Cayley DHTs — A Group-Theoretic Framework for Analyzing DHTs Based on Cayley Graphs

Changtao Qu, Wolfgang Nejdl
L3S and University of Hannover
Expo Plaza 1, D–30539 Hannover, Germany
{qu, nejdl}@l3s.de

Matthias Kriesell
Institute of Mathematics, University of Hannover
Welfengarten 1, D-30167 Hannover, Germany
kriesell@math.uni-hannover.de

Abstract

Static DHT topologies influence important features of such DHTs such as scalability, communication load balancing, routing efficiency and fault tolerance. Nevertheless, it is commonly recognized that the primary difficulty in designing DHT is not in static DHT topologies, but in the dynamic DHT algorithm which adapts various static DHT topologies to a dynamic network at Internet. As a direct consequence, the DHT community has been paying more attention to the dynamic DHT algorithm design, resulting in a variety of DHT systems lacking of a common view for analysis and interoperation.In this paper we reiterate the importance of static DHT topologies in the DHT system design by analyzing and classifying current DHTs in terms of their static topologies based on a grouptheoretic model: Cayley graphs. We show that most of current DHT proposals use Cayley graphs as static DHT topologies, thus taking advantage of several important Cayley graph properties such as vertex/edge symmetry, decomposability, optimal fault tolerance and hamiltonicity. We observe that several non-Cayley-graph based DHT proposals such as Koorde/D2B/Distance Halving and Pastry/Tapestry also rely on techniques in their dynamic DHT algorithm design trying to imitate desirable Cayley graph properties. Based on Cayley graphs, we propose the class of Cayley DHTs as a unified group-theoretic model for investigating DHTs from a graph theoretic perspective. The significance of Cayley DHTs is in their explicit inspiration to a uniform dynamic DHT algorithm design, which can directly leverage algebraic design methods thus is able to generate sets of high-performance DHTs adopting various Cayley graph based static DHT topologies but still sharing the same dynamic DHT algorithm.

Keywords: *DHT, Peer-to-Peer, Cayley Graphs.*

1. Introduction

P2P (Peer–to–Peer) systems are generally characterized by the lack of centralized control or hierarchical organization. A critical research issue in such distributed systems is the *lookup problem*, namely, how to "find any given data item in a large P2P system in a scalable manner" [4]. Looking up data in early file sharing P2P systems such as Gnutella or Freenet is not scalable. These P2P systems are mostly based on unstructured content delivery overlay networks, in which the lookup method is either *flooding* (the query is broadcast to all neighbors) or *random walking* (the query is forwarded to randomly chosen neighbors until the object is found) [19]. As neither method can guarantee data location, or ensure that querying terminates once the data are located, or prevent one node from receiving the same query several times, looking up data in such unstructured P2P systems becomes increasingly hard with growing network size.

To address the lookup problem, new types of P2P systems using structured content delivery overlay networks have been proposed, which use distributed hash tables (DHTs) as underlying design concept. DHTs provide strictly controlled network topologies to organize data and computing nodes, in which data placement and lookup algorithms are precisely defined based on a DHT data structure [4]. Data items are associated with keys, and each node is responsible for storing a certain range of keys. Data lookup in DHTs is based on key lookup. As dynamic DHT algorithms build upon static DHT topologies which allow any data item to be located using a bounded number of routing hops (based on small per–node routing tables), data lookup in DHTs is scalable.

In practice, DHT design often starts with the decision for an appropriate static DHT topology with good properties regarding node location, and then adopts and decides on static DHT topology properties such as the *network diameter* (number of hops needed to get a message from source to destination) or *network degree* (number of neighbors with which a node maintains continuous contact) as crucial criteria to measure DHTs. The static DHT topologies employed actually influence important features such as scalability, communication load balancing, routing efficiency, fault tolerance, and proximity. Nevertheless, it is commonly recognized that the primary difficulty in designing DHT is not in static DHT topologies, but in the dynamic DHT algorithm which adapts various static DHT topologies to a dynamic network at Internet. As a direct consequence, the DHT community has been paying more attention to the dynamic DHT algorithm design, resulting in a variety of DHT systems lacking of a common view for analysis and interoperation. In this paper we reiterate the importance of static DHT topologies in the DHT system design by analyzing and classifying current DHTs in terms of their static topologies based on a grouptheoretic model: Cayley graphs [1]. We propose the class of Cayley DHTs as a unified group-theoretic model for investigating DHTs from a graph theoretic perspective. The paper builds upon previous results for (static) symmetric interconnection networks and graph theory, and brings these together in a unifying framework for current DHT designs. The significance of Cayley DHTs is in their explicit inspiration to a uniform dynamic DHT algorithm design, which can directly leverage algebraic design methods thus is able to generate sets of high-performance DHTs adopting various Cayley graph based static DHT topologies but still sharing the same dynamic DHT algorithm.

