The Generalized DCell Network Structures and Their Graph Properties

Markus Kliegl*, Jason Lee†, Jun Li‡, Xinchao Zhang§, David Rincón¶, Chuanxiong Guo♯
*Swarthmore College, †Duke University, ‡Fudan University, §Shanghai Jiaotong University,
¶Universitat Politècnica de Catalunya, ♯Microsoft Research Asia

***Abstract*—DCell [7] has been proposed as a server centric network structure for data centers. DCell can support millions of servers with high network capacity and provide good fault tolerance by only using commodity mini-switches. In this paper, we show that DCell is only a special case of a more generalized DCell structure. We give the generalized DCell construction rule and several new DCell structures. We analyze the graph properties, including the closed form of number of servers, bisection width, diameter, and symmetry, of the generalized DCell structure. Furthermore, we show that the new structures are more symmetric, have much smaller diameter, and provide much better load-balancing than the original DCell by using shortest-path routing. We demonstrate the load-balancing property of the new structures by analysis and extensive simulations.**

I. Introduction

Data centers are becoming increasingly important and complex. For instance, data centers are critical to the operation of companies such as Microsoft, Yahoo!, and Google, which already run data centers with several hundreds of thousands of servers. Furthermore, data center growth for e.g. Microsoft exceeds even Moore's Law [18]. It is clear that the traditional tree structure employed for connecting servers in data centers will no longer be sufficient for future cloud computing and distributed computing applications. There is, therefore, an immediate need to design new network topologies that can meet these rapid expansion requirements.

Current network topologies that have been studied include Ring, Torus, Butterfly, HyperCube, FatTree, BCube [6], and FiConn [12]. However, except for the last three, these were proposed in the context of parallel processing and do not meet all of the requirements for large data center use. Some do not scale quickly enough or cannot be incrementally deployed. Others are not robust enough, require too much wiring, have too large diameters, or suffer from bottlenecks. The last three address different issues: For large data centers, FatTree requires the use of expensive high-end switches to over come bottleneck problems, and is therefore more useful for smaller data centers. BCube is meant for container-based data center networks, which are of the order of only a few thousand servers. FiConn is designed to utilize currently unused backup ports in already existing data center networks. See also [7] for discussion of these topologies except BCube and FiConn.

Researchers at Microsoft Research Asia have recently proposed in [7] a novel network structure called DCell, which addresses the needs of a mega data center. Its desirable properties include that it

- scales doubly exponentially;
- supports a high bandwidth;
- has a large bisection width;
- has a small diameter;
- is very fault-tolerant;
- can be built from commodity network components;
- supports an efficient and scalable routing algorithm.

One problem that remains in DCell is that the load is not evenly balanced among the links in all-to-all communication. This is true for the proposed DCellRouting algorithm as well as shortest path routing. This could be an obstacle to the use of the DCell topology for applications such as MapReduce, Google's framework for distributed computing [3].

In this report, we address this problem by showing that DCell is but one member of a family of graphs satisfying all of the good properties listed above. After introducing this family of generalized DCell graphs, we explore the graph properties common to all of them as well as some differences between individual members of the family. In particular, we provide better bounds than [7] for the number of servers, the diameter, and the bisection width of DCells; and we explore the symmetries of the graphs.

Next, we provide an autoconfiguration algorithm and a generalized version of the DCellRouting algorithm. These are crucial to making DCells a viable candidate for data center networks. We also prove a number of results concerning the path length distribution in one-to-one communication and the flow distribution in all-to-all communication when using generalized DCellRouting.

Finally, we show simulation results on the path length distribution and flow distribution for both DCellRouting and shortest path routing for several realistic parameter values. The most important finding here is that other members of the generalized DCell graph family have significantly better load-balancing properties than the original DCell graph.

The rest of the report is organized as follows. In Section II, we introduce the generalized DCell design. In Section III, we present our results on the graph properties of generalized DCells. In Section IV, we give autoconfiguration and routing algorithms. In Section V, we present simulation results for path length and flow distribution using shortest path routing

and DCellRouting. We conclude the report in Section VI and provide ideas for future research.

II. Generalized DCell

A. Construction

The general construction principle of the generalized DCell is the same as that of the original DCell [7]. A DCell_0 consists of n servers connected to a common switch—as an abstract graph, we model this as K_n, the complete graph on n vertices. From here, we proceed recursively. Denote by t_k the number of servers in a DCell_k. Then, to construct a DCell_k, we take $t_k + 1$ DCell_{k-1}'s and connect them in such a way that

(a) there is exactly one edge between every pair of distinct DCell_{k-1}'s, and
(b) we have added exactly one edge to each vertex.

Requirement (a) means that, if we contract each DCell_{k-1} to a single point, then the DCell_k is a complete graph on $t_k + 1$ vertices. This imitation of the complete graph is what we believe gives the DCell structure many of its desirable properties. Requirement (b) is the reason why we must have exactly $t_k + 1$ DCell_{k-1}'s in a DCell_k. It ensures that every server has the same number of links and is the reason why DCell scales doubly exponentially.

This is precisely the point of divergence from the original DCell proposal. There, one specific way of meeting requirements (a) and (b) was proposed, which we name the "α connection rule" later on. But there are many other possibilities. Before we can make this idea more precise, we need to discuss how we label the vertices.

Each server is labeled by a vector id $[a_k, a_{k-1}, ..., a_0]$. Here a_k specifies which DCell_{k-1} the server is in; a_{k-1} specifies which DCell_{k-2} inside that DCell_{k-1} the server is in; and so on. So $0 \leq a_0 \leq n$, and for $i \geq 1$, we have $0 \leq a_i \leq t_{i-1}$. We can convert a vector id to a scalar *uid* (unique identifier) as follows:

$$u = a_0 + a_1 t_0 + a_2 t_1 + \cdots + a_k t_{k-1}. \tag{1}$$

Note that we have $0 \leq u \leq t_k - 1$. Most often, we will label servers just by $[a, b]$ where $a \simeq a_k$ is the number of the DCell_{k-1}, and b is the *uid* corresponding to $[a_{k-1}, ..., a_0]$.

Using these notions, we can define mathematically what a connection rule is. Namely, it is a perfect matching ρ_L of the vertices

$$\{0, ..., t_{L-1}\} \times \{0, ..., t_{L-1} - 1\}$$

that must satisfy the following two properties:

1) ρ_L^2 must be the identity, so that the graph is undirected. (This is also implicit in the term "perfect matching".)
2) For all $a \neq c$, there exist b and d such that $\rho_L([a, b]) = [c, d]$. This ensures that there is a L-level link between each pair of distinct DCell_{L-1}'s.

This encapsulates precisely the requirements (a) and (b) above.

For completeness's sake, we present a formal definition of a generalized DCell.

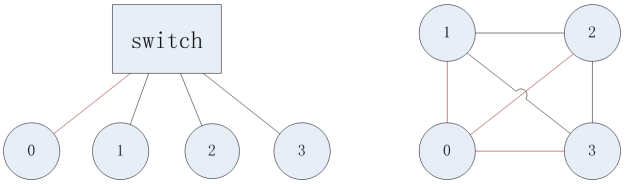


Fig. 1: Physical and abstract DCell_0 for $n = 4$.

Definition 1. *A generalized DCell with parameters $n \geq 2$, $k \geq 0$, and $R = (\rho_1, ..., \rho_k)$ is constructed as follows:*

- *A DCell_0 is a complete graph on n vertices.*
- *From here we proceed recursively until we have constructed a DCell_k: A DCell_L consists of $t_{L-1} + 1$ DCell_{L-1}'s, where t_{L-1} is the number of vertices in a DCell_{L-1}. Edges are added according to the connection rule ρ_L.*

Finally, we have two more remarks to make on this definition:

- We require $n \geq 2$, since $n = 1, k = K$ is the same as $n = 2, k = K - 1$.
- As already mentioned, in a physical network, a DCell_0 consists of n servers connected to a switch (except for $n = 2$), but for mathematical purposes, it is more convenient to model this as a complete graph. Only in the following two instances is the difference important to us:
  1) The graph is regular of degree $k + (n - 1)$, while physically each server has only $k + 1$ links.
  2) The load on a physical 0-level link is actually the sum of the loads on the corresponding $n-1$ abstract edges. See figure 1.

B. Examples of connection rules

Some examples of connection rules are:

α. The connection rule for the original DCell is

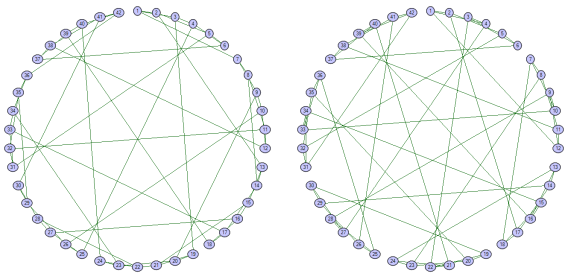
$$\alpha_L : [a, b] \leftrightarrow \begin{cases} [b + 1, a] & \text{if } a \leq b, \\ [b, a - 1] & \text{if } a > b. \end{cases} \tag{2}$$

β. A mathematically simple connection rule is

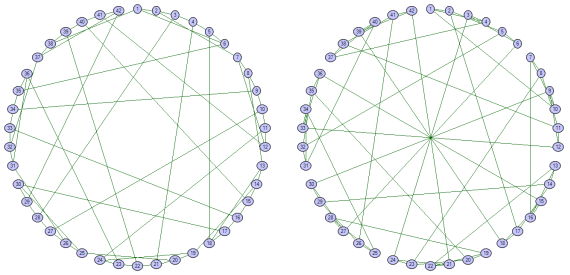
$$\beta_L : [a, b] \leftrightarrow [a + b + 1 \,(\text{mod } t_{L-1} + 1), t_{L-1} - 1 - b]. \tag{3}$$

γ. For t_{L-1} even, we can leave b unchanged by the switch, except for a change inside the DCell_0.

$$\gamma_L : [a, b] = \begin{cases} [a + b \,(\text{mod } t_{L-1} + 1), b - 1] \\ \qquad \text{if } b \text{ is odd}, \\ [a - (b + 1) \,(\text{mod } t_{L-1} + 1), b + 1] \\ \qquad \text{if } b \text{ is even}. \end{cases} \tag{4}$$



(a) α-DCell (b) β-DCell



(c) γ-DCell (d) δ-DCell

Fig. 2: Generalized DCells with different connection rules for $n = 2, k = 2$.

