

Available online at www.sciencedirect.com



Information Sciences 177 (2007) 4200-4207

INFORMATION SCIENCES

www.elsevier.com/locate/ins

Node-disjoint paths in hierarchical hypercube networks $\stackrel{\approx}{\rightarrow}$

Ruei-Yu Wu^{a,c}, Gen-Huey Chen^{a,*}, Yu-Liang Kuo^a, Gerard J. Chang^b

^a Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan ^b Department of Mathematics, National Taiwan University, Taipei 10617, Taiwan

^c Department of Management Information Systems, Hwa Hsia Institute of Technology, Taipei, Taiwan

Received 17 October 2005; received in revised form 15 February 2007; accepted 20 February 2007

Abstract

The hierarchical hypercube network is suitable for massively parallel systems. One of its appealing properties is the low number of connections per processor, which can facilitate the VLSI design and fabrication. Other alluring features include symmetry and logarithmic diameter, which can derive easy and fast algorithms for communication. In this paper, a maximal number of node-disjoint paths are constructed between every two distinct nodes of the hierarchical hypercube network. Their maximal length is not greater than $\max\{2^{m+1} + 2m + 1, 2^{m+1} + m + 4\}$, where 2^{m+1} is the diameter. The effectiveness of node-disjoint paths is further verified by experiments. © 2007 Elsevier Inc. All rights reserved.

Keywords: Container; Hierarchical hypercube network; Hypercube; Interconnection network; Node-disjoint paths

1. Introduction

Advances in hardware technology, especially the VLSI circuit technology, have made it possible to build a large-scale multiprocessor system that contains thousands or even tens of thousands of processors. For example, the Connection Machine [10] contains as many as 2^{16} processors. One crucial step on designing a large-scale multiprocessor system is to determine the topology of the interconnection network (network for short), because the system performance is significantly affected by the network topology. In recent decades, a number of networks were proposed in the literature (see [12,16]). A network is conveniently represented by a graph whose vertices represent the nodes (i.e., processors) of the network and whose edges represent the communication links of the network. Throughout this paper, we use network and graph, node and vertex, and link and edge, interchangeably.

Corresponding author. Tel.: +886 2 23625336; fax: +886 2 23628167.

^{*} This work was supported in part by the National Science Council under grant NSC95-2221-E-002-125-MY3. Taida Institute for Mathematical Sciences, National Taiwan University, Taipei 10617, Taiwan. National Center for Theoretical Sciences.

E-mail addresses: fish@inrg.csie.ntu.edu.tw (R.-Y. Wu), ghchen@csie.ntu.edu.tw (G.-H. Chen), laner@csie.ntu.edu.tw (Y.-L. Kuo), gjchang@math.ntu.edu.tw (G.J. Chang).

Let G = (V, E) be a connected graph, where V and E represent the vertex set and edge set of G, respectively. The *degree* of a vertex in G is the number of edges incident with it. If all vertices have the same degree d, then G is called *regular* or *d*-*regular*. The *distance* between two vertices u and v, denoted by d(u, v), is the length of the shortest path between u and v. The *diameter* of G is the maximal distance between any two vertices of G. The *vertex* (or *node*) *connectivity* of G is the minimal number of vertices in G whose removal can cause G disconnected or trivial.

The hypercube network [24] is one of the most popular networks. We use Q_n to denote an *n*-dimensional hypercube network. There are 2^n nodes contained in a Q_n network; each is uniquely represented by a binary sequence $b_{n-1}b_{n-2} \dots b_0$ of length *n*. Two nodes in a Q_n network are adjacent if and only if they differ at exactly one bit position. An edge of a Q_n network is said of *dimension* k if its two end vertices differ at b_k , where $0 \le k \le n-1$. The hypercube network suffers from a practical limitation: as n increases, it becomes more difficult to design and fabricate the nodes of a Q_n network because of the large fanout.