The remainder of this paper is structured in the following way. In Section 2 we provide the overview of current DHT proposals and corresponding static DHT topologies employed. In Section 3 we prove that most of current DHT proposals are Cayley DHTs using Cayley graphs as static DHT topologies. In Section 4 we associate Cayley graph properties with DHT system features and further provide insight into current DHTs in terms of our unified analytical framework. In Section 5 we discuss several major results we have arrived at and sequentially provide clear guidelines for the future DHT design. In Section 6 we present several related work to our research. In Section 7 we highlight the significance of Cayley DHTs as concluding remarks.

2. DHTs and Static DHT Topologies

Current DHT proposals use a variety of static topologies to structure their content delivery overlay networks. The following table lists several DHT systems and their static DHT topologies.

DHTs and Static DHT Topologies

DHTs	static topology	(av.) degr.	diameter
HyperCup	hypercubes	$O(\log N)$	$O(\log N)$
Chord	ring graphs	$O(\log N)$	$O(\log N)$
Pastry/Tapestry	Plaxton trees	$O(\log N)$	$O(\log N)$
Viceroy	butterfly	$O(1)$	$O(\log N)$
Cycloid	cube conn. cycles	$O(1)$	$O(d)$, $N = d \cdot 2^d$
Koorde/D2B	de Bruijn	$O(1)$	$O(\log N)$
DistanceHalving	k-base de Bruijn	$O(k)$	$O(\log N/\log k)$
CAN	d-dim. torus	$O(d)$	$O(d \cdot N^{1/d})$

Two important characteristics of DHTs are network degree and network diameter. As DHTs are formed and maintained completely on the fly through dynamic DHT algorithms, high network degree means that joining, leaving and failing nodes affect more other nodes. Based on network degree, we group static DHT topologies into two types: *non–constant degree DHT topologies*, whose network degree increases (logarithmically) with the number of nodes in the network, and *constant (average) degree DHT topologies*, whose network degree stays constant even when the network grows. Consequently, DHTs can be classified into *non-constant degree DHTs* such as HyperCup(hypercubes), Chord (ring graphs), Pastry/Tapestry (Plaxton trees), etc., and *constant degree DHTs* such as Viceroy (butterfly), Cycloid (cube connected cycles), Koorde/D2BDistance Halving (de Bruijn) and CAN (tori)[1].

Though this classification is certainly useful, the listed DHT topologies seem to have nothing more in common. Each topology exhibits specific graph properties resulting in specific DHT system features. Consequently, DHTs have so far been analyzed comparing individual systems, without a unified analytical framework which allows further insight into DHT system features and DHT system design.

The unified analytical framework discussed in this paper – *Cayley DHTs* – allows us to compare DHT topologies on a more abstract level and characterizes common features of current DHT designs. In a nutshell, we will show that most current static DHT topologies such as hypercubes, ring graphs, butterflies, cube-connected cycles, and d-dimensional tori fall into a generic group-theoretic model, Cayley graphs, and thus can be analyzed as one class. These Cayley graph based DHTs (hereafter *Cayley DHTs*), including both non-content degree DHTs and constant degree DHTs intentionally or unintentionally take advantage of several important Cayley graph properties such as vertex/edge symmetry, decomposability, good connectivity and hamiltonicity to achieve DHT design goals such as scalability, communication load balancing, optimal fault tolerance, and routing efficiency. Several non-Cayley DHTs also utilize techniques in their dynamic DHT algorithms that try

1 Distance Halving and CAN can be viewed as adjustable, constant degree DHTs since their network degrees depend on the network dimension

to imitate desirable Cayley graph properties, again showing the close relationship between Cayley graph properties and desirable DHT system features.

3. Cayley DHTs — A Group-Theoretic Model for Analyzing DHTs

3.1. Groups and Cayley Graphs

Cayley graphs were originally proposed as a generic group-theoretic model for analyzing symmetric interconnection networks [1]. The most notable feature of Cayley graphs is their universality. Cayley graphs embody almost all symmetric interconnection networks, since every vertex transitive interconnection network can be represented as the quotient of two Cayley graphs [30]. They represent a class of high performance interconnection networks with small degree and diameter, good connectivity, and simple routing algorithms.The following paragraphs give the formal definitions of groups and Cayley graphs.

Definition 1 *A* group *is a pair* $\Gamma := (V, \cdot)$ *such that* V *is a (nonempty) set and* $\cdot : V \times V \longrightarrow V$ *maps each pair* (a, b) *of elements of* V *to an element* $a \cdot b$ *of* V *with* $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ *for all* $a, b, c \in V$*, such that there exists an element* $1 \in V$ *with the following properties:*

(i) $e \cdot a = a$ *for all* $a \in V$ *and*

(ii) for every $a \in V$*, there exists some* $b \in V$ *with* $b \cdot a = 1$.

