Deconstructing DHTs

David Ratajczak* and Joseph M. Hellerstein*†
*University of California at Berkeley, †Intel Research Berkeley

Abstract— **Nearly all proposed DHTs have echoes – either explicit or implicit – of parallel interconnect networks such as *butterfly*, *torus*, *hypercube*, and *de Bruijn* graphs. However, unlike interconnection networks, DHTs define an overlay graph for all network sizes, and allow the overlay graph to evolve as nodes join and leave as participants. Most of the well-cited DHT designs obscured two basic concerns in DHT design: the choice of topology (the links and routes in "ideal" cases), and of interconnect "emulation" strategy (how they deal with dynamism and a sparsely filled identifier space). In this paper, we separate these concerns into two formal frameworks. We first observe several idealized DHT topologies from an algebraic standpoint, and discuss the utility of such an approach in creating and understanding new DHT topologies. We then examine several emulation schemes and consider their applicability to algebraic graphs. Given these pieces, we demonstrate a promising new DHT design that emerges from this separation of concerns: *IHOP* (Internet Hashing Over Pancake graphs), a Cayley graph of the symmetric group emulated with a scheme of Abraham et al.**

I. Introduction

Distributed hash tables (DHTs) have been proposed as a generic building block for large-scale distributed applications [1], [2], [3], [4], [5], [6]. Because the set of participating nodes in a DHT is assumed to be large and dynamic, it is too costly to synchronize each node's view of the current set of participants. Instead, the DHT approach is to form a *routing overlay* on which requests for data entries are routed toward the nodes currently managing them.

There are a slew of DHT designs and implementations, each claiming relevance on theoretical or practical grounds. But while the area of DHT design has been the focus of considerable research, there remains no consensus on the question, "What is a good DHT?" In this paper, we survey the field and address the more basic question, "What is a DHT?", with the hope that a better understanding of the anatomy of DHT designs will engender a more systematic comparison between them, as well as a more principled approach to future designs.

A careful study of the literature yields two distinct but complementary concerns in DHT design, which have been implicitly blended in prior proposals. The first concern is the *ideal topology*, which is the family of routing overlays that result when there is a *full assignment* of nodes to each point of a finite identifier space. For example, in Chord [6] with 2^n nodes assigned to all possible n-bit identifiers, the subsequent routing overlays correspond to a family of graphs called *Chord graphs* [7]: a ring composed with all chords of length 2^i for $i < n$. Nearly all of the proposed DHTs have ideal topologies that are familiar from the parallel computing literature on interconnection networks (ICNs). As further examples, Viceroy [2] has a butterfly topology, Pastry [5] and Kademlia [3] have hypercube topologies, CAN [4] has a d-dimensional torus topology, and Koorde [1] and the Distance Halving network [8] have de Bruijn graph topologies.

Whereas traditional ICNs assume a full (or nearly full) assignment, DHTs are designed to work when the identifier space is only sparsely filled due to nodes joining, leaving, and failing. Furthermore, DHT routing overlays are designed to enjoy similar congestion, degree, and diameter to their fully-assigned counterparts. Thus the second aspect of DHTs that we consider is the *emulation scheme*: the portion concerned with the incremental maintenance of the routing overlay. Whereas there are concrete metrics for comparing topologies, emulation schemes must be subjectively judged by their complexity, their assumptions on node reliability, and their ability to faithfully maintain congestion, degree, and diameter close to that afforded by the topology. The relationship between ideal topology and emulation scheme has been implicit in many DHT designs. However, recent work has given rise to explicit, more general purpose emulation techniques [8], [9], [10]. These techniques are not all fully generic, in that they may not apply to certain ideal topologies.

