Journal of Parallel and Distributed Computing **61**, 838–849 (2001)
doi:10.1006/jpdc.2001.1729, available online at http://www.idealibrary.com on IDEAL®

Routing Properties of a Recursive Interconnection Network

Rocky K. C. Chang[1]

Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

and

Hong Y. Wang

Department of Information Science and Electronics Engineering, Hangzhou and Zhejiang University, Hangzhou, Republic of China

Received March 9, 2000; revised July 27, 2000; accepted November 7, 2000

In this paper, we consider a highly recursive interconnection network known as the fully connected cubic network (FCCN). By exploiting its recursive properties, we thoroughly analyze the performance of a simple routing algorithm for the FCCN. We show that at least 80% of the routes obtained from this simple algorithm are shortest paths, and this percentage increases further with the network size. Subsequently, we obtain the network diameter and average internodal distance, taking into account the communication locality that is exhibited in many parallel computations. The presence of the communication locality significantly reduces the average internodal distance.

Key Words: interconnection networks; recursive topologies; shortest-path routing algorithms; communication locality; cubic networks.

1. INTRODUCTION

Packet routing is a critical component in the design of interconnection networks. An ideal routing algorithm should be very efficient to execute and optimal in terms of finding a shortest path. Indeed, it is well known that simple and shortest-path routing algorithms exist in a class of interconnection networks, notably the hypercube and mesh networks. The hypercube, however, is not a practical candidate for scalable network design, due to its nonconstant nodal degree. Mesh networks, on the other hand, suffer from a relatively large network diameter for a small nodal

[1] To whom correspondence should be addressed.

0743-7315/01 $35.00

degree. As a result, a number of more scalable interconnection networks have been proposed, e.g., [1–6]. These interconnection networks also possess simple routing algorithms that require only the packet's destination address and the processing node's address. However, unlike the hypercube and mesh networks, these simple routing algorithms do not guarantee shortest paths. On the other hand, shortest-path routing algorithms also exist for these interconnection networks, but each node is required to obtain more topological information and the processing of the algorithms is much more involved. Moreover, distributed mechanisms are needed to make sure that the topological information kept in each node is up-to-date.

Because of the complexity involved in implementing shortest-path routing algorithms, simple routing algorithms usually prevail in actual implementation. Nonetheless, there is a lack of analysis and a failure to probe further into important performance issues, such as "how many routes computed by the simple routing algorithms are indeed shortest paths?"; "how does the percentage of shortest paths change with network size?"; and "how much longer are those nonshortest paths when compared with shortest paths?" Our main focus in this paper is to address these issues for a simple routing algorithm in a recently proposed interconnection network known as the fully connected cubic network (FCCN) [7]. Owing to its recursive structure, we are able to obtain recurrence equations and recursive algorithms to describe the routing algorithm's properties. The main result is a criterion under which the routing algorithm guarantees shortest paths. We then apply the results to obtain network diameter and average internodal distance, taking into consideration the communication locality that is exhibited in parallel computations. Our conclusion for the FCCN, after a detailed analysis, is that it is justified to employ the simple routing algorithm although it does not guarantee shortest paths. Numerical results show that no more than 20% of routes obtained from the algorithm are nonshortest paths, and they are longer than the corresponding shortest paths by no more than 30% on average. Furthermore, both percentages decrease with network size. As a result, it does not seem to be worthwhile from a practical point of view to obtain shortest paths just for at most 20% of routes by adding significant complexity to the routing algorithm and to the processing nodes.

The rest of this paper is organized as follows. In Section 2, we briefly review the definition of the FCCN. In Section 3, we describe a simple routing algorithm that uses only a processing node's address and a packet's destination address to make routing decisions. We then address the performance issues of the routing algorithm in Section 4 and the effect of communication locality on the average internodal distance in Section 5.

2. FULLY CONNECTED CUBIC NETWORK

The FCCN is a product of three design goals: using 3-cubes as basic building blocks, maintaining a constant nodal degree, and balancing between nodal degree and diameter. We choose 3-cubes as the building blocks for the FCCN, because they also serve the same purpose for hypercube, which can effectively execute a large class of parallel algorithms. Our approach to achieving the design goals is to

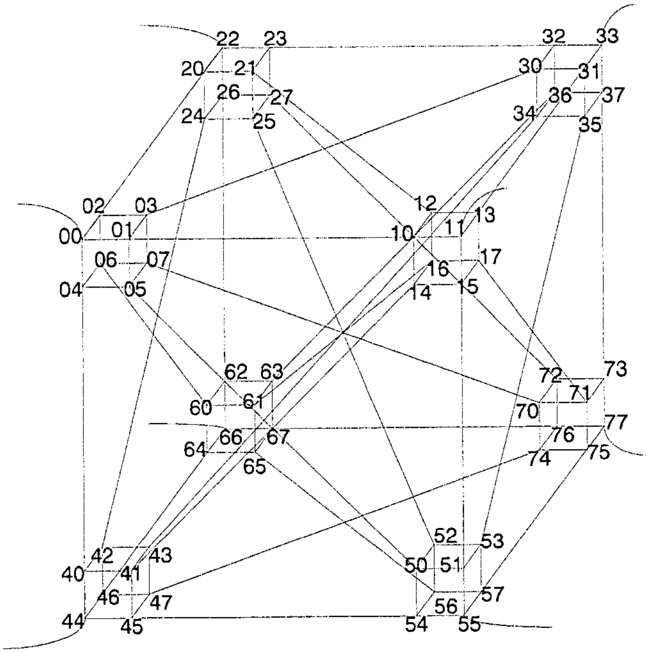


FIG. 1. A 2-FCCN.

