The Journal of Supercomputing (2019) 75:3621–3639
https://doi.org/10.1007/s11227-018-2712-z

The divide-and-swap cube: a new hypercube variant with small network cost

Jong-Seok Kim[1] · Donghyun Kim[2] · Ke Qiu[3] · Hyeong-Ok Lee[4]

Published online: 7 December 2018

Abstract

The hypercube is one of the most popular interconnection networks. Its network cost is $O(n^2)$. In this paper, we propose a new hypercube variant, the divide-and-swap cube $DSC(n)$ ($n = 2^d$, $d \geq 1$), which reduces the network cost to $O(n \log n)$ while maintaining the same number of nodes and the same asymptotic performances for fundamental algorithms such as the broadcasting. The new network has nice hierarchical properties. We first show that the diameter of $DSC(n)$ is lower than or equal to $\frac{5n}{4} - 1$. However, unlike the hypercube of dimension n whose degree is n, the node degree of the network is $\log n + 1$, resulting in a network cost of $O(n \log n)$. We then examine the one-to-all and all-to-all broadcasting times of $DSC(n)$, based on the single-link-available and multiple-link-available models. We also present an upper bound on the bisection width of the $DSC(n)$ and show that $DSC(n)$ is Hamiltonian. Finally, we introduce the folded divide-and-swap cube, $FDSC(n)$, a variant of the $DSC(n)$ and study its many properties including its hierarchical structure, routing algorithm, broadcasting algorithms, bisection width, and its Hamiltonicity. All the broadcasting algorithms presented in this paper are asymptotically optimal.

Keywords Interconnection network · Divide-and-swap cube · Folded divide-and-swap cube · Network cost · Diameter · Routing · Broadcasting · Bisection width · Hamiltonian cycle

✉ Hyeong-Ok Lee
oklee@scnu.ac.kr; oklee@sunchon.ac.kr

1 Department of Mathematics and Physics, North Carolina Central University, Durham, NC 27707, USA

2 Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA

3 Department of Computer Science, Brock University, St. Catharines, ON, Canada

4 Department of Computer Education, Sunchon National University, Sunchon, Jeonnam 57922, South Korea

1 Introduction

Major research areas in supercomputers in massive parallel computing systems include the development of central processing units (CPUs), interconnection networks, and routing algorithms. When building a supercomputer, the interconnection network, which involves connections among tens, thousands or even millions of processors, is extremely important.

An interconnection network is composed of a set of processors, each with its own local memories, and communication links for data transmission between processors. It can be modeled as an undirected graph $G = (V, E)$ where $V(G)$ and $E(G)$ are the set of nodes and the set of edges of graph G, respectively. Each processor p_i is an element of $V(G)$, and $E(G)$ contains all the node-pairs (p_i, p_j) if nodes p_i and p_j are connected directly by a communication link. That is, each processor of an interconnection network is represented as a node of G and a communication link between two processors is represented as an edge. The parameters for measuring the efficiency of interconnection networks are the degree, diameter, fault-tolerance, bisection width, and the broadcasting time [2]. The degree is related to hardware costs required for node connection and the diameter is related to transmission times of messages between nodes. There is a trade-off between degree and diameter; accordingly, the network cost, which is defined as degree $\times$ diameter, is typically used as an evaluation parameter for interconnection networks [13,14,19,29]. Numerous diverse interconnection networks for parallel processing systems exist. In [3–5], reconfigurable interconnection networks were introduced as an application.

The hypercube is a popular interconnection network topology, with several notable properties, including regularity, symmetry, simple routing, strong connectivity, and recursive structure. A number of variants of the hypercube have been proposed, including folded hypercubes [15,36], twisted cubes [1,7], crossed cubes [14,16], Möbius cubes [9], augmented cubes [8], shuffle cubes [23], locally twisted cubes [33], spined cubes [35], exchanged hypercube [25], and exchanged crossed cube [24]. The hierarchical cubic network (HCN) [18,34] and hierarchical folded-hypercube network (HFN) [12], which are based on a hierarchical structure using the hypercube as a basic module, have also been proposed. The network cost of the hypercube, including all existing variations, is $O(n^2)$.

In this paper, we propose a new hypercube variant, the divide-and-swap cube $DSC(n)$ $(n = 2^d, d \geq 1)$, which reduces the network cost to $O(n \log n)$. This is achieved by reducing its degree to $\log n + 1$ while retaining the same number of nodes. We also show that the network has nice hierarchical properties. We examine the network cost of $DSC(n)$ by analyzing its diameter using a simple routing algorithm. In addition, we develop and analyze the one-to-all and all-to-all broadcasting algorithms based on the single-link-available (SLA) and multiple-link-available (MLA) models. We also present an upper bound on the network's bisection width and show that the network is Hamiltonian. Finally, we introduce the folded divide-and-swap cube, $FDSC(n)$, a variant of the $DSC(n)$ and study its many properties including its hierarchical structure, routing algorithm, broadcasting algorithms, bisection width, and its Hamiltonicity.

This paper is organized as follows. Section 2 presents the divide-and-swap cube together with its general properties. Section 3 develops and analyzes a simple routing algorithm and the diameter of the $DSC(n)$. Section 4 describes the one-to-all and all-to-all broadcastings in $DSC(n)$ based on the SLA and MLA models. Section 5 studies the bisection width and Hamiltonicity of the network while a variant of the $DSC(n)$, folded divide-and-swap network, is proposed and its properties and algorithms studied in Sect. 6. Section 7 summarizes and concludes the paper.

2 Divide-and-swap cube *DSC*(*n*)

$DSC(n)$ $(n = 2^d,\ d \geq 1)$ is defined as an n-dimensional binary cube where the nodes are all binary n-tuples. Each node is represented by a binary n-bit string, $s_1 s_2 s_3 \ldots s_i \ldots s_{n-1} s_n$ $(s_i \in \{0, 1\},\ 1 \leq i \leq n)$, and the edge that connects two arbitrary nodes u and w is represented by (u, w). The divide-and-swap cube can be defined by

$$DSC(n) = (V(DSC(n)), E(DSC(n))),$$

where $V(DSC(n))$ and $E(DSC(n))$ are the set of nodes and the set of edges, respectively. In $DSC(n)$, edges are defined by Definition 1.

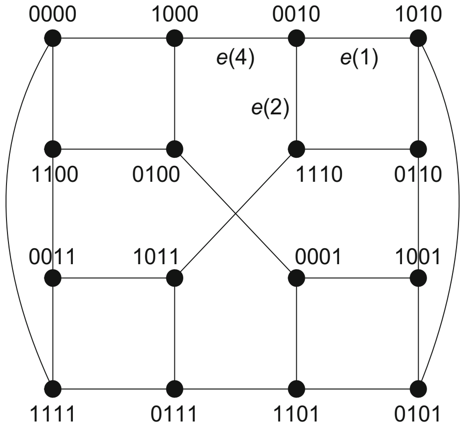
Definition 1 For any n, where $n = 2^d$, $d \geq 1$, let two arbitrary nodes of $DSC(n)$ be $u(= s_1 s_2 s_3 \ldots s_n = t_1 t_2 t_3)$ and w, where $t_1 = s_1 s_2 \ldots s_{\frac{n}{2^k}}$, $t_2 = s_{\frac{n}{2^k}+1} s_{\frac{n}{2^k}+2} \ldots s_{\frac{n}{2^{k-1}}}$, and $t_3 = s_{\frac{n}{2^{k-1}}+1} s_{\frac{n}{2^{k-1}}+2} \cdots s_n$ $(1 \leq k \leq \log n$; if $k = 1$, then $t_3 = \{\})$. If the node w satisfies one of the following conditions, the node w is adjacent to node u:

- **Condition 1** $w = \overline{s_1} s_2 \ldots s_n$ where $\overline{f}$ indicates the complement of f. This type of edge is denoted as an $e(1)$-edge.
- **Condition 2** If $t_1 = t_2$, then $w = \overline{t_1 t_2} t_3$. If $t_1 \neq t_2$, then $w = t_2 t_1 t_3$. This type of edge is denoted as an $e(\frac{2n}{2^k})$-edge.