To remove the limitation, the cube-connected cycles (CCC for short) network [22] was designed as a substitute for the hypercube network. The node degree of a CCC network is restricted to three. However, this restriction degrades the performance of a CCC network at the same time. For example, a CCC network has a greater diameter than a hypercube network having the same number of nodes. Taking both the practical limitation and the performance into account, the hierarchical hypercube (HHC for short) network [19–21] was proposed as a compromise between the hypercube network and the CCC network.

An HHC network, which has a two-level structure, takes hypercubes as basic modules and connects them in a hypercube manner. An HHC network has a logarithmic diameter, which is the same as a hypercube network. Since the topology of an HHC network is closely related to the topology of a hypercube network, it inherits some favorable properties from the latter.

It is practically important to construct node-disjoint paths (disjoint paths for short) in networks, because they can be used to increase the transmission rate and enhance the transmission reliability. Besides, disjoint paths have applications in multipath routing (such as Rabin's information dispersal algorithm [23]), fault tolerance (see [3,8]), and communication protocols (see [12]). Disjoint paths in a variety of networks can be found in the literature [6,9,12,14,24]. Among them, one-to-one disjoint paths are also named *containers*, which we formally define in the next section.

In this paper, a maximal number of disjoint paths whose maximal length is not greater than $\max\{2^{m+1} + 2m + 1, 2^{m+1} + m + 4\}$ are constructed between every two distinct nodes of an HHC network, where 2^{m+1} is the diameter of the HHC network. In the next section, the structure of an HHC network is first reviewed. In Section 3, a maximal number of disjoint paths inside it are constructed. The effectiveness of disjoint paths is further verified by experiments. Finally, this paper concludes with some remarks in Section 4.

2. HHC networks and containers

A CCC network can be obtained by replacing each node of a Q_k network with a cycle of k nodes so that they are connected to the k neighbors of the original node in the Q_k network. Actually, an HHC network is a modification of a CCC network in which the k-node cycle is replaced with a hypercube. Assume $k = 2^m$. An HHC network can be constructed as follows: start with a Q_{2^m} network and then replace each node of it with a Q_m network.

Since there are a total of $2^{2^m} \times 2^m = 2^{2^m+m}$ nodes, each node in the HHC network can be uniquely represented by a binary sequence $b_{n-1}b_{n-2} \dots b_0$, where $n = 2^m + m$. Refer to Fig. 1, where an example with m = 2 is shown. For convenience, $b_{n-1}b_{n-2} \dots b_0$ is expressed as a two-tuple (S, P), where $S = b_{n-1}b_{n-2} \dots b_m$ tells which Q_m network the node is located in and $P = b_{n-1}b_{n-2} \dots b_n$ gives the address of the node in the located Q_n network

work the node is located in and $P = b_{m-1}b_{m-2}...b_0$ gives the address of the node in the located Q_m network. Let $P^{(l)} = b_{m-1}...b_{l+1}\bar{b}_lb_{l-1}...b_0$ ($S^{(m+l)} = b_{n-1}...b_{m+l+1}\bar{b}_{m+l-1}...b_m$) denote the binary sequence obtained by complementing b_l (b_{m+l}) of P (S). An HHC network can be defined in terms of graph as follows.

Definition 2.1. The node set of an *n*-dimensional HHC (*n*-HHC for short) network is $\{(S,P) \mid S = b_{n-1} \\ b_{n-2} \dots b_m, P = b_{m-1}b_{m-2} \dots b_0$, and $b_i \in \{0,1\}$ for all $0 \le i \le n-1\}$, where $n = 2^m + m$ and $m \ge 1$. Node adjacency of an *n*-HHC network is defined as follows: (S,P) is adjacent to (1) $(S,P^{(l)})$ for all $0 \le l \le m-1$ and (2) $(S^{(m+dec(P))}, P)$, where dec(P) is the decimal value of *P*.



Fig. 1. Construction of an HHC network from a Q_{2^2} network.

Edges defined by (1) are referred to as *internal edges*, and those defined by (2) are referred to as *external edges*. Internal edges are within Q_m networks and each of external edges connects two Q_m networks. Note that an *n*-HHC network is (m + 1)-regular, symmetric, and has a diameter of 2^{m+1} (see [20]). In subsequent discussion, whenever a node A of an *n*-HHC network is mentioned, we use A_S and A_P to denote the S part and P part of A, respectively.