δ. For t_{L-1} even:

$$\delta_L : [a, b] \leftrightarrow \begin{cases} [a + b + 1 \,(\text{mod } t_{L-1} + 1), b + \frac{t_{L-1}}{2}] \\ \qquad \text{if } b < \frac{t_{L-1}}{2}, \\ [a - b + \frac{t_{L-1}}{2} - 1 \,(\text{mod } t_{L-1} + 1), b - \frac{t_{L-1}}{2}] \\ \qquad \text{otherwise.} \end{cases} \tag{5}$$

For $n = 2, k = 2$, these graphs are shown in figure 2.

It is not hard to see that these are indeed valid connection rules. As an example, we show the details for the β connection rule.

1) We have

$$\begin{aligned} &[a + b + 1 \,(\text{mod } t_{k-1} + 1), t_{k-1} - 1 - b] \\ &\quad \leftrightarrow [(a + b + 1) + (t_{k-1} - 1 - b) + 1 \,(\text{mod } t_{k-1} + 1), \\ &\qquad t_{k-1} - 1 - (t_{k-1} - 1 - b)] \\ &\quad = [a + (t_{k-1} + 1) \,(\text{mod } t_{k-1} + 1), b] \\ &\quad = [a, b]. \end{aligned}$$

So β_k^2 is indeed the identity.

2) $a + b + 1 \,(\text{mod } t_{k-1} + 1)$ ranges, in cyclic order, from $a + 1$ to $a - 1$ modulo $t_{k-1} + 1$ as b ranges from 0 to $t_{k-1} - 1$. So there is indeed a k-level link from a to every DCell$_{k-1}$ distinct from a.

In the rest of this report, when the specific connection rule used is not important, we will speak of just DCells. If we need to make reference to a specific connection rule, we will speak e.g. of α-DCells, meaning DCells with $R = (\alpha_1, ..., \alpha_k)$. In this context, we should clarify why the requirement that t_{k-1} be even is not a practical problem for the γ and δ connection rules. The reason is the following fact.

Fact 1. *t_k is even for $k \geq 1$. For $k = 0$, t_0 is even if and only if n is even.*

Proof: This follows at once from the definition $t_0 = n$ and the recurrence $t_k = t_{k-1}(t_{k-1} + 1)$ for $k \geq 1$. ■

It follows that we could define γ-DCell's for odd n by $R = (\rho_1, \gamma_2, ..., \gamma_k)$, where ρ_1 is any 1-level connection rule that works for odd n. (And similarly for δ.) However, almost all real switches have an even number of ports. So we will focus on even n for the remainder of the paper.

Fact 1 also explains why a connection rule of the form $[a, b] \leftrightarrow [f(a, b), b]$ is not possible. For if there are an odd number of DCell$_{k-1}$'s, then there are an odd number of vertices for each value of b; but it is impossible to find a perfect matching for an odd number of vertices. Hence, the slightly more complicated form $[a, b] \leftrightarrow [f(a, b), g(b)]$ is required if the second component is to be independent of a. Observe that this is the form of the β, γ, and δ connection rules.

III. Graph Properties

In this chapter, we give expressions and bounds for graph properties such as the number of servers, the diameter, and the bisection width. We also investigate the symmetries of the different connection rules.

A. The number of servers

1) Previous results: No closed-form expression for the exact number of servers t_k in a DCell$_k$ is known. However, it is clear from the DCell construction that t_k satisfies the following recurrence relation:

$$\begin{aligned} t_{k+1} &= t_k(t_k + 1), \qquad \text{for } k \geq 1, \\ t_0 &= n. \end{aligned} \tag{6}$$

This permits t_k to be easily and quickly computed for small n and k. Using this recurrence, the following bounds have been proved in [7]:

$$\left(n + \frac{1}{2}\right)^{2^k} - \frac{1}{2} \leq t_k \leq (n + 1)^{2^k} - 1. \tag{7}$$

These bounds show that t_k grows doubly exponentially.

2) An exact expression: Following a hint by D. E. Knuth in [15], we use the methods of [1] to solve the recurrence (6), leading to the following theorem.

Theorem 1. *We have*

$$t_k + \frac{1}{2} < c^{2^k} < t_k + \frac{6}{10}, \tag{8}$$

and hence

$$t_k = \lfloor c^{2^k} \rfloor, \tag{9}$$

where

$$c = \left(n + \frac{1}{2}\right) \prod_{i=0}^{\infty} \left(1 + \frac{1}{4\left(t_i + \frac{1}{2}\right)^2}\right)^{1/2^{i+1}}. \tag{10}$$

We remark two things about this theorem:

- Since we know from (7) that t_k grows doubly exponentially, the infinite product in (10) will converge to c very rapidly. Thus, although in principle knowledge of the sequence t_k is required to compute c—defeating the whole purpose of this endeavor—in practice, knowing only the first few terms of t_k will allow for c to be computed to extraordinary accuracy.
- We see that, as n increases, c asymptotically approaches $n+\frac{1}{2}$ from above; that is, it approaches the lower bound in (7).

The rest of this section is concerned with the proof of this theorem. First, letting

$$s_k = t_k + \frac{1}{2}, \tag{11}$$

we end up with the simpler recurrence relation

$$s_{k+1} = s_k^2 + \frac{1}{4} = s_k^2\left(1 + \frac{1}{4s_k^2}\right), \qquad \text{for } k \geq 1, \tag{12}$$
$$s_0 = n + \frac{1}{2}.$$

Next, letting

$$y_k = \log s_k, \tag{13}$$
$$\alpha_k = \log\left(1 + \frac{1}{4s_k^2}\right), \tag{14}$$

and taking the logarithm of both sides of (12), we find

$$y_{k+1} = 2y_k + \alpha_k. \tag{15}$$

Thus, we have

$$y_1 = 2y_0 + \alpha_0, \tag{16}$$
$$y_2 = 2(2y_0 + \alpha_0) + \alpha_1, \tag{17}$$
$$\vdots \tag{18}$$
$$y_k = 2^k\left(y_0 + \frac{\alpha_0}{2} + \frac{\alpha_1}{2^2} + \cdots + \frac{\alpha_{k-1}}{2^k}\right). \tag{19}$$

Now look instead at

$$Y_k = 2^k\left(y_0 + \sum_{i=0}^{\infty}\frac{\alpha_i}{2^{i+1}}\right). \tag{20}$$

We will show in a moment that the error arising from looking at Y_k instead of y_k is small. Exponentiating, we find

$$S_k = e^{Y_k} = c^{2^k}, \tag{21}$$

where

$$\begin{aligned} c &= \exp\left(y_0 + \sum_{i=0}^{\infty}\frac{\alpha_i}{2^{i+1}}\right) \\ &= s_0\prod_{i=0}^{\infty}\left(1 + \frac{1}{4s_i^2}\right)^{1/2^{i+1}} \\ &= \left(n+\frac{1}{2}\right)\prod_{i=0}^{\infty}\left(1 + \frac{1}{4\left(t_i+\frac{1}{2}\right)^2}\right)^{1/2^{i+1}}. \end{aligned} \tag{22}$$

It remains to show that S_k closely approximates t_k. Write

$$Y_k = y_k + r_k, \tag{23}$$
$$r_k = 2^k\sum_{i=k}^{\infty}\frac{\alpha_i}{2^{i+1}}. \tag{24}$$

From (12), it follows that $s_{k+1} > s_k$ and hence that $\alpha_k > \alpha_{k+1}$. Thus, we have

$$r_k = 2^k\sum_{i=k}^{\infty}\frac{\alpha_i}{2^{i+1}} \tag{25}$$
$$< 2^k\,\alpha_k\sum_{i=k}^{\infty}\left(\frac{1}{2}\right)^{i+1} \tag{26}$$
$$= \alpha_k, \tag{27}$$

and so

$$S_k = e^{y_k}e^{r_k} \tag{28}$$
$$< s_k e^{\alpha_k} \tag{29}$$
$$= s_k\left(1 + \frac{1}{4s_k^2}\right) \tag{30}$$
$$= s_k + \frac{1}{4s_k} \tag{31}$$
$$\leq s_k + \frac{1}{10} \tag{32}$$
$$= t_k + \frac{6}{10}. \tag{33}$$

where for the last inequality we used $s_k \geq s_0 \geq 2.5$. So we see that, in fact,

$$t_k = \lfloor S_k \rfloor = \lfloor c^{2^k} \rfloor, \tag{34}$$

where c is given by (22).

