Graph-Theoretic Analysis of Structured Peer-to-Peer Systems: Routing Distances and Fault Resilience

Dmitri Loguinov, *Member, IEEE*, Juan Casas, and Xiaoming Wang, *Student Member, IEEE*

***Abstract*—This paper examines graph-theoretic properties of existing peer-to-peer networks and proposes a new infrastructure based on optimal-diameter de Bruijn graphs. Since generalized de Bruijn graphs exhibit very short average distances and high resilience to node failure, they are well suited for distributed hash tables (DHTs). Using the example of Chord, CAN, and de Bruijn, we study the routing performance, graph expansion, clustering properties, and bisection width of each graph. Having confirmed that de Bruijn graphs offer the best diameter and highest connectivity among the existing peer-to-peer structures, we offer a very simple incremental building process that preserves optimal properties of de Bruijn graphs under uniform user joins/departures. We call the combined peer-to-peer architecture *optimal diameter routing infrastructure*.**

***Index Terms*—De Bruijn graphs, diameter-degree tradeoff, peer-to-peer networks.**

I. Introduction

OVER the last few years, peer-to-peer networks have rapidly evolved and have become an important part of the existing Internet culture. All current peer-to-peer proposals are built using application-layer overlays, each with a set of graph-theoretic properties that determine its routing efficiency and resilience to node failure. Graphs in peer-to-peer networks range from star-like trees (centralized approaches such as Napster) to complex k-node-connected graphs (such as Chord [42], CAN [33], and Pastry [37]). The performance of each peer-to-peer architecture is determined by the properties of these graphs, which typically possess $\Theta(\log N)$ diameter and $\Theta(\log N)$ degree at each node (where N is the number of peers in the system). Until recently [13], [20], [30], understanding whether these bounds were optimal and whether there existed *fixed-degree* graphs with $\Theta(\log N)$ diameter was an important topic of distributed hash table (DHT) research [34], [46].

In the first part of this paper, we examine the problem of obtaining a logarithmic routing diameter in fixed-degree peer-to-peer networks. Our work relies on generalized de Bruijn graphs [19] of degree k and asymptotically optimal diameter $\log_k N$. However, since the diameter itself does not tell the whole story, we also study the *average* distances between all pairs of nodes since this metric (rather than the diameter) often determines the responsiveness and capacity[1] of a peer-to-peer network. We also study the optimality of greedy routes constructed by each graph and compare them to those obtained through BFS.

We next analyze clustering and small-world properties of several P2P networks and explain how they relate to graph expansion. We derive that de Bruijn graphs have an order of magnitude smaller clustering coefficients than Chord, which partly explains the differences in expansion, fault tolerance, and diameter between the two graphs. We then study the resilience of these networks against node failure, or simply their *connectivity*. In general, connectivity determines the number and location of failures that a graph can tolerate without becoming disconnected. We focus on the edge bisection width of each graph and demonstrate that de Bruijn graphs are several times more difficult to disconnect than the traditional approaches.

Having confirmed that de Bruijn graphs offer an appealing framework for P2P systems, we provide an algorithm called *optimal diameter routing infrastructure* (ODRI) for building and load-balancing such graphs incrementally as peer nodes join/leave the system. We conclude the paper by showing that under uniform user joins, the diameter of ODRIs *peer-to-peer* graph remains asymptotically optimal, and its degree can be bounded with a proper choice of load-balancing steps.

The paper is organized as follows. Section II discusses the background. Section III introduces static de Bruijn graphs and shows their advantages over earlier proposals. Section IV derives the average distance of Chord, CAN, and de Bruijn and studies their greedy routing. Section V examines clustering/expansion of each graph, and Section VI investigates their bisection width. Section VII introduces distributed de Bruijn graphs and ODRI. Section VIII concludes the paper.

II. Background

Many current peer-to-peer networks [33], [37], [42], [47] are based on DHTs, which provide a decentralized, many-to-one mapping between user objects and peers. This mapping is accomplished by organizing the peers in some virtual coordinate space and hashing each object to these virtual coordinates. The information about each object (such as the IP address of the owner) is kept by the peer to whose coordinates the object hashes. Their distributed structure, excellent scalability, short routing distances, and failure resilience make DHTs highly suitable for P2P networks.

Manuscript received November 12, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Govindan. This work was supported by the National Science Foundation under Grants CCR-0306246, ANI-0312461, CNS-0434940, and REU-0353957.

D. Loguinov and X. Wang are with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: dmitri@cs.tamu.edu; xmwang@cs.tamu.edu).

J. Casas is with the Department of Computer Science, University of Texas—Pan American, Edinburg, TX 78539 USA (e-mail: jcasas@rgv.rr.com).

Digital Object Identifier 10.1109/TNET.2005.857072

[1]Capacity is a term used in wireless ad-hoc networks [17] to measure the maximum throughput of a network under all-to-all communication.

A. Peer-To-Peer DHTs

Many current DHTs [37], [39], [47] rely on the concept of prefix-based routing introduced by Plaxton *et al.* in [32]. Plaxton's framework is extended in [47] and [37] to accommodate dynamic join/departure of peers and provide necessary failure-recovery mechanisms. Using an alphabet of size b, both approaches build a k-regular graph of diameter $\log_b N$ and node degree $(b-1)\log_b N, b \geq 2$. Among other methods, Ratnasamy *et al.* [33] propose a peer-to-peer architecture called *content-addressable network* (CAN) that maps the DHT to a d-dimensional Cartesian space. CAN's diameter is $dN^{1/d}/2$ and the degree of each node is $2d$. Stoica *et al.* [42] propose a distributed graph called Chord that uses a ring with diameter and degree both equal to $\log_2 N$.

Recent proposals start to address the issue of routing in logarithmic time in *fixed-degree* graphs. For example, Considine *et al.* [9] expand on Chord's ring structure by constructing a digraph (directed graph) of fixed degree; however, the proposed structure needs to estimate the number of active nodes to properly build the application-layer graph. Among tree-based structures, Freedman *et al.* [14] propose a DHT based on distributed tries and Tran *et al.* [44] organize peers into a multicast tree of degree $O(k^2)$ and diameter $O(\log_k N)$. Xu *et al.* [46] study diameter-degree tradeoffs of current DHTs and propose a graph based on a modified static butterfly. Another peer-to-peer architecture based on butterfly networks (*Viceroy*) is shown in [26].

Independently of this work, several recent papers have also proposed de Bruijn graphs for peer-to-peer networks [13], [20], [30]. These developments are complementary to our investigation and provide implementation details and additional analysis not covered in this paper. For example, Koorde [20] has a different set of linking/routing rules for incomplete de Bruijn graphs, D2B [13] extensively studies the binary version of the graph, and distance halving [30] shifts node labels in the reverse direction (i.e., left to right).

B. Fault Tolerance of DHTs

Fault tolerance of peer-to-peer networks is an equally important topic. Liben-Nowell *et al.* [24] examine error resilience dynamics of Chord when nodes join/leave the system and derive lower bounds on the rate of neighbor acquisition necessary to maintain a connected graph with high probability. Fiat *et al.* [12] build a *censorship-resistant network* that can tolerate massive adversarial node failures and random object deletions. Saia *et al.* [38] create another highly fault-resilient structure with $O(\log^3 N)$ state at each node and $O(\log^3 N)$ per-message routing overhead. Gummadi *et al.* [16] find that ring-based graphs (such as Chord) offer more flexibility with route selection and provide better performance under random node failure compared to several other traditional DHTs.

C. Random Graphs

Another direction for building DHTs relies on properties of random graphs. The main thrust in this area is to build logarithmic-time routing structures with constant degree. Pandurangan *et al.* [31] propose a random DHT graph with a constant degree and (almost certainly) logarithmic diameter; however, the paper does not provide an efficient routing algorithm for the proposed structure that can deterministically explore the low diameter of the graph. Aspnes *et al.* [1] examine random graphs of fixed degree $l+1$ and derive upper and lower bounds on the expected routing distance in such graphs. Their results show that both bounds are proportional to $\log^2 N/(l \log\log N)$. Manku [27] considers random graphs of degree $3l+3$ and asymptotically optimal expected distance $O(\log N/\log l)$. Law *et al.* [22] build random expander graphs based on Hamiltonian cycles with $O(\log N)$ diameter and $O(\log N)$ degree. Manku *et al.* [29] analyze several randomized systems (i.e., Randomized Chord, Randomized Hypercube, and Symphony [28]) and conclude that the usage of neighbors-of-neighbors (NoNs) in routing decisions reduces the average distance in the corresponding graph from $\Theta(\log N)$ to $\Theta(\log N/\log\log N)$ with high probability.

D. Optimal-Diameter Graphs

The problem of designing an optimal-diameter graph of fixed degree has been extensively studied in the past. In one formulation of this problem, assume a graph of fixed degree k and diameter D (the maximum distance between any two nodes in the graph). What is the maximum number of nodes N that can be packed into any such graph? A well-known result is the *Moore bound* [7], [8]

$$N \leq 1 + k + k^2 + \cdots + k^D = \frac{k^{D+1}-1}{k-1} = N_M. \quad (1)$$

Interestingly, the Moore bound N_M is only achievable for trivial values of k and D. In fact, the Moore bound is *provably* not achievable for any nontrivial graph [7]. Directed de Bruijn graphs come close to the Moore bound and can be built with $N = k^D$ nodes [19] or even with $N = k^D + k^{D-1}$ nodes [35]; however, in general, it is not known how close we can approach the upper bound N_M for nontrivial graphs [8]. In the context of peer-to-peer DHTs, we are concerned with a different formulation of the problem: given N nodes and fixed degree k, what is the minimum diameter in any graph built on top of these N nodes? The answer follows from (1) as

$$D \geq \lceil \log_k(N(k-1)+1) \rceil - 1 = D_M. \quad (2)$$

Imase and Itoh [19] construct nearly optimal de Bruijn graphs of diameter $D = \lceil \log_k N \rceil$, which is at most $D_M + 1$; however, for large k, the two diameters become asymptotically equal. In this paper, we use the same basic algorithms [19] even though they can be slightly improved [35] at the cost of losing greedy shortest-path routing.

Another important metric related to the routing performance of a graph is its average distance μ_d between any pair of nodes.[2] The lower bound on μ_d in any k-regular graph is given by the average distance in the corresponding Moore graph and is also not achievable for nontrivial values of N and k [40]

$$\mu_d \geq D_M - \frac{k(k^{D_M}-1)}{N(k-1)^2} + \frac{D_M}{N(k-1)} \approx D_M - \frac{1}{k-1}. \quad (3)$$

[2]Note that we include the distance from each node to itself in μ_d while some of the related work does not.