1 is the unique element having properties (i) and (ii). It is called the *neutral element* of Γ, and $a \cdot e = a$ holds for all $a \in V$. b as in (ii) is uniquely determined by a and is called the *inverse* of a, written as $b = a^{-1}$. It is the unique element b for which $a \cdot b = 1$ holds. If $a \cdot b = b \cdot a$ holds for all $a, b \in V$ then Γ is called an *abelian* group. This is usually expressed by *additive notation*, i. e. by writing $\Gamma = (V, +)$, 0 for the neutral element, and $-a$ for the inverse of a. Groups are fundamental objects of mathematics, and the foundation for Cayley graphs. We will see several examples for them, based on the the following definition.

Definition 2 *Let* $\Gamma := (V, \cdot)$ *be a finite group,* 1 *its neutral element, and let* $S \subseteq V - \{1\}$ *be closed under inversion (i. e.* $x^{-1} \in S$ *for all* $x \in S$*). The* Cayley graph $G(\Gamma, S) = (V, E)$ *of* $(V, \cdot)$ *and* S *is the graph on* V *where* x, y *are adjacent if and only if* $xy^{-1} \in S$.

We can also define *directed versions* of this concept, which are obtained by omitting the symmetry condition $S^{-1} = S$ to S. The condition $1 \notin S$ keeps Cayley graphs *loopless*. Note that Cayley graph are sometimes called *group graphs*. For some historical notes of these definitions, see [17].

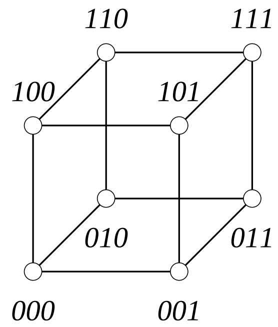

Figure 1. 3**-dimensional binary hypercube**

3.2. Non-constant Degree Cayley DHTs

HyperCup [31] Though HyperCup itself is not a DHT system, it is a topology for structured P2P networks which could also be used for DHT design, and which represents an important type of Cayley graphs, hypercubes. So far there are no DHTs which use pure hypercubes as static DHT topologies, even though some literature (i.e. [32, 16]) argue that Pastry/Tapestry and Chord emulate approximate hypercubes when taking into account the dynamic DHT algorithm design. However, differentiating cleanly between static DHT topologies and dynamic DHT algorithms, it is more appropriate to describe their static topologies as Plaxton trees and ring graphs respectively.

Hypercubes are typical Cayley graphs. For a natural number m, let $(\mathcal{Z}_m, +)$ denote the group of residuals $\mod m$. Consider the group $\Gamma := (\mathcal{Z}_2^d, +)$, where $\mathcal{Z}_2^d$ denotes the set of all $0, 1$-words of length d and $+$ is the componentwise addition $\mod 2$. We want to make a, b adjacent whenever they differ in exactly one digit, i.e. whenever $a - b$ is a word containing precisely one letter 1. So if S is the set of these d words then S is closed under inversion, and $H_2^d := G(\Gamma, S)$ is called the (binary) *d-dimensional (binary) hypercube*. Figure 1 represents H_2^3.

It is also possible to give a *hierarchical description* of H_2^d by means of the following recursion. Set $H_2^1 = (\{0, 1\}, \{01\})$, and for $d > 1$ define H_2^d recursively by $V(H_2^d) := \{xv : x \in \{0, 1\}, v \in V(H_2^{d-1})\}$ and $E(H_2^d) := \{xvyw : xv, yw \in V(H_2^d)$ and: $(x = y \wedge vw \in E(H_2^{d-1}))$ or $(x \neq y \wedge v = w)\}$. Roughly, in every step, we take two disjoint copies of the previously constructed graphs and add edges between pairs of corresponding vertices.

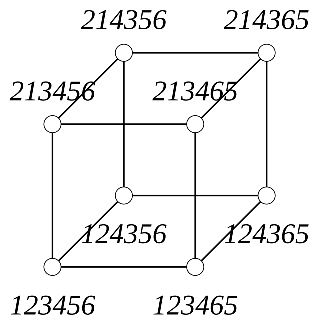

Figure 2. 3**-dimensional binary hypercube (alternative representation)**

This concept can be generalized by looking at *cartesian products* of graphs: For graphs G, H, let their *product* $G \times H$ be defined by $V(G \times H) := V(G) \times V(H)$ and $E(G \times H) := \{(w,x)(y,z) : (w = y \in V(G) \wedge yz \in E(H))$ or $(y = z \in V(H) \wedge wx \in E(G))\}$. Clearly, $G \times H$ and $H \times G$ are isomorphic (take $(x, y) \mapsto (y, x)$ as an isomorphism). Defining $K_2 := (\{0,1\}, \{01\})$ to be the complete graph on two vertices, we see that H_2^1 is isomorphic to K_2 and H_2^d is isomorphic to $H_2^{d-1} \times K_2$ for $d > 1$, which is in turn isomorphic to $\underbrace{K_2 \times \cdots \times K_2}_{d \text{ times}}$.