B. Bisection Width

Ideally, data center networks should have very large bisection widths. There are two reasons for this. First, the bisection width is a measure of the robustness of a network, since a high bisection width ensures that communication within a network will remain possible and efficient even when some links fail. Second, in distributed computing applications, the bisection width is, as Leighton puts it, "often a critical factor in determining the speed with which a network can perform a calculation" [10].

1) A lower bound: Using the methods of [10, §1.9], a lower bound on the bisection width may be found that takes the following form:

$$BW \geq \frac{t_k^2}{4F_{max}}, \tag{35}$$

where F_{max} is the maximum number of flows carried by an edge in an all-to-all communication scenario. In [7], it is shown that, when using DCellRouting, we have $F_{max} < 2^k t_k$, and hence that

$$BW \geq \frac{t_k}{4\cdot 2^k}. \tag{36}$$

Later on, in corollary 4, we will prove a more accurate upper bound on F_{max} for a generalized recursive routing scheme, namely

$$F_{max} < \frac{n-1}{n+\frac{1}{2}} 2^k (t_k + 0.6). \tag{37}$$

Using this, we find the following, slightly improved lower bound on the bisection width.

Corollary 1. *We have*

$$BW \geq \frac{n+\frac{1}{2}}{n-1} \cdot \frac{t_k^2}{4 \cdot 2^k (t_k + 0.6)} = \frac{n+\frac{1}{2}}{n-1} \cdot \frac{t_k}{4 \cdot 2^k} (1 - o(1)). \tag{38}$$

Note that, as n increases, this asymptotically approaches the bound (36).

We can improve this bound slightly by counting the flows between the two sides of the bisection more carefully. For the bound (37) on F_{max} we just used a bound on the flow across a 0-link in an all to all communication scheme (called F_0), since higher-level links carry fewer flows. However for the purpose of using the technique given in [10], we only need to upper bound the number of flows on the 0-link due to communication between the two sides of a bisection. Many of the flows counted by F_0 are due to communication between a pair (u, v) where u and v are in the same side of the bisection. We can tighten the bisection width lower bound by enumerating some of these flows and subtracting them off from F_0.

Theorem 2. *We have*

$$BW \geq \frac{t_k^2}{4(F_0 + t_k - 2\frac{n-1}{n} t_k)} \tag{39}$$

Proof: Let σ_0 denote the number of nodes that are reached from a server through its 0-link using DCellRouting. Due to symmetry,

$$\sigma_0 = \frac{n-1}{n} t_k \geq \frac{t_k}{2}. \tag{40}$$

Thus at least

$$\frac{n-1}{n} t_k - \frac{t_k}{2}$$

of the flows are from communication between servers on one of the sides of the bisection. The same applies to the other side of the bisection and hence at least

$$2\left(\frac{n-1}{n} t_k - \frac{t_k}{2}\right)$$

of the flows are due to communication within a single side of the bisection. Subtracting these flows off in inequality 35 gives the inequality in the theorem. ∎

Notice that this gives us no improvement for the $n = 2$ case and the bound gets better for larger n. An exact expression for F_0 is derived later on, in theorem 9.

n	k	(35)	(41) for α	(41) for β	(41) for γ	(41) for δ
2	2	7	5	6	7	7
4	2	50	43	53	64	55
6	2	189	178	201	264	213
2	3	148	103	185	226	182

TABLE I: Comparison of lower bounds for bisection width. In inequality (35), we used the exact value of F_0 for DCell-Routing (see theorem 9).

n	k	α	β	γ	δ
2	2	11	11	11	11
4	2	72	98	108	110
6	2	355	431	551	541
2	3	288	402	426	420

TABLE II: Upper bounds on bisection width found using the Kernighan-Lin heuristic. At least 1000 trials were used for each of these numbers.

2) A spectral lower bound: A well-known (e.g. [4, p.293]) spectral lower bound on the bisection width of a graph with an even number of vertices is given by

$$BW \geq \frac{N\lambda_2}{4}, \tag{41}$$

where N is the number of vertices and λ_2 is the second smallest eigenvalue of the Laplacian matrix. (Recall that t_k is even for $k \geq 1$.)

The performance of bounds (41) and (35) is compared in table I for some small values of n and k. For the γ connection rule in particular, the spectral bound appears to be significantly better than the bound (35). However, we emphasize that these lower bounds are not tight, and that table I hence does not permit to draw conclusions on whether the bisection width is larger for some connection rules than for others.

3) Some heuristic upper bounds: Using the Kernighan-Lin (K-L) heuristic for graph partitioning [9], we found small bisections for some small values of n and k. The results are shown in table II. All values differ from the best lower bound in the previous section by less than a factor of 3.

We also tried the Randomized Black Hole (RBH) heuristic [5], but this produced larger bisections. K-L and RBH are also reviewed in [8].

C. Diameter

It is desirable for a data center network to have as small a diameter as possible. For if the diameter is large, then communication between some pairs of servers will necessarily be slow, no matter which routing algorithm is used.

1) Previous α-DCell specific results: In [7], it is shown that the diameter satisfies

$$D \leq 2^{k+1} - 1. \tag{42}$$

However, it is also shown that this bound is not tight. For example, for $n = 2$ and $k = 4$ the graph has diameter 27, but the bound yields 31. Finally, it should be remarked that it was not even known yet whether the diameter depends on n.

2) Generalized upper bound: We first restate and reprove the upper bound (42) for generalized DCells.

Theorem 3. *For fixed n, the diameter D_k of a DCell$_k$ satisfies*

$$D_{k+1} \leq 2D_k + 1. \tag{43}$$

If D_{k_0} is known for some k_0, we have

$$D_{k+1} \leq (D_{k_0} + 1)2^{k-k_0} - 1. \tag{44}$$

Proof: The first statement follows from the recursive definition of the DCell structure: Take any two elements in a DCell$_{k+1}$. If they are in the same DCell$_k$, the distance between them is at most D_k. If they are in different DCell$_k$'s, we can find a path of length at most $2D_k+1$ between them by joining each by a path of length at most D_k to the respective vertices linking the two DCell$_k$'s.

As for the second statement, just add 1 to both sides of the first inequality, yielding

$$D_{k+1} + 1 \leq 2(D_k + 1). \tag{45}$$

Then $S_k = D_k + 1$ clearly satisfies $S_k \leq 2^{k-k_0} S_{k_0}$. ∎

Noting that $D_0 = 1$, this theorem leads immediately to inequality (42).

3) A lower bound: A well-known (e.g. [10, p.238]) lower bound on the diameter of a graph G with N vertices and maximum degree Δ is

$$D \geq \frac{\log N}{\log \Delta}. \tag{46}$$

It is not hard to see that a DCell$_k$ is regular of valency $n+k-1$. Recall from theorem 1 that

$$N = \lfloor c^{2^k} \rfloor, \tag{47}$$

where c is approximately $n + \frac{1}{2}$ for large n.

Using this information, the inequality (46) yields the following lemma.

Theorem 4. *The diameter D is bounded below by*

$$D \geq 2^k \frac{\log c}{\log(n + k - 1)}. \tag{48}$$

Note that, by theorem 1, we have

$$\frac{\log c}{\log(n + k - 1)} \geq \frac{\log\left(n + \frac{1}{2}\right)}{\log(n + k - 1)}. \tag{49}$$

Asymptotically, as $n/k \to \infty$, this bound approaches 2^k. More generally, this lower bound is useful whenever k is small compared to n. For example, we have

$$\frac{\log\left(n + \frac{1}{2}\right)}{\log(n + k - 1)} \geq \frac{1}{2} \tag{50}$$

when $k \leq n^2 + \frac{5}{4}$. Since k is an integer, we can write this result as follows.

Corollary 2. *For $k \leq n^2 + 1$, the diameter D is bounded below by*

$$D \geq 2^{k-1}. \tag{51}$$

n	k	α	β	γ	δ
2	2	7	6	6	6
4	2	7	7	7	7
2	3	15	10	10	10
4	3	15	13	12	12
2	4	27	17	15	16

TABLE III: Comparison of the diameter for different connection rules.

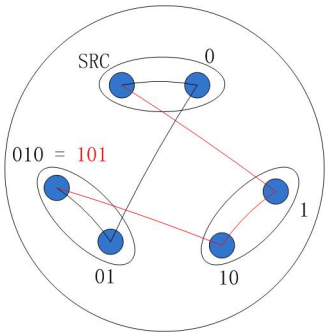


Fig. 3: Notation for proof of fact 2.

Together with the upper bound (42), this narrows the diameter down to within a factor of 4 for the cases when $k \leq n^2 + 1$. Since $n^2 + 1$ is never smaller than 5 (since we require $n \geq 2$), this corollary applies to all realistic cases, since those will have $k = 3$ or $k = 4$.

4) Dependence on n and R: We can now answer in the affirmative the question whether the diameter of an α-DCell depends on n. It is known that the diameter of an α-DCell with $n = 2$ and $k = 4$ is 27 [7]. However, we found computationally that the distance between vertices 0 ($[0, 0, 0, 0, 0]$) and 537468775 ($[21944, 104, 8, 2, 1]$) in a $n = 3, k = 4$ α-DCell is 31. Since we know from the upper bound (42) that the diameter of a $k = 4$ DCell can be at most 31, this shows that the diameter for $n = 3, k = 4$ is exactly 31.