take advantage of the smallest diameter offered by a clique while keeping a small and constant nodal degree. Informally, each node of the clique is substituted by a 3-cube, and the eight 3-cubes are fully interconnected according to Fig. 1; the resulting network is referred to as a 2-level FCCN, or just 2-FCCN. A 3-FCCN, shown in Fig. 2, is in turn constructed from eight 2-FCCNs in a similar fashion.

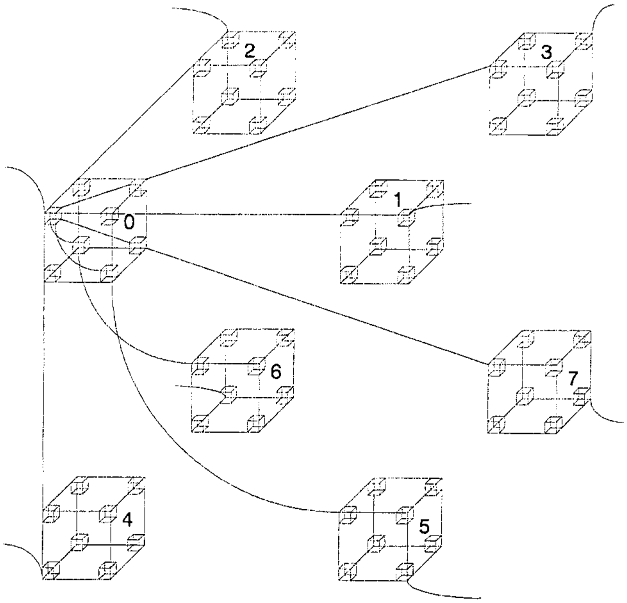


FIG. 2. A 3-FCCN (with partial links).

Generally, an FCCN with m levels (m-FCCN) is constructed by fully connecting eight $(m-1)$-FCCNs in the manner of an 8-node clique.

Each node in an m-FCCN is uniquely identified by a vector of m octal numbers $\boldsymbol{\beta}_m$: $(\beta_m \beta_{m-1} \cdots \beta_1)$. To differentiate octal values from decimal values, we use Greek (English) symbols to refer to octal (decimal) constants and variables that take on octal (decimal) values. Moreover, we use $\boldsymbol{\beta}_{m \cdots k}$, $1 < k \leqslant m$, to refer to $(\beta_m \cdots \beta_k)$, the kth to mth element in $\boldsymbol{\beta}_m$. Note that $\boldsymbol{\beta}_{m \cdots k}$ uniquely identifies the $(k-1)$-FCCN in which the node resides, and β_1 is the node's position on the first level (3-cube).

Definition 2.1. An *unconnected gateway node* (UGN) in an m-FCCN is one with an unused link, and its address is given by $(\upsilon\, \upsilon \cdots \upsilon)$ or (υ^m) in short. Therefore, there are always eight UGNs in an m-FCCN.

Definition 2.2. A kth level *connected gateway node* in an m-FCCN, $1 < k \leqslant m$, denoted by k-CGN, is one whose address is given by $(k\, \upsilon^{m-1})$ for m-CGN, by $(\boldsymbol{\beta}_{m \cdots \kappa+1}\, \kappa\, \upsilon^{k-1})$ *f*or other cases, and $\kappa \neq \upsilon$ for all cases. It is easy to show that a FCCN node is either a UGN or a CGN.

Definition 2.3. Denote an m-FCCN by $F_m \equiv (V_m, E_m)$, where V_m and E_m are the sets of nodes and links, respectively. A 1-FCCN is equivalent to a 3-cube. An m-FCCN, $m > 1$, consists of eight identical $(m-1)$-FCCNs; the κth $(m-1)$-FCCN is denoted by $F_m(\kappa) \equiv (V_m(\kappa), E_m(\kappa))$. Then

$$V_m = \bigcup_{\kappa = 0, \ldots, 7} V_m(\kappa)$$

with the node addresses changed from $\boldsymbol{\beta}_{m-1}$ to $(\kappa\, \boldsymbol{\beta}_{m-1})$ for nodes in $F_m(\kappa)$. Besides the links $\bigcup_{\kappa=0, \ldots, 7} E_m(\kappa)$, a new *intercubic link* is set up between each pair of m-CGNs: $(\upsilon\, \kappa^{m-1})$ and $(\kappa\, \upsilon^{m-1})$, $\upsilon \neq \kappa$.