Figure 1 shows $DSC(4)$; if node $u = 1101$, then $k = 1$ and 2 $(1 \leq k \leq \log n)$. Here, the node u has one adjacent node by Condition 1 and two adjacent nodes by Condition 2. From Condition 1, $w = 0101$. From Condition 2, if $k = 2$, then $t_1 = s_1 = 1$, $t_2 = s_2 = 1$, and $t_3 = 01$, because $\frac{n}{2^k} = 1$. Therefore, $w = 0001$. If $k = 1$, then $t_1 = s_1 s_2 = 11$, $t_2 = s_3 s_4 = 01$, and $t_3 = \{\}$, because $\frac{n}{2^k} = 2$. Hence, $w = 0111$. Consequently, in $DSC(4)$, nodes u $(= 1101)$ and w $(= 0101)$ are connected via an $e(1)$-edge, nodes u $(= 1101)$ and w $(= 0001)$ via an $e(2)$-edge, and nodes u $(= 1101)$ and w $(= 0111)$ through an $e(4)$-edge. From Definition 1, we can see that $DSC(n)$ is a regular graph with degree $\log n + 1 = d + 1$, where $n = 2^d$. By the definition of $DSC(n)$, $DSC(n)$ has these properties.

Property 1 1. The number of nodes in $DSC(n)$ is 2^n.
2. The number of $e(1)$-edges = the number of $e(2)$-edges $= \cdots =$ the number of $e(n)$-edges $= 2^{n-1}$.
3. The number of edges in $DSC(n)$ is $\frac{1}{2} \times$ the number of nodes $\times$ degree $= 2^{n-1}(\log n + 1)$.

Fig. 1 $DSC(4)$



4. The girth of $DSC(n)$, i.e., the length of a shortest cycle in the graph, is 4.

$DSC(n)$ has a hierarchical structure. This will be shown by Lemma 1.

Lemma 1 *$DSC(n)$ is decomposed into $2^{\frac{n}{2}} DSC(\frac{n}{2})$'s.*

Proof We refer to each of $DSC(\frac{n}{2})$ in $DSC(n)$ as a module. Let us assume that an arbitrary node of $DSC(n)$ is $u = s_1 s_2 \ldots s_{\frac{n}{2}} s_{\frac{n}{2}+1} s_{\frac{n}{2}+2} \ldots s_n = AB$ ($A = s_1 s_2 \ldots s_{\frac{n}{2}}$ and $B = s_{\frac{n}{2}+1} s_{\frac{n}{2}+2} \ldots s_n$). The address of a node in a module is denoted by AB, where A represents the address of the node inside the module and B is the address of the module. The number of nodes having the bit string of A is $2^{\frac{n}{2}}$, and the number of modules having the bit string of B is also $2^{\frac{n}{2}}$. For example, $DSC(4)$ consists of 16 nodes and each node is connected to others via an $e(1)$-, $e(2)$-, or $e(4)$-edge. If we group the nodes of $DSC(4)$ with the same module address B, the nodes can be classified into four groups: 00, 01, 10, 11. That is, $DSC(4)$ consists of four modules, and the address of each node is also one of 00, 01, 10, 11. Nodes 00 and 10, and nodes 11 and 01 are connected via an $e(1)$-edge, and nodes 00 and 11, and nodes 10 and 01 are connected through an $e(2)$-edge. The edges (00, 10), (01, 11), (10, 01), (11, 00) that connect to the node addresses of each module 00, 01, 10, 11 and the nodes of the modules in $DSC(4)$ are the same as those (00, 10), (01, 11), (10, 01), (11, 00) that connect to the node addresses 00, 01, 10, 11 of $DSC(2)$ and the nodes of $DSC(2)$ where $DSC(2)$ is simply a 2-cube. Accordingly, each module B of $DSC(4)$ is $DSC(2)$. Clearly, we can generalize the above decomposition easily. Therefore, there exist $2^{\frac{n}{2}}$ modules in $DSC(n)$ and $2^{\frac{n}{2}}$ nodes in each module. That is, $DSC(n)$ consists of $DSC(\frac{n}{2})$ modules with a hierarchical structure. (Figure 1 shows an example of DSC(4) with a hierarchical structure.) □

Lemma 2 *Let the address of each module of $DSC(n)$ be $B_1, B_2, \ldots, B_i, \ldots, B_j, \ldots, B_{2^{\frac{n}{2}}}$ ($1 \leq i \neq j \leq 2^{\frac{n}{2}}$). If $B_i = \overline{B}_j$ at the address of a module, then the modules B_i and B_j are connected by two edges. Otherwise ($B_i \neq \overline{B}_j$), B_i and B_j are connected by one edge.*

Proof Let the address of each node that contains the address of module B_i be

$$A_1B_i, \ldots, A_kB_i, \ldots, A_{2^{\frac{n}{2}}}B_i \left(1 \leq k \neq 1 \leq 2^{\frac{n}{2}}, A_1 = \overline{A_k}\right).$$

The number of such B_i's is the same as the number of A_kB_i, namely $2^{\frac{n}{2}}$, from Lemma 1. By the definition of $DSC(n)$, there exists one node where $B_i = A_k$ in each module and each node of module B_i is connected to nodes $B_iA_1, B_iA_2, \ldots, B_iA_k, \ldots,$ and $B_iA_{2}^{\frac{n}{2}}$, respectively. Hence, module B_i is connected to all the other modules by at least one edge. Node $B_iB_i(= A_kB_i)$ where $B_i = A_k$ is connected to node $B_jB_j(= \overline{B_iB_i})$, and node A_1B_i is connected to node B_iA_1. In addition, since $A_1 = \overline{A_k}$, we have $A_1 = \overline{A_k} = \overline{B_i} = B_j$. In other words, nodes A_1B_i and B_iB_j are connected to each other. Hence, modules $B_i(= \overline{B_i})$ and B_j are connected by two edges. □

Corollary 1 *Let the module $DSC(\frac{n}{2})$ of $DSC(n)$ be a super node of $DSC(n)$. If we represent $DSC(n)$ with $2^{\frac{n}{2}}$ super nodes, $DSC(n)$ is a complete graph.*

3 A simple routing algorithm and the diameter of $DSC(n)$

Here, we present and analyze a simple routing algorithm and examine the diameter of $DSC(n)$. First, let us observe the routing techniques of $DSC(4)$. We assume a starting node $u = s_1s_2s_3s_4$, a destination node $w = s'_1s'_2s'_3s'_4$, and that $u \bigoplus w = r = r_1r_2r_3r_4$, where the symbol $\bigoplus$ denotes the XOR operator. Let us divide the nodes of $DSC(4)$ into four groups: $G_1 = \{0000, 1010, 0101, 1111\}$, $G_2 = \{1100, 0110, 1001, 0011\}$, $G_3 = \{1000, 0010, 1101, 0111\}$, and $G_4 = \{0100, 1110, 0001, 1011\}$. If $u \in G_1$, then the routing from u to w is performed by the routing process shown in Fig. 2a. When $u \in G_2$, $u \in G_3$, or $u \in G_4$, the routing is processed as shown in Fig. 2b–d, respectively. Each bit string in Fig. 2 is denoted by the symbol r.