As every finite group is isomorphic to some group of permutations, it is possible to unify the Cayley graph notion once more. Without loss of generality, we could assume that the generating group Γ is a permutation group. This is certainly useful when describing algorithms on general Cayley graphs. For the presentation here, it is, however, more convenient to involve other groups as well. Figure 2 gives the following, alternative representation of H_2^3. For Γ, we take the subgroup of the permutation group $S_6 := (\{f : \{1,\ldots,6\} \longrightarrow \{1,\ldots,6\} : f \text{ bijective}\}, \circ)$ generated by $S := \{213456, 124356, 123465\}$ (here $a_1 \cdots a_6$ denotes the permutation f of $\{1,\ldots,6\}$ with $f(1) = a_1, f(2) = a_2, \ldots, f(6) = a_6$). $f \circ g$ is the permutation defined by $(f \circ g)(x) = f(g(x))$ for all possible x.)

A third group which can be involved here to represent hypercubes as Cayley graphs comes from coding theory. Let V be the set of all 2^d subsets of some d-element set, and let $A \Delta B := (A \cup B) - (A \cap B)$ denote the *symmetric difference* of A and B. The cardinality of $A \Delta B$ is often called the *Hamming distance* of A and B Then $\Gamma = (V, \Delta)$ is a group, its neutral element is the empty set, and taking S to be the set of all 1-element subsets of V produces a Cayley graph $G(\Gamma, S)$ isomorphic to H_2^d. The distance of two vertices in this graph is precisely their *Hamming distance*.

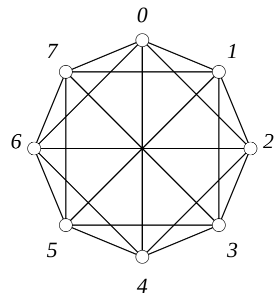

Figure 3. Chord ring with 3**-bit key space**

Chord [33] Chord uses a 1-dimensional circular key space, in which the node responsible for a key is the node whose identifier most closely follows the key in the numeric order (the key's *successor*). All nodes in Chord are arranged into a *ring graph*. In a m-bit Chord key space, each Chord node maintains two sets of neighbors: a successor list of k nodes that immediately follow it in the key space, and a finger list of $O(\log N)$ nodes spaced exponentially around the key space. The ith entry of the finger list points to the node that is 2^i away from the current node, or to that node's successor if that node does not exist.

The graphs approximated here are special *circulant graphs*, i. e. Cayley graphs obtained from the cyclic group $(\mathcal{Z}_n, +)$ and an arbitrary (inversion–closed) generator set.The most prominent example is the *cycle* $C_n := G(\mathcal{Z}_n, \{\pm 1\}) = (\mathcal{Z}_n, \{01, 12, 23, \ldots, (n-1)n, n0\})$ of length n. (An old Theorem of Turner [34] states that every vertex transitive graph with a prime number of vertices must be such a circulant.)

For the topology of the ideal d-bit Chord key space, we simply take the Cayley graph $R_d := G((\mathcal{Z}_{2^d}, +), \{\pm 2^k : k \in \{0, \ldots, d-1\}\})$. Figure 3 shows the topology of an ideal 3-bit Chord key space i. e. the graph $G(\mathcal{Z}_8, \{\pm 1, \pm 2, 4\})$.

The Chord cycle graph with 3-bit key space maintains 20 edges. The topology can be covered by a hypercube and a skew-symmetric variant of the hypercube called *hyperskewbe* [15], as illustrated in Figure 4. This *skew* variant of a cube is defined recursively as follows: $\tilde{H}_2^1 = (\{0,1\}, \{01\})$, and for $d > 1$ define $\tilde{H}_2^d$ recursively by $V(\tilde{H}_2^d) := \{xv : x \in \{0,1\}, v \in V(\tilde{H}_2^{d-1})\}$ and $E(\tilde{H}_2^d) := \{xvyw : xv, yw \in V(\tilde{H}_2^d)$ and: $(x = y \wedge vw \in E(\tilde{H}_2^{d-1}))$ or $(x \neq y \wedge v = s(w))\}$. To calculate $s(w)$, we consider the binary word w as a residual $z \mod 2^{d-1}$, and let $s(w)$ be the binary word representing $z + 1 \mod 2^{d-1}$. Note that this is very similar to the recursive definition of H_2^d given above. Note also that, for $d > 2$, the diameter of $\tilde{H}_2^d$ is smaller than the diameter of H_2^d.