3. A SIMPLE ROUTING ALGORITHM

In Algorithm 3.1, we present a recursive routing function for a pair of source and destination nodes, denoted by $\boldsymbol{\alpha}_m$: $(\alpha_m\, \alpha_{m-1} \cdots \alpha_1)$ and $\boldsymbol{\beta}_m$: $(\beta_m\, \beta_{m-1} \cdots \beta_1)$, respectively. *Max_level_differ*$(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$ determines the number of the highest level that their addresses differ; that is, *Max_level_differ*$(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) = k$ such that $\alpha_k \neq \beta_k$ and $\alpha_i = \beta_i$ for all $k < i \leqslant m$. The entire routing path consists of a route from the source node to a k-CGN node (labeled A.1), a one-hop route between two k-CGNs (labeled A.2), and a route from a k-CGN node to the destination node (labeled A.3). By recursively expanding *R-Route*() in (A.1) and (A.3), we obtain a nonrecursive version of Algorithm 3.1, presented as Algorithm 3.2, with which a source node or an intermediate node makes routing decisions based on its address and the destination address. Note that the cubic routing in (B.1) and (B.3) incurs 1, 2, or 3 hops.

Algorithm 3.1 (A recursive routing algorithm)

```
        Procedure R-Route($\boldsymbol{\alpha}_m$, $\boldsymbol{\beta}_m$)
        begin
          $k \leftarrow$ Max_level_differ($\boldsymbol{\alpha}_m$, $\boldsymbol{\beta}_m$);
          if $k = 1$, route the packet to the destination using cubic routing;
          else do
(A.1)       R-Route($\boldsymbol{\alpha}_m$, ($\boldsymbol{\alpha}_{m\cdots k+1}\,\alpha_k\,\beta_k^{k-1}$));
(A.2)       R-Route(($\boldsymbol{\alpha}_{m\cdots k+1}\,\alpha_k\,\beta_k^{k-1}$), ($\boldsymbol{\alpha}_{m\cdots k+1}\,\beta_k\,\alpha_k^{k-1}$));
(A.3)       R-Route(($\boldsymbol{\alpha}_{m\cdots k+1}\beta_k\,\alpha_k^{k-1}$), $\boldsymbol{\beta}_m$);
          end;
        end {R-Route}
```

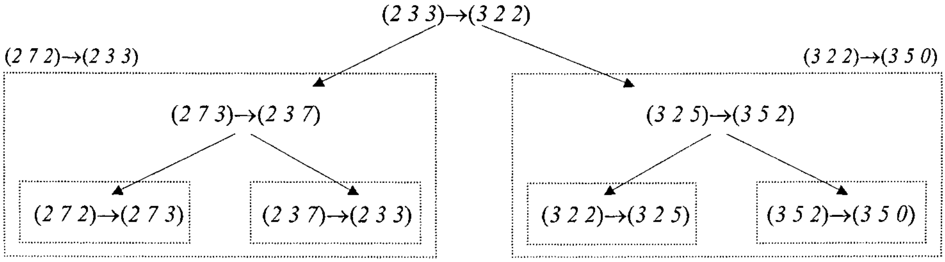
Algorithm 3.2 (A nonrecursive routing algorithm)

```
        Procedure S-Route($\boldsymbol{\alpha}_m$, $\boldsymbol{\beta}_m$)
        begin
          if $\boldsymbol{\alpha}_m = \boldsymbol{\beta}_m$, the packet is received by $\boldsymbol{\beta}_m$;
          else do
            $k \leftarrow$ Max_level_differ($\boldsymbol{\alpha}_m$, $\boldsymbol{\beta}_m$);
(B.1)       if $k = 1$, route the packet from $\boldsymbol{\alpha}_m$ to $\boldsymbol{\beta}_m$ using cubic routing;
            else do
(B.2)         if $\alpha_1 = \beta_k$, route the packet to the $k$-CGN at the other end of the
              intercubic link;
(B.3)         else route the packet to ($\alpha_m\,\alpha_{m-1}\cdots\alpha_2\,\beta_k$) using cubic routing;
            end;
          end;
        end {S-Route}
```

The routing algorithm is very efficient, requiring only two comparison operations and a function call of *Max_level_differ*() at each hop. The computational requirement for *Max_level_differ*() can be drastically reduced by having the source node first execute the function fully, and then use a pointer to remember the returned value of the function. The pointer value is sent together with the data. Therefore, other nodes do not have to execute the function all over again, but they need to decrement the pointer value if the condition for (B.2) is satisfied.

We illustrate the routing algorithm with a simple example below, in which we depict the route sequence in the form of a binary tree. Specifically, the route sequence from the source node to the destination node is given by an inorder traversal of the tree nodes. The interior tree nodes correspond to the one-hop routes between two k-CGNs (B.2). The three leftmost leaf nodes correspond to (B.3), and the rightmost leaf node corresponds to (B.1). Note that the routing pattern involves alternating between intra- (leaf nodes) and intercubic routing (interior nodes). Relating it to Algorithm 3.1, the bigger dashed box on the left subtree corresponds to (A.1) and the one on the right subtree, (A.3). The smaller dash boxes, on the other hand, correspond to the cases of $k = 1$ that end the recursive calls of *R-Route*().

Example: (2 7 2) → (3 5 0)



It is clear that the routing algorithm guarantees that a packet will always be delivered to the destination, but the resulting path is not necessarily a shortest path. A nonshortest route for a 2-FCCN, for example, is given by (0 7) → (3 7), for which the routing algorithm gives a distance of 5; however, the route (0 7) → (7 0) → (7 1) → (7 3) → (3 7), which passes through $F_1(7)$ before reaching $F_1(3)$, is one hop shorter.