The other connection rules similarly exhibit n-dependence, as can be seen in table III. Furthermore, the diameter appears to depend significantly on the choice of connection rule R. It seems, therefore, that a search for tighter bounds on the diameter will have to take R into account.

As an example, we consider the case $n = 2, k = 2$. For this case we can prove that $5 \leq D \leq 7$, and that it is impossible to find tighter bounds that are independent of R. This is shown in the following two facts.

Fact 2. *For $n = 2, k = 1$, the generalized DCell is isomorphic to a 6-cycle, and hence has diameter 3; for $n = 2, k = 2$, it is at least 5 and at most 7.*

Proof: For $n = 2, k = 1$, the graph is connected and 2-regular; hence it is a cycle.

Now look at $n = 2, k = 2$. Fix a source vertex, and denote by 01 the vertex reached by first taking the 0-link from the source vertex and then the 1-link from the next vertex, etc. See figure 3. There are three vertices that are one hop away: 0,1,2. There are 6 vertices that are two hops away: 01, 02,

n	k	(52)	D_k	(53)
2	1	0.667	3	3
2	2	0.204	7	8
2	3	0.010	15	24
4	1	0.200	3	5
4	2	0.024	7	15

(a) α-DCell

n	k	(52)	D_k	(53)
2	1	0.667	3	3
2	2	0.185	6	8
2	3	0.005	10	18
4	1	0.200	3	5
4	2	0.019	7	14

(b) β-DCell

n	k	(52)	D_k	(53)
2	1	0.667	3	3
2	2	0.151	6	7
2	3	0.004	10	17
4	1	0.200	3	5
4	2	0.016	7	12

(c) γ-DCell

n	k	(52)	D_k	(53)
2	1	0.667	3	3
2	2	0.151	6	7
2	3	0.006	10	18
4	1	0.200	3	5
4	2	0.018	7	13

(d) δ-DCell

TABLE IV: Comparison of the spectral bounds (52) and (53) to the actual diameter.

10, 12, 20, 21. Since the case $n = 2, k = 1$ is a 6-cycle, we have $010 = 101$, and hence there are at most 11 vertices that are three hops away. Consequently, there can be at most 20 vertices that are four hops away, since $0101 = 1011 = 10$, $1010 = 0100 = 01$, $0102 = 1012$, and $2010 = 2101$. But $3 + 6 + 11 + 20 = 40 < 41$. So at least one vertex must be at distance 5 or more.

The upper bound is just the familiar $2^{k+1} - 1$ bound. ∎

Fact 3. *This lower bound of $D = 5$ when $n = 2, k = 2$ is achieved for $R = (\alpha_1, \beta_2)$. As can be seen in table III, $D = 6$ is achieved e.g. for a β-DCell, while $D = 7$ is achieved for an α-DCell.*

5) Spectral bounds: Mohar proved the following lower bound for the diameter D of a graph of N vertices [13] (also cited in [4, p.305]):

$$D \geq \frac{4}{N\lambda_2}, \tag{52}$$

where λ_2 is the second smallest eigenvalue of the Laplacian.

The following spectral upper bound was proved by Chung [2]:

$$D \leq \left\lceil \frac{\cosh^{-1}(N-1)}{\cosh^{-1}\frac{\lambda_N+\lambda_2}{\lambda_N-\lambda_2}} \right\rceil. \tag{53}$$

Here λ_N is the largest eigenvalue of the Laplacian.

The difficulty of finding λ_2 and λ_N for large n or k aside, it appears that these bounds do not perform well even for small values of n and k, as table IV shows.

Given this discouraging performance, we have decided to not further pursue the spectral approach to bounding the diameter.

D. Symmetry

The symmetries of a data center network are of importance for two reasons. On the one hand, more symmetry makes for a more regular network and facilitates the initial wiring. On the other hand, a high degree of asymmetry could allow for nearly perfect autoconfiguration.

1) Generalized DCell: It turns out that, at least for $n \geq 3$, every graph automorphism of a generalized DCell respects its leveled structure; that is, a DCell_L will be mapped to another DCell_L for all L, and all link levels are preserved.

Definition 2. *A link level preserving graph automorphism (LLPGA) is a graph automorphism that maps L-level links to L-level links for each L.*

Lemma 1. *For $n \geq 3$, every graph automorphism of a DCell is a LLPGA.*

Proof: The proof is by induction on the link level L. The base case is $L = 0$. If two adjacent vertices are connected by a 0-link, then they have $n - 2 \geq 1$ common neighbors. If they are connected by a link of degree $k \geq 1$, however, they have no common neighbors. This shows that 0-links must be mapped to 0-links, and hence DCell_0's must be mapped to DCell_0's.

Now suppose a graph automorphism τ preserves link levels for all levels $L < k$ and maps DCell_{k-1}'s to DCell_{k-1}'s. Then, contracting each DCell_{k-1} to a single point is invariant under τ. In the contracted graph, a DCell_k is a complete graph, and so adjacent vertices have $t_{k-1} - 1$ common neighbors if they are connected by a k-link, but no common neighbors if they are connected by a link of greater degree. As desired, we see that k-links must be mapped to k-links by τ, and hence DCell_k's must be mapped to DCell_k's by τ. ∎

Definition 3. *A sequence of nonzero link levels $P = (L_1, ..., L_m)$ induces a function ρ_P from the graph to itself as follows: $\rho_P(v)$ is the vertex we arrive at if we start from vertex v and take at the ith step the L_i-level link. We call such a function a path function.*

Lemma 2. *Path functions are set automorphisms. Furthermore, path functions commute with LLPGA's.*

Proof: First of all, note that path functions are well-defined since each node has a unique L-level link for each L satisfying $1 \leq L \leq k$. Next, note that a path function ρ_P where $P = (L_1, ..., L_m)$ has as its inverse the path function ρ_Q where $Q = (L_m, ..., L_1)$. Finally, that path functions commute with LLPGA's is clear from the two definitions. ∎

2) α-DCell: Despite its simple arithmetical definition, it seems that α-DCell is highly asymmetric. In the following, we discuss the only symmetry we could find.

Definition 4. *The complement of a vertex $\mathbf{v} = [a_k, ..., a_1, a_0]$ is*

$$\bar{\mathbf{v}} = [t_{k-1} - a_k, ..., t_0 - a_1, n - 1 - a_0].$$

The corresponding uid's satisfy $\bar{u} = t_k - 1 - u$.

Note that $\bar{\bar{u}} = u$. Furthermore, it is not hard to show that mapping each vertex to its complement results in an isomorphic graph.

Theorem 5. *The map $\sigma : \mathbf{v} \mapsto \bar{\mathbf{v}}$ is a graph automorphism of α-DCell.*

Proof: The proof is by induction on k. A DCell_0 is just a complete graph, so for $k = 0$ the statement is certainly true. If $k \geq 1$, by the α connection rule, we have

$$[a, b] \leftrightarrow \begin{cases} [b+1, a] & \text{if } a \leq b, \\ [b, a-1] & \text{if } a > b. \end{cases} \tag{54}$$

Consequently, we have

$$\overline{[a,b]} = [t_{k-1} - a, (t_{k-1} - 1) - b] \tag{55}$$

$$\leftrightarrow \begin{cases} [t_{k-1} - 1 - b + 1, t_{k-1} - a] \\ \qquad \text{if } t_{k-1} - a > (t_{k-1} - 1) - b, \\ [t_{k-1} - 1 - b, t_{k-1} - a - 1] \\ \qquad \text{if } t_{k-1} - a \leq (t_{k-1} - 1) - b, \end{cases} \tag{56}$$

$$= \begin{cases} [t_{k-1} - (b+1), (t_{k-1} - 1) - a] & \text{if } \bar{a} \leq \bar{b}, \\ [t_{k-1} - b, (t_{k-1} - 1) - (a-1)] & \text{if } \bar{a} > \bar{b}, \end{cases} \tag{57}$$

$$= \begin{cases} \overline{[b+1, a]} & \text{if } \bar{a} \leq \bar{b}, \\ \overline{[b, a-1]} & \text{if } \bar{a} > \bar{b}. \end{cases} \tag{58}$$

■

Based on some empirical evidence, we are led to conjecture that this is in fact the only symmetry of the graph.

Conjecture 1. *For $k \geq 2$, the automorphism group of an α-DCell$_k$ consists of just the identity and σ.*

So far, this conjecture has been verified for $k = 2$, $2 \leq n \leq 6$ by directly computing the automorphism group. It has also been verified indirectly for $k = 3$, $2 \leq n \leq 4$ by looking at the shortest-path routing distribution for each vertex. In these cases, each pair $(\mathbf{v}, \bar{\mathbf{v}})$ had a unique distribution.

Fact 4. *The conjecture holds for $k = 2$ and $2 \leq n \leq 6$.*

The rest of this section describes our work in progress on proving the conjecture.