In the following, we define containers and Gray codes, which will be used in the next section.

Definition 2.2 [12]. Suppose that A and B are two distinct nodes of a graph G. An (A, B)-container in G is a set of disjoint paths between A and B. The width (length) of a container is the number (maximal length) of the paths it contains.

Definition 2.3 [11]. An *m*-bit Gray code, denoted by G_m , defines an ordering among all the *m*-bit binary numbers. G_1 is defined as (0, 1), and for m > 1, G_m is defined recursively in terms of G_{m-1} as $(0G_{m-1}, 1G_{m-1}^r)$, where G_{m-1}^r stands for the reverse ordering of G_{m-1} and $0G_{m-1}$ $(1G_{m-1}^r)$ stands for prefixing each binary number in G_{m-1} (G_{m-1}^r) with 0 (1).

For example, G_2 is (00,01,11,10) and G_3 is (000,001,011,010,110,111,101,100). Note that every two adjacent binary numbers, including the first one and the last one, in G_m differ in exactly one bit position.

3. A container of width m + 1

Suppose that A and B are two distinct nodes of an *n*-HHC network. According to Menger's theorem [1], there are at most m + 1 disjoint paths between A and B. In this section, a maximal number (i.e., m + 1) of disjoint paths from A to B are constructed. Since an *n*-HHC network is vertex-symmetric, we assume $A = (0^{2^m}, 0^m)$ without losing generality, where $0^{2^m} (0^m)$ represents $2^m (m)$ consecutive 0's. In the rest of this section, we first introduce external edge sequences, which can be used to simplify the path representation. Then, the construction of the container is shown, and an illustrative example is provided.

3.1. External edge sequences

Each path in an *n*-HHC network contains internal edges and external edges alternately. Since each path of the container is desired to be as short as possible, each subpath of it within a Q_m network is maintained shortest.

 \mathbf{v} two distinct nodes of a $\mathbf{0}$ - network. So, if the submaths within

It is easy to obtain a shortest path between any two distinct nodes of a Q_m network. So, if the subpaths within Q_m networks are ignored, then a path in an *n*-HHC network can be simply represented by a sequence of external edges, called an *external edge sequence* (EES).

For example, $A = (0000, 00) \stackrel{*}{\rightarrow} (0000, 11) \rightarrow (1000, 11) \stackrel{*}{\rightarrow} (1000, 10) \rightarrow (1100, 10) \stackrel{*}{\rightarrow} (1100, 01) \rightarrow (1110, 01) = B$ expresses a path from A = (0000, 00) to B = (1110, 01), where $\stackrel{*}{\rightarrow}$ denotes a shortest path within a Q_2 network. The path contains three external edges that can be represented by their *P* parts, i.e., 11, 10 and 01 in sequence. Hence, the path can be simply represented by an EES (11, 10, 01). Any path from *A* to *B* contains at least $|B_S|$ external edges, where $|B_S|$ is the number of bits 1 in B_S . An EES is *shortest* if it contains $|B_S|$ external edges. In order to reduce the path length, a particular shortest EES, denoted by π , can be obtained from B_S , as elaborated below.

First, collect the indices *i* of $B_S = b_{2^m+m-1}b_{2^m+m-2} \dots b_m$ with $b_i = 1$. Second, decrease each *i* by *m*, and so $0 \le i \le 2^m - 1$. Third, construct π by arranging all indices *i* (expressed in binary form) into a subsequence of an *m*-bit Gray code (refer to Definition 2.3). That is, π contains only those collected indices minus *m*, arranged in the order as they appear in an *m*-bit Gray code. For example, when m = 3 and $B_S = 10101111$, the set of indices obtained after the second step is $\{0, 1, 2, 3, 5, 7\}$. Considering a 3-bit Gray code (000,001,011,010,110,111,101), we have $\pi = (000,001,011,010,111,101)$ finally. Note that by Definition 2.3, we have an *m*-bit Gray code begins with 0^m .