Let's have a look at the particular graph of $\tilde{H}_2^3$; it is a tri-

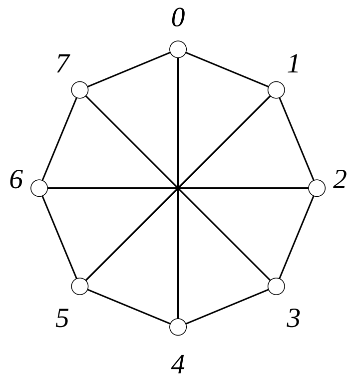

Figure 4. 3**-dimensional hyperskewbe**

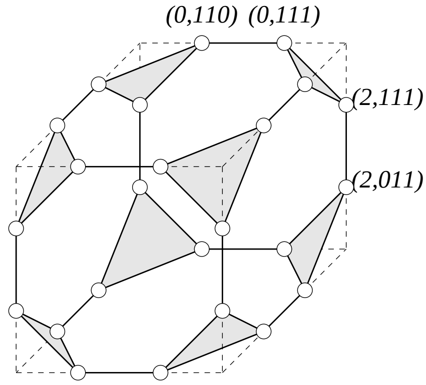

Figure 5. Cube connected cycle of a 3**-dimensional hypercube**

angle free 3-regular graph on eight vertices of diameter 2. Up to isomorphism, there is only one such graph, namely $G(\mathcal{Z}_8, \{\pm 1, 4\})$ (see Figure 4), which is sometimes called the *Wagner graph*. In particular, $\tilde{H}_2^3$ is a Cayley graph as well.

3.3. Constant Degree Cayley DHTs

Cycloid [32] Cycloid is a constant degree DHT emulating a cube connected cycle as its static DHT topology. In Cycloid, each node is specified by a pair of cyclic and cube indices. In order to dynamically maintain connectivity of the DHT topology, the dynamic DHT algorithm of Cycloid forces each node to keep a routing table consisting of 7 entries. Among them, several entries (so-called leaf sets) only make sense for the dynamic DHT algorithm to deal with network connectivity in sparsely populated identifier spaces.

A d*-dimensional cube connected cycle graph* is obtained from a d-dimensional cube by replacing each vertex with a cycle of d nodes. It contains $d \cdot 2^d$ nodes of degree d each. Each node is represented by a pair of indices (k, v), where $k \in \mathcal{Z}_d$ is a cyclic index and $v \in \mathcal{Z}_2^d$ is a cube index. A cube connected cycle graph can be viewed as a specific case of Cayley Graph Connected Cycles (CGCC) [26], defined as:

Definition 3 *Let* $\Gamma = (V, \cdot)$ *be a group and* $S := \{s_1, \ldots, s_d\} \subseteq V - \{1\}$ *closed under inversion with* $d \geq 3$. *The Cayley graph connected cycles network* $CGCC(\Gamma, S) = (V, E)$ *is the graph defined by* $V' := \mathcal{Z}_d \times V$ *and* $E' := \{(i,x)(j,y) : (x = y \wedge i = j \pm 1)$ *or* $(i = j \wedge x = s_i \cdot y)\}$.

$CGCC(\Gamma, S)$ can be obtained by replacing each vertex of the Cayley graph $G(\Gamma, S)$ with a cycle of length d and replacing each edge of $G(\Gamma, S)$ with an edge connecting two members of a cycle in a certain way. The edges $(i,x)(j,y)$ with $i = j$ form *cycle connections*, the others form *cayley graph connections*. [17] proves that these graphs are Cayley graphs.

Following the definition of CGCC, the n-dimensional cube connected cycle is a graph built from a n-cube replacing each of its nodes with a cycle of length n. In Figure 5, the 3-dimensional cube-connected cycle is displayed. The 3-cycles are shaded, and the dotted lines help to recognize the 3-dimensional cube.

Viceroy [22] Viceroy is a constant degree DHT emulating an approximate butterfly graph as its static DHT topology. The dynamic DHT algorithm of Viceroy is rather involved. It works based on a rough estimate of the network size and forces each node to keep a routing table containing 5 to 7 entries [22]. Similar to Cycloids, part of the entries only make sense for the dynamic DHT algorithm to deal with a sparsely populated identifier space (i.e. ring links [22]). For Viceroy we can only guarantee with high probability that the constructed DHT topology is a butterfly graph.

The d*-dimensional binary wrapped directed butterfly* $\vec{B}_2^d$ is a graph with vertices $V = V(\vec{B}_2^d) = \mathcal{Z}_{d-1} \times \mathcal{Z}_2^d$ such that there is an edge from $a = (i, v_1 \cdot v_d) \in V$ to $b = (j, w_1 \cdot w_d) \in V$ if and only if $i \in \{0, \ldots, d-1\}$, $j = i+1$ and $v_k = w_k$ for all $k \in \{0, \ldots, d-1\} - \{i\}$.One can think of i, j as of the *levels* of a and b, respectively, and some level i vertex (i, v) has precisely two neighbors $(i+1, v)$ and $(i+1, v')$, where v' is obtained from v by adding 1 $(\bmod 2)$in the ith component of v. The d*-dimensional binary wrapped butterfly* B_2^d is the underlying graph of the digraph $\vec{B}_2^d$, where there is a (single) edge ab whenever there is an edge (a, b) or an edge (b, a) in $\vec{B}_2^d$. As we can see, B_2^d is 4-regular for $d \geq 3$.