Lemma 3. *For $k \geq 2$, the special case $[c, 0, 0]$ where $c = \bar{0}$ or $c = [a, b]$ is routed via the link sequence $(k, k-1, k, k-1, k, k-1)$ as follows:*

$$[c, 0, 0] \Rightarrow^* \begin{cases} [2t_{k-2} + 1, 2, 1] & \text{if } a = 0, b = 0 \\ [t_{k-2}, a - 1, \bar{0}] & \text{if } a \geq 1, b = 0 \\ [t_{k-2} + 1, a + 1, b - 1] & \text{if } \bar{0} > a \geq 1, b = 1 \\ [t_{k-2} + 1, 0, 0] & \text{if } a = \bar{0}, b = 1 \\ [t_{k-2}, a, b - 1] & \\ \qquad \text{if } a < b, b \geq 1 \text{ or } a \geq b, b \geq 2 & \\ [t_{k-2}, \bar{0}, \bar{0}] & \text{if } c = \bar{0} \end{cases} \tag{59}$$

The only value of c for which $[c, 0, 0]$ is routed to $[c, 0, x]$ for some x is $t_{k-2} = [1, 0]$.

Proof: This can be computed directly from the definitions of k and $k-1$ level links. The second statement is found by checking the six cases in equation (59). ■

Lemma 4. *For $k \geq 2$ and $2 \leq n \leq 6$, the only LLPGA's are the identity and σ.*

Proof: The proof is by induction on k. The base cases are stated in fact 4.

Now suppose $k \geq 3$ and the theorem holds for $k - 1$. Let τ be a LLPGA. τ induces a link-preserving $k-1$ graph automorphism $\tilde{\tau}$ as follows:

$$\tilde{\tau}(x) = y, \qquad \text{where } [c, y] = \tau([t_{k-2}, x]). \tag{60}$$

By the induction hypothesis, it follows that $\tilde{\tau}(0) = 0$ or $\tilde{\tau}(0) = \bar{0}$. So it remains to show only that we must have $c = t_{k-2}$ or $c = \overline{t_{k-2}}$, respectively. Then, by the autoconfiguration algorithm 1 proved later on, τ is completely determined and is the identity in the former and σ in the latter case.

First suppose $\tilde{\tau}(0) = 0$. Since τ is link level preserving, by lemmas 2 and 3, we see that c must be t_{k-2}, as desired. Similarly, taking complements in lemma 3, we see that if $\tilde{\tau}(0) = \bar{0}$, then we must have $c = \overline{t_{k-2}}$. ■

Let us summarize our progress on the conjecture so far and what remains to be done.

- Combining lemmas 1 and 4, we see that the conjecture is proven for $3 \leq n \leq 6$.
- For $n = 2$, we have only shown that the only LLPGA's are the identity and σ. To finish the proof of the conjecture, we would need to show that lemma 1 also holds for $n = 2$ when $k \geq 2$ (for $n = 2, k = 1$, the lemma is false). A different proof (or at least a different family of base cases) is needed to do this.
- To prove the conjecture for all $n \geq 7$, we would have to find a way of proving the base case in lemma 4 for arbitrary n.

3) Other connection rules:

Theorem 6. *Suppose the k-level connection rule of a DCell is of the form:*

$$\rho_k : [a, b] \leftrightarrow [a + b + 1 \,(mod\ t_{k-1} + 1), g(b)], \tag{61}$$

where g is any permutation on $\{0, ..., t_{k-1} - 1\}$. Then the map

$$\tau : [a, b] \mapsto [a + 1 \,(mod\ t_{k-1} + 1), b] \tag{62}$$

is a graph automorphism. τ generates a cyclic subgroup of the automorphism group of order $t_{k-1} + 1$.

Proof: We have

$$\tau([a, b]) = [a + 1 \,(\text{mod } t_{k-1} + 1), b] \tag{63}$$

$$\leftrightarrow [(a + 1) + b + 1 \,(\text{mod } t_{k-1} + 1), g(b)] \tag{64}$$

$$= [(a + b + 1) + 1 \,(\text{mod } t_{k-1} + 1), g(b)] \tag{65}$$

$$= \tau([a + b + 1 \,(\text{mod } t_{k-1} + 1), g(b)]). \tag{66}$$

As for the second assertion, clearly τ is of order $t_{k-1}+1$, since for no smaller number c do we have $a + c \equiv a \,(\text{mod } t_{k-1} + 1)$. ■

Note that β, γ, and δ are all of this form. Hence, these connection rules lead to significantly more symmetric graphs

than the α rule. This group of symmetries is very apparent in figure 2.

For β-DCell, the map $\sigma : u \mapsto \bar{u}$ of theorem 5 is also a graph automorphism. As is apparent in figure 2, σ can be viewed as a flip over an axis of the figure. So σ and τ together generate a subgroup isomorphic to the dihedral group of order $2(t_{k-1}+1)$.

IV. Applications

In the previous chapter, we have proved important properties of the graph. To use a DCell network in practice, it is important to be able to configure the servers automatically. Furthermore, an efficient routing algorithm is needed, as shortest path routing can only be performed in small networks. In this chapter, we present an autoconfiguration algorithm and generalize the DCellRouting algorithm of [7].

A. Autoconfiguration

If a data center network of several hundreds of thousands or even millions of servers were deployed, it would be nearly impossible to correctly hardcode the *uid* on each individual computer. For this reason, it is important that the network can autoconfigure itself given only minimal initial information—and do so efficiently in the face of power outages or other network wide failures. We present in this subsection an algorithm that accomplishes this purpose. Assuming only that each server knows the level of each of its links, we show that it is sufficient to hardcode the *uid* of only the first n servers.

Algorithm 1 (Autoconfiguration)**.** *Given a full DCell$_k$ and the labels for a single DCell$_0$ we can uniquely assign the uid's for the entire DCell$_k$, assuming each server knows the level of each of its links.*

Proof: We prove this by providing an algorithm that correctly labels DCell.

The algorithm proceeds level by level. We are assuming one DCell$_0$ is already labeled, so we need only provide an algorithm for labeling a DCell$_k$ given a fully labeled DCell$_{k-1}$. First, we use the labeled DCell$_{k-1}$ to fully label the 0^{th} node in every other DCell$_{k-1}$ by following the level-k link of each node in the DCell$_{k-1}$. This gives the first entry in every node's *uid*.

Now consider a level-k link $[a_k, x] \leftrightarrow [b_k, y]$. Then x and y are completely determined by the connection rule, since we know that there is a unique k-level link between the DCell$_{k-1}$'s a_k and b_k. Thus, the entire *uid* of every node is now determined. ∎

B. Routing

Since shortest path routing is feasible only for small networks [14], it is important to have an efficient, locally computable routing algorithm. In [7], a recursive routing algorithm called DCellRouting is presented for the α-DCell. In this subsection, we show that DCellRouting can be made to work for any connection rule. We also prove a number of results concerning the path length distribution and flow distribution when using DCellRouting.

1) Generalized DCellRouting: Generalized DCellRouting is quite simple, and works almost exactly like the original DCellRouting algorithm.

- If the source and destination are identical, do nothing.
- If the source and destination are within the same DCell$_0$, just take the link connecting the two.
- Otherwise, determine the largest L such that the source and destination are not within the same DCell$_{L-1}$. We know that there is a unique link between the two DCell$_{L-1}$'s. Call the nodes at the two ends of the link a and b. Then we route from the source to a using DCellRouting on a DCell$_{L-1}$, then take the L-level link, and finally route from b to the destination, again using DCellRouting on a DCell$_{L-1}$.

The only aspect of this algorithm that is connection-specific is the computation of a and b. For simple rules such as α, β, γ, and δ, this is a quick computation. If random connection rules are used, this would necessarily involve a lookup in a potentially very large table, which would significantly detract from the usefulness of the algorithm.

2) Path-Length Distribution: As shown in [7], the longest path using DCellRouting is $2^{k+1}-1$. In fact, the proof is essentially identical to that of theorem 3.

Fix a vertex v in a DCell$_k$ and let N_i^k denote the number of servers that are exactly i hops away from v in DCellRouting. It turns out that N_i^k is independent of the choice of v, as the following theorem shows. It is remarkable that DCellRouting is so symmetric, given how asymmetric it is possible for a DCell to be, especially for the α connection rule.

Theorem 7. *N_i^k satisfies*

$$N_0^k = 1, \tag{67}$$

$$N_i^0 = \delta_{i0} + (n-1)\delta_{i1}, \tag{68}$$

$$N_i^k = N_i^{k-1} + \sum_{j=0}^{i-1} N_j^{k-1} N_{i-1-j}^{k-1}\ , \qquad \text{for } k, i \geq 1. \tag{69}$$

Here δ_{ij} is the Kronecker delta, which is 1 if $i = j$ and 0 otherwise.

Proof: Equations (67) and (68) are just the boundary conditions: there is only one vertex at distance 0 from v, namely v itself; and in a DCell$_0$, all other vertices are at distance 1, since a DCell$_0$ is a complete graph.

Equation (69) is explained as follows: The number of vertices i hops away is equal to the number of vertices i hops away that are in the same DCell$_{k-1}$ as v plus the number of vertices i hops away that are in a different DCell$_{k-1}$. The former number is just N_i^{k-1}. For the latter number, we know that exactly one of the hops must be a k-level link, since we are using DCellRouting. The sum displayed counts for each j the number of vertices we can reach by making j hops in the DCell$_{k-1}$ that v is in, then taking a k-level link, and finally making $(i-1)-j$ more hops in the other DCell$_{k-1}$. ∎

For $n = 2$, these numbers are described in [16]. Following the proof in [11], we arrive at the following result.