With respect to μ_d, de Bruijn graphs are again asymptotically optimal and converge to the bound in (3) for sufficiently large N and k [40].

III. De Bruijn Graphs

A. Motivation

One of the goals of this work is to build a DHT on top of fixed-degree graphs with provably optimal routing diameter. Since nontrivial Moore graphs do not exist [7], we use de Bruijn graphs [19] of diameter $\lceil \log_k N \rceil$ and often call them "optimal" since, among the class of practically useful graphs, they *are* optimal. To illustrate the impressive reduction in diameter compared to the classical DHT structures, assume one million nodes and degree k fixed at $\lceil \log_2 N \rceil = 20$. Under these circumstances, Chord offers a graph with diameter D equal to $\lceil \log_2 N \rceil = 20$, while a de Bruijn graph with the same number of neighbors has a diameter four times smaller: $D = \lceil \log_{20} N \rceil = \lceil 4.61 \rceil = 5$. Note that the diameter of the corresponding Moore graph is essentially the same: $D_M = \lceil 4.59 \rceil = 5$. The reduction in the average distance is not as impressive, but nevertheless significant: $\mu_d = 10$ hops in Chord and 4.6 in de Bruijn.

Throughout the paper, we are concerned with the properties of the underlying graph of each peer-to-peer network. Consequently, we examine the diameter and resilience of these graphs assuming that the hashing function equally spreads users along the DHT space and that all graphs are populated with the maximum number of nodes (this assumption is relaxed in Section VII). We further assume, for simplicity of notation, that the total number of nodes N is a power of node degree and omit ceiling functions whenever appropriate.

B. Structure

De Bruijn graphs [6], [19], [23], [40] are nearly optimal, fixed-degree digraphs of diameter $\log_k N$, where k is the fixed degree of each node and N is the total number of nodes. Note that de Bruijn graphs are *directed* graphs with k outgoing and k incoming edges at each node, which also holds for many current DHTs [37], [42], [47]. Assume that each node x is hashed to a string H_x drawn from some alphabet Σ of size k. The classical directed de Bruijn graph [19] contains $N = k^D$ nodes where D is the diameter of the graph. Each node H_x in the graph is a string $(h_1, \ldots, h_D)$ of length D linked to k other nodes $(h_2, \ldots, h_D, \alpha)$, for all possible $\alpha \in \Sigma$. For examples and discussion of routing rules, see [25].

C. Comparison With Existing Graphs

In this section, we briefly examine diameter-degree tradeoffs of the existing protocols and compare them to those of de Bruijn graphs. We leave a thorough analysis of numerous recently proposed graphs [15], [22], [26], [28], [44], [46] for future work and conduct a detailed study of two classical approaches (Chord [42] and CAN [33]) in Sections IV–VI. This section also shows results for Pastry and the static butterfly graph (without detailed analysis). Note that our treatment of the butterfly follows the traditional definition [23], which is the basis of two recent proposals, Viceroy [26] and Ulysses [46]; however, neither of these two graphs exactly implements the classic butterfly. Therefore, as we discuss below, the individual diameter-degree tradeoffs of these approaches are slightly different from that in the classic graph. We further remove all fault-resilient additions of each structure (such as the predecessor pointer and r-element successor list in Chord) and only analyze the "raw" performance of each graph.

TABLE I
Asymptotic Degree/Diameters of Popular Graphs

Graph	Degree	Diameter D
de Bruijn	k	$\log_k N$
Trie	$k+1$	$2\log_k N$
Chord	$\log_{1+1/d} N$	$\log_{1+d} N$
CAN	$2d$	$dN^{1/d}/2$
Pastry / Tapestry	$(b-1)\log_b N$	$\log_b N$
Classic butterfly	k	$2\log_k N(1-o(1))$

As an illustration of fixed-degree tree structures, we also examine k-ary tries as they have been recently proposed for DHTs [14]. A k-ary tree uses prefix-based routing over a tree where each parent maintains k children, one child for each symbol in the alphabet. Consequently, the maximum degree of any node in the trie is $k+1$ and the diameter of the graph is $2\lceil \log_k N \rceil$ (i.e., the distance to the root and back).

Finally, recall that the traditional butterfly network contains $N = mk^m$ nodes (where k is again the degree of each node) and has diameter $D = 2m - 1$. Notice that D can be expressed in terms of degree k using Lambert's function W [23]

$$D = 2m - 1 = 2\frac{W(N \ln k)}{\ln k} - 1 = 2\log_k N(1 - o(1)) \quad (4)$$

which is asymptotically twice the diameter of de Bruijn graphs. Even though butterflies are appealing structures, there are nontrivial difficulties in building them as nodes join and leave the system. In one example, Viceroy [26] implements a binary butterfly with diameter $3\log_2 N$ and degree 7 and further requires estimation of the number of nodes in the system. In another example, Ulysses [46] adds $\log N$ neighbors to each node and is no longer a fixed-degree graph. As we show in Section VII, distributed de Bruijn graphs possess no more conceptual complexity than Chord, achieve optimal diameter in the peer-to-peer graph, and can be built with a fixed application-layer degree.

Table I shows asymptotic diameter and node degree of de Bruijn graphs and several existing structures. Even though Chord allows a generalization to $\log_{1+1/d} N$ neighbors and diameter $\log_{1+d} N$ [42], it is most frequently used with the default value of $d = 1$ studied throughout this paper. Also note that the tree maintains its *average* degree over all nodes equal to only 2 (since approximately $(k-1)/k$ fraction of the nodes are leaves); however, the imbalance in the middle of the tree with nodes of degree $k+1$ creates a rather pessimistic diameter-degree tradeoff.

We next examine the performance of these graphs in a hypothetical peer-to-peer system of $N = 10^6$ nodes. Table II shows the diameter of each graph as a function of its degree k. Notice that, for low-degree networks ($k \leq 20$), even the trie offers a better diameter than the three classical approaches (i.e., CAN,

TABLE II
GRAPH DIAMETER FOR $N = 10^6$ PEERS

k	de Bruijn	Trie	Chord	CAN	Pastry	Butterfly
2	20	–	–	5×10^5	–	31
3	13	40	–	–	–	20
4	10	26	–	1,000	–	16
10	6	13	–	40	–	10
20	5	10	20	20	20	8
50	4	8	10	–	7	7
100	3	6	7	–	5	5

Chord, and Pastry). In fact, the trie routes in *half* the time compared to the classical Chord (i.e., $d = 1$) or CAN. Also notice that de Bruijn graphs with $k = 20$ offer a diameter four times smaller than those in Chord or CAN. Furthermore, de Bruijn graphs can route between any pair of nodes in 20 hops with only two neighbors, which is ten times less than that required by CAN, Chord, or Pastry to achieve the same diameter. Finally, the traditional butterfly offers a diameter 50%–60% larger than in de Bruijn graphs, but 30%–60% smaller than in base-d Chord.

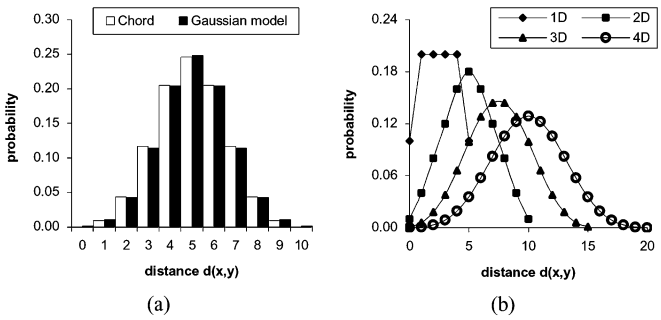
One interesting observation about CAN points to the fact that selection of the number of dimensions d is an important decision for a given number of nodes N. It is noted in [33] that d is likely to be fixed while N changes; however, as Table II shows, many small values of $d \ll \log_2 N$ result in greatly suboptimal diameters. This observation is easy to explain since CAN's diameter $dN^{1/d}/2$ is a strictly convex function with a unique minimum located at $d = \ln N$ (epeers per dimension). Keeping in mind that each dimension must contain an integer number of peers, the best practical diameter is achieved for $d = \log_3 N$. Thus, for $N = 10^6$, the optimal number of dimensions d is 12 ($k = 24$ neighbors) and the optimal diameter is 19. Additionally, note that, for $d = \log_2 N/2$, CAN's degree and diameter are *both* equal to that of Chord (this is shown in Table II for $k = 20$ and is noted in [33] and [42]).

Further examining Table II, notice that Pastry offers a good diameter only for large $b \gg 2$. In fact, to come within one hop of the optimal diameter for $N = 10^6$, Pastry requires *at least* 160 neighbors (not shown in the table). Such large routing tables may sometimes (i.e., over modem links) be impractical in the real Internet due to the high volume of traffic required to maintain peer-level connections and repair broken links when existing neighbors fail.

IV. ROUTING ANALYSIS

De Bruijn graphs have desirable properties for peer-to-peer networks that stem from their small diameter. However, the diameter of a graph is simply the largest distance between any pair of nodes, which only provides an *upper bound* on the number of hops between any pair of users. A much more balanced metric is the *average* distance between any pair of nodes since this is the performance a user can expect from the peer-to-peer system when searching for objects.

Define $d(x, y)$ to be the shortest distance (using greedy routing rules) between nodes x and y in a given graph. In what follows below, we first derive the probability mass function of $d(x, y)$ and then compute its expectation μ_d.



(a) (b)

Fig. 1. (a) Distribution of shortest paths $d(x, y)$ in Chord for $N = 1\ 024$ fitted with a Gaussian model. (b) Shortest path distribution in CAN for $N = 10\ (d = 1), N = 10^2 (d = 2), N = 10^3\ (d = 3), N = 10^4\ (d = 4)$.

A. Chord

Stoica *et al.* [42] demonstrated in simulation that the average inter-node distance μ_d in Chord is $D/2$ and offered a brief explanation of this fact. They further showed that the distribution of $d(x, y)$ is bell-shaped, as illustrated in Fig. 1(a). The histogram appears to be Gaussian as supported by an almost-perfect fit of a Gaussian distribution in the figure. It has been noted before that certain real-world graphs (such as those describing webpage linkage structure [4]) exhibit Gaussian distributions of $d(x, y)$, but no explanation of why this happens has been offered. Below, we analyze Chord's distribution of shortest distances, understand why it appears to be Gaussian, and provide additional qualitative insight into the structure of the graph using "small-world" terminology.