Definition 3.1. Denote an *internodal distance matrix* for an m-FCCN, of dimension $8^m \times 8^m$, by $\mathbf{H}_m$. The entry specified by a $\boldsymbol{\alpha}_m$-row and a $\boldsymbol{\beta}_m$-column gives the internodal distance for $\boldsymbol{\alpha}_m \rightarrow \boldsymbol{\beta}_m$. A $\boldsymbol{\alpha}_m$-row vector is denoted by $\mathbf{h}^r_m(\boldsymbol{\alpha}_m) \equiv [c_{\boldsymbol{\alpha}_m,1}\, c_{\boldsymbol{\alpha}_m,2} \cdots c_{\boldsymbol{\alpha}_m,8^m}]$, and a $\boldsymbol{\beta}_m$-column vector is denoted by $\mathbf{h}^c_m(\boldsymbol{\beta}_m) \equiv [d_{1,\boldsymbol{\beta}_m}\, d_{2,\boldsymbol{\beta}_m} \cdots d_{8^m,\boldsymbol{\beta}_m}]^T$. $\mathbf{H}_1$ is the well-known internodal distance matrix for a 3-cube. Another way to write $\mathbf{H}_m$, $m > 1$ is based on its recursive structure

$$\mathbf{H}_m = [\mathbf{H}_{m-1}(\kappa, \upsilon)]_{\kappa, \upsilon = 0, \ldots, 7} = \begin{bmatrix} \mathbf{H}_{m-1}(0,0) & \mathbf{H}_{m-1}(0,1) & \cdots & \mathbf{H}_{m-1}(0,7) \\ \mathbf{H}_{m-1}(1,0) & \mathbf{H}_{m-1}(1,1) & \cdots & \mathbf{H}_{m-1}(1,7) \\ \vdots & \vdots & & \vdots \\ \mathbf{H}_{m-1}(7,0) & \mathbf{H}_{m-1}(7,1) & \cdots & \mathbf{H}_{m-1}(7,7) \end{bmatrix},$$

where $\mathbf{H}_{m-1}(\kappa, \upsilon)$ is an $8^{m-1} \times 8^{m-1}$ dimensional matrix giving internodal distances for source nodes in $F_{m-1}(\kappa)$ and destination nodes in $F_{m-1}(\upsilon)$.

Definition 3.2. Define an operator Θ for two vectors $\mathbf{h}^c_m(\boldsymbol{\beta}_m)$ and $\mathbf{h}^r_m(\boldsymbol{\alpha}_m)$, such that $\mathbf{h}^c_m(\boldsymbol{\beta}_m)\ \Theta\ \mathbf{h}^c_m(\boldsymbol{\alpha}_m) = [z_{i,j}]_{i,j=1,\ldots,8^m}$, a matrix of dimension $8^m \times 8^m$, where $z_{i,j} = d_{i,\boldsymbol{\beta}_m} + c_{\boldsymbol{\alpha}_m,j}$.

Theorem 3.1. *Given* $\mathbf{H}_m$, $m \geqslant 1$, $\mathbf{H}_{m+1} = [\mathbf{H}_m(\kappa, \upsilon)]_{\kappa, \upsilon = 0, \ldots, 7}$, where

$$\mathbf{H}_m(\kappa, \upsilon) = \begin{cases} \mathbf{H}_m & \textit{if} \quad \kappa = \upsilon \\ \mathbf{h}^c_m(\upsilon^m)\ \Theta\ \mathbf{h}^r_m(\kappa^m) + \mathbf{1} & \textit{if} \quad \kappa \neq \upsilon, \end{cases} \tag{3.1}$$

and $\mathbf{1}$ *is an* $8^m \times 8^m$ *dimensional identity matrix.*

Proof. Both source and destination nodes belong to the same m-FCCN for $\kappa = \upsilon$; thus, $\mathbf{H}_m(\kappa, \kappa) = \mathbf{H}_m$. If $\kappa \neq \upsilon$, the route from a source node $\boldsymbol{\alpha}_{m+1} \in F_m(\kappa)$ to a destination node $\boldsymbol{\beta}_{m+1} \in F_m(\upsilon)$, as obtained from Algorithm 3.1, consists of *R-Route*$(\boldsymbol{\alpha}_{m+1}, (\kappa\, \upsilon^m))$, *R-Route*$((\kappa\, \upsilon^m), (\upsilon\, \kappa^m))$, and *R-Route*$((\upsilon\, \kappa^m), \boldsymbol{\beta}_{m+1})$. Since both $\boldsymbol{\alpha}_{m+1} \equiv (\kappa\, \boldsymbol{\alpha}_{m \cdots 1})$ and $(\kappa\, \upsilon^m)$ reside in $F_m(\kappa)$, *R-Route*$(\boldsymbol{\alpha}_{m+1}, (\kappa\, \upsilon^m))$ is identical to *R-Route*$(\boldsymbol{\alpha}_{m\cdots 1}, (\upsilon^m))$, and its length is given in $\mathbf{H}_m$. Furthermore, because (υ^m) is a common destination for any source nodes in $F_m(\kappa)$, the internodal distances from all source nodes in $F_m(\kappa)$ to (υ^m) are given by a column vector $\mathbf{h}^c_m((\upsilon^m))$. Similar arguments apply also to *R-Route*$((\upsilon\ \kappa^m), \boldsymbol{\beta}_{m+1})$; thus, the internodal distances from (κ^m) to all nodes in $F_m(\upsilon)$ are given by a row vector $\mathbf{h}^r_m((\kappa^m))$. As a result, the internodal distance is given by the sum of the two and one hop for the intercubic link. ▌

