J. Parallel Distrib. Comput. 96 (2016) 38–44

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Vertex-disjoint paths in DCell networks

Xi Wang [a], Jianxi Fan [a,*], Cheng-Kuan Lin [a], Xiaohua Jia [b]

[a] School of Computer Science and Technology, Soochow University, Suzhou 215006, China
[b] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

HIGHLIGHTS

- Construct $n + k - 1$ vertex-disjoint paths between every two distinct vertices of the DCell. Their longest length is not greater than $2^{k+1} + 3$, where it was proved that the diameter of a k-dimensional DCell, $DCell_k$, has an upper bound $2^{k+1} - 1$.
- Propose an $O(k2^k)$ algorithm for finding vertex-disjoint paths between every two distinct vertices in DCell.
- Give the simulation result of the maximal length of disjoint paths gotten by our algorithm.

ARTICLE INFO

Article history:
Received 13 August 2014
Received in revised form
28 March 2016
Accepted 1 May 2016
Available online 6 May 2016

Keywords:
Disjoint paths
DCell networks
Data center networks
Algorithm

ABSTRACT

The DCell network is suitable for massively scalable data centers with high network capacity by only using commodity switches. In this paper, we construct $n+k-1$ vertex-disjoint paths between every two distinct vertices of the DCell network. Their longest length is not greater than $2^{k+1} + 3$, where it was proved that the diameter of a k-dimensional DCell, $DCell_k$, has an upper bound $2^{k+1} - 1$. Furthermore, we propose an $O(k2^k)$ algorithm for finding vertex-disjoint paths between every two distinct vertices in DCell networks. Finally, we give the simulation result of the maximal length of disjoint paths gotten by our algorithm.

1. Introduction

Data centers are critical to the business of companies such as Amazon, Google, Facebook, and Microsoft, which have already owned data centers with more than hundreds of thousands of servers. Their operations are important to offer both many on-line applications such as web search, on-line gaming, email, cloud storage, and infrastructure services such as GFS [10], Map-reduce [5], and Dryad [16]. In particular, a data center network plays important roles in the data center, which is composed of servers, switches, and links connecting servers and switches.

The topologies of data center networks can be modeled using graphs. In research on various properties of data center networks, switches can be regarded as transparent network devices [13]. As a result, a data center network can generally be represented by a simple graph $G = (V(G), E(G))$, where $V(G)$ represents the vertex set and $E(G)$ represents the edge set, each vertex represents a server, and each edge represents a link between servers. In this paper, all graphs are simple and undirected, we use graphs and data center networks interchangeably. Recently, graph theory is a greatly powerful mathematical tool for designing and analyzing topological properties of data center networks such as routing [13,12,18,19], fault tolerance [13,12], Hamiltonian properties [9,25], and robustness analyzing [3,2].

A path in G is a sequence of vertices, $P = \langle u_0, u_1, \ldots, u_j, \ldots, u_{n-1}, u_n\rangle$, in which no vertices are repeated and u_j, u_{j+1} are adjacent for any integer $0 \leq j < n$. G is connected when at least one path exists between any two vertices. We use $\mathrm{dist}(G, x, y)$ to denote the distance between two vertices x and y of the graph G. The diameter of G is defined as $d(G) = \max\{dist(G, x, y)|x, y \in V(G), x \neq y\}$. The connectivity of G is defined as the minimum number of vertices whose removal renders it disconnected or trivial. Let $\kappa(G)$ (κ for short) be the connectivity of G. For any pair of vertices in G, say u and v, by the Menger's theorem [4], we can find κ vertex-disjoint paths (disjoint paths for short).

In recent years, finding and implementing disjoint paths in data center networks have become increasingly important in studies of fault tolerant ability, load balancing, congestion control, and multi-path TCP [12,1,11,15,24]. We use $BCube_k$ to denote a k-dimensional BCube, Guo et al. gave a $O(k^2)$ algorithm of finding the $k + 1$

* Corresponding author.
E-mail address: jxfan@suda.edu.cn (J. Fan).

http://dx.doi.org/10.1016/j.jpdc.2016.05.001
0743-7315/

disjoint paths between any two vertices in $BCube_k$, and they have proved that the longest length of these disjoint paths is $k+1$ [12]. Jayaram et al. provided a new disjoint paths forwarding solution over arbitrary data center networks [23]. Hussam et al. proposed a custom routing protocol using disjoint paths in CamCube [20].