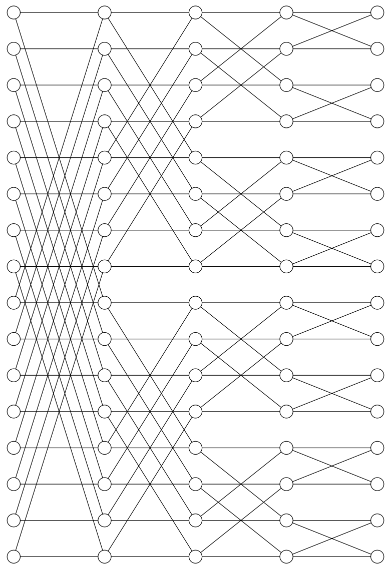

Figure 6. The wrapped butterfly B_4

Figure 6 shows B_2^4, where we have to identify the rightmost level with the leftmost one. In the *unwrapped version* of the butterfly, we would *not* identify the two border levels. This can be modelled formally by taking V to be $\mathcal{Z}_d \times \mathcal{Z}_2^d$ instead of $\mathcal{Z}_{d-1} \times \mathcal{Z}_2^d$ and then following literally the definitions of the preceding paragraph (so $j = i + 1$ will be taken $\mod d + 1$, not $\mod d$).

The advantage of taking the wrapped rather than the unwrapped version of the butterfly is that B_2^d is a Cayley graph, whereas unwrapped ones are not even regular, since for $d \geq 3$ the vertices on the border levels have degree 2 and the others have degree 4. We can represent B_2^d as a Cayley graph of the *wreath product* of the groups $(\mathcal{Z}_d, +)$ and $(\mathcal{Z}_2^d, +)$. For (i, v) and (j, w) in V, we define $(i, v) \bullet (j, w)$ $:= (i + j, (v_0 + w_{-\ell}, v_1 + w_{-\ell+1}, \ldots, v_{d-1} + w_{-\ell+d-1}))$. Note that $i + j$ and the indices at the components of v and w are to be taken $\mod d$. This operation $\bullet : V \times V \longrightarrow V$ constitutes a *group* $\Gamma = (V, \bullet)$, with neutral element $(0, 0)$. By taking $S = \{(1, 0), (1, (1, 0, \ldots, 00 \cdots 0))\} \subseteq V$ we obtain the representation $B_2^d = G(\Gamma, S)$ of B_2^d as a Cayley graph. For the mathematical details we refer to [17].

CAN [27] CAN is an (adjustable) constant degree DHT using a virtual d-dimensional Cartesian coordinate space to store $(key, value)$–pairs. The topology under this Cartesian coordinate space is a d-dimensional torus.

Let $T_{m,n} := C_m \times C_n$ of length m and n be the Cartesian product of two cycles C_m, C_n. The componentwise addition $+$ establishes a group $\Gamma(\mathcal{Z}_m \times \mathcal{Z}_n, +)$ on its vertices, and clearly $T_{m,n} = G(\Gamma, \{(0, \pm 1), (\pm 1, 0)\})$. Hence the torus is a Cayley graph as well. One could consider such a toroidal graph as a rectangular grid, where the points on opposite borders are identified. We can extend this definition easily to higher dimensions: Let $n_1, \ldots, n_d$ be numbers ≥ 2. Componentwise addition $+$ of elements in $V := \mathcal{Z}_{n_1} \times \cdots \times \mathcal{Z}_{n_d}$ establishes a group $\Gamma(V, +)$, and by taking S to be the set $\{(z_1, \ldots, z_d) \in P$: there is an $i \in \{1, \ldots, d\}$ such that $z_i = \pm 1$ and $z_j = 0$ for all $j \neq i$ in $\{1, \ldots, d\}\}$ and we obtain a *d-dimensional torus* $T_{n_1,\ldots,n_d} = G(\Gamma, S)$. Explicitly, $T_{n_1,\ldots,n_d}$ is a graph on the vertex set V, where $(v_1, \ldots, v_d)$ and $(w_1, \ldots, w_d)$ are adjacent if and only if they differ in exactly one component and the difference in this component is either $+1$ or -1. As the presence of i's with $n_i = 2$ stretches formal arguments slightly (for example, when considering degrees), some authors force $n_i \geq 3$ for all $i \in \{1, \ldots, d\}$. (So a torus of the latter type had always degree $2 \cdot k$, whereas a torus in our definition had degree k plus the number of n_i with $n_i \geq 3$.) They lose then, however, the possibility to consider the d-dimensional hypercube as a special torus, namely as $T_{\underbrace{2, \ldots, 2}_{d \text{ times}}}$.

3.4. Non-Cayley DHTs

There are only a few DHT designs, which use asymmetric static DHT topologies, and thus cannot be classified as Cayley DHTs. However, even for these topologies, our Cayley DHTs analytical framework is significant.