We believe that separating these concerns can lead to significant clarity in future research on DHTs. This outlook illuminates the relationship between DHTs and ICNs, placing the contributions of DHT designers in proper context, and allowing additional results from the ICN literature to be "ported" to peer-to-peer environments. As a constructive example, we present a promising new DHT design of this sort – *IHOP* (Internet Hashing Over Pancake graphs) – which combines a pancake graph [11] topology with the emulation scheme of Abraham et al. [9]. Our separation of concerns also admits the possibility that as designers target specific applications over DHTs, they can carefully choose a topology and emulation scheme to suit their applications' communication patterns.

A. Outline

We begin by revisiting a group-theoretic model for network topologies proposed by Akers and Krishnamurthy [11]. They observed that many topologies have a natural algebraic representation as coset graphs [12], of which *Cayley graphs* are a special case. This stems directly from the symmetric nature of topologies and a result by Sabidussi [13] showing that all vertex-symmetric graphs can be represented as coset graphs. We examine DHT topologies in this context, giving explicit algebraic representations of currently proposed DHT topologies, as well as some that have not yet been used in DHTs. The lack of physical wiring constraints in DHTs means that algebraically derived topologies, sometimes too complex to cast in hardware, have renewed promise in DHTs and should be reconsidered in that light.

We then discuss several emulation schemes proposed both for specific and general topologies. We examine them in an algebraic context and evaluate their effectiveness in emulating coset graphs. Specifically we consider the schemes of Manku [10], Abraham et al. [9] and Naor and Wieder [8]. We conclude with the design of IHOP, and a discussion of the possibilities for new DHT designs that are tailored to specific communication patterns.

II. Algebraic Graphs

In this section we assume some familiarity with the basic terminology of group theory.

A. Cayley Graphs

Given a finite group G and a set of elements (generators) S in that group, the elements of G representable as a product of generators and their inverses is the "subgroup of G generated by S." When S generates G, then we can draw the *Cayley graph* of (G, S), where the nodes of the graph correspond to elements of G, and there is an edge (g_1, g_2) if and only if there is a generator $s \in S$ such that $g_1 s = g_2$.

We first consider subgroups of the symmetric group S_n, whose elements are permutations of n elements and whose operation is index permutation.[1] In the sequel, we may represent permutation elements either with their one-row representation (ex: e="12345"), their cycle representation (ex: (12)(345)="21453"), or their transposition representation (ex: (1,3)="32145").

Example 1: Let G be the subgroup of S_{2n} of 2^n elements generated by the transpositions $(2i-1, 2i)$ for $i = 1$ to n. The Cayley graph of G is isomorphic to the n-dimensional hypercube, and has diameter and degree of n.

Example 2: Let G be the subgroup of S_{2n} of $n2^n$ elements generated by $(12\dots 2n)^2$ and $(12\dots 2n)^2(12)$. The Cayley graph is isomorphic to a butterfly graph [14].

The following two examples involve the complete symmetric group S_n.

Example 3: Let G be the symmetric group S_n with $n!$ elements generated by $(1, i)$ for $i = 2$ to n. The resulting Cayley graph is called a star *graph [11], and has $O(n)$ diameter and degree $n-1$.*

Example 4: Let G be S_n generated by the permutations "$i\ i-1 \dots 1\ i+1\ i+2 \dots n$" for $i = 1$ to n. These correspond to flipping the first i elements in a permutation. The resulting Cayley graph is called a pancake *graph [11], and has $O(n)$ diameter and degree n.*

Of the above examples, the hypercube is distinguished by the additional property that it is abelian: for any two generators s_1, s_2, $s_1 s_2 = s_2 s_1$. In considering further abelian groups, we make use of the additive cyclic group[2] of n elements Z_n with identity 0 and cyclic generator 1. In the cartesian product Z_n^d we denote 1_i to be the d-tuple with 1 in the ith position and 0 elsewhere.[3]

Example 5: Let G be the group Z_n^d with n^d elements generated by 1_i for $i = 1$ to d. The Cayley graph is isomorphic to the d-dimensional torus and has diameter dn and degree $2d$ (recall that the inverses of the generators also produce an edge).