Lemma 1: Each node in Chord can reach exactly $\binom{D}{i}$ nodes at shortest distance i.

Proof: Recall that, in Chord, each node x with hash index H_x has $k = \log_2 N$ neighbors located at indexes $(H_x + 2^j) \bmod N$, for $j = 0, 1, \ldots, k-1$. Notice that any shortest path to a given destination is a sequence of *unique* jumps, each of which is a power of two. The uniqueness of jumps is easy to see since any two jumps of length 2^j within the same shortest path can be replaced with a single (more optimal) jump of size 2^{j+1}. Consequently, any path of length i is formed by drawing i *unique* elements from set $\{1, 2, 4, \ldots, 2^{D-1}\}$, where D is the diameter of the graph as discussed earlier. The number of possibilities to draw i distinct objects from a set of size D is $\binom{D}{i}$, which immediately leads to the statement of the lemma. ■

Using symmetry of nodes in Chord and the result of this lemma, the probability mass function (PMF) of $d(x, y)$ is given by a binomial distribution with parameters $p = q = 1/2$

$$f(i) = P(d(x,y) = i) = \frac{\binom{D}{i}}{2^i 2^{D-i}} = \binom{D}{i} p^i q^{D-i}. \quad (5)$$

Our simulation results confirm that (5) gives the *exact* distribution of shortest path lengths in Chord. The expected value $E[d(x, y)] = \mu_d$ of a binomial random variable is a well-known result and equals Dp or simply $D/2$. This provides an alternative derivation of the result previously shown in [42].

The reason why the distribution of shortest distances in Chord appears to be Gaussian is explained by the de Moivre–Laplace theorem, which states that the binomial distribution in (5) asymptotically tends to a Gaussian distribution with mean $D_p = D/2$ and variance $D_{pq} = D/4$ for sufficiently large D. Even though we have not provided an insight into why certain Internet graphs exhibit Gaussian distributions of shortest paths, we found a clear explanation of this phenomenon in Chord.

There is also a simple intuitive link between the bell shape of the curve in Fig. 1(a) and the expansion properties of the graph. As the distance from any given node x increases, the number of *new* neighbors found by the search slowly saturates and starts declining after half of the nodes have been reached. This means that many of the newly found nodes link to some of the previously discovered nodes. This leads to a situation where the new neighbors "know" many of the old neighbors, which is often called the *small-world property* (or *clustering*) of the graph [4], [5]. In graph theory, the growth in the number of new neighbors discovered at a certain distance is related to *node expansion* of the graph. Quickly expanding graphs maintain an exponentially increasing number of new neighbors up to the diameter of the graph, which means that very few of the new neighbors "know" the old ones (and hence their clustering coefficients are virtually zero). We study these phenomena more carefully in Section V, but currently conjecture that we should expect reasonably high clustering and low expansion from Chord.

B. CAN

Recall that CAN organizes its nodes into a d-dimensional Cartesian space. We first examine the average distance in this graph and then show that, for the same degree, CAN's distribution of routing distances becomes identical to that in Chord. The next lemma generalizes the result previously shown without proof in [33].

Lemma 2: The expected distance between any two CAN nodes is $D/2$ for even N and $(2D+d)/4 - o(1)$ for odd N.

Proof: Examine a d-dimensional CAN space. The distance between each source node $s = (s_1, \ldots, s_d)$ and each destination $w = (w_1, \ldots, w_d)$ can be decomposed into d random variables

$$X_j = \|s_j - w_j\|, \qquad j = 1, 2, \ldots, d \tag{6}$$

where $\|\cdot\|$ is a one-dimensional (1-D) distance norm that depends on whether CAN is toroidal or not. Since we examine the distribution of *all* pairs (s, w), each coordinate assumes all possible values and thus is independent of the other coordinates. Hence, all X_j are i.i.d. random variables with some PMF $f_1(i)$, which is the distribution of $d(x, y)$ in a 1-D space. Consequently, the probability that a given path from s to w has length i in a d-dimensional CAN is

$$f_d(i) = P(X_1 + X_2 + \cdots + X_d = i). \tag{7}$$

Recalling that the mass function of a sum of random variables is the convolution of their densities, we get

$$f_d(i) = f_{d-1}(i) * f_1(i), \qquad d \geq 2 \tag{8}$$

where * denotes discrete convolution. Unlike Chord, CAN does not have a single distribution that describes its shortest paths for all dimensions d.

To complete the picture, we next derive distribution $f_1(i)$. Assume that the number of peers in each dimension is $M = N^{1/d}$. Then the PMF of shortest distances in a 1-D toroidal CAN of diameter $\Delta = \lceil (M-1)/2 \rceil$ is given by

$$f_1(i) = \frac{1}{M} \begin{cases} 1, & i = 0 \\ 2, & 0 < i < \Delta \\ M \bmod 2 + 1, & i = \Delta \end{cases} \tag{9}$$

The four cases in (9) are very simple. A node s can reach exactly one vertex (itself) in zero hops, two vertices in $0 < i < \Delta$ hops, and either one or two vertices in Δ hops depending on the value of M. The result in (9) matches simulation results and produces symmetric curves that progressively become bell-shaped, as shown in Fig. 1(b).

In order to derive the expected distance in CAN, notice that $E[X_1 + X_2 + \cdots + X_d] = dE[X_1]$ and that the average 1-D distance in the CAN graph is given by

$$E[X_1] = \sum_{i=0}^{\Delta} i f_1(i) = \frac{\Delta}{M}(\Delta + M \bmod 2). \tag{10}$$

For odd N, 1-D diameter Δ equals $(M-1)/2$ and (10) becomes

$$E[X_1] = \frac{\Delta(\Delta+1)}{2\Delta+1}. \tag{11}$$

Keeping in mind that diameter D of the d-dimensional graph is $d\Delta$, the expected distance in CAN is

$$\mu_d = dE[X_1] = \frac{D(D+d)}{2D+d} = \frac{2D+d}{4} - \frac{d^2}{4(2D+d)}. \tag{12}$$

For even N, 1-D diameter $\Delta = M/2$ and

$$E[X_1] = \frac{\Delta^2}{2\Delta} = \frac{\Delta}{2} \tag{13}$$

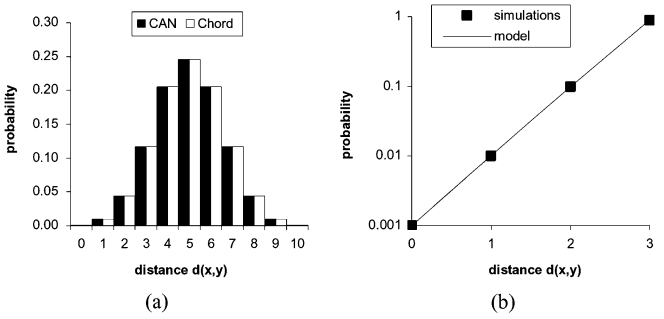
which remains the same regardless of the number of dimensions: $\mu_d = dE[X_1] = D/2$. ∎

Our next lemma shows that, as d increases, CAN's distribution of shortest paths becomes Gaussian as well.

Lemma 3: For large d, CAN's distribution of shortest distances $f(i)$ is Gaussian.

Proof: The Gaussian shape in Fig. 1(b) follows from the Central Limit Theorem as the sum of i.i.d. random variables X_j in (7) tends to a Gaussian distribution. The formal proof utilizes Berry-Esseen's theorem [43] and shows that a d-fold convolution of (9) tends to a Gaussian distribution as $d \to \infty$. We skip the details for brevity. ∎

We now have sufficient evidence that demonstrates that both Chord and CAN exhibit Gaussian distributions of shortest distances. Thus, it is natural to wonder whether the two graphs are in fact the *same* structure? Although it is easy to verify that Chord and CAN are not isomorphic, is it possible that they offer the same path length distributions to end users? As discussed in Section III-C, when $d = \log_2 N/2$, CAN's degree and diameter



(a) (b)

Fig. 2. (a) Comparison between Chord's and CAN's shortest path distributions for $N = 1\,024$ and $d = 5$. (b) Distribution of shortest distances in de Bruijn for $N = 1\,000$ and $k = 10$ in comparison with model (14).

are both $\log_2 N$, or those of Chord. We call such CAN "logarithmic" and note that the size of its dimensions is $N^{1/d} = 4$ peers. The next lemma directly follows from (9).

Lemma 4: The distribution of shortest distances in logarithmic CAN is binomial and identical to that in Chord.

The result of this lemma is illustrated in Fig. 2(a) for $N = 1\,024$, $d = 5$, and $D = 10$, which shows a perfect match between the two graphs (the distributions also match numerically).

C. De Bruijn

In general, the distribution of de Bruijn's distances $d(x, y)$ is very complicated and there is no known closed-form expression for its PMF $f(i)$ [40]. Below, we derive a simple formula for $f(i)$ that is *exact* for all graphs of diameter $D \leq 3$ and is very close to the real $f(i)$ for the rest of the graphs.

Lemma 5: The asymptotic distribution of shortest distances in de Bruijn graphs is given by

$$f(i) \approx \frac{k^i}{N} - \frac{k^{2i-1}}{N^2} \geq \frac{k^i - k^{i-1}}{N}. \tag{14}$$

Proof: Recall the peering rules of de Bruijn graphs. Each node $v = (v_1, v_2, \ldots, v_D)$ links to all possible neighbors $(v_2, v_3, \ldots, v_D, z), z \in \Sigma$. Examine a graph of diameter D and degree k. We next derive how many neighbors of any given vertex v are at *shortest* distance i from v. Denote by $S_i = \{u : u = (v_{i+1}, \ldots, v_D, z_1, \ldots, z_i), z_j \in \Sigma\}$ the set of all neighbors at (not necessarily shortest) distance i from v, which is produced by shifting vertex v left i times and filling the remaining coordinates with arbitrary symbols $z_1, \ldots, z_i \in \Sigma$. There are obviously k^i such vertices. Consider one vertex $u = (v_{i+1}, \ldots, v_D, z_1, \ldots, z_i)$ from S_i. Our immediate goal is to find out how many such vertices u do *not* belong to $S_m, 0 \leq m \leq i-1$ (i.e., there is no path from v to u shorter than i hops).

First, consider an arbitrary vertex $w = (v_i, \ldots, v_D, y_1, \ldots, y_{i-1})$ from S_{i-1}. Now examine the number of different possibilities that $u = w$, which provides the size of the overlap between sets S_{i-1} and S_i. It is easy to see that, among the free variables $z_1, \ldots, z_i$ and $y_1, \ldots, y_{i-1}$, z_1 is always equal to v_D, while the remaining variables $z_2, \ldots, z_i$ lead to k^{i-1} possible pairs (u, w) such that $u = w$.