Consider a starting node $u = s_1s_2 \ldots s_n = a_1a_2 \ldots a_i \ldots a_{\frac{n}{4}}$ and a destination node $w = s'_1s'_2 \ldots s'_n = b_1b_2 \ldots b_j \ldots b_{\frac{n}{4}}$, where $a_1 = s_1s_2s_3s_4$, $a_2 = s_5s_6s_7s_8, \ldots,$ $a_{\frac{n}{4}} = s_{n-3}s_{n-2}s_{n-1}s_n$ and $b_1 = s'_1s'_2s'_3s'_4$, $b_2 = s'_5s'_6s'_7s'_8, \ldots, b_{\frac{n}{4}} = s'_{n-3}s'_{n-2}s'_{n-1}s'_n$. To perform routing in $DSC(n)$, we convert the bit string a_i to the bit string b_j where $j = \frac{n}{4} - i + 1$. The string conversion of the bit string a_i to the bit string b_j is represented as $a_i \Longrightarrow b_j$. The method for converting a partial bit string of u to a partial bit string of w is as follows: $a_1 \Longrightarrow b_{\frac{n}{4}}$, $a_2 \Longrightarrow b_{\frac{n}{4}-1}, \ldots, a_i \Longrightarrow b_j, \ldots, a_{\frac{n}{4}} \Longrightarrow b_1$. The conversion $a_i \Longrightarrow b_j$ indicates the routing of $DSC(4)$, because this is the conversion between nodes consisting of four bits. The swapping of bit strings $s_1s_2 \ldots s_h \ldots s_p$ and $s_{p+1}s_{p+2} \ldots s_{p+h} \ldots s_{2p}$ is represented by $swap(s_p, s_{2p})$ $(1 \leq h \leq p)$. The routing paths between nodes, and the distance of each node in $DSC(4)$, are evident in Fig. 2. The maximum distance between any two nodes in the set {1000, 0100, 0010, 1110, 1011, 0111, 0001, 1101} is 3 and the maximum distance between any two nodes in the set {0000, 1100, 1010, 0110, 0011, 1111, 1001, 0101} is 4. That is, in $DSC(4)$, the maximum distance from a_i to b_j by the conversion $a_i \Longrightarrow b_j$ is 4.

Algorithm 1 shows the simple routing algorithm of $DSC(n)$. The equation for converting from decimal y to binary $y_zy_{z-1} \ldots y_m \ldots y_1$ is represented by $y \xrightarrow{2}$

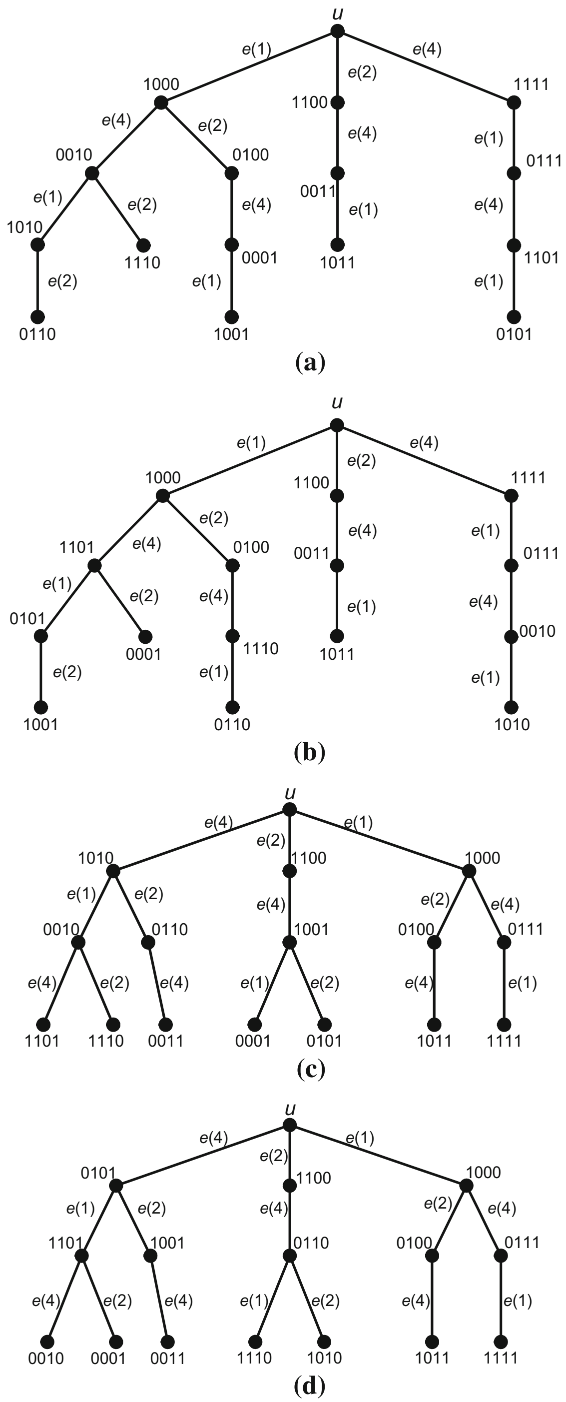


Fig. 2 Routing paths in $DSC(4)$. **a** Routing from u to w when $u \in G_1$ in $DSC(4)$. **b** Routing from u to w when $u \in G_2$ in $DSC(4)$. **c** Routing from u to w when $u \in G_3$ in $DSC(4)$. **d** Routing from u to w when $u \in G_4$ in $DSC(4)$

Algorithm 1 Simple routing algorithm (u, w, y)

Note that $u = s_1 s_2 \ldots s_n = a_1 a_2 \ldots a_i \ldots a_{\frac{n}{4}}$, $w = s'_1 s'_2 \ldots s'_n = b_1 b_2 \ldots b_j \ldots b_{\frac{n}{4}}$, $y = 0$.
repeat
  $y \leftarrow y + 1$
  $j \leftarrow \frac{n}{4} - y + 1$
  $a_1 \Longrightarrow b_j$
  $u \leftarrow b_j a_2 \ldots a_i \ldots a_{\frac{n}{4}}$
  if $u = w$ **then**
    Break; // quit this repeat-until loop
  else
    $y \xrightarrow{2} y_z y_{z-1} \ldots y_m \ldots y_1$
    $x \leftarrow min\{m | y_m = 1, 1 \le m \le z\}$
    $p \leftarrow 2^{x+1}$
    $swap(s_p, s_{2p})$ // swap s_p with s_{2p}.
    $u' \leftarrow s_{2p} s_p u''$
    $u \leftarrow u'$
  end if
until $u \ne w$

$y_z y_{z-1} \ldots y_m \ldots y_1$. x denotes the minimum value of m, where $y_m = 1$ $(1 \le m \le z)$ and u'' is the bit string obtained by removing bits s_p and s_{2p} from the bit string of u.

To route from u to w, the number of times that the conversion $a_1 \Longrightarrow b_j$ is performed is $\frac{n}{4}$, and the number of times that $swap(s_p, s_{2p})$ is performed is $\frac{n}{4} - 1$. Thus, the maximum routing distance from u to w (i.e., the diameter of $DSC(n)$) is $4 \times \frac{n}{4} + \frac{n}{4} - 1 = n + \frac{n}{4} - 1 = \frac{5n}{4} - 1$. The diameter of $DSC(2)$ is 2.

Theorem 1 *The diameter of $DSC(n)$ is $\le \frac{5n}{4} - 1$ $(n = 2^d, d \ge 2)$.*

We conjecture that this upper bound is tight, namely the diameter of $DSC(n)$ is $\frac{5n}{4} - 1$, $(n = 2^d, d \ge 2)$.

4 Broadcasting on *DSC*(*n*)

Broadcasting is a basic data communication procedure for interconnection networks and refers to message transmission between the nodes in a network [21,26]. Messages are generally transmitted in two ways: either one-to-all or all-to-all broadcasting. In one-to-all broadcasting, messages are disseminated from a source node to all other nodes in the network; in all-to-all broadcasting, messages are disseminated from all nodes to all other nodes in the network simultaneously. In addition, communication is usually accomplished in one of two ways: single-port communication, where a source node transmits a message to only one adjacent node in one unit of time, and all-port communication, where a source node transmits a message to all adjacent nodes in one unit of time [10,27]. The former is termed the single-link-available (SLA) model, and the latter the multiple-link-available (MLA) model.