COROLLARY 3.1. *The internodal distance for a given source-destination pair* $(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$, *where* $\alpha_m \neq \beta_m$, *is given by*

$$D(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) = \begin{cases} |\alpha_1 \oplus \beta_1| & \text{if} \quad m = 1 \\ |\alpha_1 \oplus \beta_m| + \sum\limits_{k=2}^{m-1} 2^{k-2}[\,|\alpha_k \oplus \beta_m|\,]^+ + 1 & \\ \qquad + \sum\limits_{k=2}^{m-1} 2^{k-2}[\,|\alpha_m \oplus \beta_k|\,]^+ + |\alpha_m \oplus \beta_1| & \text{if} \quad m > 1, \end{cases} \tag{3.2}$$

where $|\alpha \oplus \beta|$ *gives the distance of bitwise exclusive OR* $(\oplus)$ *of* α *and* β *in binary representations, i.e., the number of* 1*s in* $\alpha \oplus \beta$. *Moreover,* $[\,|\alpha \oplus \beta|\,]^+$ *is equal to* 0 *for* $\alpha = \beta$, *and* $|\alpha \oplus \beta| + 1$, *otherwise.*

Proof. The derivation is based on a recursive computation of (3.1) and the well-known $\mathbf{H}_1$. The first two terms in (3.2) give the length of the (A.1)-route; the third term gives the (A.2)-route; and the last two terms give the (A.3)-route. ▌

4. AN ANALYSIS OF ROUTING PERFORMANCE

The main result in this section, given in Theorem 4.1, is a criterion under which Algorithm 3.1 guarantees shortest paths. The proof for the theorem is based on Lemmas 4.1–4.3. Lemma 4.1 first gives the internodal distance of a route that goes through one or two intermediate $(m-1)$-FCCNs before reaching the destination $(m-1)$-FCCN. Without loss of generality, we again assume $\alpha_m \neq \beta_m$ in this section. Because of the network's structure, it is important to note that any routing algorithms employed for FCCN must first route packets to an m-CGN, but the methods of routing packets to the m-CGN and the choices of this m-CGN may differ.

LEMMA 4.1. *For a given source-destination pair* $(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$, *consider Algorithm* 3.1 *with a modified* (A.2)-*route*: *at the end of* (A.1)-*route*, *packets will be routed to one or two intermediate* $(m-1)$-*FCCNs before reaching* $F_{m-1}(\beta_m)$. *The internodal distance for the case of traversing one intermediate* $(m-1)$-*FCCN, say* $F_{m-1}(\gamma)$, $\gamma \neq \alpha_m, \beta_m$, *is given by*

$$\begin{aligned} D_\gamma(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) = |\alpha_1 \oplus \gamma| &+ \sum_{k=2}^{m-1} 2^{k-2}[\,|\alpha_k \oplus \gamma|\,]^+ + 1 \\ &+ 2^{m-2}(|\alpha_m \oplus \beta_m| + 1) + \sum_{k=2}^{m-1} 2^{k-2}[\,|\beta_k \oplus \gamma|\,]^+ \\ &+ |\beta_1 \oplus \gamma|, \qquad m > 1. \end{aligned} \tag{4.1}$$

The internodal distance for the case of traversing two intermediate $(m-1)$*-FCCNs, say* $F_{m-1}(\gamma)$ *first and then* $F_{m-1}(\zeta)$*, where* $\gamma, \zeta \neq \alpha_m, \beta_m$ *and* $\gamma \neq \zeta$*, is given by*

$$\begin{aligned} D_{\gamma,\zeta}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) = {} & |\alpha_1 \oplus \gamma| + \sum_{k=2}^{m-1} 2^{k-2}[\,|\alpha_k \oplus \gamma|\,]^+ \\ & + 2^{m-2}(|\alpha_m \oplus \zeta| + 1) + 1 + 2^{m-2}(|\gamma \oplus \beta_m| + 1) \\ & + \sum_{k=2}^{m-1} 2^{k-2}[\,|\beta_k \oplus \zeta|\,]^+ + |\beta_1 \oplus \zeta|, \qquad m > 1. \end{aligned} \tag{4.2}$$

Proof. For $D_\gamma(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$, (A.1), (A.2), and (A.3) become *R-Route*$(\boldsymbol{\alpha}_m, (\alpha_m \gamma^{m-1}))$, *R-Route*$((\alpha_m \gamma^{m-1}), (\beta_m \gamma^{m-1}))$, and *R-Route*$((\beta_m \gamma^{m-1}), \boldsymbol{\beta}_m)$, respectively. Unlike the previous case, the (A.2)-route now consists of multiple hops. The derivation for $D_{\gamma,\zeta}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$ can be performed similarly. ▮

LEMMA 4.2. $D(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) < D_{\gamma,\zeta}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$, $m > 1$, *and a similar inequality holds for the case of traversing more than two intermediate* $(m-1)$*-FCCNs.*