Example 6: Let G be the group Z_{2^n} of 2^n elements with generators 2^{i-1} for $i = 1$ to n. The Cayley graph is

[1] By Cayley's Group Theorem, all groups of order n are isomorphic to subgroups of S_n, though this is not always the most natural representation.

[2] All cyclic groups of order n are isomorphic to each other.

[3] By the Kronecker Decomposition Theorem, every finite abelian group is a direct product of cyclic groups of prime power group order.

a Chord graph [6] with diameter $\lfloor n/2 \rfloor$ [7] and degree $2n$.

Notice that in the above example, the entire group can be generated by a single generator (1). Any group with a single generator is a cyclic group.

We finally point out that representations are not unique, as the following example illustrates.

Example 7: Consider the group Z_2^n of 2^n elements with generators 1_i for $i = 1$ to n. The Cayley graph is isomorphic to the n-dimensional hypercube.

B. Properties

A graph is *vertex symmetric* if there is an automorphism of the graph mapping any node of the graph into any other. It is not difficult to show that every Cayley graph is regular and vertex symmetric. Vertex symmetry is strongly desirable in a DHT for two reasons. The first reason is that a path from x to y that uses generators $s_1 s_2, \ldots, s_k$ is also a path from $y^{-1}x$ to the identity e. In the setting of permutation groups, this implies that routing between x and y is equivalent to "sorting" the permutation $y^{-1}x$. More importantly, it means that each node need only store (or compute) the optimal routes from all elements to the identity, implying they can execute the same algorithm. Secondly, a vertex symmetric graph (with symmetric routing) exhibits balanced node congestion, since there are equally many distinct paths through each vertex.

Cayley graphs tend to have a number of other desirable properties as well, including low diameter for a given degree [15], and high expansion and fault-tolerance [16]. There is, however, a fundamental difference between Cayley graphs of abelian and nonabelian groups, as Babai et al. [15] show that every nonabelian finite simple group has a set of at most 7 generators resulting in logarithmic diameter, whereas an abelian group of n elements and c generators must have diameter $\Omega(n^{1/c})$.

Unfortunately, it was shown by Even and Goldreich [17] that computing the diameter of an arbitrary Cayley graph over a set of generators is NP-hard. Furthermore, constructing an optimal routing algorithm is PSPACE-hard [18]. Even for simple examples, such as pancake graphs, the exact diameter is still unknown [11], and the addition of new generators to a group can dramatically change the optimal routing algorithm, as shown in a recent analysis of Chord graphs [7].

It is for this reason that the primary utility of an algebraic model is not in the creation of topologies, but rather in understanding their structure and capturing the relationships between seemingly unrelated topologies. We enumerate three areas in which an algebraic perspective has increased the understanding of network topologies.

1) Coset Graphs: While every Cayley graph is vertex symmetric, not all vertex symmetric graphs are representable as Cayley graphs.[4] However, the result of Sabidussi [13] shows that every vertex symmetric graph is a *coset graph*, obtained from a group G with generators S and a subgroup $H \subset G$, where the vertices correspond to left cosets of H in G, and an edge exists from xH to yH if and only if there exists $h_1, h_2 \in H$ such that $xh_1 = yh_2$.[5]

Coset graphs also allow us to describe some other familiar "semi-uniform" graphs which are not vertex symmetric, but which do exhibit algebraic structure. For example, Annexstein et al. [12] showed that de Bruijn graphs are coset graphs of butterfly graphs.

Example 8: Let G be the subgroup of S_{2n} of $n2^n$ elements generated by $(12\ldots 2n)^2$ and $(12\ldots 2n)^2(12)$, as in Example 2. Let H be the subgroup of n elements generated by $(12\ldots 2n)^2$. The coset graph is isomorphic to de Bruijn graphs.

A similar relationship exists between cube-connected cycles (a Cayley graph) and shuffle-exchange networks [12].