For any network G with N nodes, there exist several trivial lower bounds for broadcasting algorithms. For example, the diameter of G is clearly a lower bound. Another trivial lower bound is $\Omega(\log N)$ for the SLA model since the number of

Algorithm 2 **One-to-all broadcasting algorithm of** $DSC(n)$ **using the SLA model**

1: $d \leftarrow 3$
2: **for** $d \leq n$ **do**
3: u_0 of the module $DSC(\frac{2^d}{2})$ with message M performs broadcasting to all the other nodes in the module.
4: The message M is transmitted from every node in the module to nodes that are connected by an $e(2^d)$-edge.
5: Perform Step 3 in each module.
6: $d \leftarrow d + 1$
7: **end for**

nodes with the message after each broadcasting step can be at most doubled. For $DSC(n)$, both bounds are $\Omega(n)$ since the diameter of $DSC(n)$ is $O(n)$ and $N = 2^n$.

4.1 One-to-all broadcasting for $DSC(n)$ using the SLA and MLA models

Let the message to be transmitted be M, the source node be u_0, the module to which the source node belongs be T, and all nodes in $DSC(n)$ except u_0 be $u_1, u_2, \ldots, u_{2^n-1}$. We use $c \rightarrow d$ to represent a message transmission from node c to node d.
Let us assume that an arbitrary node of the $DSC(n)$ is $u = s_1 s_2 \ldots s_{\frac{n}{2}} s_{\frac{n}{2}+1} s_{\frac{n}{2}+2} \ldots s_n = AB$ ($A = s_1 s_2 \ldots s_{\frac{n}{2}}$, $B = s_{\frac{n}{2}+1} s_{\frac{n}{2}+2} \ldots s_n$). If $A = B$, then the node u does not perform Step 2. The one-to-all broadcasting on $DSC(4)$ using the SLA model is as follows:

- Step 1: The source node u_0 of module T with message M sends the message to all other nodes in module T, which is a $DSC(2)$,
  - Step 1-1: u_0 transmits the message to its neighbor using an $e(1)$-edge: $u_0 \rightarrow u_1$.
  - Step 1-2: Each node with the message M transmits the message to its neighbor using an $e(2)$-edge: $u_0 \rightarrow u_2$, $u_1 \rightarrow u_3$.
- Step 2: Each node with the message M transmits the message to its neighbor using an $e(4)$-edge: $u_0 \rightarrow u_4$, $u_1 \rightarrow u_5$, $u_2 \rightarrow u_6$.
- Step 3: Perform Step 1 in each module.
  - Step 3-1: Each node with the message M transmits the message to its neighbor using an $e(1)$-edge: $u_4 \rightarrow u_7$, $u_5 \rightarrow u_8$, $u_6 \rightarrow u_9$.
  - Step 3-2: Each node with the message M transmits the message to its neighbor using an $e(2)$-edge: $u_4 \rightarrow u_{10}$, $u_5 \rightarrow u_{11}$, $u_6 \rightarrow u_{12}$, $u_7 \rightarrow u_{13}$, $u_8 \rightarrow u_{14}$, $u_9 \rightarrow u_{15}$.

From Lemma 2, we can see that each module of $DSC(n)$ is connected to all the other modules in $DSC(n)$. Algorithm 2 shows the one-to-all broadcasting algorithm on $DSC(n)$ using the SLA model.

It can be seen that $DSC(n)$ has $2^{\frac{n}{2}}$ $DSC(\frac{n}{2})$'s, $DSC(\frac{n}{2})$ has $2^{\frac{n}{4}}$ $DSC(\frac{n}{4})$'s, $DSC(\frac{n}{4})$ has $2^{\frac{n}{8}}$ $DSC(\frac{n}{8})$'s, …, and $DSC(8)$ has 2^4 $DSC(4)$'s, respectively, from Lemma 1. Therefore, the broadcasting time for the one-to-all broadcasting on $DSC(n)$ using the

SLA model is as follows, where $T_o(n)$ is the broadcasting time for $DSC(n)$ using the SLA model:

$T_o(4) = 5,$
$T_o(8) = 5 + 1 + 5 = 11,$
$T_o(16) = 11 + 1 + 11 = 23,$
..., and in general, the broadcasting time on $DSC(n)$ is
$T_o(n) = n + \frac{n}{2} - 1 = \frac{3n}{2} - 1$, which can be proved easily by induction.

The broadcasting time for the one-to-all broadcasting of $DSC(2)$ using the SLA model is 2.

The one-to-all broadcasting algorithm for $DSC(n)$ using the MLA model is as follows:

- Step 1: Each node that has message M transmits the message to all of its neighbors.
- Step 2: Perform Step 1 until all nodes in $DSC(n)$ receive the message M.

As the diameter of $DSC(n)$ is $\leq \frac{5n}{4} - 1$, the broadcasting time of $DSC(n)$ using the MLA model is $\leq \frac{5n}{4} - 1$. The broadcasting time for the one-to-all broadcasting of $DSC(2)$ using the MLA model is 2.

Theorem 2 *The one-to-all broadcasting time of $DSC(n)$ using the SLA model is lower or equal than $\frac{3n}{2} - 1$, and the one-to-all broadcasting time of $DSC(n)$ using the MLA model is lower or equal than $\frac{5n}{4} - 1$ ($n = 2^d$, $d \geq 2$).*

4.2 All-to-All Broadcasting for *DSC*(*n*) using the SLA and MLA Models

In this section, we denote the address of a node inside each module by a binary number and the address of a module by a decimal. We use $c \rightarrow d$ to represent a message transmission from node c to node d. $DSC(n)$ consists of $2^{\frac{n}{2}}$ modules with $2^{\frac{n}{2}}$ nodes in each module, from Lemma 1, and each module of $DSC(n)$ is connected to all the other modules in $DSC(n)$ from Lemma 2. Consequently, $DSC(4)$ consist of four $DSC(2)$ modules, the address of each module is {0, 1, 2, 3}, and the address of each node inside each module is {00, 01, 10, 11}.
All-to-all broadcasting in $DSC(4)$ using the SLA model is as follows:

- Step 1: Nodes 00 and 01 inside each module transmit message M using the edges $e(1)$, $e(2)$, and $e(1)$ in order and nodes 10 and 11 inside each module transmit message M using the edges $e(2)$, $e(1)$, and $e(2)$ in order.
- Step 2: Perform message transmission between each module using an $e(4)$-edge as follows; each message transmission is performed in parallel.
  - Step 2-1: $0 \rightarrow \{1, 2, 3\}$, $1 \rightarrow \{2, 3\}$, $2 \rightarrow 3$.
  - Step 2-2: $3 \rightarrow \{0, 1, 2\}$, $2 \rightarrow \{0, 1\}$, $1 \rightarrow 0$.
- Step 3: Repeat Step 1.

The all-to-all broadcasting time for $DSC(4)$ using the SLA model is 3 (by Step 1) + 2 (by Step 2) + 3 (by Step 3) = 8. Figures 3 and 4 show Step 2-1 and Step 2-2, respectively.