DCell [13] has been proposed as a novel data center network. It has many desirable properties including exponential scalability, high network capacity, small diameter, and high fault tolerance. DCell only requires mini-switches and can support an efficient and scalable routing algorithm [13]. We use $DCell_k$ to denote a k-dimensional DCell built with n port switches. Some basic properties $DCell_k$, such as connectivity [13,22], restricted connectivity [27], diameter [13,19], symmetry [19], broadcasting [13], and Hamiltonian properties [25], have been studied. Recently, we proved that a $DCell_k$ is one-to-one r-disjoint path coverable for $k \geq 0$ and $n \geq 2$ with one exception and proposed an $O(t_k)$ algorithm for finding a one-to-one r-disjoint path cover in $DCell_k$, where t_k is the number of servers in $DCell_k$ [26]. These measurement results demonstrate that a $DCell_k$ has excellent topological properties.

In particular, Guo et al. [13] proved that the connectivity of $DCell_k$ is $n+k-1$. Hence, by Merger's theorem [4], at least $n+k-1$ disjoint paths exist between any two distinct vertices u and v in a $DCell_k$. In this paper, we will study the disjoint paths in DCell networks. Our results are fundamental and essential for fault tolerance, congestion control, and multi-path TCP in DCell networks. The major contributions are as follows:

(1) We construct $n+k-1$ disjoint paths between every two distinct vertices in $DCell_k$. Moreover, we prove that length of the longest path of the $n+k-1$ disjoint paths is no more than $2^{k+1}+3$.
(2) We provide an $O(k2^k)$ algorithm for finding $n+k-1$ disjoint paths between every two distinct vertices in $DCell_k$.

The remainder of this paper is organized as follows. Section 2 provides the preliminary knowledge. Section 3 gives an algorithm to construct $n+k-1$ disjoint paths between every two distinct vertices in $DCell_k$ and discusses the lengths of obtained paths. We make a conclusion in Section 4.

2. Preliminaries

For $U \subseteq V(G)$, we use $G[U] = (U, E')$ to denote the subgraph induced by U in G where $E' = \{(u,v) \in E(G) | u, v \in U\}$. The edge between vertices u and v is denoted by (u,v). If graph G_1 is isomorphic with graph G_2, then we write as $G_1 \cong G_2$. We use $V(P)$ to denote the set of all vertices appearing in P and $E(P)$ to denote the set of all edges appearing in P. The notation $l(P)$ refers to the length of the path P. We say $P = \emptyset$ if the path P satisfy $l(P) = 0$. We also write the path $\langle u_1, u_2, \ldots, u_k \rangle$ as $\langle u_1, Q_1, u_i, u_{i+1}, \ldots, u_j, Q_2, u_t, \ldots, u_k \rangle$, where Q_1 is the path $\langle u_1, u_2, \ldots, u_i \rangle$ and Q_2 is the path $\langle u_j, u_{j+1}, \ldots, u_t \rangle$. Hence, it is possible to write a path as $\langle u_1, Q, u_1, u_2, \ldots, u_k \rangle$ if $Q = \emptyset$.

DCell uses recursive defined structure to interconnect servers. Each server connects to different levels of DCell through multiple links. We build high-level DCell recursively to form many low-level ones. Due to this structure, DCell uses only mini-switches instead of using high-end switches.

$DCell_0$ is a complete graph on n vertices for any integer $n \geq 2$. We use t_0 to denote the number of vertices in $DCell_0$ with $t_0 = n$. Furthermore, we use t_k to denote the number of vertices in $DCell_k$ for any integer k with $k \geq 1$, where $t_k = t_{k-1}(t_{k-1}+1)$. The vertex x of $DCell_k$ can be labeled by $[x_k, x_{k-1}, \ldots, x_i, \ldots, x_0]$ with $x_0 \in \{0, 1, \ldots, n-1\}$ and $x_i \in \{0, 1, \ldots, t_{i-1}\}$ for all $i = 1, 2, \ldots, k$. According to the definition of $DCell_k$ [13], we provide a recursive definition of it.

Definition 1. The $DCell_k$ is defined recursively as follows.