2) Hierarchical Structure: Implicit in our discussion thus far, is that a DHT topology is not just a graph, but rather a family of graphs $G_0, G_1, G_2, \ldots$ defined for a set of "ideal" network sizes. An algebraic model can elucidate the relationships between successive graphs G_i and G_{i+1}. For *hierarchical* Cayley graphs [11], whose generators can be ordered as $s_1, s_2, \ldots, s_d$ such that each s_{k+1} is outside the subgroup generated by $s_1, \ldots, s_k$, then G_{i+1} can be viewed as a collection of copies of G_i with edges between them corresponding to the actions of the additional generators in G_{i+1}. For example, a Chord graph of n-bit identifiers C_n can be viewed as two interleaved copies of C_{n-1} with "ring" links between adjacent vertices.

3) Simulation: Annexstein et al. [12] show through algebraic techniques that certain topologies can efficiently simulate algorithms designed for much larger topologies without a significant performance slowdown (e.g. a de Bruijn graph can efficiently simulate a butterfly graph). This type of result may become increasingly

[4] The Petersen graph is a counterexample.

[5] Notice that if H is a normal subgroup, then the coset graph is again a Cayley graph of the quotient group G/H.

important in the DHT setting, as DHTs may soon support higher-level functionality (such as aggregation) that is optimized for a specific topology (such as a hypercube) for simplicity, but which must then be appropriately simulated on other DHT topologies in the field.

III. Emulation Schemes

In this section, we describe several emulation schemes and highlight their common elements. Note that until recently, emulation schemes were never an explicitly defined component of a DHT.

A. Continuous-Discrete Approach [8]

We begin by describing a fairly intuitive scheme proposed by Naor and Wieder which is similar to the schemes implicitly used in several DHTs. In this approach, one constructs a *continuous graph* G_∞ on an infinite space (typically $[0, 1)$) so that every point has a set (possibly infinite) of edges to other points in the space. Since there are only a finite number of nodes, each node v is associated with a partition P_v of the space, and maintains an edge to another node u if and only if there is an edge $(x, y) \in E(G_\infty)$ such that $x \in P_v$ and $y \in P_u$. When a node joins, another node's partition is split, similarly when a node leaves, its partition is merged with another node's partition.

The Distance Halving Network [8] applies this approach to an infinite de Bruijn graph, where every point $x \in [0, 1)$ has edges to $x/2$, $x/2+1/2$, and $2x$. Because these are continuous functions, two points that are close to each other will have neighbors that are close to each other. As a result, the authors show that assigning intervals from $[0, 1)$ to each node yields average degree of at most 6. Furthermore, if these intervals are nearly equal in size, the maximum node degree is also constant. However, the routing strategy is non-optimal, as it only uses the $2x$ edges.

Koorde [1] utilizes a very similar approach, though instead of keeping all of the $2x$ edges associated with its interval, a node only keeps the $2x$ edge of the *lowest* point of its interval, and a pointer to the "successor" node managing the abutting higher interval. Thus a hop of the Distance Halving Network is simulated in Koorde by a traversal on a $2x$ edge, and then a sequence of successor hops. Thus the even distribution of intervals is required for logarithmic path lengths, but not for bounding the degree.

Many of the other early DHTs, including Kademlia [3], Pastry [5], CAN [4], and Chord [6] have similar approaches, with small optimizations that are topology-specific.

While this scheme is conceptually simple, it is not clear if the infinite graph is well-defined for some topologies, such as for pancake graphs and star graphs, which are Cayley graphs of the symmetric group.

B. Abraham et al. [9]

In contrast, this scheme does not require an infinite graphs, and produced better load-balancing than the scheme above. It works by viewing the set of node identifiers as a search tree with keys at the leaves of the tree, hence a joining node will pick a node in a ahallow part of the tree (corresponding to a long interval) and together they will split the interval, each taking tree positions that are deeper by one level. A leaving node will merge its interval with its "sibling" and the remaining sibling will take a tree position that is shallower by one level. Balancing the length of intervals thus corresponds to keeping a balanced tree, and the authors show a local scheme that keeps the tree balanced to within an $O(1)$ additive factor w.h.p.