Suppose that *t* is an *m*-bit binary sequence contained in π . Define π^t to be the shortest EES that is obtained by cyclically shifting π toward the left until the resulting EES begins with *t*. For example, if $\pi = (000, 001, 011, 010, 111, 101)$, then $\pi^{001} = (001, 011, 010, 111, 101, 000)$ and $\pi^{111} = (111, 101, 000, 001, 011, 010)$. For those paths that contain more than $|B_S|$ external edges, their corresponding EESs are not shortest. For example, $A = (0000, 00) \stackrel{*}{\rightarrow} (0000, 11) \rightarrow (1000, 11) \stackrel{*}{\rightarrow} (1000, 00) \rightarrow (1001, 00) \stackrel{*}{\rightarrow} (1001, 11) \rightarrow (0001, 11) = B$ contains $3 > 1 = |B_S|$ external edges, and so its corresponding EES, i.e., (11, 00, 11), is not shortest. It will be clear later that EESs can greatly help the construction of disjoint paths.

3.2. Construction of the container

Now we show the construction of the container. First, we consider a special situation in which A and B are located within the same Q_m network, i.e., $A_S = B_S$. According to Saad and Schultz's work [24], m disjoint paths from A to B can be obtained within the Q_m network. Moreover, these m disjoint paths have maximal length not greater than m + 1. One more disjoint path can be obtained according to the EES $(0^m, B_P, 0^m, B_P)$ whose length is computed as $4 + 3d_H(0^m, B_P) \le 4 + 3m$, where $d_H(0^m, B_P)$ denotes the Hamming distance between 0^m and B_P . In the rest of this section, we assume $A_S \neq B_S$.

Since an *n*-HHC network is (m + 1)-regular, each edge incident with A or B must be included in some disjoint path. First, we compute $\pi = (c_0, c_1, \dots, c_{r-1})$ according to A_S and B_S . One or two disjoint paths that contain the two external edges incident with A or B are constructed below, depending on four cases.

- Case 1. $A_P \in \pi$ and $B_P \in \pi$. We have $c_0 = 0^m$. Assume $c_z = B_P$, where $0 \le z \le r 1$. Two (or one if z = r 1) disjoint paths are constructed according to π^{c_0} and $\pi^{c_{(z+1) \mod r}}$, respectively.
- Case 2. $A_P \notin \pi$ and $B_P \in \pi$. Then, $\tau = (0^m, c_0, c_1, \dots, c_{r-1})$ is a subsequence of an *m*-bit Gray code. Two disjoint paths are constructed according to $(\tau, 0^m)$ and $\pi^{c_{(z+1) \mod r}}$, respectively, where $c_z = B_P$ is assumed.
- Case 3. $A_P \in \pi$ and $B_P \notin \pi$. We have $c_0 = 0^m$. Construct $\theta = (c_0, \ldots, c_{u-1}, B_P, c_u, \ldots, c_{r-1})$, where $0 \le u \le r$, so that it is a subsequence of an *m*-bit Gray code. One disjoint path is constructed according to (θ^{B_P}, B_P) , where θ^{B_P} is obtained by cyclically shifting θ toward the left until the resulting EES begins with B_P . Another disjoint path is constructed according to π^{c_0} .
- Case 4. $A_P \notin \pi$ and $B_P \notin \pi$. One disjoint path is constructed according to $(\tau, 0^m)$, where τ was defined in Case 2. If $B_P \neq 0^m$, another disjoint path is constructed according to (θ^{B_P}, B_P) , where θ^{B_P} was defined in Case 3.

Next, the other m - 1 or m disjoint paths are constructed. We show only the construction for Case 2 (i.e., $A_P \notin \pi$ and $B_P \in \pi$), in which m - 1 additional disjoint paths are required; the construction for the other cases is similar. When $r \ge m$, the other m - 1 disjoint paths can be obtained according to the first m - 1 unused

EESs of $\pi^{c_0}, \pi^{c_1}, \ldots, \pi^{c_{r-1}}$. When r < m, r - 1 disjoint paths are first constructed according to the unused EESs of $\pi^{c_0}, \pi^{c_1}, \ldots, \pi^{c_{r-1}}$. Then, the remaining m - r disjoint paths are constructed by the aid of Rabin's work [23], as explained below.