However, notice that not all source vertices v allow such matching. The necessary condition on v for $S_n \cap S_{n-1}$ to be nonempty is $v_i = v_{i+1} = \cdots = v_D$, which leaves $i-1$ variables $v_1, \ldots, v_{i-1}$ free and allows an arbitrary choice for v_i. Therefore, there are k^i vertices v in the graph that allow a match between u and w. Hence, the expected overlap $R(i, i-1)$ between S_i and S_{i-1} is the probability of v allowing such overlap times the number of overlapped vertices

$$R(i, i-1) = \frac{k^i}{N} k^{i-1}. \tag{15}$$

Using similar reasoning and subtracting additional fraction $1/k$ of nodes that belong to both S_m with S_{m-1} for all $m \leq i-1$, an interested reader can show that $R(i, m)$ can be approximated with high accuracy by[3]

$$R(i, m) \approx \frac{k^i}{N} k^m \left(1 - \frac{1}{k}\right), \qquad 0 \leq m \leq i-1. \tag{16}$$

Next subtract $R(i, m)$ for all levels $m = 0, 1, \ldots, i-1$ from the maximum number of nodes possible at distance i (i.e., k^i) and normalize by N

$$\begin{aligned} f(i) &\approx \frac{1}{N}\left(k^i - \sum_{m=0}^{i-1} R(i, m)\right) \\ &= \frac{1}{N}\left(k^i - \frac{k^i}{N}\left[\left(1 - \frac{1}{k}\right)\sum_{m=0}^{i-2} k^m + k^{i-1}\right]\right) \\ &= \frac{k^i}{N}\left(1 - \frac{k^i + k^{i-1} - 1}{kN}\right), n \geq 1. \end{aligned} \tag{17}$$

Note that $i = 0$ is a special case of the distance from node v to itself, in which case $f(0) = 1/N$. Keeping the dominating terms in (17), we get the required expansion in (14). ■

It immediately follows from the lemma that de Bruijn graphs expand *exponentially* and that the majority of nodes are reachable at shortest distance D from each node v. This is demonstrated in Fig. 2(b) that shows de Bruijn's $f(i)$ for $N = 1\,000$ and $k = 10$ (note the log scale of the y axis). Intuitively, it is clear that the average distance in de Bruijn graphs must be very close to diameter D and that the local structure of the graph at each node looks like a tree (i.e., very few short cycles and low clustering). We examine the cyclic structure of each graph in Section V and in the meantime focus on de Bruijn's average distance μ_d.

Lemma 6: The average distance in de Bruijn graphs is asymptotically

$$\mu_d \approx D - \frac{1}{k-1}. \tag{18}$$

Proof: Examine the expected distance between any pair of nodes

$$\mu_d = \sum_{i=0}^{D} i f(i). \tag{19}$$

[3]To keep the formula exact for $D > 3$, one must take into account additional pair-wise overlap between S_m and S_j, for all $j < m - 1$. There is no known closed-form expression for this overlap.

TABLE III
AVERAGE GRAPH DISTANCE FOR $N = 10^6$

k	Moore graph	de Bruijn	Chord	CAN	Butterfly
2	17.9	18.3	–	2.5×10^5	22.4
3	11.7	11.9	–	–	14.7
4	9.4	9.5	–	500	11.8
10	5.8	5.9	–	19.8	7.3
20	4.5	4.6	10.0	10.0	5.7
50	3.5	3.5	6.1	–	4.3
100	2.98	2.98	4.75	–	3.65

To simplify the expansion, first consider series $\sum_{i=0}^{D} ix^i$ and notice that it can be computed by differentiating geometric series $\sum_{i=0}^{D} x^i$ and multiplying it by x

$$\sum_{i=0}^{D} ix^i = \frac{Dx^{D+2} - (D+1)x^{D+1} + x}{(x-1)^2} = \varsigma(x). \qquad (20)$$

Substituting (20) into (19) yields

$$\mu_d \approx \frac{\varsigma(k)}{N} - \frac{1}{kN^2}\left[\varsigma(k^2) + \frac{\varsigma(k^2)}{k} - \varsigma(k)\right]. \qquad (21)$$

For small k, this result improves previously known [6], [18], [40] lower and upper asymptotic bounds on μ_d. For large values of k and N, (21) simplifies to become (18). We leave this verification to the reader.

D. Butterfly

The final graph we examine in this section is the classic butterfly. Even though its diameter and average distance are close to optimal, they are always higher than those in (nontrivial) de Bruijn graphs. Recall that the average distance in the butterfly graph is given by the following [18]

$$\mu_d = \frac{3m-1}{2} - \frac{1}{k-1} + \frac{m}{N(k-1)} \approx \frac{3\log_k N}{2} \qquad (22)$$

which, for large N and k, is 50% larger than the same metric in de Bruijn graphs.

E. Discussion

The results of this section indicate that de Bruijn graphs offer not only provably-optimal diameter D, but also smaller average routing times compared to Chord, CAN, and the static butterfly. As shown in Table III for $N = 10^6$, the average distance in de Bruijn graphs is still smaller than half of that in Chord and CAN for the same number of neighbors and 22% smaller than that in the butterfly. Also notice that, for large k, μ_d in de Bruijn graphs converges to the best possible average distance of Moore graphs shown in the first column of Table III.

This result has several practical implications. First, μ_d determines the expected distance (and sometimes delay) in the graph and represents a measure of the overhead needed to find data. Second, the average distance determines the *capacity* of a peer-to-peer network, where the capacity is a term widely used in interconnection and wireless networks to define the throughput available to each node under random communication patterns within the network. Since each peer must forward requests for other peers, the expected useful capacity of a node is determined by the inverse of μ_d (i.e., for each useful request a node makes, it must forward on average μ_d other requests).

TABLE IV
COMPARISON OF GREEDY AND BFS DISTANCES

	de Bruijn	Chord	Pastry	Randomized Chord
BFS diameter	3	10	5.18	5.05
Greedy diameter	3	10	10	11.26
BFS μ_d	2.87	5	3.37	3.33
Greedy μ_d	2.87	5	5	4.83

Assuming fixed transmission bandwidth and discounting interference effects, the average capacity $c(G)$ of wireless ad-hoc networks is $O(1/\sqrt{N})$ due to spatial restrictions on connectivity [17], while both Chord and logarithmic CAN maintain an average capacity of $1/\mu_d = 2/\log_2 N$. Compared to wireless networks, this is a much better bound; however, it is still several times lower than that in de Bruijn graphs. Even assuming a worst-case average distance $\mu_d = D$ in de Bruijn graphs, their average capacity with $\log_2 N$ neighbors is superior to Chord's for all $N > 16$

$$c(G) = \frac{1}{\log_k N} = \frac{\log_2 \log_2 N}{\log_2 N}. \qquad (23)$$

In fact, the ratio of these two capacities grows infinitely large (albeit very slowly) for large N. Compared to the static butterfly, de Bruijn graphs offer 50% more capacity in asymptotics and at least 22% more capacity in graphs of practical size examined in this work.

In the current Internet, each search request typically carries a small amount of information and it is not clear at this point whether future peer-to-peer systems will be utilized to the point of their ultimate capacity. Nevertheless, we believe that it is desirable to design the underlying structure of the application-layer graph to be able to carry as many concurrent requests as possible. Thus, we must conclude that de Bruijn graphs offer clear benefits in terms of expected capacity and routing distances over the existing approaches.

F. Greedy Routing

One issue left to examine is the ability of each graph to find the *actual* shortest paths using its greedy rules. We use simulations for this study and compare the quality of the paths built in deterministic approaches de Bruijn and Chord with that in nondeterministic techniques Pastry and Randomized Chord [29]. The last two methods are used to illustrate a common property of random P2P graphs—the default routing algorithms often find paths much longer than the actual shortest routes. Table IV shows a comparison between the four methods in graphs of 1000 (de Bruijn) and 1024 (Chord, Pastry, and Randomized Chord) nodes and degree at each peer $k = 10$ (note that, for nondeterministic methods, all metrics are averaged over 100 runs). Observe that de Bruijn's and Chord's greedy routing is optimal (the same holds for CAN), while that of Pastry and Randomized

TABLE V
COMPARISON OF NONGREEDY AND BFS DISTANCES

	de Bruijn	Chord	Pastry	Randomized Chord
BFS diameter	2	5	3	3
Greedy diameter	2	5	5	5
BFS μ_d	1.89	2.75	1.91	1.91
Greedy μ_d	1.89	2.75	2.24	2.23

Chord can be substantially improved. Also notice that Randomized Chord exhibits larger average greedy diameter than Chord (i.e., 11.26), but smaller average distance (i.e., 4.83).

Recent work [29] has shown that the addition of NoN routing can significantly improve the asymptotic performance of many randomized methods such as Randomized Hypercube, Randomized Chord, and Symphony. We examine this hypothesis in simulation for Randomized Chord and the same set of parameters as in Table IV. Recall that NoN routing involves not only the neighbors of each node, but also all nodes at distance two. For de Bruijn, this is equivalent to increasing node degree to k^2 and leads to the reduction of the diameter by a factor of two (if D is odd, the corresponding ceiling function must be applied to $D/2$). Since not all path lengths are divisible by two, the average distance typically does not enjoy the same improvement (see below).

The results of NoN simulations are shown in Table V. As expected, the diameter of both deterministic graphs is reduced to $\lceil D/2 \rceil$ and their average distance became correspondingly smaller. On the other hand, Pastry and Randomized Chord are able to use NoN routing to reduce their greedy μ_d by a factor of 2.24 and 2.17, respectively. In fact, their performance becomes almost identical, although it is still 18% worse than that of de Bruijn graphs.

One question comes to mind—can the performance of NoN-Pastry be improved by constructing a 100-regular version of the graph instead of using NoN routing over a 10-regular network? The answer is positive—using 100 neighbors ($b = 50$), Pastry can route in two hops to any peer in a 2500-node graph, which is already a substantial improvement over the values shown in Table V. It thus becomes unclear whether given a family of k-regular graphs $G(k)$ (such as Pastry or base-d Chord), NoN-greedy routing in $G(k)$ can be more optimal than simple greedy routing in $G(k^2)$. We leave this analysis for future work.

Next, we investigate clustering and then resilience features of de Bruijn graphs before addressing their practical use in peer-to-peer networks.

