L be a subpath of $P_B(x, y)$ given by

$$L = \langle x^1, \ldots, x^m = y\rangle.$$

Since (x, x^1) is an edge, it follows that $(x_1x_0, x_1^1x_0^1) \in R$ and moreover, $(\bar{x}_1\bar{x}_0, x_1^1\bar{x}_0^1) \in R$. Therefore, we obtain

$$P_B(x', y) = \langle \hat{x}^0\bar{x}_1\bar{x}_0, \hat{x}^1x_1^1\bar{x}_0^1, x^1, L\rangle.$$

It follows that

$$(\hat{x}^0\bar{x}_1\bar{x}_0, y) \in \chi[u, v]$$

and $\tilde{P}_B(\hat{x}^0\bar{x}_1\bar{x}_0, y) = \tilde{P}_B(x, y)$, i.e., $(\hat{x}^0\bar{x}_1\bar{x}_0, y) \cong (x, y)$. Therefore, $(\hat{x}^0x'_1x'_0, y) \cong (x, y)$ for all x'_1, x'_0.

Since $\rho_0(x, y) = 0$ implies $\rho_0(x, y') = 1$, there are three possibilities: $\bar{x}_1x_0 \overset{d.-p.}{\sim} y'_1y'_0$, $x_1\bar{x}_0 \overset{d.-p.}{\sim} y'_1y'_0$, or $\bar{x}_1\bar{x}_0 \overset{d.-p.}{\sim} y'_1y'_0$. For $\bar{x}_1x_0 \overset{d.-p.}{\sim} y'_1y'_0$ or $x_1\bar{x}_0 \overset{d.-p.}{\sim} y'_1y'_0$, the vertex u is not on $P_B(x, y')$ since we can first change the zeroth double bit and $u_1u_0 \overset{d.-p.}{\not\sim} y'_1y'_0$. For $\bar{x}_1\bar{x}_0 \overset{d.-p.}{\sim} y'_1y'_0$, since (x, x^1) is a *dim-p* edge for $p \geq 2$, it follows that $x_1^1\bar{x}_0^1 \overset{d.-p.}{\sim} y'_1y'_0$. Then $P_B(x, y')$ is obtained by traversing from x to x^1 and then changing the zeroth double bit of x^1. Therefore, $P_B(x, y')$ will not route through $[u, v]$, i.e., $(x, y') \notin \chi(u, v)$ for all $y' \neq y$.

Since $\rho_0(x', u) = 1$ and $\rho_0(u, y') = 1$ for $x' \neq x$ and $y' \neq y$, it follows that $P_B(x', y')$ will not route through vertex u and therefore, $(x', y') \notin \chi(u, v)$ for all $x' \neq x$ and $y' \neq y$. Hence, the equivalence class in $\chi[u, v]$ containing (x, y) has exactly four elements $(\hat{x}^0x_1x_0, y)$, where $x_1, x_0 = 0, 1$.

We note that (x, y)-pairs discussed in Case 1 are different from those in Case 2 since $\tilde{P}_B(x, y)$ in Case 1 is not equal to $\tilde{P}_B(x, y)$ in Case 2. Therefore, these two cases define different equivalence classes. We can show, using similar arguments, that $\chi(u, v)$ containing (x, y) with $(p_{k-2}(v), p_{k-2}(u)) = [\hat{x}^i, \hat{x}^{i+1}]$ for $i \geq 0$ defines different equivalence classes from others. Thus, each equivalence class in $\chi(u, v)$ contains exactly four elements, and the lemma is proven. □

ACKNOWLEDGMENTS

This work was supported in part by the National Science Council of the Republic of China under contract NSC86-2213-E001-026.

REFERENCES

[1] K. Efe, "The Crossed Cube Architecture for Parallel Computing," *IEEE Trans. Parallel and Distributed Systems*, vol. 3, no. 5, pp. 513-524, Sept. 1992.

[2] K. Efe, "A Variation on the Hypercube with lower Diameter," *IEEE Trans. Computers*, vol. 40, no. 11, pp. 1,312-1,316, Nov. 1991.

[3] K. Efe, P.K. Blackwell, W. Slough, and T. Shiau, "Topological Properties of the Crossed Cube Architecture," *Parallel Computing* vol. 20, pp. 1,763-1,775, 1994.

[4] C.M. Fiduccia and P.J. Hedrick, "Edge Congestion of Shortest Path Systems for All-to-All Communication," *IEEE Trans. Parallel and Distributed Systems*, vol. 8, no. 10, pp. 1,043-1,054, Oct. 1997.

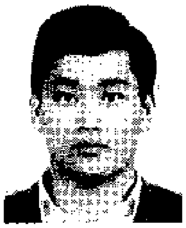
[5] D.F. Hsu, "On Container Width and Length in Graphs, Groups, and Networks," *IEICE Trans. Fundamentals*, vol. E77-A, no. 4, pp. 668-680, 1994.

[6] P. Kulasinghe, "Connectivity of the Crossed Cube," *Information Processing Letters*, vol. 61, pp. 221-226, Feb. 1997.

[7] P. Kulasinghe and S. Bettayeb, "Embedding Binary Trees into Crossed Cubes," *IEEE Trans. Computers*, vol. 44, no. 7, pp. 923-929, July 1995.

[8] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. San Mateo, Calif.: Morgan Kaufmann, 1992.

[9] S.Q. Zheng and S. Latifi, "Optimal Simulation of Linear Multiprocessor Architectures on Multiply-Twisted Cube Using Generalized Gray Code," *IEEE Trans. Parallel and Distributed Systems*, vol. 7, no. 6, pp. 612-619, June 1996.



Chien-Ping Chang received a BS degree in electrical engineering from Chung-Cheng Institute of Technology in 1986, and a PhD degree in computer and information science from National Chiao Tung University, Taiwan, Republic of China, in 1998. He is currently a senior engineer at Chung Shan Institute of Science and Technology, Taiwan, Republic of China. His research interests include interconnection networks, graph algorithms, and medical images.

Ting-Yi Sung received a BS degree in management science from National Chiao Tung University, Taiwan, Republic of China, in 1980, and a PhD degree in operations research from New York University in 1989. She is currently an associate research fellow at the Institute of Information Science, Academia Sinica, Taiwan, Republic of China. Her research interests include fault tolerance and architectures for interconnection networks, graph algorithms, and mathematical programming. She is a member of the IEEE.

Lih-Hsing Hsu received his BS degree in mathematics from Chung Yuan Christian University, Taiwan, Republic of China, in 1975, and his PhD degree in mathematics from the State University of New York at Stony Brook in 1981. He is currently a professor in the Department of Computer and Information Science, National Chiao Tung University, Taiwan, Republic of China. His research interests include interconnection networks, algorithms, graph theory, and VLSI layout.