Theorem 8. $N_i^k = [x^{i+1}]f_k(x)$, *where*

$$f_0(x) = x + (n-1)x^2, \tag{70}$$

$$f_k(x) = f_{k-1}(x) + (f_{k-1}(x))^2, \qquad \text{for } k \geq 1. \tag{71}$$

In words: For fixed k, the sequence N_i^k has a generating function f_k satisfying the above recurrence relation.

Proof: Boundary condition (68) is satisfied because of equation (70). From (70) and (71) it is clear that $[x]f_k(x) = 1$ for all k, which means that boundary condition (67) is also satisfied. Finally, by (71) we have

$$[x^{i+1}]f_k(x) = [x^{i+1}]f_{k-1}(x) + [x^{i+1}](f_{k-1}(x))^2 \tag{72}$$

$$= [x^{i+1}]f_{k-1}(x) + \sum_{j=0}^{i+1}[x^j]f_{k-1}(x)[x^{i+1-j}]f_{k-1}(x) \tag{73}$$

$$= [x^{i+1}]f_{k-1}(x) + \sum_{j=0}^{i-1}[x^{j+1}]f_{k-1}(x)[x^{(i-1-j)+1}]f_{k-1}(x), \tag{74}$$

where for the last inequality we changed the summation index $j \mapsto j+1$ and used the fact that $[x^0]f_k(x) = 0$ for all k. This shows that the recurrence (69) is satisfied and concludes the proof. ∎

Using theorem 8, we may easily compute some special cases of N_i^k.

Corollary 3. *The following holds:*

1) $N_1^k = n - 1 + k$.
2) $N_{2^{k+1}-2}^k = 2^k(n-1)^{2^k-1}$.
3) $N_{2^{k+1}-1}^k = (n-1)^{2^k}$.

Proof: (1) is just the valency of a DCell_k. (3) and (2) are the leading two coefficients of $\left(x + (n-1)x^2\right)^{2^k}$. ∎

Now we turn to some empirical studies of the mean and mode of the DCellRouting path length distribution. Table V shows the ratio of the mode to the diameter $2^{k+1}-1$. It appears to approach a constant depending only on n. This was also observed for the special case $n = 2$ in [17]. The same appears to be true for the variance of the distribution—see table VI.

Table VII shows the ratio of the mean to the mode. For sufficiently large k, they are nearly identical. So it would appear that a good estimate for the one would immediately lead to a good estimate for the other.

Conjecture 2. *There exist constants a_n and b_n such that $\frac{1}{2} < a_n < 1$, $\lim_{n\to\infty} a_n = 1$, and*

$$\textit{mean} = (a_n + o(1)) \cdot 2^{k+1}, \tag{75}$$

$$\textit{mode} = (a_n + o(1)) \cdot 2^{k+1}, \tag{76}$$

$$\textit{variance} = (b_n + o(1)) \cdot 2^{k+1}. \tag{77}$$

for DCellRouting.

$n\backslash k$	2	3	4	5	6	7	8
2	0.5714	0.5333	0.5806	0.5714	0.5748	0.5725	0.5734
3	0.7143	0.7333	0.7097	0.6984	0.7008	0.6980	0.6986
4	0.7143	0.8000	0.7742	0.7778	0.7717	0.7686	0.7691
5	0.8571	0.8667	0.8387	0.8254	0.8110	0.8118	0.8121
6	1.0000	0.8667	0.8710	0.8413	0.8425	0.8431	0.8415
7	1.0000	0.8667	0.8710	0.8730	0.8661	0.8667	0.8630
8	1.0000	0.8667	0.9032	0.8889	0.8819	0.8824	0.8806

TABLE V: Ratio of mode to diameter for DCellRouting.

$n\backslash k$	2	3	4	5	6	7	8
2	0.3138	0.3556	0.3477	0.3422	0.3395	0.3382	0.3375
3	0.3187	0.3258	0.3157	0.3107	0.3083	0.3070	0.3064
4	0.2867	0.2810	0.2720	0.2677	0.2656	0.2646	0.2640
5	0.2532	0.2432	0.2354	0.2317	0.2298	0.2289	0.2285
6	0.2242	0.2132	0.2063	0.2030	0.2014	0.2006	0.2002
7	0.2002	0.1892	0.1831	0.1802	0.1788	0.1781	0.1778
8	0.1804	0.1699	0.1644	0.1618	0.1605	0.1599	0.1596

TABLE VI: Ratio of variance to diameter for DCellRouting.

3) All-To-All Communication:

Theorem 9. *In all-to-all communication using DCellRouting, the number of flows F_L carried by a L-level link is*

$$F_L = \begin{cases} t_{k-1}^2 & \text{for } L = k, \\ t_{L-1}^2 \prod_{j=L}^{k-1}(1+2t_j) & \text{for } 1 \leq L \leq k-1, \\ (n-1)\prod_{j=0}^{k-1}(1+2t_j) & \text{for } L = 0. \end{cases} \tag{78}$$

Proof: For convenience, define a DCell_{-1} to be a single node (so that $t_{-1} = 1$) and consider an abstracted DCell_0 represented by a complete graph. Each physical 0-level link then corresponds to $n-1$ abstract 0-level links, and so we need only multiply F_0 by $n-1$ in the end to get the physical number of flows carried by a 0-level link. (Note that for $n = 2$, the abstract and physical representations are identical from a graph-theoretic point of view.)

Consider an L-level link. We prove the expression for F_L by induction on k. First suppose $L = k$. Then note that a k-level link carries precisely all the flows between the two DCell_{k-1}'s it connects. Since each DCell_{k-1} has t_{k-1} servers, the link must support t_{k-1}^2 flows.

Now suppose that $k > L$ and that the expression for F_L holds for $k-1$. Since we are using DCellRouting, the only additional flows carried by the link as k increases by 1 are those that have one server in a different DCell_{k-1} and then have to cross the link. Each pair of vertices utilizing the link in the original DCell_{k-1} thus leads to $2t_{k-1}$ further flows, since each of the vertices has one k-level link that will be utilized by all t_{k-1} vertices in the DCell_{k-1} that the link connects to. So we see that the expression for $k-1$ has to be increased by a factor of $(1+2t_{k-1})$. ∎

Once again, it is remarkable that the number of flows carried by a link depends only on the link level in DCellRouting. This is another example of the symmetry of DCellRouting on top of a potentially highly asymmetric DCell.

$n\backslash k$	2	3	4	5	6	7	8
2	0.9329	1.0228	0.9639	0.9917	0.9918	0.9987	0.9987
3	0.9277	0.9257	0.9711	0.9938	0.9939	0.9995	0.9995
4	1.0325	0.9405	0.9821	0.9825	0.9927	0.9978	0.9978
5	0.9166	0.9215	0.9600	0.9792	0.9985	0.9985	0.9985
6	0.8191	0.9582	0.9598	0.9968	0.9968	0.9968	0.9991
7	0.8436	0.9849	0.9855	0.9858	0.9949	0.9949	0.9994
8	0.8623	1.0053	0.9692	0.9870	0.9959	0.9960	0.9982

TABLE VII: Ratio of expected value to mode for DCellRouting.

Using theorems 9 and 1, we can derive from the exact expression for F_L a fairly tight upper bound that is more readily compared to the previously known bound $2^{k-L}t_k$.

Corollary 4. *We have*

$$F_0 < \frac{n-1}{n+\frac{1}{2}}2^k(t_k+0.6) = \frac{n-1}{n+\frac{1}{2}}2^k t_k(1+o(1)). \quad (79)$$

For $1 \le L < k$, we have

$$F_L < \frac{t_L - t_{L-1}}{t_L+\frac{1}{2}}2^{k-L}(t_k+0.6) = \frac{t_L - t_{L-1}}{t_L+\frac{1}{2}}2^{k-L}t_k(1+o(1)). \quad (80)$$

Proof: We know from theorem 1 that $t_k + 0.6 > c^{2^k} > t_k + \frac{1}{2}$. It follows that $1 + 2t_j < 2c^{2^k}$, and hence

$$\begin{aligned} F_0 &= (n-1)(1+2t_0)\cdots(1+2t_{k-1}) && (81)\\ &< (n-1)2^k c^{2^0}\cdots c^{2^{k-1}} && (82)\\ &= (n-1)2^k c^{2^0+2^1+\cdots 2^{k-1}} && (83)\\ &= (n-1)2^k c^{2^k-1} && (84)\\ &= \frac{n-1}{c}2^k c^{2^k} && (85)\\ &< \frac{n-1}{n+\frac{1}{2}}2^k(t_k+0.6) && (86)\\ &= \frac{n-1}{n+\frac{1}{2}}2^k t_k(1+o(1)). && (87) \end{aligned}$$

The expression for F_L is found using the same approximations. ∎

Note that, as n gets large, this bound asymptotically approaches the bound of $2^{k-L}t_k$ proved in [7]. In figure 4, our exact expression for F_L is compared graphically with the upper bound $2^{k-L}t_k$. As we can see, F_L indeed approaches the upper bound for large n. Also, we see that the load balancing across link levels is better for small n.

Finally, we show that the expected value of the path-length distribution is related to the flow distribution.