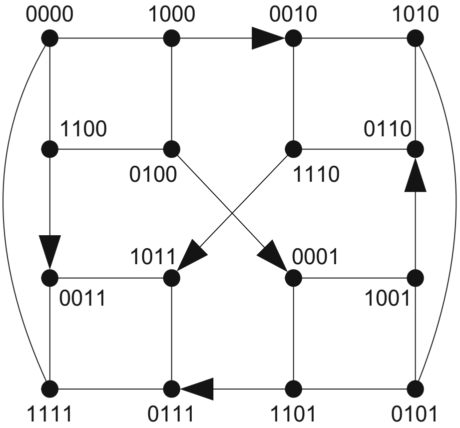


Fig. 3 Message transmission in Step 2-1

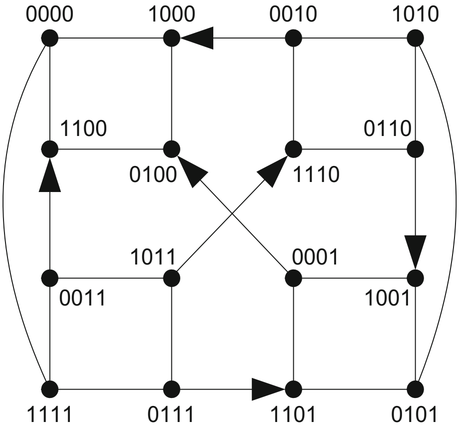


Fig. 4 Message transmission in Step 2-2

Algorithm 3 All-to-all broadcasting algorithm of $DSC(n)$ using the SLA model

$d \leftarrow 3$
for $d \le n$ **do**
  Step 1: Each node inside each module transmits message M using the broadcasting technique of $DSC(\frac{2^d}{2})$.
  Step 2: Message transmission is performed between each module using an $e(2^d)$-edge as follows, and each message transmission is performed in parallel.
  Step 2-1: $0 \rightarrow \{1, 2, \ldots, 2^{2^d-1}\}, 1 \rightarrow \{2, 3, \ldots, 2^{2^d-1}\}, \ldots, 2^{2^d-3} \rightarrow \{2^{2^d-2}, 2^{2^d-1}\}, 2^{2^d-2} \rightarrow 2^{2^d-1}$.
  Step 2-2: $2^{2^d-1} \rightarrow \{0, 1, \ldots, 2^{2^d-2}\}, 2^{2^d-2} \rightarrow \{0, 1, \ldots, 2^{2^d-3}\}, \ldots, 2 \rightarrow \{0, 1\}, 1 \rightarrow 0$.
  Step 3: Repeat Step 1.
  $d \leftarrow d + 1$
end for

Algorithm 3 shows the all-to-all broadcasting algorithm on $DSC(n)$ using the SLA model.

The broadcasting time for all-to-all broadcasting in $DSC(2)$ using the SLA model is 3. The broadcasting time for all-to-all broadcasting in $DSC(n)$ using the SLA model is as follows, where $T_a(n)$ is the broadcasting time for $DSC(n)$ using the SLA model:

$T_a(4) = 8$,
$T_a(8) = 8 + 2 + 8 = 18$,
$T_a(16) = 18 + 2 + 18 = 38$,
$\ldots$, and in general, the broadcasting time on $DSC(n)$ is $T_a(n) = \frac{5n}{2} - 2$, which can be proved easily by induction.

All-to-all broadcasting algorithm in $DSC(n)$ using the MLA model is as follows:

- Step 1: Each node of $DSC(n)$ transmits message to all adjacent nodes.
- Step 2: Repeat Step 1 until each node in $DSC(n)$ has received the messages from all nodes of $DSC(n)$.

Since the diameter of $DSC(n)$ is $\leq \frac{5n}{4} - 1$, the broadcasting time for the all-to-all broadcasting of $DSC(n)$ using the MLA model is $\leq \frac{5n}{4} - 1$. The broadcasting time for all-to-all broadcasting in $DSC(2)$ using the MLA model is 2.

Theorem 3 *The all-to-all broadcasting time of $DSC(n)$ using the SLA model is lower or equal than $\frac{5n}{2} - 2$, and the all-to-all broadcasting time of $DSC(n)$ using the MLA model is lower or equal than $\frac{5n}{4} - 1$.*

In view of the lower bounds, all algorithms in this section are asymptotically optimal.

5 Bisection width and Hamiltonian cycle

One of the metrics for evaluating interconnection networks is the bisection width. The bisection width is the minimum number of edges that need to be removed to segregate one connected network into two networks [6,22,28,31]. These two segregated networks would have the same number of nodes or have one node difference with each other. A smaller bisection width is less desirable than a larger one, because a small number of edge faults can disconnect the network. It is an important parameter for processor communications [6]. It is also related to edge congestion in routing algorithms [20]. The decision version of the problem of finding the bisection width is known to be NP-complete [11,17]. Now, we show the upper bound of the bisection width of $DSC(n)$ is 2^{n-2}.

Theorem 4 *The upper bound of the bisection width of $DSC(n)$ is 2^{n-2}.*

Proof $DSC(n)$ is decomposed into $2^{\frac{n}{2}} DSC(\frac{n}{2})$'s by Lemma 1. We refer to each of $DSC(\frac{n}{2})$ in $DSC(n)$ as a module. Let us assume that an arbitrary node of $DSC(n)$ is $u = s_1 s_2 \ldots s_{\frac{n}{2}} s_{\frac{n}{2}+1} s_{\frac{n}{2}+2} \ldots s_n = AB$ ($A = s_1 s_2 \ldots s_{\frac{n}{2}}$ and $B = s_{\frac{n}{2}+1} s_{\frac{n}{2}+2} \ldots s_n$). The address of a node in a module is denoted by AB, where A represents the address of the node inside the module and B is the address of the module. Let the modules in

$DSC(n)$ be $B_1, B_2, \ldots, B_i, \ldots, B_j, \ldots, B_2^{\frac{n}{2}}$ $(1 \leq i, j \leq 2^{\frac{n}{2}}, i \neq j)$ and two modules B_i and B_j $(B_i = \bar{B_j})$ be a pair, then there are $2^{\frac{n}{2}-1}$ pairs in $DSC(n)$. Let $2^{\frac{n}{2}-2}$ pairs be a group, then there are two groups, α and β, in $DSC(n)$. By the definition of $DSC(n)$, the number of nodes in α is the same as that of β. If all edges connecting α and β are removed, the two groups are divided into two graphs, α and β. The edges connecting α and β are $e(n)$-edges and the number of the edges connecting α and β in each module is $2^{\frac{n}{2}-1}$. Therefore the upper bound of the bisection width of $DSC(n)$ is the number of the edges in each module $\times$ the number of modules $\times \frac{1}{2} = 2^{n-2}$. □

We conjecture that this upper bound is tight, namely, the bisection width of $DSC(n)$ is 2^{n-2}.

A Hamiltonian path in network is a path that passes all the nodes within the network only once. A Hamiltonian cycle is a cycle that traverses all the nodes precisely once [30,32]. If there is a Hamiltonian path or Hamiltonian cycle in an interconnection network, the network can be a pipeline that makes parallel processing easy because it becomes a linear array or ring.

Definition 2 A graph G is called a *complete Hamiltonian graph* if there exists a Hamiltonian path between any two nodes u and v from G.

Lemma 3 $DSC(4)$ *is a complete Hamiltonian graph.*

Proof When $u = 0000$, the Hamiltonian paths from u to all other nodes are as follows.