Proof. Let $\Lambda_{\boldsymbol{\alpha}_m}(\gamma)$ be a set of indices, such that $k \in \Lambda_{\boldsymbol{\alpha}_m}(\gamma)$ if $\alpha_k = \gamma$, $k = 2, ..., m-1$. $\bar{\Lambda}_{\boldsymbol{\alpha}_m}(\gamma)$ is the complement of $\Lambda_{\boldsymbol{\alpha}_m}(\gamma)$. Thus, $[\,|\alpha_k \oplus \gamma|\,]^+ = 0$ for $k \in \Lambda_{\boldsymbol{\alpha}_m}(\gamma)$ and $[\,|\alpha_k + \gamma|\,]^+ = |\alpha_k \oplus \gamma| + 1$ for $k \in \bar{\Lambda}_{\boldsymbol{\alpha}_m}(\gamma)$. Define similar sets for $\boldsymbol{\beta}_m$: $\Lambda_{\boldsymbol{\beta}_m}(\zeta)$ and $\bar{\Lambda}_{\boldsymbol{\beta}_m}(\zeta)$. As a result, $D_{\gamma,\zeta}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$, $m > 1$, can be written as

$$\begin{aligned} D_{\gamma,\zeta}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) = {} & |\alpha_1 \oplus \gamma| + \sum_{k \in \bar{\Lambda}_{\boldsymbol{\alpha}_m}(\gamma)} 2^{k-2}(|\alpha_k \oplus \gamma| + 1) + 1 \\ & + 2^{m-2}(|\alpha_m \oplus \zeta| + 1) + 2^{m-2}(|\gamma \oplus \beta_m| + 1) \\ & + \sum_{k \in \bar{\Lambda}_{\boldsymbol{\beta}_m}(\zeta)} 2^{k-2}(|\beta_k \oplus \zeta| + 1) + |\beta_l \oplus \zeta|. \\ \geqslant {} & |\alpha_1 \oplus \beta_m| + 1 + \sum_{k \in \bar{\Lambda}_{\boldsymbol{\alpha}_m}(\gamma)} 2^{k-2}(|\alpha_k \oplus \beta_m| + 1) \\ & + \sum_{k \in \Lambda_{\boldsymbol{\alpha}_m}(\gamma)} 2^{k-2}(|\gamma \oplus \beta_m| + 1) + 1 \\ & + \sum_{k \in \bar{\Lambda}_{\boldsymbol{\beta}_m}(\zeta)} 2^{k-2}(|\beta_k \oplus \alpha_m| + 1) \\ & + \sum_{k \in \Lambda_{\boldsymbol{\beta}_m}(\zeta)} 2^{k-2}(|\zeta \oplus \alpha_m| + 1) + |\beta_1 \oplus \alpha_m| + 1. \end{aligned} \tag{4.3}$$

The inequality in (4.3) is derived primarily based on triangular inequalities, such as $|\alpha_1 \oplus \gamma| + |\gamma \oplus \beta_m| \geqslant |\alpha_1 \oplus \beta_m|$ for both $\alpha_1 = \beta_m$ and $\alpha_1 \neq \beta_m$. Based on the earlier assumptions, we also have

$$\begin{aligned} D(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) \leqslant |\alpha_1 \oplus \beta_m| &+ \sum_{k \in \bar{\Lambda}_{\boldsymbol{\alpha}_m}(\gamma)} 2^{k-2}(|\alpha_k \oplus \beta_m| + 1) \\ &+ \sum_{k \in \Lambda_{\boldsymbol{\alpha}_m}(\gamma)} 2^{k-2}(|\gamma \oplus \beta_m| + 1) + 1 \\ &+ \sum_{k \in \bar{\Lambda}_{\boldsymbol{\beta}_m}(\zeta)} 2^{k-2}(|\beta_k \oplus \alpha_m| + 1) \\ &+ \sum_{k \in \Lambda_{\boldsymbol{\beta}_m}(\zeta)} 2^{k-2}(|\zeta \oplus \alpha_m| + 1) + |\beta_1 \oplus \alpha_m|. \end{aligned} \tag{4.4}$$

A lower bound for $D_{\gamma,\zeta}(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) - D(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$, given by (4.3)–(4.4), is positive, thus completing the proof. It should be clear that this result also applies to the case of passing through more than two intermediate $(m-1)$-FCCNs. ▮

LEMMA 4.3. *If either* $\boldsymbol{\alpha}_m$ *or* $\boldsymbol{\beta}_m$ *in a given source-destination pair* $(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$ *is a UGN, where* $\alpha_m \neq \beta_m$, *the route computed by Algorithm* 3.1 *is a shortest path.*

Proof. We prove this lemma by induction. It is clearly true for $m = 1$. For $m = 2$, it turns out that we can obtain a more general result: Algorithm 3.1 gives shortest paths for any pair of nodes in a 2-FCCN with $\beta_2 \neq \alpha_2$ if $|\alpha_1 \oplus \beta_2| + |\alpha_2 \oplus \beta_1| - |\alpha_2 \oplus \beta_2| - |\alpha_1 \oplus \beta_1| - 1 \leqslant 0$. To prove this result, we have to prove three things: (1) the corresponding (A.1)-route computed by the algorithm is a shortest path; (2) the corresponding (A.3)-route computed by the algorithm is a shortest path; (3) the length of the overall path is shortest. Note that both (1) and (2) are true because of the cubic routing. Based on the result of Lemma 4.2, (3) is also true for a given source–destination pair if Algorithm 3.1 gives a shorter path than the modified algorithm that is required to traverse one intermediate 1-FCCN. By comparing (3.2) and (4.1) for $m = 2$, and using $|\alpha_1 + \gamma| + |\gamma \oplus \beta_1| \geqslant |\alpha_1 \oplus \beta_1|$, we obtain the sufficient condition stated earlier. As a result, the lemma is true also for $m = 2$.