Theorem 10. *The expected value of the path-length distribution is given by*

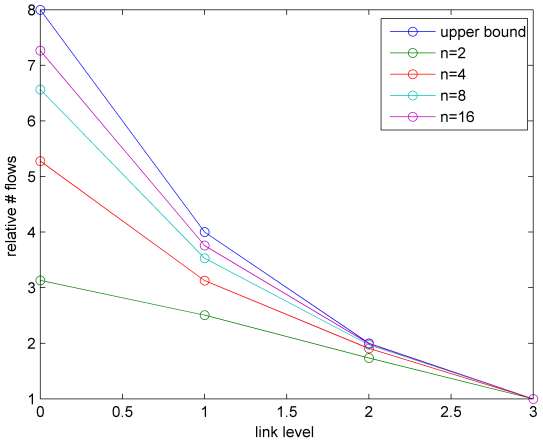
$$E = \frac{\sum_{L=0}^{k} F_L}{t_k - 1}. \quad (88)$$

Proof: Let $d(u,v)$ denote the distance between nodes u and v in DCellRouting. A flow between u and v is carried by $d(u,v)$ edges in DCellRouting. So the sum of $d(u,v)$ over all distinct u and v is equal to twice the sum over all edges of the number of flows carried by the edge. Since there are $t_k/2$ links of each level, we have

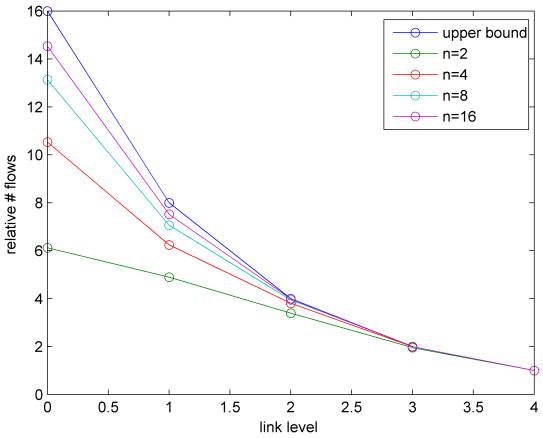
$$\sum_{u\neq v} d(u,v) = 2\sum_{L=0}^{k} F_L\left(\frac{t_k}{2}\right). \quad (89)$$

The theorem follows upon dividing both sides by $t_k(t_k-1)$, the number of pairs (u,v) with $u \neq v$. ∎

As we have seen, F_L asymptotically approaches $2^{k-L}t_k$ for large n and $L < k$. Furthermore, noting that $F_k = t_{k-1}^2 =$



(a) $k = 3$



(b) $k = 4$

Fig. 4: Number of flows by link level in all-to-all communication using DCellRouting for various n and k. Also shown is the upper bound from [7]. The numbers are relative to the number of flows carried by a k-level link.

n	k	DCR	SP-α	SP-β	SP-γ	SP-δ
2	2	3.73	3.48	3.50	3.46	3.46
4	2	5.16	4.87	4.71	4.68	4.67
6	2	5.73	5.48	5.30	5.26	5.28
8	2	6.04	5.82	5.66	5.59	5.64
2	3	8.18	6.95	6.58	6.44	6.49
4	3	11.29	9.96	8.99	8.68	8.81

(a) Mean

n	k	DCR	SP-α	SP-β	SP-γ	SP-δ
2	2	1.48	1.23	1.25	1.23	1.23
4	2	1.42	1.27	1.15	1.12	1.13
6	2	1.25	1.18	1.09	1.05	1.08
8	2	1.12	1.09	1.04	1.00	1.04
2	3	2.31	1.63	1.41	1.32	1.37
4	3	2.05	1.64	1.22	1.08	1.14

(b) Standard deviation

TABLE VIII: Expected value and standard deviation of path length distribution. DCR and SP stand for DCellRouting and shortest path routing, respectively.

$t_k - t_{k-1}$, we see that

$$E \leq \frac{\sum_{L=0}^{k-1} 2^{k-L} t_k + (t_k - t_{k-1})}{t_k - 1} \tag{90}$$

$$= \frac{2^{k+1} t_k - t_{k-1}}{t_k - 1} \tag{91}$$

$$\approx 2^{k+1} \left(1 - \frac{t_{k-1} - 1}{t_k}\right) \tag{92}$$

$$= 2^{k+1}(1 - o(1)), \tag{93}$$

with asymptotic agreement. This agrees with our claim in conjecture 2 that

$$\lim_{n \to \infty} a_n = 1. \tag{94}$$

V. Simulation

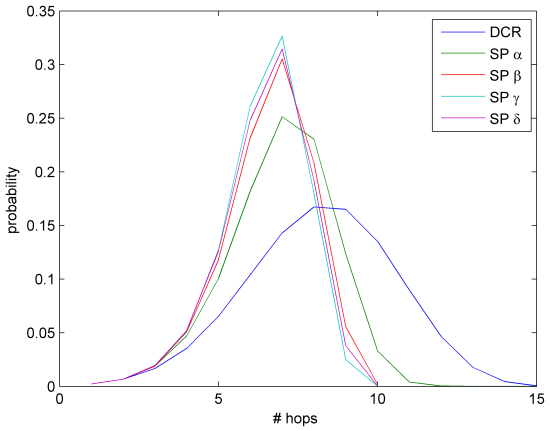
In this chapter, we compare empirically the performance of DCellRouting and shortest path routing for the various connection rules. The simulations were necessarily restricted to small n and k; but given the doubly exponential growth of DCells, these are the only realistic values anyway.

A. Path-Length Distribution

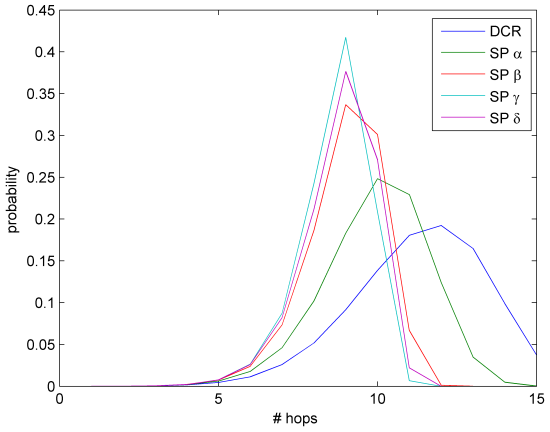
Table VIII compares, for some small n and k, the mean and standard deviation of the path length distribution when using DCellRouting or shortest path routing. Shortest path routing for the γ connection rule has the lowest expected value and standard deviation, making it the rule of choice for shortest path routing. Figure 5 shows the different path length distributions for the two $k = 3$ cases.

B. Flow Distribution

The flow distributions by link level using shortest path routing and DCellRouting are shown in figure 6 for $n = 2, k = 3$, and in figure 7 for $n = 4, k = 3$. We observe that DCellRouting does a poor job of load-balancing. Shortest path routing for α-DCell does better than DCellRouting on



(a) $n = 2, k = 3$



(b) $n = 4, k = 3$

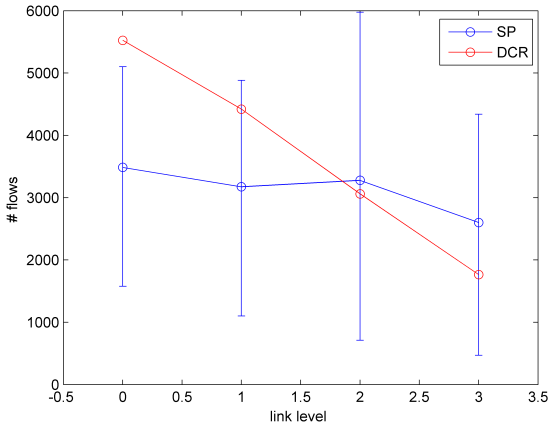
Fig. 5: Path length distributions for $k = 3$. DCR and SP stand for DCellRouting and shortest path routing, respectively.

average, but has significant bottlenecks that exceed even those of DCellRouting. Shortest path routing for the β, γ, and δ connection rules does better on average and also exhibits very good load-balancing: there are no significant bottleneck links. It appears that γ is again the rule of choice for all-to-all communication using shortest path routing.

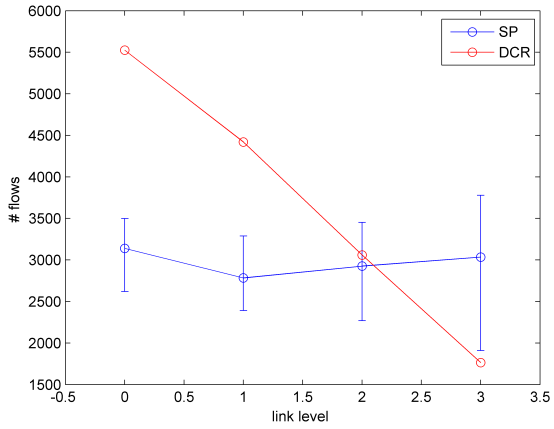
VI. Conclusion

In constructing generalized DCells, we have exhibited a new family of graphs that could be useful for large-scale data center networks. We have proven that almost all properties of the original DCell design carry over to the generalized DCell, and in many cases we have provided improved bounds or even exact expressions that were not formerly known. Furthermore, we have presented an adapted version of DCellRouting as well as an autoconfiguration algorithm that make the use of generalized DCells feasible for real networks.