(1) 0000-1100-0100-1000-0010-1010-0110-1110-1011-0011-1111-0111-1101-0101-1001-0001
(2) 0000-1000-0100-1100-0011-1111-0111-1011-1110-0110-1001-0001-1101-0101-1010-0010
(3) 0000-1111-0111-1011-1110-0110-1001-0001-1101-0101-1010-0010-1000-0100-1100-0011
(4) 0000-1100-0011-1111-0111-1011-1110-0110-1001-0001-1101-0101-1010-0010-1000-0100
(5) 0000-1100-0100-1000-0010-1010-0110-1110-1011-0011-1111-0111-1101-0001-1001-0101
(6) 0000-1100-0100-1000-0010-1110-1011-0011-1111-0111-1101-0001-1001-0101-1010-0110
(7) 0000-1100-0100-1000-0010-1010-0101-1101-0001-1001-0110-1110-1011-0011-1111-0111
(8) 0000-1100-0100-0001-1001-0101-1101-0111-1111-0011-1011-1110-0110-1010-0010-1000
(9) 0000-1000-0010-1010-0110-1110-1011-0111-1111-0011-1100-0100-0001-1101-0101-1001
(10) 0000-1000-0010-1110-1011-0111-1111-0011-1100-0100-0001-1101-0101-1001-0110-1010
(11) 0000-1100-0100-1000-0010-1110-0110-1010-0101-1001-0001-1101-0111-1111-0011-1011
(12) 0000-1000-0100-0001-1101-0101-1001-0110-1010-0010-1110-1011-0111-1111-0011-1100

(13) 0000-1000-0010-1010-0110-1110-1011-0111-1111-0011-1100-0100-0001-1001-0101-1101
(14) 0000-1000-0010-1010-0110-1001-0101-1101-0001-0100-1100-0011-1111-0111-1011-1110
(15) 0000-1100-0100-1000-0010-1110-0110-1010-0101-1001-0001-1101-0111-1011-0011-1111

Similar to the above 15 paths, there is a Hamiltonian path between every two nodes u, v in $DSC(4)$. So $DSC(4)$ is a complete Hamiltonian graph. □

We prove $DSC(n)$ is a complete Hamiltonian graph in the next theorem where $\rightarrow$ is a path between two neighbors, and $\Longrightarrow$ a path between arbitrary two nodes in $DSC(n)$.

Theorem 5 *$DSC(n)$ is a complete Hamiltonian graph.*

Proof We prove it by induction on n. We proved $DSC(4)$ is a complete Hamiltonian graph in Lemma 3. So we prove it for $n \geq 8$. We suppose that there is a Hamiltonian path between arbitrary two nodes in $DSC(k)$ when $k < n$. Let two arbitrary nodes be AB, CD in $DSC(n)$. A, B, C, D are bit strings of length $\frac{n}{2}$, respectively. Then there are two possible cases to consider.

1. $B \neq D$: We can think the next bit string progression has the condition, if $i \neq j$, then $S_i \neq S_j$.
   $B = S_1, S_2, S_3, \ldots, S_{M-1}, S_M = D,\ S_2 \neq A,\ S_{M-1} \neq C,\ M = 2^{\frac{n}{2}}$.
   By the induction, there is a Hamiltonian path $T \Longrightarrow R$ between arbitrary two nodes T and R. So the next path is a Hamiltonian path.
   $AB = S_0S_1 \Longrightarrow S_2S_1 \rightarrow S_1S_2 \Longrightarrow S_3S_2 \rightarrow S_2S_3 \Longrightarrow \cdots \Longrightarrow S_MS_{M-1} \rightarrow S_{M-1}S_M \Longrightarrow S_{M+1}S_M = CD$.
2. $B = D$: By the induction hypothesis, there is a Hamiltonian path or cycle in $DSC(\frac{n}{2})$, $M = 2^{\frac{n}{2}}$.
   $A = T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow \cdots \rightarrow T_{M-1} \rightarrow T_M = C (A \neq C)$ or
   $A = T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow \cdots \rightarrow T_{M-1} \rightarrow T_M \rightarrow T_{M+1} = C (A = C)$.
   Then we can find the next bit string progression has the condition, if $i \neq j$, then $S_i \neq S_j$ $(1 \leq i_0 \leq M$ when $B \neq T_{i_0}, B \neq T_{i_0+1})$.
   $T_{i_0} = S_1, S_2, \ldots, S_{M-1} = T_{i_0+1}, S_j \neq B (1 \leq j \leq M-1)$.
   So the next path is a Hamiltonian path in $DSC(n)$.
   $AB \Longrightarrow T_{i_0}B \rightarrow BT_{i_0} = BS_1 \Longrightarrow S_2S_1 \rightarrow S_1S_2 \Longrightarrow S_3S_2 \rightarrow \cdots \rightarrow S_{M-2}S_{M-1} \Longrightarrow BS_{M-1} \rightarrow S_{M-1}B = T_{i_0+1}D \Longrightarrow CD$.

This completes the proof. □

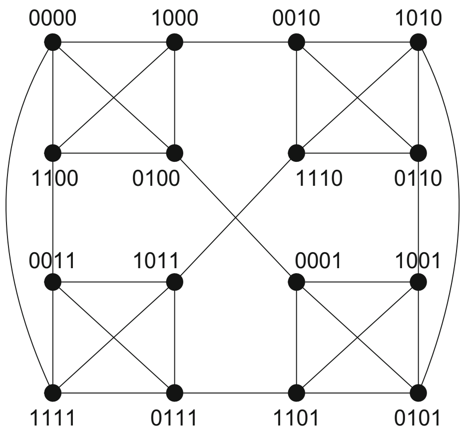
When AB is a neighbor node of CD in Theorem 5, we can get the next corollary.

Corollary 2 *$DSC(n)$ has a Hamiltonian cycle.*

6 Folded divide-and-swap cube *FDSC*(*n*)

For the hypercube, one of its variants is the folded hypercube [15]. Similarly, we introduce a variant of the DSC, the folded divide-and-swap cube, in this section.

Fig. 5 *FDSC*(4)



Therefore, the folded DSC is also a variant of the original hypercube. The folded divide-and-swap cube $FDSC(n)$ ($n = 2^d$, $d \geq 1$) is obtained by adding an edge to each node of a $DSC(n)$ to improve its diameter. We first provide the definition of $FDSC(n)$ and then analyze its various properties.

6.1 Definition of *FDSC*(*n*)

The formal definition of $FDSC(n)$ is as follows.

Definition 2 $V(FDSC(n)) = V(DSC(n))$, $E(FDSC(n)) = E(DSC(n)) \cup e$, where $e = \{(u, w) | u = s_1 s_2 s_3 \ldots s_n, w = s_1 \overline{s_2} \ldots s_n\}$.

An edge in the set of edges e is denoted as an $e(f)$-edge. Figure 5 shows $FDSC(4)$. From Definitions 1 and 2, we can see that $FDSC(n)$ is a regular graph with degree $\log n + 2 = d + 2$, where $n = 2^d$. By the definition of $FDSC(n)$, $FDSC(n)$ has these properties.

Property 2 1. The number of nodes in $FDSC(n)$ is 2^n.
2. The number of $e(1)$-edges = the number of $e(2)$-edges $= \cdots =$ the number of $e(n)$-edges = the number of $e(f)$-edges = 2^{n-1}.
3. The number of edges in $FDSC(n)$ is $\frac{1}{2}\times$ the number of nodes $\times$ degree = $2^{n-1}(\log n + 2)$.
4. The minimum length of a cycle in $FDSC(n)$ is 3.

By Lemmas 1, 2, and the definition of $FDSC(n)$, we can get the following.

Lemma 4 *$FDSC(n)$ is decomposed into $2^{\frac{n}{2}}$ $FDSC(\frac{n}{2})$'s.*

Lemma 5 *Let the address of each module of $FDSC(n)$ be $B_1, B_2, \ldots, B_i, \ldots, B_j, \ldots, B_{2^{\frac{n}{2}}}$ $(1 \leq i \neq j \leq 2^{\frac{n}{2}})$. If $B_i = \overline{B_j}$ at the address of a module, then the modules B_i and B_j are connected by two edges. Otherwise ($B_i \neq \overline{B_j}$), B_i and B_j are connected by one edge.*

Corollary 3 *Let the module $FDSC(\frac{n}{2})$ of $FDSC(n)$ be a super node of $FDSC(n)$. If we represent $FDSC(n)$ with $2^{\frac{n}{2}}$ super nodes, $FDSC(n)$ is a complete graph.*

By Lemma 4, $FDSC(n)$ has a hierarchical structure.