In the induction step, we consider a k-FCCN, $2 < k \leqslant m$, and assume that the lemma is true for a $(k-1)$-FCCN. In this case, both (1) and (3) are also true due to the assumption for $(k-1)$-FCCN. Therefore, once again the shortest path is given either by Algorithm 3.1 or by the modified algorithm that is required to traverse one intermediate $(k-1)$-FCCN. By applying an approach similar to the proof of Lemma 4.2, one can show that $D_\gamma(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) - D(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) > 0$, thus completing the proof. ▮

THEOREM 4.1. *Algorithm* 3.1 *guarantees shortest paths for any given source-destination pair* $(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$, *where* $\alpha_m \neq \beta_m$, *if and only if* $D(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m) \leqslant \min_{\gamma \neq \alpha_m, \beta_m} D_\gamma(\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$.

Proof. The cases where either $\boldsymbol{\alpha}_m$ or $\boldsymbol{\beta}_m$ is a UGN have been handled in Lemma 4.3. For other cases, note that Algorithm 3.1 yields shortest paths for both (A.1)- and (A.3)-routes, because each route involves a UGN in either the source or destination $(m-1)$-FCCNs. As for the length of the overall path, Lemma 4.1 states that going through another immediate $(m-1)$-FCCN is the only possibility of giving a shorter path than the one generated from Algorithm 3.1. ▮

TABLE I

Performance of Algorithm 3.1

m	Percentages of routes that satisfy Theorem 4.1	Average internodal distance for those paths that do not satisfy Theorem 4.1	Average internodal distance for the corresponding shortest paths
2	81.55%	5.3 (+28.8 %)	4.1
3	84.12%	11.0 (+20.8 %)	9.1
4	91.76%	22.9 (+15.8 %)	19.8
5	93.16%	46.4 (+13.4 %)	40.9

Based on the result in Theorem 4.1, one could devise a shortest-path routing algorithm for the FCCN, but every node needs to store additional information and more computation is required. To evaluate the trade-off between routing complexity and routing performance in terms of the path's length, we present numerical results for Theorem 4.1 and average internodal distances in Table 1. In the second column, we show the percentages of routes computed by Algorithm 3.1 that are shortest paths. In the third and fourth columns, we show the average internodal distance for the nonshortest paths computed by Algorithm 3.1, as well as the average for the corresponding shortest paths, respectively. The differences between the two in percentages are recorded within parentheses in the third column.

According to Table I, the case of $m = 2$ represents the worst performance for the simple routing algorithm: almost 20% of the routes are not shortest paths and, on average, are one hop longer than the shortest paths. However, the performance in both columns improves as the network size increases. As a result, using a simple routing algorithm that does not always guarantee shortest paths is quite justified for the FCCN. In other words, using a more complex routing algorithm to guarantee shortest paths for the remaining 7–18% of the routes may not be worthwhile.

5. NETWORK DIAMETER AND AVERAGE INTERNODAL DISTANCE

In this section, we apply the results in the last section to obtain network diameter and average internodal distance for an FCCN. When computing the latter, we also take into consideration the communication locality that is exhibited in many parallel computations.

THEOREM 5.1. *The diameter of the m-FCCN is given by* $W_m = 2^{m+1} - 1$, $m \geqslant 1$.

Proof. $W_m = 2^{m+1} - 1$ is obviously true for $m = 1$. For $m > 1$, we claim that the diameter is given by $2 \times W_{m-1} + 1$; that is, the diameter is given by a path that incurs maximal shortest paths in both (A.1)- and (A.3)-routes. To prove this claim, we first note that such a path exists, for example, between two UGNs in different $(m-1)$-FCCNs, and we know from Lemma 4.3 that these paths are shortest paths. Second, it is clear that all shortest paths between any two nodes are not longer than $2 \times W_{m-1} + 1$. Finally, it is straightforward to show that $W_m = 2^{m+1} - 1$ is a solution of $W_m = 2W_{m-1} + 1$, $m > 1$. ▮

Next, we derive the average internodal distance for the FCCN, taking into consideration the communication locality. Let p_k, $k=2, ..., m$, be the probability that both source and destination nodes belong to the same $(k-1)$-FCCN and $\mathbf{p}_m \equiv (p_m \cdots p_2)$. Thus, the probability of incurring intercubic communication at the kth level is given by $(1-p_k)$. These events at different levels are assumed to be independent of one another. We further assume the following:

- For intracubic communication, a packet is equally likely to be destined to any nodes inside the FCCN.