Arbitrarily select m - r neighboring nodes of A within the same Q_m network so that they were not included in π . Without losing generality, assume that $A_p^{(0)}, A_p^{(1)}, \ldots, A_p^{(m-r-1)}$ are selected. For $0 \le l \le m - r - 1$, let $\rho_l = (c_0, \ldots, c_{w_l-1}, A_p^{(l)}, c_{w_l}, \ldots, c_{r-1})$ be a subsequence of an *m*-bit Gray code, where $0 \le w_l \le r$. The remaining m - r disjoint paths can be obtained according to $(\rho_l^{A_p^{(l)}}, A_p^{(l)})$ s, where $\rho_l^{A_p^{(l)}}$ is the EES derived from cyclically shifting ρ_l toward the left until the resulting EES begins with $A_p^{(l)}$. It was shown in [23] that there are *m* disjoint paths from *A* (or *B*) to $A_p^{(0)}, A_p^{(1)}, \ldots, A_p^{(m-r-1)}, c_0, c_1, \ldots, c_{r-1}$, respectively, whose maximal length is not greater than m + 1.

A total of m + 1 paths were constructed above. They are disjoint because they traversed distinct Q_m networks, exclusive of the first Q_m network and the last Q_m network where A and B reside, respectively. Their lengths are computed as follows. The paths obtained by $\pi^{c_0}, \pi^{c_1}, \ldots, \pi^{c_{r-1}}$ have lengths not greater than $r + (m+1) + (2^m - 1) + (m+1)$, where r is the number of external edges traversed and m + 1 is an upper bound on the number of internal edges traversed in the first (or last) Q_m network. Since there are 2^m binary numbers in an *m*-bit Gray code, the total number of internal edges traversed in the other Q_m networks is not greater than $2^m - 1$.

The path obtained by $(\tau, 0^m)$ has length not greater than $(r+2) + 2^m + (m+1)$, as explained below. There are r+2 external edges traversed. No internal edges were traversed in the first Q_m network and not more than m+1 internal edges were traversed in the last Q_m network. At most 2^m internal edges were traversed in the other Q_m networks. Similarly, the path obtained by $(\rho_l^{A_p}, A_p^{(l)})$ has length not greater than $(r+2) + 1 + 2^m + (m+1)$. To sum up, the m+1 paths above have maximal length equal to $\max\{2^m + 2m + r + 1, 2^m + m + r + 4\}$.

The discussion above is based on Case 2. For the other cases, the path construction and length computation are similar. The constructed m+1 disjoint paths have maximal length equal to $\max\{2^m + 2m + r + 1, 2^m + m + r + 4\}$, $\max\{2^m + 2m + r + 1, 2^m + m + r + 4\}$ and $\max\{2^m + 2m + r + 1, 2^m + m + r + 4\}$, if Case 1, Case 3 and Case 4 are considered, respectively. Consequently, the m+1 disjoint paths from A to B have maximal length equal to $\max\{2^m + 2m + r + 1, 2^m + m + r + 4\} \le \max\{2^{m+1} + 2m + 1, 2^{m+1} + m + 4\}$. Therefore, we have the following theorem.

Theorem 3.1. There exists a container of width m + 1 between any two distinct nodes of an n-HHC network whose length is not greater than $\max\{2^{m+1} + 2m + 1, 2^{m+1} + m + 4\}$, where 2^{m+1} is the diameter.

3.3. An illustrative example

Consider two nodes A = (00000000, 000) and B = (01010100, 101) in an 11-HHC network (i.e., m = 3). Clearly, A and B belong to two different Q_m networks and $\pi = (010, 110, 100)$. The construction of the (A, B)-container falls into Case 4 $(A_P \notin \pi \text{ and } B_P \notin \pi)$. The following two paths are constructed according to $((\tau, 000) =) (000, 010, 110, 100, 000)$ and $((\theta^{101}, 101) =) (101, 100, 010, 110, 101)$, respectively.