V. CLUSTERING AND EXPANSION

Following significant research effort to model the structure of the current Internet, it was discovered that many of the existing topology generators did not accurately match the "small-world" (clustering) properties of the Internet graph [4], [5]. Clustering is a very interesting concept that is found in many natural phenomena and that determines how tightly neighbors of any given node link to each other. In what follows, we examine clustering in Chord, CAN, and de Bruijn graphs, study graph-theoretic semantics behind the clustering coefficient, and show why metrics related to clustering are important concepts for peer-to-peer systems.

Given graph $G = (V, E)$, node $v \in V$, and its neighborhood $\Gamma(v) = \{u : (v, u) \in E\}$, clustering coefficient $\gamma(v)$ is defined as the ratio of the number of links $L(\Gamma(v))$ that are entirely contained in $\Gamma(v)$ to the maximum possible number of such links (if the graph is undirected, each link in $L(\Gamma(v))$ is counted twice)

$$\gamma(v) = \frac{L(\Gamma(v))}{|\Gamma(v)|(|\Gamma(v)| - 1)}. \tag{24}$$

Graph clustering $\gamma(G)$ is the average of $\gamma(v)$ for all vertices v with degree at least 2. The two questions we study below are: *what exactly does clustering mean and how does it affect the properties desirable in peer-to-peer networks?*

A. Clustering Coefficients

We first present the values of clustering coefficients of all three graphs and then explain the meaning of our results.

Lemma 7: Chord's clustering coefficient is $1/\log_2 N$.

Proof: In Chord, all nodes are symmetric (modulo N) in terms of their connectivity rules. Without loss of generality, consider node $v = 0$ and its k neighbors $1, 2, \ldots, 2^{k-1}$. Examine any two nodes $x = 2^i$ and $y = 2^j, (i < j)$, from this list. It is easy to notice that x and y are neighbors of each other if and only if $2^j - 2^i$ is a power of two. Hence, we must have $2^i(2^{j-i} - 1) = 2^m$ for some integer m. This can only occur when $2^{j-i} - 1 = 1$, or $j - i = 1$. This means that all k neighbors of v are sequentially chain-linked to one another (note that the links are uni-directional from smaller nodes to larger ones). Hence, clustering in Chord is

$$\gamma(G) = \frac{k-1}{k(k-1)} = \frac{1}{k} = \frac{1}{\log_2 N}. \tag{25}$$

This completes the proof. ■

Lemma 8: De Bruijn's clustering coefficient is $(k-1)/N$ for $k \geq 3$ and $1/(N-2)$ for $k = 2$.

Proof: De Bruijn graphs are also fairly simple to analyze. Below, we show that there are exactly $k(k-1)$ nodes with nonzero clustering coefficients, while the remaining $N - k(k-1)$ nodes have $\gamma(v) = 0$. Examine the conditions necessary to achieve nonzero clustering for a given node $v = (v_1, \ldots, v_D)$. For any two of v's neighbors $a = (a_1, \ldots, a_D)$ and $b = (b_1, \ldots, b_D)$, the necessary condition for nonzero clustering is either a links to b or b links to a. Due to symmetric neighboring rules, it is sufficient to analyze the case of a linking to b.

First, notice that $(a_1, \ldots, a_D) = (v_2, \ldots, v_D, \alpha)$ and $(b_1, \ldots, b_D) = (v_2, \ldots, v_D, \beta)$ for some symbols α and β from Σ. Note that the value of v_1 does not affect whether a and b can be neighbors of each other. When there is a directed link from a to b, we have for some $z \in \Sigma : (a_2, \ldots, a_{D-1}, z) = b$, or, in other words, $(v_3, \ldots, v_D, \alpha, z) = (v_2, \ldots, v_D, \beta)$. This can only occur when $v_2 = v_3 = \cdots = v_D$, which means that there are exactly k^2 nodes v with this property (since only two out of D coordinates are free variables). Ensuring that $a \neq v, a \neq b$, and $b \neq v$, there are $k^2 - k$ vertices v that have nonzero clustering. If v allows clustering, then α must be equal to v_D, but β is a free variable that determines how

many pairs (a, b) are neighbors of each other. Again, making sure that $a \neq b$, we are left with $k-1$ choices for β. Hence, clustering $\gamma(v)$ is $(k-1)/k(k-1) = 1/k$ and the average graph clustering $\gamma(G)$ is given by

$$\gamma(G) = \frac{1}{N}\left(k(k-1)\frac{1}{k} + 0\right) = \frac{k-1}{N}. \tag{26}$$

A special case of $k = 2$ is handled similarly (except that degree-1 nodes must be excluded from the summation) and leads to $\gamma(G) = (k-1)/(N-2) = 1/(N-2)$. We skip the proof for brevity. ■

Simulation results confirm that (25) and (26) are exact. Next notice that de Bruijn's $\gamma(G)$ decays to zero much quicker than Chord's confirming our earlier conjecture based on the distribution of shortest paths in Section IV.

The derivation of $\gamma(G)$ for CAN is much simpler as one can easily notice that *none* of the nodes in any neighborhood link to each other. Hence, CAN's $\gamma(G)$ is zero. This is somewhat counterintuitive since CAN's number of new neighbors becomes saturated at $D/2$ just as in Chord, and therefore its clustering properties should be similar to Chord's. We next examine the reasons behind this phenomenon and generalize clustering to become a global metric.

B. Cycles

There are two ways to better understand what clustering means and assess its importance for peer-to-peer networks. The first insight is based on cycles. Given a k-regular *undirected* graph G, it is easy to notice that the number of 3-cycles per node determines the clustering coefficient of the graph. Recall that an n-cycle is a path that starts and ends in the same node and contains exactly n edges[4]. Hence, any 3-cycle must involve two direct neighbors of node v, which results in clustering.

Since one goal in peer-to-peer networks is to reach as many nodes as possible within a certain number of hops, cycles that lead back to the original node where the request started are not very helpful. Another goal of peer-to-peer networks is to provide a fault-resilient environment where a simultaneous collapse of several nodes does not separate the graph into disjoint components. Short cycles mean that paths from any node x through different neighbors leading to any destination y must *overlap* with each other. This is not desirable since multiple parallel paths to y may be compromised when nodes in the neighborhood fail. This is shown in Fig. 3(a) where failure of node 1 leaves x with no path leading outside of its neighborhood. In fact, when node 1 fails, nodes 2 and 3 are also disconnected from the rest of the graph since all of their outgoing (as well incoming) edges are locally clustered.

Now we come back to the issue of why CAN has zero clustering, but identical shortest-path properties to those found in Chord. The absence of 3-cycles in CAN is explained by the fact that it has no *odd* cycles whatsoever, but it does have plenty of *even* cycles. In fact, the number of 4-cycles in CAN is roughly the same as in undirected Chord with the same number of neighbors. Consequently, *local* properties captured by the clustering coefficient do not necessarily mean much for graphs like CAN where only "friends of friends" have common acquaintances, while direct friends of node x never know each other. This is illustrated in Fig. 3(b), where clustering coefficient $\gamma(x)$ is zero, but nodes 2, 3, and 4 all link to the same "friend of a friend" node 5.

[4]Usually, these paths are required to be edge and/or node disjoint, but this always holds for 3-cycles.

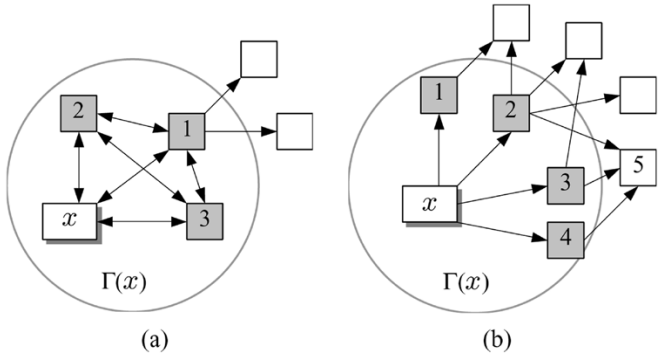


Fig. 3. (a) High clustering leads to weak connections outside a neighborhood. (b) A more generic definition of clustering.

The concept of n-cycles applies to *directed* graphs as well; however, it does not directly produce the clustering coefficient because of a stricter nature of directed cycles. These difficulties lead us to generalize the framework of clustering using expansion analysis below.

C. Graph Expansion

Again examine Fig. 3(b), which shows how undirected 4-cycles contribute to a graph's global clustering properties. *Global clustering* is a concept of "friends knowing each other" generalized to "friends knowing each others' friends." Although the previous discussion of cycles allows one to account for these cases, we seek a more generic and useful definition of clustering that goes beyond n-cycles ($n \geq 3$) and has a simple closed-form analytical expression for all three graphs.

We next study graph *expansion*, which determines how quickly the graph finds "unknown" nodes. Consider a graph $G = (V, E)$ and select some of its nodes into set $S \subset V$. Define the set of all edges between S and the rest of the graph $V \backslash S$ to be $\partial S = \{(u, v) : (u, v) \in E, u \in S, v \in V \backslash S\}$. Set ∂S is called the *edge boundary* of S. Edge expansion $i(S)$ is defined as the ratio of the size of ∂S to the size of S

$$i(S) = \frac{|\partial S|}{|S|}. \tag{27}$$

It is easy to see the relationship of $i(S)$ to clustering. Select S to be the neighborhood $\Gamma(v)$ of some node v. Therefore, $|S| = k$ and the number of edges contained within S is $k^2 - |\partial S|$, generically assuming a k-regular graph. Then the clustering coefficient of v is given by

$$\gamma(v) = \frac{k^2 - |\partial S|}{k(k-1)} = \frac{k^2 - i(\Gamma(v))k}{k(k-1)} = \frac{k - i(\Gamma(v))}{k-1}. \tag{28}$$

Edge expansion determines the strength of the graph in the presence of edge failure. Clearly, a larger clustering coefficient in a k-regular graph implies smaller $i(\Gamma(v))$, as seen in (28), and generally leads to weaker graphs.

Definition 1: Graph edge expansion (sometimes called the isoperimetric number of the graph) $i(G)$ is the minimum of $i(S)$ for all nonempty sets $S \subset V, |S| \leq |V|/2$.