For a topology $G_0, G_1, \ldots,$ positions of the search tree at level i correspond to vertices of G_i. A node at depth i will therefore have a set of links prescribed by G_i and its position in the search tree. Since not all nodes are at the same depth, the scheme requires some care in the way vertices are mapped to positions in the tree. In particular, the sequence $G_0, G_1, \ldots$ must posess a recursive structure dubbed *parent-child commutativity*, so that the vertices of G_{i+1} that are children (in the tree) of a vertex v in G_i must have neighbors that are children of neighbors of v. The authors show that this property is satisfied by de Bruijn graphs, butterfly graphs, and hypercubes. We have shown that it is true for pancake graphs, though we suspect it to be untrue for most other algebraic graphs.

C. Manku [10]

This scheme differs from the previous ones in that it allows emulation of arbitrary families of graphs that need not possess any recursive structure. It does this by relying instead on a precise estimate of the size of the network, obtained in a decentralized manner, so that nearly every node has an estimate of $\log n$ within 1 of every other node's estimate (this is specific to graphs of size 2^n but a similar technique works for graphs of other sizes). Each node has edges associated with G_{n-1}, G_n, and G_{n+1}, where n is its estimate, and thus there will be some level $G_{n'}$ whose edges will exist at nearly every node. Routing initially follows the link corresponding to the smallest level at the source, and it will switch to a higher level if necessary along the way. Since nodes are not evenly spaced in the identifier space, the choice of level n is reduced so that there may be a cluster of nodes mapping to each vertex of G_n, and at least one node will map to each vertex with high probability. Thus the routing prescribed by the topology is actually between clusters, and routing within a cluster is handled by a local link structure.

IV. IHOP

We now wish to merge the pancake graph topology of Example 4, with the emulation scheme of Abraham et al.[6] For a given permutation π_i of i elements, we define the *parent* of π_i to be the permutation of $i-1$ elements obtained by removing the ith element from π_i (e.g. P("12354")="1234"). In this manner, we can embed the permutation groups S_n in a tree, with the $i!$ elements of S_i mapping to all nodes at depth i. Next, we must show that the parent-child commutativity property is satisfied by such an embedding. In particular, we must show that for any permutation π_i, any neighbor of any child of π_i is the child of some neighbor of π_i.

Let $C_k(\pi_i)$ denote the child of π_i with the $i+1$st element in the kth position. Let $F_j(C_k(\pi_i))$ denote the permutation obtained by flipping the first j elements of that child. Then we can see that $F_j(C_k(\pi_i)) = C_k(F_j(\pi_i))$ if $j < k$, or $C_{j-k+1}(F_j(\pi_i))$ otherwise.

Nodes joining and leaving the network induce splits and merges in the tree of identifiers (as described in [9]). However, since the tree of permutations is not a binary tree, during a split the two affected nodes may have to simulate several nodes "virtually", and further joining nodes must then assume responsibility for these children until they are all associated with real nodes. A similar process must occur with merging. We refer the reader to [9] for a full description of the scheme.

Within a level, we route by using a naive "back-to-front" flipping strategy that flips the final element of the target to the front and then to the back, then repeats this process recursively. This yields a diameter of $2k-3$ on S_k, or if there are n nodes, implies a degree and diameter of $O(\log n / \log\log n)$. To our knowledge, this is the first DHT design providing this diameter-degree tradeoff without assuming an *a priori* upper-bound on the size of the network.

[6] This section requires a familiarity with [9].