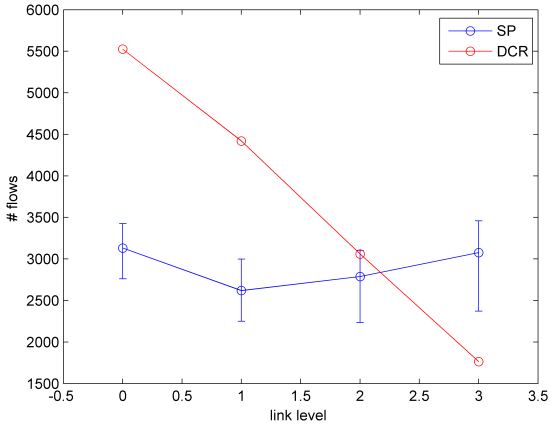
Furthermore, we have proposed three specific new instances of the generalized DCell family, termed β, γ, and δ, and we



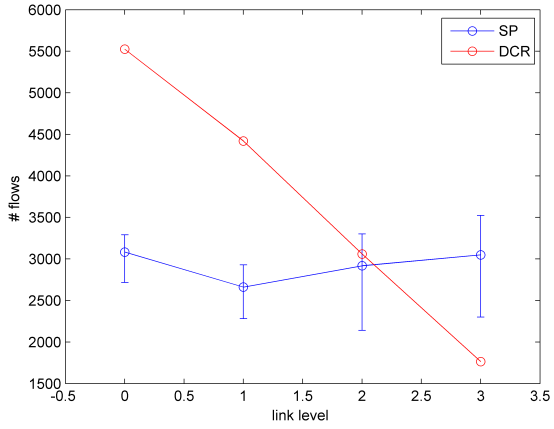
(a) α-DCell



(b) β-DCell



(c) γ-DCell



(d) δ-DCell

Fig. 6: Distribution of flows by link level using all-to-all communication for $n = 2$ and $k = 3$. DCR and SP stand for DCellRouting and shortest path routing, respectively. The error bars indicate the maximum and minimum values.

have shown that these may have more desirable properties than the original α-DCell design. In particular, we emphasize that the β, γ, and δ designs

1) exhibit significantly better load-balancing properties in all-to-all communication using shortest path routing than the original design;
2) are a lot more symmetric than the original design.

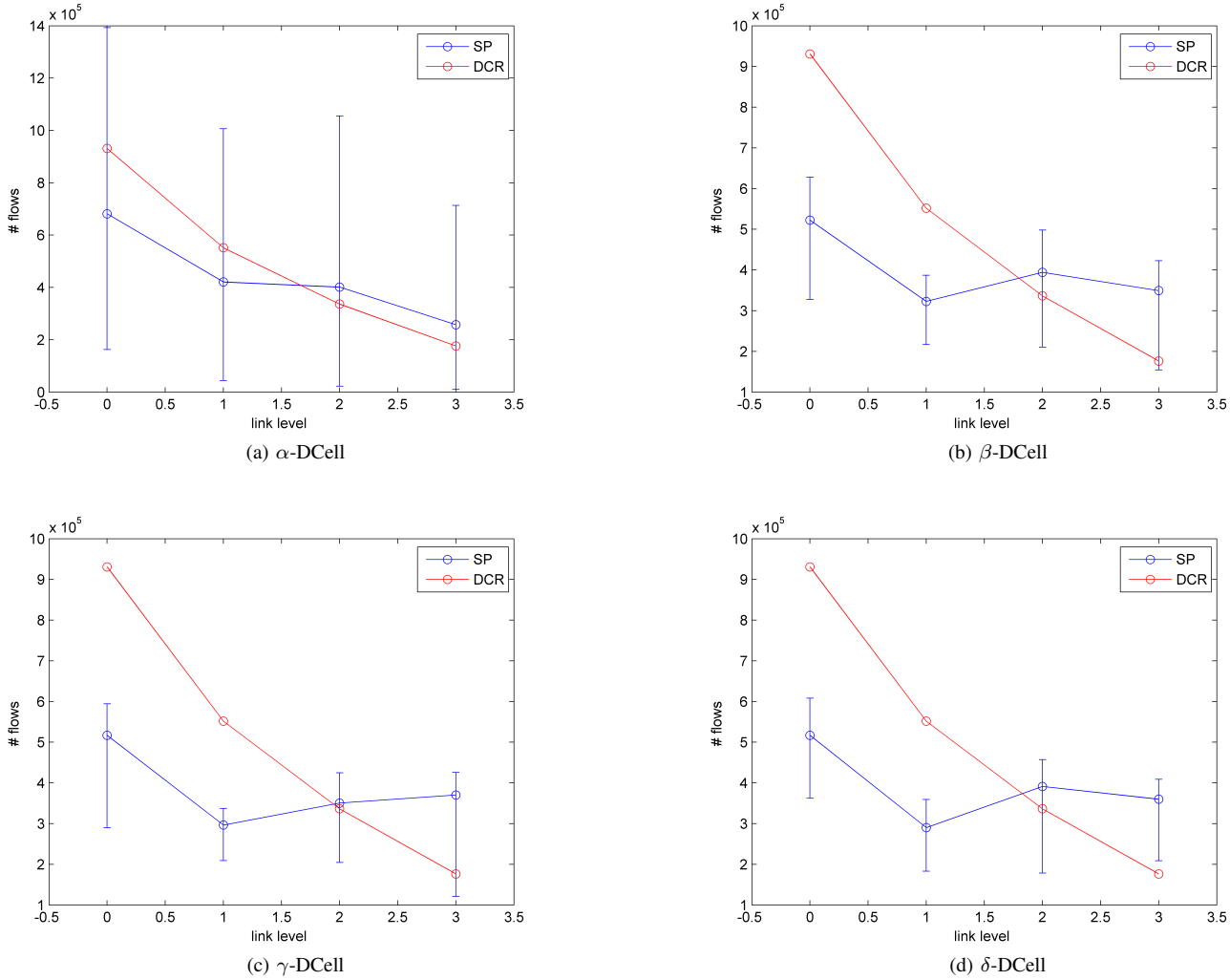
Point 1 is of importance to data center networks in general and to applications such as MapReduce [3] in particular. The importance of point 2 is two-fold. First, more symmetry should ease the wiring of the network. Second, heuristically speaking, more symmetry means more regularity, and this increased regularity, we hope, will facilitate the design of new algorithms for DCell networks.

Our future research will focus on the following questions.

1) What other connection rules are possible, and what properties should they have to make them most useful for data center networks?
2) Is there an efficient, load-balancing routing algorithm for one of the new connection rules? Shortest path routing is only feasible for small networks, and without a new routing algorithm we may not be able to reap the benefits of the specific connection rule.
3) What are the minimal requirements for (robust) auto-configuration? Can we draw on properties of a specific connection rule to design a better autoconfiguration algorithm? For example, the original DCell is highly asymmetric and in principle nearly perfect autoconfiguration should thus be possible. But so far we have not found an efficient way of exploiting the asymmetry.

VII. Acknowledgement

This work was performed when Markus Kliegl, Jason Lee, Jun Li, and Xinchao Zhang were visiting students and David Rincón was a visiting academic mentor for the MSRA and UCLA IPAM RIPS-Beijing 2009 program at Microsoft Research Asia. The four students are equal contributors. Funding was provided by the NSF and MSRA.



(a) α-DCell (b) β-DCell

(c) γ-DCell (d) δ-DCell

Fig. 7: Distribution of flows by link level using all-to-all communication for $n = 4$ and $k = 3$. *DCR* and *SP* stand for DCellRouting and shortest path routing, respectively. The error bars indicate the maximum and minimum values.

REFERENCES

[1] A. V. Aho and N. J. A. Sloane. Some Doubly Exponential Sequences. *Fibonacci Quarterly*, Vol. 11 (1970), pp. 429–437.

[2] F. K. Chung. *Spectral Graph Theory.* CBMS no. 92. AMS, 1997.

[3] J. Dean and S. Ghemawat. MapReduce: Simplified Data Procesing on Large Cluster. In *OSDI'04*, 2004.

[4] C. Godsil and G. Royle. *Algebraic Graph Theory.* Springer, 2001.

[5] A. Ferencz, R. Szewczyk, J. Weinstein, and J. Wilkening. Graph Bisection. Final Report. University of California at Berkeley, 1999.

[6] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers In *ACM SIGCOMM'09*, 2009.

[7] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers. In *ACM SIGCOMM'08*, 2008.

[8] G. Jäger. An Efficient Algorithm for Graph Bisection of Triangularizations. *Applied Mathematical Sciences* **25**(1):1203-1215, 2007.

[9] B. W. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, Vol. 49 (1970), pp. 291–307.

[10] F. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays. Trees. Hypercubes.* Morgan Kaufmann, 1992.

[11] J. B. Lewis. *The Art of Problem Solving*, 2008. http://www.artofproblemsolving.com/Forum/viewtopic.php?t=203662

[12] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, and S. Lu. FiConn: Using Backup Port for Server Interconnection in Data Centers. In *IEEE INFOCOMM'09*, 2009.

[13] B. Mohar. Eigenvalues, diameter, and mean distance in graphs. *Graphs and Combinatorica* **7**(1), 1991.

[14] J. T. Moy. *OSPF: Anatomy of an Internet Routing Protocol.* Addison-Wesley Professional, 1998.

[15] N. J. A. Sloane, Ed. Sequence A007018. *The On-Line Encyclopedia of Integer Sequences*, 2009.

[16] *Ibid.* Sequence A122888.

[17] *Ibid.* Sequence A122893.

[18] J. Snyder. Microsoft: Datacenter Growth Defies Moore?s Law. *PCWorld*, 2007. http://www.pcworld.com/article/130921/microsoft_datacenter_growth_defies_moores_law.html