 $P_1: A = (00000000, 000) \rightarrow (0000001, 000) \xrightarrow{*} (00000001, 010) \rightarrow (00000101, 010) \xrightarrow{*}$

 $(00000101, 110) \rightarrow (01000101, 110) \xrightarrow{*} (01000101, 100) \rightarrow (01010101, 100) \xrightarrow{*}$

 $(01010101, 000) \rightarrow (01010100, 000) \xrightarrow{*} (01010100, 101) = B.$

 $(01110100, 110) \xrightarrow{*} (01110100, 101) \rightarrow (01010100, 101) = B.$

Since $r = 3 \ge 3 = m$, two more disjoint paths are constructed according to $(\pi^{010} =)(010, 110, 100)$ and $(\pi^{100} =)$ (100, 010, 110), respectively, as shown below.

$$P_{3}: A = (0000000, 000) \stackrel{*}{\rightarrow} (0000000, 010) \rightarrow (00000100, 010) \stackrel{*}{\rightarrow} (00000100, 110) \rightarrow (01000100, 110) \stackrel{*}{\rightarrow} (01000100, 100) \stackrel{*}{\rightarrow} (01010100, 100) \stackrel{*}{\rightarrow} (01010100, 101) = B.$$

$$P_{4}: A = (00000000, 000) \stackrel{*}{\rightarrow} (00000000, 100) \rightarrow (00010000, 100) \stackrel{*}{\rightarrow} (00010000, 010) \rightarrow (00010100, 010) \stackrel{*}{\rightarrow} (01010100, 101) = B.$$

The length of P_1 is computed as $5 + d_H(000, 010) + d_H(010, 110) + d_H(110, 100) + d_H(100, 000) + d_H(000, 101) = 11$, where 5 is the number of external edges traversed. Similarly, P_2 , P_3 and P_4 have lengths equal to 13, 7 and 9, respectively.

3.4. Experimental results

In this section, we compare the performance and cost of the crossbar [25], fat-tree [17] and HHC network, each with 64 processors. Table 1 shows their bisection widths, latencies, numbers of switches, and numbers of ports per switch. The *bisection width* [16] of a network is the minimum number of links whose removal can partition the network into two halves with identical (within one) numbers of processors. The bisection width is often a critical factor in determining the speed with which a network can perform a calculation. This is due to the fact that for many problems, the data contained and/or computed by one half of the network may be needed by the other half before the overall computation can be completed. On the other hand, the latency [15] is the time required for a (destination) node to receive all the messages transmitted from another (source) node. In the experiment of Table 1, the latency was calculated for transmitting 10000 messages between two farthest nodes. Moreover, there were one, two and three node-disjoint paths between the source and the destination nodes of the crossbar, fat-tree and HHC network, respectively. They all were used for message transmission.

Fig. 2 further shows the latencies of the crossbar, fat-tree and HHC network for transmitting 1, 10, 100, 1000 and 10000 messages. All messages were transmitted between two farthest nodes, using the

Table 1 Performance and cost of the crossbar, fat-tree and HHC

Crossbar	Fat-tree	ННС
64	4	8
12501.00	6260.75	4180.75
64×64	60	64
2	6	4
	Crossbar 54 12501.00 64 × 64 2	Crossbar Fat-tree 54 4 12501.00 6260.75 64 × 64 60 2 6



Fig. 2. Construction of an HHC network from a Q_{2^2} network.

store-and-forward strategy. We assumed that it took $0.25 \,\mu\text{s}$ for a message to pass through each switch. The length of each message was assumed to be 20 bytes, and the transmission rate for each link was assumed to be 20 MB/s. For the fat-tree and HHC network, the messages to be transmitted were distributed evenly over the node-disjoint paths. The time required for a node to transmit or receive a message was ignored. As revealed from Fig. 2, the HHC network has lower latency than the crossbar and the fat-tree as the numbers of messages are greater than 100 and 1000, respectively.