Pastry/Tapestry [29] [36] The static DHT topology emulated by Pastry/ Tapestry are Plaxton trees. However, when taking the dynamic DHT algorithms of Pastry/Tapestry into account, we find that the static DHT topology of Pastry/Tapestry behaves quite similar to a randomized approximation of hypercubes. As analyzed in [16], in the Pastry/Tapestry identifier space, each node on the Plaxton tree differs from its ith neighbor on only the ith bit, dynamic routing is done by correcting a single bit at a time in the left-to-right order. This turns out to be the same routing mechanism adopted by DHTs using hypercubes as static DHT topologies, even though hypercube based DHTs allow bits to be corrected in any order. In terms of our Cayley DHT analytical framework, we can highlight several disadvantages of this non-Cayley graph based design notwithstanding any other advantages it might have(see Section 5).

Koorde/D2B/Distance Halving [12] [13] [24] The static DHT topologies emulated by Koorde/D2B/Distance Halving are de Bruijn graphs[2]. These graphs share with hypercubes most of the good network properties such as the same number of vertices and the same diameter, but ex-

2 Distance Halving emulates a generalized k-ary de Bruijn graph

hibit constant graph degrees. From the mathematical point of view, De Bruijn graphs again are closely related to butterfly/Cayley graphs [3].We need several ingredients to demonstrate this interconnection.

First, for a set V and a set $P \subseteq S_V$ of permutations of V, we define the *(group) action digraph* $A(V, P)$ of V and P by $V(A(V, P)) = V$ and $E(P) := \{(x, f(x)) : x \in V, f \in P\}$. (If we label the edge (x, y) with all f such that $y = f(x)$ we may reconstruct V and P from its labelled action digraph.)

Second, we assign a Cayley digraph to V and P in a natural way. Although P itself is not a group in general, we can take $\Gamma :=< P >_{S_V}$ to be the group *generated by* P *in* S_V, i.e. the smallest subgroup of S_V which contains P. Let $\vec{G}(V, P) := \vec{G}(\Gamma, P)$ be the *Cayley digraph* associated to V and P, where we have an edge (x, y) whenever $x \circ y^{-1} \in P$.

The *d-dimensional binary De Bruijn Digraph* $\vec{D}_2^d$ is defined by $V = V(\vec{D}_2^d) := \mathcal{Z}_2^d$, where there is an edge from $v = (v_1, \ldots, v_d) \in V$ to $w \in V$ if w is one of $v' := (v_2, \ldots, v_d, v_1)$ and $v'' := (v_2, \ldots, v_d, v_1 + 1)$. This *$d$-dimensional binary De Bruijn Graph* D_2^d is obtained by taking the underlying undirected multigraph of $\vec{D}_2^d$.

Clearly, $\vec{D}_2^d$ is an action digraph of $V = \mathcal{Z}_2^d$ and the permutations f', f'' of V defined by $f'(v) = v'$ and $f''(v) := v''$ ($v \in V$). So the interconnection of butterfly graphs and de Bruijn graphs can be stated as follows.

Theorem 1 *[3] For $d \geq 3$, $\vec{B}_2^d$ is isomorphic to the Cayley digraph associated to $\mathcal{Z}_2^d$ and $P = \{f', f''\}$, where f', f'' are defined as in the two preceeding paragraphs.*

Figure 7 displays $\vec{D}_2^3$, where the edges caused by f'' are emphasized. One immediately sees that f' is a permutation with two fixed points and two 3-cycles, and that f'' consists of a 2-cycle and a 6-cycle. As one can check f' and f'' generate a 24-element subgroup Γ of the 40.320-element group of permutations of $\mathcal{Z}_2^3$, and the associated Cayley graph will be $\vec{B}_2^3$. Again, for generalizations and the mathematical details we refer to [17].

This relationship between de Bruijn graphs and Cayley graphs clarifies why de Bruijn graphs are an important type of static DHT topologies even if they are not Cayley DHTs. Still, comparing them to Cayley DHTs, we observe disadvantages of de Bruijn based static DHT topologies such as communication congestion and a higher dependency on the dynamic DHT algorithm employed to ensure fault tolerance[12]. Such analysis strongly suggests the possibility and feasibility to "Cayley-lize" non-Cayley graph based static DHT topologies (i.e. de Bruijn graphs) to build high performance Cayley DHTs (see Section 5).

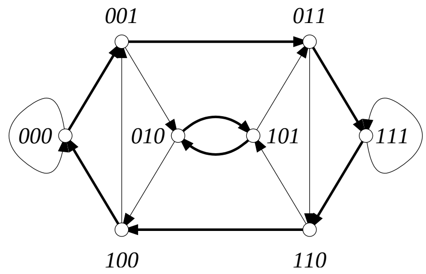

Figure 7. The de Bruijn graph $\vec{D}_2^3$

4. Cayley Graph Properties and DHTs

Cayley graphs have a specific set of properties which can be closely associated with important DHT system features. The following paragraphs include a discussion of these Cayley DHT properties and features thus providing a good insight into Cayley DHT design in general.