V. Discussion

The early and most influential DHT designs were described with metric-space analogies (routing on rings, torii, etc) that obscured the underlying topologies. The result, in some cases, was that the proposed routing algorithms did not utilize all of the links available or the flexibility of routes that the topology allowed; examples have been identified clearly by Ganesan and Manku [7] in the context of routing on Chord graphs, and by Gummadi et al. [19] in comparing the routing flexibility of trees and hypercubes.

We suspect that as DHTs are considered for increasingly complex tasks – such as broadcast, aggregation, and database querying – and on networks of limited connectivity – such as sensor networks – it will be necessary to more fully understand the ways in which topology and emulation interact, and to more clearly assess the limitations imposed by a particular choice of topology. We view this paper as merely a small step in one possible direction. There remain several important issues such as atomicity, fault-tolerance, and routing flexibility that will also crucially impact design choices, but which we have not specifically addressed. However, we hope that an algebraic approach may yield new insights in these areas as well, and that results from the ICN literature can be brought to bear in the context of DHTs.

Acknowledgments

The authors would like to thank Christos Papadimitriou and David Karger for helpful input.

References

[1] F. Kaashoek and D. Karger, "Koorde: A simple degree-optimal distributed hash table," in *International Peer-to-Peer Symposium*, 2003.

[2] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: A scalable and dynamic emulation of the butterfly," in *Proc. 21st ACM Symposium on Principles of Distributed Computing*, 2002.

[3] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proceedings of International Peer-to-Peer Symposium*, 2002.

[4] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," in *Proceedings of ACM SIGCOMM 2001*, 2001.

[5] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," *Lecture Notes in Computer Science*, vol. 2218, pp. 329–350, 2001.

[6] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "A scalable content addressable network," in *Proceedings of ACM SIGCOMM 2001*, 2001.

[7] P. Ganesan and G. S. Manku, "Optimal routing in Chord," in *ACM SIAM Symposium on Distributed Algorithms(SODA)*, 2004.

[8] M. Naor and U. Wieder, “Novel architectures for p2p applications: the continuous-discrete approach,” in *ACM Symposium on Parallel Algorithms and Architectures*, 2003.
[9] I. Abraham, B. Awerbuch, Y. Azar, Y. Bartal, D. Malkhi, and E. Pavlov, “A generic scheme for building overlay networks in adversarial scenarios,” in *International Parallel and Distributed Processing Symposium*, Nice, France, April 2003.
[10] G. S. Manku, “Routing networks for distributed hash tables,” in *Proc. 22nd ACM Symposium on Principles of Distributed Computing*, June 2003.
[11] S. Akers and B. Krishnamurthy, “A group-theoretic model for symmetric interconnection networks,” *IEEE Transactions on Computers*, vol. 38, no. 4, pp. 555–566, 1989.
[12] F. Annexstein, M. Baumslag, and A. L. Rosenberg, “Group action graphs and parallel architectures,” *SIAM Journal on Computing*, vol. 19, pp. 544–569, 1990.
[13] G. Sabidussi, “Vertex-transitive graphs,” *Monatsheft für Mathematik*, vol. 68, pp. 426–438, 1964.
[14] G. Chen and F. Lau, “Comments on ’a new family of cayley graph interconnection networks of constant degree four’,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 8, no. 12, pp. 1299–1300, 1997.
[15] L Babai, G. Hetyei, W. M. Kantor, A. Lubotzky, and A. Seress, “On the diameter of finite groups,” in *IEEE Symposium on Foundations of Computer Science*, 1990, pp. 857–865.
[16] Alon and Roichman, “Random cayley graphs and expanders,” *RSA: Random Structures and Algorithms*, vol. 5, 1994.
[17] S. Even and O. Goldreich, “The minimum-length generator sequence problem is NP-hard,” *J. Algorithms*, vol. 2, no. 3, pp. 311–313, 1981.
[18] M. R. Jerrum, “The complexity of finding minimum length generator sequences,” *Theoretical Computer Science*, vol. 36, pp. 265–289, 1985.
[19] 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 *Proceedings of the ACM SIGCOMM*, 2003.