Notice that, by examining $i(G)$, we no longer focus on *local* clustering, but rather on global properties of the graph and its resilience to edge failure over *all possible* sets S. Edge expansion tells us how many edges link outside any set S; however, it does *not* tell us if the outgoing edges link to the same node multiple times. For example, in Fig. 3(b), there are eight edges leaving neighborhood $\Gamma(x)$, but they link to only four unique nodes, which indicates a good amount of path overlap. Edge expansion tells us the size of the edge cut between $\Gamma(x)$ and the rest of the graph, which is a useful analysis tool for studying a graph's resilience when *edges* are expected to fail (i.e., eight edges in the cut are better than four). In peer-to-peer systems, *node* failure is much more common than edge failure, in which case regardless of how many edges cross the cut, the strength of the neighborhood is determined by the number of nodes on the other side of ∂S. Hence, from the resilience perspective of peer-to-peer networks, it makes more sense to examine *node* expansion of the graph as we define below.

Definition 2: Consider a graph $G = (V, E)$ and some subset of nodes $S \subset V$. Define the *node* boundary of S to be $\partial S = \{v : (u, v) \in E, u \in S, v \in V \backslash S\}$. Node expansion $h(G)$ of the graph is given by

$$h(G) = \min_{\{S:|S|\leq|V|/2\}} \frac{|\partial S|}{|S|}. \tag{29}$$

Metrics $i(G)$ and $h(G)$ are related to edge and node bisection widths, respectively, of the graph, the determination of which is generally an NP-complete problem. Furthermore, even after many years of research, the exact expression of these metrics for de Bruijn graphs remains unknown. Below, we limit our analysis to sets S that are neighborhoods of a given node (i.e., balls centered at the node) and study graph expansion that explains how well each ball is connected to the rest of the graph. Note that these balls do not necessarily represent the weakest sets S of each graph and do not, in general, achieve the minimum bound in (29). Derivation of better bounds on $h(G)$ is the topic of on-going research.

Recall that ball $B(v, n)$ of radius n centered at node v contains all nodes reachable from v in no more than n hops. In other words, $B(v, n) = \{u : d(v, u) \leq n\}$. It is easy to notice that the boundary of a ball is simply $\partial B(v, n) = \{u : d(v, u) = n+1\}$ and that our derivations in Section IV can be applied to study expansion (and global clustering) of each graph. Both logarithmic CAN and Chord have the same expansion properties since their distributions of $d(x, y)$ are identical. Hence, from now on, we only consider Chord.

Lemma 9: Chord's ball expansion $h_B(G)$ is asymptotically $\Theta(1/\sqrt{\log_2 N})$.

Proof: Using our prior result in (5), the size of each ball of radius n in Chord is given by

$$|B(v, n)| = N \sum_{i=0}^{n} f(i) = \sum_{i=0}^{n} \binom{D}{i} = \sum_{i=0}^{n} \frac{D!}{i!(D-i)!}. \tag{30}$$

The number of nodes in boundary $\partial B(v, n)$ is $\binom{D}{n+1}$ and the resulting *ball expansion* $h_B(G)$ of the graph is given by

$$h_B(G) = \min_{\{n\leq D/2\}} \frac{\binom{D}{n+1}}{|B(v, n)|}. \tag{31}$$

From observing prior plots of the distribution of $d(x, y)$, it is clear that Chord has a low expansion value $h_B(G)$ since the number of nodes in $\partial B(v, n)$ saturates at $n = D/2$. Omitting cases when D is even and no single ball contains exactly half the nodes, we briefly consider the odd values of D as they allow us to approach the worst-case bound on $h_B(G)$. For odd D, $h_B(G)$ reaches its minimum when ball radius n is $(D-1)/2$. Keeping in mind that the size of this ball is $N/2 = 2^{D-1}$ and using Stirling's approximation in (31), we have

$$h_B(G) = \frac{\binom{D}{(D-1)/2+1}}{2^{D-1}} \approx \sqrt{\frac{8}{\pi \log_2 N}} \tag{32}$$

which leads to the statement of the lemma. ■

This function slowly decays from 0.45 for $N = 2\,048$ to 0.27 for $N = 2^{30}$. Contrast this result with that for de Bruijn graphs (see below), which maintain constant connectivity $h_B(G)$ for all ball sizes and all values of N.

Lemma 10: De Bruijn's ball expansion $h_B(G)$ is no less than $k - 1$.

Proof: De Bruijn graphs are also easy to analyze given their expansion model in (14)–(17). From (14), write the size of each ball of radius n as k^n (assuming large k and neglecting insignificant terms) and notice that the largest ball smaller than the entire graph contains only a small fraction of all nodes

$$|B(v, D-1)| \approx k^{D-1} = \frac{N}{k}. \tag{33}$$

Further estimating edge boundary $|\partial B(v, D-1)|$ using (14), we obtain

$$h_B(G) = \frac{|\partial B(v, D-1)|}{|B(v, D-1)|} \geq \frac{k^D - k^{D-1}}{k^{D-1}} = k - 1 \tag{34}$$

which completes the proof. ■

De Bruijn graphs expand so quickly that they actually approach the maximum possible bound on $h_B(G)$ and keep all balls $B(v, n)$ connected to the rest of the graph through at least $(k-1)|B(v, n)|$ external nodes. In fact, this not only explains the low diameter of de Bruijn graphs, but also leads to two important results. First, clustering in de Bruijn graphs is minimal at both local *and* global levels since exponential neighborhood expansion is preserved for all balls smaller than the graph itself. Second, path overlap in the graph is virtually nonexistent due to little global clustering. This means that *shortest* parallel paths toward any given destination are expected to be node-disjoint with high probability.

In the next section, we study fault resilience of these graphs and then proceed to dynamic construction of peer-to-peer networks.

VI. Resilience

A. Generic Methods

Classical failure analysis in peer-to-peer networks (e.g., [24], [42]) focused on analyzing the probability that a given node x becomes disconnected under a p-percent node failure. This amounts to computing the probability that all k neighbors of x fail simultaneously and leads to small individual failure probabilities p^k for most practical networks. Also note that results derived using this method hold for any k-regular graph, regardless of its internal structure. Clearly, this analysis is insufficient to distinguish between all k-regular graphs since some of them may contain "weak" parts that can partition the graph into several disjoint components while no *single* node is completely disconnected from its component.

Another approach often used in classical fault resilience analysis is to examine k-node-connectivity of the graph in question. Given our graph structures, we show below that this metric does not lead to any significant insight either.

Definition 3: A k-regular graph is k-node-connected if there are k node-disjoint paths between any pair of nodes.

This implies that a k-node-connected graph can tolerate the failure of any $k-1$ nodes without becoming disconnected and that the diameter of the graph after any $k-1$ nodes have failed is at most $D+1$. Both CAN and Chord are k-node-connected[5], while de Bruijn graphs are not due to several "weak" nodes with self-loops. This classical form of de Bruijn graphs has been shown to be $(k-1)$-node connected [41]; however, we seek to achieve maximum fault tolerance, which leads us to removing the loops and linking these k "weak" nodes to each other. Consider node $(h, h, \ldots, h), h \in \Sigma$, with a self-loop. A *chain-linked* de Bruijn graph has directed links $(h, h, \ldots, h) \rightarrow (g, g, \ldots, g)$, for all $h \in \Sigma$ and $g = (h+1) \bmod k$. Recent development in consecutive-d graphs [10] also studied chain-linked de Bruijn graphs and proved that they are k-node connected.

What we know so far from classical peer-to-peer network analysis and maximum fault-tolerance metrics is that all three graphs are similar in their resilience. Hence, we seek additional methods that can distinguish between the fault tolerance offered by each graph. One such metric is *bisection width* [23], which is defined as the smallest number of (possibly directed) edges between any two equal-size partitions of the graph. Graph bisection width determines the difficulty of splitting the graph into giant components by failing individual edges. We next examine this metric in all three graphs.

B. Bisection Width

Note that, besides determining resilience, bisection width of a graph often provides tight upper bounds on the *achievable* capacity of the graph. Assume that each node sends messages to random destinations at a certain fixed rate. This communication pattern generates N messages per time unit. Each message is replicated μ_d times (on average) and each edge is expected to carry $N\mu_d/(Nk) = \mu_d/k$ messages per time unit. Note, however, that this analysis assumes that the combined load is *equally* distributed between all edges. There may be bottlenecks in which the load is significantly higher than the average, and the resulting throughput capacity of the graph may be lower than the mean value.

[5]This can be shown for CAN by generating all possible orders of traversing d-dimensional paths between any pair of nodes. Chord's connectivity is easily derived from the well-known properties of hypercubes.

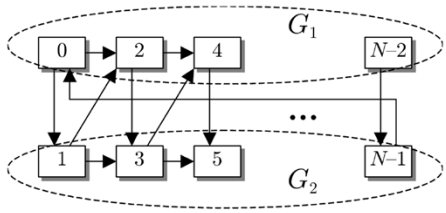


Fig. 4. Partitioning of Chord into two smaller Chord graphs.

Recall that approximately half of all communication in the graph is expected to cross the bisection cut. Thus, if this part of the graph is narrow, it will lead to congestion and inability of the graph to carry its expected load. One example of graphs with unacceptably small bisection width are trees, which are susceptible to both easy disconnects and severe congestion near the root.

Lemma 11: Chord's bisection width $bw(G)$ is N.

Proof: The proof is similar to that for hypercubes [23]. We first show that each edge in Chord is contained in $N/2$ shortest paths, which establishes (using graph-embedding arguments [23]) a lower bound on the bisection width. We then construct a particular partition of G into two subgraphs that achieves this lower bound.

Our first goal is to show that each edge is contained in exactly $N/2$ shortest paths. Assume an edge (u, v) of "length" 2^i (i.e., $\|u - v\| = 2^i$). Next, observe that each path from a node x to all destinations in the graph can be coded with a string $(X_1, \ldots, X_D)$, where $X_i = 1$ if an edge of length 2^i is used in the path and 0 otherwise. Notice that a path $(X_1, \ldots, X_D)$ contains an edge of length 2^i if and only if $X_i = 1$. Thus, there are exactly $2^{D-1} = N/2$ such paths originating from x. It then follows that the total number of paths with an edge of size 2^i (i.e., over all nodes x) is $N^2/2$. Since the load on every edge of size 2^i in Chord is symmetric, all edges of length 2^i share this load equally. Finally, recalling that there are exactly N edges of size 2^i in G, each of them has $(N^2/2)/N = N/2$ shortest paths passing through it. This directly leads to a lower bound $bw(G) \geq N$ [23].