- For intercubic communication, a packet is equally likely to be destined to the other seven FCCNs.

Let $D_m(\mathbf{p}_m)$ be the average internodal distance for the m-FCCN with communication locality characterized by $\mathbf{p}_m$, and D_m be the average internodal distance with uniform message distribution, i.e., $p_i = 1/8$, $i=2, ..., m$. Note that

$$(1-p_m)+p_m(1-p_{m-1})+\cdots+p_m p_{m-1}\cdots p_4 p_3(1-p_2)+p_m p_{m-1}\cdots p_4 p_3 p_2 = 1. \tag{5.1}$$

THEOREM 5.2. $D_1 = 1.5$ *and*

$$D_m(\mathbf{p}_m) = 2D_{m-1} - 2\sum_{i=1}^{m-2}\prod_{j=1}^{i} p_{m-j+1}(D_{m-i}-D_{m-i-1})+1$$

$$+0.5\prod_{i=2}^{m} p_i \in O(\sqrt[3]{|V_m|}) \qquad \text{for} \quad m>1, \tag{5.2}$$

where $|V_m| = 8^m$ *is the number of nodes in an* m-FCCN; *and*

$$D_m = D_{m-1} + \tfrac{55}{64}\times 2^{m-2}\times(D_1+1)+\tfrac{1}{64} \in O(\sqrt[3]{|V_m|}). \tag{5.3}$$

Proof. The proof is straightforward, therefore it is omitted. ▮

Although the presence of communication locality does not improve the order of increase in the average internodal distance, the numerical results in Table II show

TABLE II

Effect of Communication Locality on the Average Internodal Distance

m	$a_i = 0.125\ \forall i$	$a_i = 0.5\ \forall i$	$a_i = 0.6\ \forall i$	$a_i = 0.7\ \forall i$	$a_i = 0.8\ \forall i$	$a_i = 0.9\ \forall i$
2	3.66	2.75	2.50	2.25	2.00	1.75
3	7.98	5.24	4.59	3.89	3.14	2.35
4	16.59	9.98	8.64	7.14	5.46	3.58
5	33.79	19.13	16.50	13.48	10.02	6.05
6	68.18	37.05	31.89	25.93	19.01	10.95
7	136.95	72.48	62.29	50.52	36.79	20.64
8	274.46	142.91	122.71	99.37	72.10	39.91

that its presence reduces the average internodal distance significantly. For example, the case of $\alpha_i = 0.5\ \forall i$ improves the average internodal distance in the uniform case ($\alpha_i = 0.125\ \forall i$) by 25, 34, and 40% for $m = 2$, 3, 4, respectively. This improvement increases with m, and it reaches 48% for $m = 8$. Moreover, the improvement increases with α_i; for example, the case of $\alpha_i = 0.9\ \forall i$ and $m = 8$ improves the average internodal distance in the uniform case by 85%.

6. CONCLUSIONS

In this paper, we have analyzed several important routing properties for the FCCN. The most important property is the condition under which the simple routing algorithm guarantees shortest paths. The proof for this condition is based on a recursive algorithm for computing internodal distance for a source–destination pair, triangular inequalities, and other routing properties. Numerical results computed from the condition confirm that using the simple routing algorithm is well justified, because over 80% of the routes computed by the algorithm are indeed shortest paths. Moreover, this percentage increases with the network size. We then apply the routing properties to obtain the network diameter and average internodal distance. When computing the latter, we take account of the communication locality that is exhibited in many parallel computations. Numerical results show that the presence of the communication locality significantly reduces the average internodal distance in the network.

ACKNOWLEDGMENTS

This work was partially supported by Hong Kong Polytechnic University Central Research Grants 353/001 and 353/081. The authors are grateful to the anonymous reviewers for their useful comments.

REFERENCES

1. C. Chen, D. P. Agrawal, and J. R. Burke, dBCube: A new class of hierarchical multiprocessor interconnection networks with area efficient layout, *IEEE Trans. Parallel Distrib. Systems* **4**(12) (Dec. 1993), 1332–1344.
2. G. H. Chen and D. R. Duh, Topological properties, communication, and computation on WK-recursive networks, *Networks* **24** (1994), 303–317.
3. Q. M. Malluhi and M. A. Bayoumi, The hierarchical hypercube: A new interconnection topology for massively parallel systems, *IEEE Trans. Parallel Distrib. Systems* **5**(1) (Jan. 1994), 17–30.
4. F. P. Preparata and J. Vuillemin, The cube-connected cycles: A versatile network for parallel computation, *Commun. ACM* **24**(5) (May 1981), 300–309.
5. M. R. Samatham and D. K. Pradhan, The De Bruijn multiprocessor network: A versatile parallel processing and sorting network for VLSI, *IEEE Trans. Comput.* **38**(4) (Apr. 1989), 567–581.
6. P. Vadapalli and P. K. Srimani, A new family of Cayley graph interconnection networks of constant degree four, *IEEE Trans. Parallel Distrib. Systems* **7**(1) (Jan. 1996), 26–32.
7. H. Y. Wang and R. K. C. Chang, Fully connected cubic network: A highly recursive interconnection network, *in* “Proc. 11th Intl. Conf. Parallel and Distributed Computing Systems,” pp. 250–255, Sept. 1998.