4. Discussion and conclusion

The hierarchical hypercube network, which was originally proposed in [19–21] for building massively parallel systems, owns many favorable topological properties. Besides, it can perform one-to-one communication, one-to-all communication and divide-and-conquer algorithms efficiently. In [20,21], it was compared with the hypercube and the CCC networks for three metrics: degree, number of edges, and diameter. It has a low number of connections per processor, which can enhance the feasibility of VLSI design and fabrication. It also has a more compact VLSI layout than the hypercube network, which can reduce the fabrication cost and increase the chip turnout.

In this paper, we constructed a container of maximal width m + 1 in a hierarchical hypercube network with 2^{2^m+m} nodes (i.e., an *n*-HHC network with $n = 2^m + m$). The length of the container, which counts the total number of internal edges and external edges traversed, is greater than the diameter $(=2^{m+1})$ by $\max\{2m+1, m+4\}$ in the worst case. The number of external edges traversed is $r(=|B_S|)$ or r+2, whereas the number of internal edges traversed relies on the EESs used. We estimated the container length for a worst-case scenario, in which $r = 2^m$ and the total number of internal edges traversed is $2^m - 1$. Really, there are $d_{H}(c_i, c_{i+1})$ internal edges traversed for any two adjacent *m*-bit binary sequences c_i, c_{i+1} in an EES.

For the example of Section 3.3, the lengths of P_1 , P_2 , P_3 and P_4 are estimated as $(r+2) + 2^m + (m+1)$, $(r+2) + (m+1) + 2^m$, $r + (m+1) + (2^m - 1) + (m+1)$ and $r + (m+1) + (2^m - 1) + (m+1)$, respectively, whose maximum is max $\{2^m + m + r + 3, 2^m + 2m + r + 1\}$. Since m = 3 and r = 3, the length of the (A, B)-container is estimated as 18, compared with 13, the real length of the (A, B)-container. Further, the estimation of the container length in Theorem 3.1 took $r = 2^m$ with the worst-case consideration.

Recall that the numbers of internal edges traversed in the first Q_m network and the last Q_m network were estimated as m + 1 according to Rabin's work [23]. In fact, when r approaches 2^m , far fewer than m + 1 internal edges are traversed in the two Q_m networks. This overestimation causes that the estimated container length is greater than the diameter by max $\{2m + 1, m + 4\}$ in the worst case. It is a challenging problem how to make a more accurate estimation when r is large.

The $(\kappa - 1)$ -fault diameter of a graph G was defined to be the maximal diameter of G with at most $\kappa - 1$ nodes removed [12], where κ is the node connectivity of G. Theorem 3.1 provides an upper bound of $\max\{2^{m+1} + 2m + 1, 2^{m+1} + m + 4\}$ on the *m*-fault diameter of an *n*-HHC network. A lower bound on the $(\kappa - 1)$ -fault diameter of G can be easily derived by computing the distance between two particular nodes in G with $\kappa - 1$ nodes removed.

Consider two nodes $A = (0^{2^m}, 0^m)$ and $B = (1^{2^m-1}0, 0^m)$ in an *n*-HHC network, and remove the *m* neighboring nodes of *A* within the same Q_m network. The remaining neighboring node of *A* is $(0^{2^m-1}1, 0^m)$. Thus, the distance between *A* and *B* is at least $2(2^m + 1) - 1 = 2^{m+1} + 1$, which is a lower bound on the *m*-fault diameter of an *n*-HHC network. It is an interesting problem how to narrow down the gap between the upper bound and the lower bound of the *m*-fault diameter of an *n*-HHC network.

Finally, before ending this paper, more further research problems on the hierarchical hypercube network are suggested. There are some interesting combinatorial and topological problems, e.g., Rabin number [4,14], wide diameter [2,13], hamiltonicity [7,8,18] and pancyclicity [5,26]. They are still open for the hierarchical hypercube network.

Acknowledgements

We express our sincere gratitude to the anonymous reviewers and Editor-in-Chief for their constructive and valuable comments. They improved the readability of this paper a lot.