(1) $DCell_0$ is a complete graph consisting of n vertices.
(2) For any positive integer k, $DCell_k$ is built from $t_{k-1}+1$ disjoint copies $DCell_{k-1}$, according to the following steps.
  (2.1) We use $DCell^i_{k-1}$ to denote the graph obtained by prefixing the label of each vertex of one copy of $DCell_{k-1}$ with i for $i = 0, 1, \ldots, t_{k-1}$.
  (2.2) For any $x_k, y_k \in \{0, 1, \ldots, t_{k-1}\}$ with $x_k < y_k$, vertex $x = [x_k, x_{k-1}, \ldots, x_0]$ in $DCell^{x_k}_{k-1}$ is adjacent to vertex $y = [y_k, y_{k-1}, \ldots, y_0]$ in $DCell^{y_k}_{k-1}$ if and only if $x_k = y_0 + \sum_{j=1}^{k-1}(y_j t_{j-1})$ and $y_k = x_0 + \sum_{j=1}^{k-1}(x_j t_{j-1}) + 1$.

For example, when $n = 6$ and $k = 3$, vertex [36, 1, 1, 0] in $DCell^{36}_2$ is adjacent to vertex [49, 0, 6, 0] in $DCell^{49}_2$ by step (2.2) of Definition 1.

Fig. 1 shows three examples of DCell. The following definition describes how to check if a given pair vertices are adjacent in $DCell_k$. This definition can be considered as the non-recursive definition of DCell.

Definition 2. Let $x = [x_k, x_{k-1}, \ldots, x_0]$ and $y = [y_k, y_{k-1}, \ldots, y_0]$. For all integers k with $k \geq 0$, (x, y) is an edge in $DCell_k$ if and only if there is an integer l with

(1) $[x_k, x_{k-1}, \ldots, x_l] = [y_k, y_{k-1}, \ldots, y_l]$,
(2) $x_{l-1} \neq y_{l-1}$, and
(3) $x_{l-1} = y_0 + \sum_{j=1}^{l-2}(y_j t_{j-1})$ and $y_{l-1} = x_0 + \sum_{j=1}^{l-2}(x_j t_{j-1}) + 1$ if $x_{l-1} < y_{l-1}$.

When both conditions (1) and (2) of Definition 2 hold, we say that x and y have a leftmost differing element at position $l-1$. For any integer d with $d \geq 1$, when two adjacent vertices x and y have a leftmost differing element at the position d, we say that y is the d-neighbor of x or the edge (x, y) is an edge of dimension d. We use $N(x, d)$ to denote the d-neighbor of x if $d \geq 1$. For all integer i with $0 \leq i \leq n$, let $z_i = [x_k, \ldots, x_1, i]$, we say that z_i is a 0-neighbor of x or the edge (x, z_i) is an edge of dimension 0 if $z_i \neq x$. Let $N(x, 0) = \{z_i | 0 \leq i \leq n \text{ and } z_i \neq x\}$, we use $N(x, 0)$ to denote the 0-neighbors of x.

It is clear that $DCell_k$ is a regular graph with t_k vertices and $\frac{(n+k-1)t_k}{2}$ edges. Obviously, we have $DCell_{k-1} \cong DCell^0_{k-1} \cong DCell^1_{k-1} \cong \cdots \cong DCell^{t_{k-1}}_{k-1}$. For any integers $n \geq 2$ and $k \geq 1$, edges joining nodes in the same copy of $DCell_{k-1}$ are called internal edges and edges joining nodes in disjoint copies of $DCell_{k-1}$ are called external edges. Clearly, each node of $DCell^i_{k-1}$ is joined to exactly one external edge and $(n+k-2)$-internal edges for $i = 0, 1, \ldots, t_{k-1}$. Moreover, there exists and only exists one unique external edge which connects two disjoint copies of $DCell_{k-1}$.

Some properties of the $DCell_k$ ($k \geq 0$ and $n \geq 2$) have received considerable attentions in the literature.

Theorem 3 ([13]). *The number of servers in $DCell_k$ satisfies $t_k \geq (n+\frac{1}{2})^{2^k} - \frac{1}{2}$.*

Theorem 4 ([13]). *The diameter of $DCell_k$ is no more than $2^{k+1}-1$.*

3. Main results

In this section, we construct $n+k-1$ disjoint paths in $DCell_k$ in Theorem 9. Furthermore, we propose an $O(k2^k)$ algorithm for finding $n+k-1$ disjoint paths in $DCell_k$.

DCellRouting is simple and efficient one-to-one routing algorithm in DCell without failure [13]. We use $\Delta(k)$ to denote $2^{k+1}-1$ and $R(x, y, DCell_k)$ to denote a path in DCellRouting between any two distinct vertices x and y in $DCell_k$.