6.2 A simple routing algorithm and the diameter of *FDSC*(*n*)

Now we suggest a simple routing algorithm, and examine the diameter of $FDSC(n)$. First, let us observe the routing techniques of $DSC(4)$. The routing techniques for $FDSC(4)$ are similar to those for $DSC(4)$. We assume a starting node $u = s_1s_2s_3s_4$, a destination node $w = s'_1s'_2s'_3s'_4$, and that $u \bigoplus w = r = r_1r_2r_3r_4$, where the symbol $\bigoplus$ denotes the XOR operator. Let us divide the nodes of $FDSC(4)$ into four groups: $FG_1 = \{0000, 1010, 0101, 1111\}$, $FG_2 = \{1100, 0110, 1001, 0011\}$, $FG_3 = \{1000, 0010, 1101, 0111\}$, and $FG_4 = \{0100, 1110, 0001, 1011\}$. For cases when $u \in FG_1$, $u \in FG_2$, $u \in FG_3$, and $u \in FG_4$, the routing is processed as shown in Fig. 6a–d, respectively. Each bit string in Fig. 6 is denoted by the symbol r. By Fig. 6, the maximum distance between any two nodes in $FDSC(4)$ is 3. That is, the diameter of $FDSC(4)$ is 3.

The simple routing algorithm on $FDSC(n)$ is the same as that on $DSC(n)$. To route from u to w, the number of times that the conversion $a_1 \Longrightarrow b_j$ is performed is $\frac{n}{4}$, and the number of times that $swap(s_p, s_{2p})$ is performed is $\frac{n}{4} - 1$. Thus, the maximum routing distance from u to w (i.e., the diameter of $DSC(n)$) is $3 \times \frac{n}{4} + \frac{n}{4} - 1 = \frac{3n}{4} + \frac{n}{4} - 1 = n - 1$. The diameter of $FDSC(2)$ is 1.

Theorem 6 *The diameter of $FDSC(n)$ is lower or equal than $n - 1$ ($n = 2^d, d \geq 2$).*

Tables 1 and 2 show a comparison among $DSC(n)$, $FDSC(n)$ and the hypercube (including its variants thereof) on the number of nodes, degree, diameter, and network cost. The diameters of the hypercube variants, although smaller, are asymptotically the same as those of $DSC(n)$ and $FDSC(n)$; however, the degree and the network cost of $DSC(n)$ and $FDSC(n)$ are much smaller than those of the hypercube and its variants.

The broadcasting algorithms for $FDSC(n)$ are the same as those for $DSC(n)$. So we have the next theorem.

Theorem 7 *The one-to-all broadcasting time of $DSC(n)$ using the SLA and MLA model is $\frac{3n}{2} - 1$ and $n - 1$, respectively. And the all-to-all broadcasting time of $DSC(n)$ using the SLA and MLA model is $\frac{5n}{2} - 2$ and $n - 1$, respectively.*

Clearly, all these algorithms are asymptotically optimal. Now, we have the following theorem.

Theorem 8 *The upper bound of the bisection width of $FDSC(n)$ is 2^{n-2}.*

Proof This theorem can be easily proven by Theorem 4, Corollary 2 and the definition of $FDSC(n)$. □

Corollary 4 *$FDSC(n)$ has a Hamiltonian cycle.*

Similar to the $DSC(n)$, we conjecture that our bounds for both the diameter and the bisection width of the $FDSC(n)$ are tight.

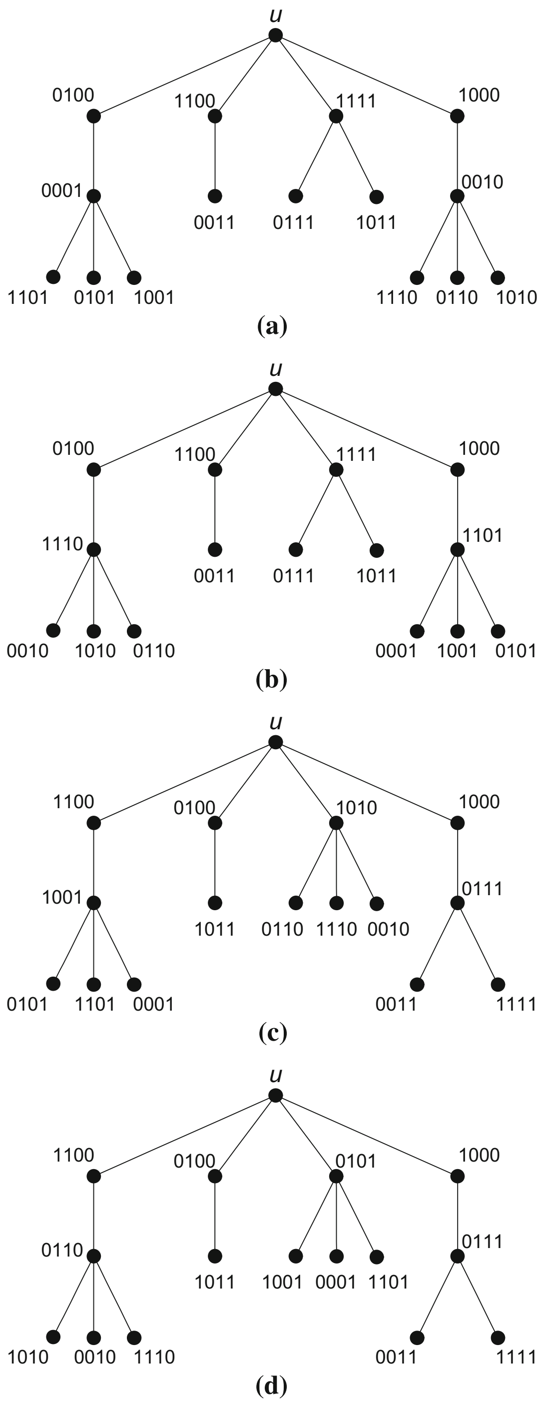


Fig. 6 Routing paths in *FDSC*(4). **a** Routing from u to w when $u \in G_1$ in *FDSC*(4). **b** Routing from u to w when $u \in G_2$ in *FDSC*(4). **c** Routing from u to w when $u \in G_3$ in *FDSC*(4). **d** Routing from u to w when $u \in G_4$ in *FDSC*(4)

Table 1 Comparison of the number of nodes, degree, diameter, and network cost among $DSC(n)$, $FDSC(n)$ and other hypercube variants ($n = s + t + 1$)

Interconnection network	Number of nodes	Degree	Diameter	Network cost
Hypercubes	2^n	n	n	$O(n^2)$
Folded hypercubes	2^n	$n+1$	$\lceil \frac{n}{2} \rceil$	$O(n^2)$
Twisted cubes	2^n	n	$\lceil \frac{n+1}{2} \rceil$	$O(n^2)$
Crossed cubes	2^n	n	$\lceil \frac{n+1}{2} \rceil$	$O(n^2)$
Möbius cubes	2^n	n	$\lceil \frac{n+1}{2} \rceil$ or $\lceil \frac{n+2}{2} \rceil$	$O(n^2)$
Augmented cubes	2^n	$2n-1$	$\lceil \frac{n}{2} \rceil$	$O(n^2)$
Locally twisted cubes	2^n	n	$\lceil \frac{n+3}{2} \rceil$	$O(n^2)$
Shuffle cubes	2^n	n	$\lceil \frac{n}{4} \rceil + 3$	$O(n^2)$
Spined cubes	2^n	n	$\lceil \frac{n}{3} \rceil + 3$	$O(n^2)$
Exchanged hypercubes	2^n	$s+1$ or $t+1$	$s+t+1$	$O(n^2)$
Exchanged crossed cubes	2^n	$s+1$ or $t+1$	$\lceil \frac{s+1}{2} \rceil + \lceil \frac{t+1}{2} \rceil + 2$	$O(n^2)$
$DSC(n)$	2^n	$\log n + 1$	$\frac{5n}{4} - 1$	$O(n \log n)$
$FDSC(n)$	2^n	$\log n + 2$	$n-1$	$O(n \log n)$