For the second half of the proof, is easy to see that each Chord graph of diameter D can be split into two Chord graphs $G_1 \cup G_2 = G$, each of diameter $D-1$, with exactly $N = 2^D$ edges between the smaller versions of the graph. This is illustrated in Fig. 4, where all odd nodes are in graph G_1 and all even nodes are in graph G_2. ■

Note that Chord's $bw(G)$ is double the bisection width of binary hypercubes since Chord uses *directed* links while hypercubes are undirected.

Lemma 12: Assuming the size of each dimension is even, CAN's bisection width $bw(G)$ is $2N^{(d-1)/d}$.

Proof: CAN is optimally split in half when all edges crossing two parallel $(d-1)$-dimensional planes are failed. Since the size of each dimension in CAN is $N^{1/d}$, each $(d-1)$-dimensional planes contains $N^{1-1/d}$ peers. This leads

to the bisection width of $2N^{(d-1)/d}$. Note that an alternative derivation can be found in [3]. ■

Applying this result to logarithmic CAN ($d = \log_2 N/2$), notice that its bisection width is $N/2$. Furthermore, if we view undirected links of logarithmic CAN as being composed of two directed edges, its bisection width matches that of Chord. Also note that CAN achieves its maximum $bw(G)$ when $d = \log_3 N$ and that all *sub-logarithmic* values of $d \ll \log_2 N$ result in "weaker" graphs. This is another way of showing that CAN with small fixed values of d may not be competitive to Chord in practical settings.

The bisection width of the butterfly is $kN/(2m)$ [23], where m is given by Lambert's function W in (4). Asymptotically, this bisection becomes $kN/(2\log_k N)$, although for small N it is slightly better. Finally, the exact value of $bw(G)$ of de Bruijn graphs is unknown and the best available upper and lower bounds differ by a factor of four [36]

$$\frac{kN}{2\log_k N}(1-o(1)) \le bw(G) \le \frac{2kN}{\log_k N}(1+o(1)). \quad (35)$$

Using the lower bound in (35), the bisection width of de Bruijn graphs for $k = \log_2 N$ is larger than that in Chord or CAN by a factor of $\log_2 \log_2 N/2$ (which is 2.2 for $N = 10^6$) and is generally no worse than that in the butterfly. It is further conjectured that the actual bisection width of de Bruijn graphs is at least 40% higher than the pessimistic lower bound used in the above comparison [11].

In summary, larger values of bisection width in de Bruijn graphs point toward higher resilience against graph partitioning and lower congestion in the bisection cut in addition to their optimal routing established earlier.

VII. Optimal Diameter Routing Infrastructure

We have accumulated sufficient evidence that shows that de Bruijn graphs possess both short routing distances and high fault tolerance. In this section, we discuss ODRI, which builds de Bruijn graphs incrementally and preserves their nice properties at the application layer. Fortunately, de Bruijn graphs are very simple to build incrementally and many of the details (some of which we skip) are similar to those in recent proposals D2B [13] and distance halving [30]. We also feel that the algorithmic structure of ODRI is much simpler than that of other recently proposed fixed-degree graphs [26], [46].

Let $N_{\max}$ be the maximum possible number of nodes in the system (note that $N_{\max}$ must be a power of node degree k). Organize the space of all possible nodes between $[0, N_{\max}-1]$ into a virtual de Bruijn graph and notice that each node x in this structure is a base-k integer H_x and that its neighboring rules can be expressed as

$$H_x \to (kH_x + i) \bmod N_{\max}, \qquad i = 0, 1, \ldots, k-1 \quad (36)$$

since a shift left by one digit is equivalent to multiplication of H_x by k.

In ODRI, each existing peer holds a consecutive stretch of the number space, which can be denoted by $[z_1, z_2]$, for some $z_1, z_2 \in [0, N_{\max}-1]$. To join the network, a node routes to the area of the circle where its hash index H_x is located and asks the previous owner of the zone to split it in half. Notice that building the routing table for a newly joined node requires only $O(1)$ message complexity as it can be copied from the previous owner of the zone. Notification of existing neighbors has another $O(k)$ message overhead. Further notice that unlike Koorde [20], each existing zone is split in half, which leads to significantly better bounds on the size of the smallest zone [45].

Peer-to-peer linking rules are also straightforward. Consider node x that owns zone $[z_1, z_2]$. Each of the integer values in $[z_1, z_2]$ corresponds to the virtual de Bruijn graph of size $N_{\max}$. Hence, to preserve de Bruijn linkage at the application layer, x must connect to all peers holding the other end of each edge originating in $[z_1, z_2]$. This means that there is an *application-layer* edge (x, y) if and only if there is an edge (u, v) in the virtual de Bruijn graph such that $u \in Z_x$ and $v \in Z_y$, where Z_x and Z_y are the corresponding zones held by x and y. Again observe that these rules are different from those in Koorde, which explains the difference in the application-layer diameter of these graphs (see below).

We next present several useful results about ODRI. We first address the issue of whether the application-layer graph maintains fixed degree and optimal diameter under the condition of equal-size zones. We then extend this analysis to random zones created by a uniform hashing function.

A. Equal-Size Zones

Lemma 13: If all zones have the same fixed size, ODRI maintains the application-layer degree equal to k.

Proof: Denote by $M = N_{\max}/N$ the fixed size of each zone after N nodes have joined (this condition further implies that $N_{\max}$ is divisible by N and there exists a peer whose left boundary is $z = 0$). Consider an arbitrary node x with zone $Z_x = [z, z+M-1]$ and notice that x links to every peer whose zone falls between zk and $(z+M-1)k+k-1$. Since this stretch spans $Mk-1$ de Bruijn vertices from the virtual graph, there are exactly k different peers in this stretch. Using similar reasoning, it is easy to show that the in-degree of each node is always k (for more details, see Lemma 15 below). ■

Given the assumptions of the previous lemma, notice that the application-layer graph in ODRI is a scaled-down version of the virtual de Bruijn graph. Thus, the diameter of the peer-to-peer graph under these conditions remains optimal as we show in the next lemma.

Lemma 14: If all zones have the same fixed size, ODRI builds an N-node application-layer de Bruijn graph with diameter $\lceil \log_k N \rceil$.

Proof: Assume that $N_{\max} = k^{D_{\max}}$ and $N = k^D$, where $D_{\max}$ and D are the diameters of the virtual and application-layer graphs, respectively. Then, the size of each zone is $M = k^{D_{\max}-D} = k^m$, and each node holds enough consecutive de Bruijn vertices to arbitrarily select the last m digits of the vertex from which it starts jumping toward any destination. Since the last m digits of the source can be selected to always match the first m digits of the destination *before* any routing starts, the longest path in the virtual de Bruijn graph must match the remaining $D_{\max} - m = D$ digits. This clearly requires no more than $D = \log_k N$ hops and provides the necessary bound on the application-layer diameter. ■

B. Random Zones

Achieving constant-size zones using distributed join and leave processes is a nontrivial but well-studied problem [2], [30]. Equal zone sizes are desirable as they maintain a fixed out-degree at the application layer and provide better balancing of user objects between the peers. Assuming uniform random hashing, it can be shown [30] that after a sequence of N random joins, the maximum zone held by a peer is larger than average by a factor of $O(\log N)$ with high probability (note that the same bound applies to the maximum *out-degree* of each peer). While the existence of large zones has no direct effect on the diameter, the ability of a graph to avoid generating *small* zones is of paramount importance to distributed de Bruijn graphs. Fortunately, under center-splits, it can be shown [45] that the smallest zone size does not deviate "too much" from the ideal average size and the diameter of ODRI remains asymptotically optimal.

While the lower bound on the peer out-degree is simply 1, the following lower bound on the application-layer *in-degree* is less obvious.

Lemma 15: Under a uniform hashing function, ODRI's in-degree at each peer is no less than k with high probability.

Proof: Select any vertex $v, v \in [0, N_{\max} - 1]$, in the virtual graph and examine the set of de Bruijn vertices $u_i, i = 0, 1, \ldots, k-1$, all linking to v. Sort u_i in ascending order and notice that the distance between each pair of adjacent nodes in this list is exactly $N_{\max}/k$. From this fact, it follows that with high probability all vertices u_i must belong to *different* peers since the largest zone held by a peer is no more than $N_{\max} \log(N)/N \ll N_{\max}/k$ [30]. ∎

Our next result shows that the imbalance in zone sizes has little impact on the asymptotic diameter of the peer-to-peer graph.

Lemma 16: Under a uniform hashing function, ODRI constructs a peer-level graph of diameter $\lceil \log_k N \rceil (1 + o(1))$ with high probability.

Proof: Assume the notation in the Proof of Lemma 14 and notice that the size of the smallest zone is $\Omega(M/2^{\sqrt{\log N}}) = \Omega(k^m/2^{\sqrt{\log N}})$ with high probability [45]. Thus, in the worst case, the smallest node can always match at least $m - \log_k 2^{\sqrt{\log N}} = m - \Theta(\sqrt{\log N})$ last digits of the source to those in the index of the destination before routing is started. Following the reasoning in Lemma 14, the diameter of this graph is $D_{\max} - m + \Theta(\sqrt{\log N}) = D + \Theta(\sqrt{\log N})$. Finally, recalling that D is $\log_k N$, we have that the diameter of the ODRI graph is no more than $\log_k N(1 + o(1))$. ∎

This lemma further implies that the *average* distance in the application-layer graph is asymptotically optimal. Also note that, by allowing $\Theta(\log N)$ worst-case out-degree[6] in the peer graph, ODRI improves Koorde's diameter from $\Theta(k \log_k N)$ to $\log_k N(1 + o(1))$.

C. Balancing Zones and Proximity

To overcome imbalance in zone sizes in a highly dynamic environment, ODRI implements a variation of the "power of two choices" algorithm [2], [13], [30] during peer joins and departures. To join an existing ODRI network, a node x performs a biased walk of length d through the graph starting in a random location and searching for the largest node to split. During the walk, the peer samples k neighbors of each visited node y, assuming that their size is known to y through some keep-alive mechanism. The walk is biased toward large-zone neighbors since they are more likely to "know" other large nodes. Additional use of random walks includes load-balancing of objects (i.e., the node with the largest current load is split in half) and proximity-aware graph construction (i.e., the new node joins wherever its neighbors are closest to itself in some physical sense). During departure, node x does the same biased walk looking for the smallest (or least loaded) node to take over its zone Z_x.

[6]This is not a major issue since Koorde's in-degree is similarly $\Theta(\log N)$ and thus neither graph (in its unbalanced form) is truly "fixed degree."