References

- [1] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, North-Holland, New York, 1976.
- [2] W.M. Chen, G.H. Chen, D.F. Hsu, Generalized diameters of the mesh of trees, Theory of Computing Systems 37 (4) (2004) 547–556.
- [3] D.R. Duh, G.H. Chen, Topological properties of WK-recursive networks, Journal of Parallel Distributed Computing 23 (3) (1994) 468–474.
- [4] D.R. Duh, G.H. Chen, On the Rabin number problem, Networks 30 (3) (1997) 219-230.
- [5] J. Fan, X. Lin, X. Jia, Node-pancyclicity and edge-pancyclicity of crossed cubes, Information Processing Letters 93 (3) (2005) 133– 138.
- [6] J.S. Fu, G.H. Chen, D.R. Duh, Node-disjoint paths and related problems on hierarchical cubic networks, Networks 40 (3) (2002) 142–154.
- [7] J.S. Fu, Hamiltonicity of the WK-recursive network with and without faulty nodes, IEEE Transactions on Parallel and Distributed Systems 16 (9) (2005) 853–865.
- [8] J.S. Fu, Longest fault-free paths in hypercubes with vertex faults, Information Sciences 176 (7) (2006) 759-771.
- [9] Q.P. Gu, S.T. Peng, An efficient algorithm for the k-pairwise disjoint paths problem in hypercubes, Journal of Parallel Distributed Computing 60 (6) (2000) 764–774.
- [10] W.D. Hillis, The Connection Machine, MIT Press, Cambridge, MA, 1985.
- [11] E. Horowitz, S. Sahni, S. Rajasekaran, Computer Algorithms, Computer Science Press, New York, USA, 1996.
- [12] D.F. Hsu, On container width and length in graphs, groups, and networks, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E77-A (4) (1994) 1450–1466.
- [13] T. Kojima, Wide diameter and minimum length of disjoint menger path systems, Networks 46 (3) (2005) 136-141.
- [14] C.N. Lai, G.H. Chen, Strong Rabin numbers of folded hypercubes, Theoretical Computer Science 341 (1-3) (2005) 196-215.
- [15] B.W. Lampson, Interconnecting computers: architecture technology and economics, in: J. Gutknecht (Ed.), Programming Languages and System Architectures, Lecture Notes in Computer Science, vol. 782, Springer, 1994, pp. 1–20.
- [16] T. Leighton, Introduction to Parallel Algorithms and Architectures: Networks and Algorithms, Morgan-Kaufmann, San Mateo, CA, 1992.
- [17] C.E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, IEEE Transactions on Computers 34 (10) (1985) 892–901.
- [18] T.K. Li, J.J.M. Tan, L.H. Hsu, Hyper hamiltonian laceability on edge fault star graph, Information Sciences 165 (1-2) (2004) 59-71.
- [19] Q.M. Malluhi, M.A. Bayoumi, T.R.N. Rao, On the hierarchical hypercube interconnection network, in: Proceedings of the International Parallel Processing Symposium, 1993, pp. 524–530.
- [20] Q.M. Malluhi, M.A. Bayoumi, The hierarchical hypercube: a new interconnection topology for massively parallel systems, IEEE Transactions on Parallel and Distributed Systems 5 (1) (1994) 17–30.
- [21] Q.M. Malluhi, M.A. Bayoumi, Properties and performance of the hierarchical hypercube, in: Proceedings of the International Parallel Processing Symposium, 1992, pp. 47–50.
- [22] F.P. Preparata, J. Vuillemin, The cube-connected cycles: a versatile network for parallel computation, Communications of the ACM 24 (5) (1981) 300–309.
- [23] M.O. Rabin, Efficient dispersal of information for security, load balancing, and fault tolerance, Journal of ACM 36 (2) (1989) 335– 348.
- [24] Y. Saad, M.H. Schultz, Topological properties of hypercubes, IEEE Transactions on Computers 37 (7) (1988) 867-872.
- [25] Y. Tamir, H.-C. Chi, Symmetric crossbar arbitres for vlsi communication switches, IEEE Transactions on Parallel and Distributed Systems 4 (1) (1993) 13–27.
- [26] M.C. Yang, T.K. Li, J.J.M. Tan, L.H. Hsu, On embedding cycles into faulty twisted cubes, Information Sciences 176 (6) (2006) 676– 690.