Table 2 Comparison of the number of nodes, degree, diameter, and network cost among $DSC(n)$, $FDSC(n)$ and other hierarchical hypercube variants

Interconnection network	Number of nodes	Degree	Diameter	Network cost
$HCN(n)$	2^{2n}	$n+1$	$n + \lfloor \frac{n+1}{3} \rfloor + 1$	$O(n^2)$
$HFN(n)$	2^{2n}	$n+2$	$2\lceil \frac{n}{2} \rceil + 1$	$O(n^2)$
$DSC(n)$	2^{2n}	$\log n + 2$	$\frac{5n}{2} - 1$	$O(n \log n)$
$FDSC(n)$	2^{2n}	$\log n + 3$	$2n-1$	$O(n \log n)$

7 Conclusion

We have proposed a new hypercube variation, the $DSC(n)$, which reduces the network cost compared with the hypercube (and its variations thereof) from $O(n^2)$ to $O(n \log n)$, while retaining the same number of nodes. This is achieved by generating edges whereby part of the address of the nodes is exchanged, thus reducing the degree of the network to $\log n + 1$ (where the network cost is given by the degree times the diameter). We also studied the network's properties and algorithms. Specifically, we have shown that the newly proposed network has good hierarchical properties and we have described a simple routing algorithm for $DSC(n)$ and proved that the diameter of $DSC(n)$ is no greater than $\leq \frac{5n}{4} - 1$, where $n = 2^d$ and $d \geq 2$. Furthermore, we developed one-to-all broadcasting algorithms on $DSC(n)$, whose running times are at most $\frac{3n}{2} - 1$ and $\frac{5n}{4} - 1$ based on the SLA and MLA models, respectively, and the all-to-all broadcasting algorithms with running times at most $\frac{5n}{2} - 2$ and $\frac{5n}{4} - 1$ using the SLA and MLA models, respectively. In addition, we presented an upper bound on the bisection width of the network and showed that the divide-and-swap network is Hamiltonian. A variant of the divide-and-swap network, the folded divide-and-swap network is also proposed and some of its properties and algorithms are studied and presented. All broadcasting algorithms presented are asymptotically optimal. These results demonstrate that $DSC(n)$ is a suitable interconnection network for implementation in large-scale multi-computer systems. As for the future work, there remain several open questions. For example, are our bounds on the diameters and bisection widths of both $DSC(n)$ and $FDSC(n)$ tight.

Acknowledgements This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2017R1D1A3B03032173). We are grateful to the anonymous referees for their helpful comments and suggestions.

References

1. Abraham S, Padmanabhan K (1991) The twisted cube topology for multiprocessors: a study in network asymmetry. J Parallel Distrib Comput 13(1):104–110
2. Akers SB, Harel D, Krishnamurthy B (1989) A group-theoretic model for symmetric interconnection network. IEEE Trans Comput 38(4):555–565
3. Arabnia HR, Oliver MA (1987) A transputer network for the arbitrary rotation of digitised images. Comput J 30(5):425–433
4. Arabnia HR, Smith JW (1993) A reconfigurable interconnection network for imaging operations and its implementation using a multi-stage switching box. In: Proceedings of the 7th Annual International High Performance Computing Conference. The 1993 High Performance Computing: New Horizons Supercomputing Symposium, Calgary, Alberta, Canada, pp 349–357
5. Arabnia HR, Taha TR (1998) A parallel numerical algorithm on a reconfigurable multi-ring network. J Telecommun Syst 10:185–203
6. Bornstein CF, Litman A, Maggs BM, Sitaraman RK, Yatzkar T (2001) On the bisection width and expansion of butterfly networks. Theory Comput Syst 34:491–518
7. Chang CP, Wang JN, Hsu LH (1999) Topological properties of twisted cube. Inf Sci 113(1–2):147–167
8. Choudum SA, Sunitha V (2002) Augmented cubes. Networks 40(2):71–84
9. Cull P, Larson SM (1995) The Möbius cubes. IEEE Trans Comput 44(5):647–659
10. De Azevedo MM, Bagherzadeh N (1995) Broadcasting algorithms for the star-connected cycles interconnection network. J Parallel Distrib Comput 25:209–222
11. Díaz J, Serna MJ, Wormald MC (2007) Bounds on the bisection width for random d-regular graphs. Theo Comput Sci 382:120–130
12. Duh DR, Chen GH, Fang JF (1995) Algorithms and properties of a new two-level network with folded hypercubes as basic modules. IEEE Trans Parallel Distrib Syst 6(7):714–723
13. Efe K (1991) A variation on the hypercube with lower diameter. IEEE Trans Comput 40(11):1312–1316
14. Efe K (1992) The crossed cube architecture for parallel computing. IEEE Trans Parallel Distrib Syst 3(5):513–524
15. El-Amawy A, Latifi S (1991) Properties and performance of folded hypercubes. IEEE Trans Parallel Distrib Syst 2(1):31–42
16. Fan J, Jia X (2007) Embedding meshes into crossed cubes. Inf Sci 177(15):3151–3160
17. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco
18. Ghose K, Desai KR (1995) Hierarchical cubic networks. IEEE Trans Parallel Distrib Syst 6(4):427–436
19. Harary F, Hayes JP, Wu HJ (1988) A survey of the theory of hypercube graphs. Comput Math Appl 15(4):277–289
20. Hsu LH, Lin CK (2008) Graph theory and interconnection networks. CRC Press, Boca Raton
21. Johnson SL, Ho CT (1989) Optimal broadcasting and personalized communication in hypercubes. IEEE Trans Comput 38(9):1249–1268

22. Kim JS, Kim SW, Qiu K, Lee HO (2014) Some properties and algorithms for the hyper-torus network. J Supercomput 69(1):121–138
23. Li TK, Tan JJM, Hsu LH, Sung TY (2001) The shuffle-cubes and their generalization. Inf Process Lett 77(1):35–41
24. Li K, Mu Y, Li K, Min G (2013) Exchanged crossed cube: a novel interconnection network for parallel computation. IEEE Trans Parallel Distrib Syst 24(11):2211–2219
25. Loh PKK, Hsu WJ, Pan Y (2005) The exchanged hypercube. IEEE Trans Parallel Distrib Syst 16(9):866–874
26. Mendia VE, Sarkar D (1992) Optimal broadcasting on the star graph. IEEE Trans Parallel Distrib Syst 3(4):389–396
27. Mkwawa IM, Kouvatsos DD (2003) An optimal neighborhood broadcasting scheme for star interconnection networks. J Interconnect Netw 4(1):103–111
28. Monien B, Preis R (2006) Upper bounds on the bisection width of 3- and 4-regular graphs. J Discrete Algorithms 4(3):475–498
29. Parhami B, Kwai DM (2001) A unified formulation of honeycomb and diamond networks. IEEE Trans Parallel Distrib Syst 12(1):74–80
30. Rahman MS, Kaykobad M (2005) On Hamiltonian cycles and Hamiltonian paths. Inf Process Lett 94(1):37–41
31. Tang KW, Kamoua R (2007) An upper bound for the bisection width of a diagonal mesh. IEEE Trans Comput 56(3):429–431
32. Wang D (2001) Embedding Hamiltonian cycles into folded hypercubes with faulty links. J Parallel Distrib Comput 61(4):545–564
33. Yang XF, Evans DJ, Megson GM (2005) The locally twisted cubes. Int J Comput Math 82(4):401–413
34. Yun SK, Park KH (1998) Comments on “hierarchical cubic networks”. IEEE Trans Parallel Distrib Syst 9(4):410–414
35. Zhou W, Fan J, Jia X, Zhang S (2012) The spined cube: a new hypercube variant with smaller diameter. Inf Process Lett 111(12):561–567
36. Zhu Q, Liu SY, Xu M (2008) On conditional diagnosability of the folded hypercubes. Inf Sci 178(4):1069–1077