TABLE VI
ODRI's DYNAMIC ROUTING PERFORMANCE ($N = 30\,000,\ k = 8$)

Walk length d	Diameter D	Average distance μ_d
0	7	5.91
1	6	4.88
2	6	4.84

Some of the details of this framework are presented in [45], while others are still under investigation. It is worthwhile to note that as long as $dk = \Theta(\log N)$, the largest zone and largest out-degree exceed their ideal values by a fixed factor with high probability [45]. Thus, ODRI achieves both a fixed application-layer degree and asymptotically optimal diameter.

Table VI shows ODRI's simulation results in a system with $N = 30\,000$ users, $k = 8$, and $N_{\max} = 8^7 = 2\,097\,152$ (the results are taken from a single run and represent those observed in a typical ODRI graph). The ideal (i.e., static) diameter for this case is five hops and the ideal average distance is 4.81. As the table shows, ODRI's zone balancing algorithm greatly improves the average distance and brings it close to its ideal value with just a single-hop walk.

VIII. CONCLUSION

In this paper, we studied the diameter-degree tradeoff question of DHT research and conducted an extensive graph-theoretic comparison of several existing methods in terms of their routing performance and fault resilience. We then proposed a distributed architecture based on de Bruijn graphs and demonstrated that it offered an optimal diameter for a given fixed degree, k-node connectivity, large bisection width, and good node expansion. Combining these findings with incremental construction of ODRI, we conclude that de Bruijn graphs are viable structures for peer-to-peer networks.

ACKNOWLEDGMENT

The authors are grateful to J. Byers and the anonymous reviewers for providing excellent suggestions and comments.

REFERENCES

[1] J. Aspnes, Z. Diamadi, and G. Shah, "Fault-tolerant routing in peer-to-peer systems," in *Proc. ACM PODC*, Jul. 2002, pp. 223–232.
[2] Y. Azar, A. Broder, A. Karlin, and E. Upfal, "Balanced allocations," *SIAM J. Comput.*, vol. 29, no. 1, pp. 180–200, Feb. 2000.
[3] M. C. Azizoglu and O. Egecioglu, "The isoperimetric number and the bisection width of generalized cylinders," in *Proc. 9th Quadrennial Int. Conf. Graph Theory, Combinatorics, Algorithms, and Applications, Special Issue on Electronic Notes in Discrete Mathematics*, 2002.

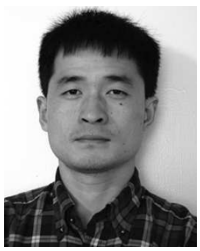
[4] A.-L. Barabasi, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: The topology of the world wide web," *Physica A*, vol. 281, pp. 69–77, 2000.
[5] T. Bu and D. Towsley, "On distinguishing between internet power law topology generators," *Proc. IEEE INFOCOM*, pp. 638–647, 2002.
[6] C. Baransel, W. Doboseiwicz, and P. Gburzynski, "Routing in multi-hop packet switching networks: Gbps challenge," *IEEE Netw. Mag.*, vol. 9, no. 3, pp. 38–61, 1995.
[7] W. G. Bridges and S. Toueg, "On the impossibility of directed moore graphs," *J. Combinator. Theory*, ser. B29, no. 3, pp. 339–341, 1980.
[8] F. Chung, "Diameters of communication networks," *AMS Proc. Symp. Applied Mathematics, Mathematics of Information Processing.*, pp. 1–18, 1984.
[9] J. Considine and T. A. Florio, "Scalable peer-to-peer indexing with constant state," Boston Univ., Boston, MA, Tech. Rep. 2002-026, Aug. 2002.
[10] D.-Z. Du, D. F. Hsu, H. Q. Ngo, and G. W. Peck, "On connectivity of consecutive-d digraphs," *Discrete Mathematics*, vol. 257, no. 2–3, pp. 371–384, 2002.
[11] R. Feldmann, B. Monien, P. Mysliwietz, and S. Tschoke, "A better upper bound on the bisection width of de bruijn networks," in *Proc. Symp. Theoretic. Aspects Comput. Sci. (STACS)*, 1997, pp. 511–522.
[12] A. Fiat and J. Saia, "Censorship resistant peer-to-peer content addressable networks," in *Proc. Symp. Discrete Algorithms*, 2002, pp. 94–103.
[13] P. Fraigniaud and P. Gauron, "An overview of the content-addressable network D2B," in *Proc. ACM PODC*, Jul. 2003, p. 151.
[14] M. J. Freedman and R. Vingralek, "Efficient peer-to-peer lookup based on a distributed trie," *Proc. IPTPS*, pp. 66–75, Mar. 2002.
[15] P. Ganesan, Q. Sun, and H. Garcia-Molina, "YAPPERS: A peer-to-peer lookup service over arbitrary topology," *Proc. IEEE INFOCOM*, pp. 1250–1260, 2003.
[16] K. P. Gummadi, R. Gummadi, S. D. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 381–394.
[17] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 48, no. 2, pp. 388–404, Mar. 2000.
[18] M. G. Hluchyj and M. J. Karol, "Shufflenet: An application of generalized perfect shuffles to multihop lightwave networks," *Proc. IEEE INFOCOM*, pp. 379–390, 1988.
[19] M. Imase and M. Itoh, "Design to minimize diameter on building-block network," *IEEE Trans. Computers*, vol. C-30, no. 6, pp. 439–442, 1981.
[20] F. Kaashoek and D. R. Karger, "Koorde: A simple degree-optimal hash table," *Proc. IPTPS*, pp. 98–107, Feb. 2003.
[21] A. Kumar, S. Merugu, J. Xu, and X. Yu, "Ulysses: A robust, low-diameter, low-latency peer-to-peer network," *Proc. IEEE ICNP*, pp. 258–267, 2003.
[22] C. Law and K.-Y. Siu, "Distributed construction of random expander graphs," *Proc. IEEE INFOCOM*, pp. 2133–2143, 2003.
[23] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. New York: Academic/Morgan Kaufmann, 1991.
[24] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the evolution of peer-to-peer networks," in *Proc. ACM PODC*, Jul. 2002, pp. 233–242.
[25] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 395–406.
[26] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: A scalable and dynamic emulation of the butterfly," in *Proc. ACM PODC*, Jul. 2002, pp. 183–192.
[27] G. S. Manku, "Routing networks for distributed hash tables," in *Proc. ACM PODC*, Jul. 2003, pp. 133–142.
[28] G. S. Manku, M. Bawa, and P. Raghavan, "Symphony: Distributed hashing in a small world," *Proc. USENIX Symp. Internet Technologies and Systems (USITS)*, 2003.
[29] G. S. Manku, M. Naor, and U. Weider, "Know thy neighbor's neighbor: The power of lookahead in randomized P2P networks," in *Proc. ACM STOC*, Jun. 2004, pp. 54–63.
[30] M. Naor and U. Wieder, "Novel architectures for P2P applications: The continuous-discrete approach," in *Proc. ACM SPAA*, Jun. 2003, pp. 50–59.
[31] G. Pandurangan, P. Raghavan, and E. Upfal, "Building low-diameter P2P networks," in *Proc. IEEE FOCS*, 2001, pp. 492–499.
[32] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environment," in *Proc. ACM SPAA*, Jun. 1997, pp. 311–320.
[33] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. ACM SIGCOMM*, Aug. 2001, pp. 161–172.
[34] S. Ratnasamy, S. Shenker, and I. Stoica, "Routing algorithms for DHTs: Some open questions," *Proc. IPTPS*, pp. 45–52, 2002.
[35] S. M. Reddy, J. G. Kuhl, S. H. Hosseini, and H. Lee, "On digraph with minimum diameter and maximum connectivity," in *Proc. Allerton Conf. Communications, Control and Computers*, 1982, pp. 1018–1026.
[36] J. Rolim, P. Tvrdik, J. Trdlicka, and I. Vrto, "Bisecting de Bruijn and Kautz graphs," *Discrete Appl. Math.*, vol. 85, no. 1, pp. 87–97, Jun. 1998.
[37] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," in *Proc. IFIP/ACM Int. Conf. Distributed Systems Platforms*, 2001, pp. 329–350.
[38] J. Saia, A. Fiat, S. Gribble, A. R. Karlin, and S. Saroiu, "Dynamically fault-tolerant content addressable networks," *Proc. IPTPS*, Mar. 2002.
[39] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl, "HyperCuP—Hypercubes, ontologies, and efficient search on P2P networks," in *Proc. Workshop on Agents and P2P Computing*, 2002.
[40] K. N. Sivarajan and R. Ramaswami, "Lightwave networks based on de Bruijn graphs," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 70–79, Jan. 1994.
[41] M. A. Sridhar and C. S. Raghavendra, "Fault-tolerant networks based on the de Bruijn graph," *IEEE Trans. Computers*, vol. 40, no. 10, pp. 1167–1174, Oct. 1991.
[42] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. ACM SIGCOMM*, Aug. 2001, pp. 149–160.
[43] D. W. Stroock, *Probability Theory, an Analytic View*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[44] D. A. Tran, K. A. Hua, and T. T. Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," *Proc. IEEE INFOCOM*, pp. 1283–1292, 2003.
[45] X. Wang, Y. Zhang, X. Li, and D. Loguinov, "On zone-balancing of peer-to-peer networks: Analysis of random node join," in *Proc. ACM SIGMETRICS*, Jun. 2004, pp. 211–222.
[46] J. Xu, A. Kumar, and X. Yu, "On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 151–163, Jan. 2004.
[47] B. Y. Zhao, J. D. Kubiatowicz, and A. Joseph, "Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing,", Univ. California Berkeley Tech. Rep., Apr. 2001.

Dmitri Loguinov (S'99–M'03) received the B.S. degree (with honors) in computer science from Moscow State University, Moscow, Russia, in 1995 and the Ph.D. degree in computer science from the City University of New York, New York, in 2002.

Since 2002, he has been an Assistant Professor of computer science with Texas A&M University, College Station. His research interests include peer-to-peer networks, video streaming, congestion control, Internet measurement, and modeling.

Juan Casas is currently working toward the B.S. degree in computer science at the University of Texas—Pan American, Edinburg, TX.

In the summer of 2004, he participated in the National Science Foundation REU research program at Texas A&M University. His research interests include distributed processing and computer networks.



Xiaoming Wang (S'04) received the B.S. degree in computer science and the M.S. degree in electronic engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1999 and 2002, respectively. He is currently working toward the Ph.D. degree at Texas A&M University, College Station.

During 2002–2003, he worked for Samsung Advanced Institute of Technology, South Korea. His research interests include peer-to-peer systems, probabilistic analysis of computer networks, and topology modeling.