4.1. Symmetry and Load Balancing

The most useful properties of Cayley graphs are *symmetry properties*. Recall that an *automorphism* of some graph G is a bijection $\varphi : V(G) \longrightarrow V(G)$ with $\varphi(x)\varphi(y) \in E(G)$ if and only if $xy \in E(G)$.The set of all these mappings form the *automorphism group* $(Aut(G), \circ)$ of G.

Definition 4 *A graph G is called* vertex symmetric *or* vertex transitive*if its automorphism group acts transitively on its vertices, i. e. for arbitrary $x, y \in V(G)$ there exists an automorphism φ of G such that $\varphi(x) = y$.*

Now if x, y are vertices of some Cayley graph $G((V, \cdot), S)$ then it is easy to see that the translation $\varphi : V \longrightarrow V$, $\varphi(z) := z \cdot (x^{-1} \cdot y)$, is an automorphism which maps x to y. In fact, *every* group automorphism produces an automorphism of $G(V, \cdot), S)$. This proves the following classical observation.

Theorem 2 *Every Cayley graph is vertex transitive.*

This property results in an important feature of Cayley graphs — routing between two arbitrary vertices can be reduced to the routing from an arbitrary vertex to a special vertex [1]. This feature is significant for Cayley DHTs because it enables an algebraic design approach for the routing algorithm.

Suppose that $\Gamma = (V, \circ)$ is a group of permutations, let $S \subseteq V - \{id_V\}$ be closed under inversion and consider the Cayley graph $G = G(\Gamma, S)$. For a path $P = x_0, \ldots, x_\ell$

from x_0 to x_ℓ set $s_i := x_{i-1}x_i^{-1}$ for $i \in \{1, \ldots, \ell\}$. Then the sequence $s_1, \ldots, s_\ell$ in S *represents* the path P, and it also represents the path from x_0x^{-1} to id_V. Consequently the routing problem G is equivalent to a certain sorting problem [1]. Taking V to be the set of all permutations of some set and $S \subseteq V$ to be the set of all transpositions will produce the so called *bubble sort graph* (see [17]).

We can leverage this property to implement optimized routing algorithms for Cayley DHTs through purely algebraic approaches supported by sets of mature algebraic methods. Furthermore, vertex transitivity provides a unified method to evaluate communication load on DHT nodes. In Cayley DHTs, the communication load is uniformly distributed on all vertices without any point of congestion. In contrast, non-Cayley DHTs exhibit congestion points. As communication load balancing is one of the principal design concerns of DHTs, this points out major drawback of non-Cayley DHTs. While for constant degree non-Cayley DHTs such as Koorde/D2B/Distance Halving, which do not have this property, the drawback is acceptable as a tradeoff against better DHT maintainability[3], for non–constant degree non-Cayley DHTs such as Pastry/Tapestry, such a drawback suggests a redesign of their static DHT topology.

In addition to vertex transitivity, Cayley graphs may also have another important property, edge transitivity.

Definition 5 *A graph G is* edge symmetric *or* edge transitive *if its automorphism group acts transitively on its edges, i. e. for edges wx, yz there exists an automorphism φ such that $\varphi(w)\varphi(x) = yz$*[4].

Clearly, every edge transitive graph without isolated vertices is vertex transitive, but the converse is not true. For example, the Wagner graph $G(\mathcal{Z}_8, \{\pm 1, 4\})$ in Figure 4 is not edge transitive, as there are two structurally distinguishable types of edges: the four diagonals, which are contained in two cycles of length 4, and the others edges, which are not.

There is, however, a sufficient condition to S which forces a Cayley graph to be edge transitive. It is taken from [1], but fails to be necessary as it was claimed to be there and as it has been observed in [17]. Recall that a *homomorphism* of a group $\Gamma = (V, \cdot)$ is a mapping $\varphi : V \longrightarrow V$ satisfying $\varphi(x \cdot y^{-1}) = \varphi(x) \cdot \varphi(y)^{-1}$ for all $x, y \in V$.

Theorem 3 *If $G = G(\Gamma = (V, \cdot), S)$ is a Caley graph and for all $s, t \in S$ there exists a homomorphism φ of Γ such that $\varphi(S) = S$ and $\varphi(s) = t$ then G is edge transitive.*

Among Cayley DHTs, HyperCup (hypercubes), CAN(d-dimensional torus), and Viceroy (butterfly) are edge transitive, whereas Chord (ring graphs) and Cycloid (cube connected cycles) are not. All non–Cayley DHTs are not edge transitive.

Edge transitivity results in a unified method to evaluate the communication load on DHT edges. In edge transitive Cayley DHTs communication load is uniformly distributed on all edges without any point of congestion. For constant degree Cayley DHTs such as Cycloid, the loss of edge transitivity can be seen as a reasonable tradeoff against the constant degree property. For non-constant degree Cayley DHTs such as Chord, the loss of edge transitivity is disadvantageous, and has to be compensated for in the design of the dynamic DHT routing algorithm.

4.2. Hierarchy, Fault Tolerance, and Proximity

Recall that $< S >_\Gamma$ is the subgroup of $\Gamma = (V, \cdot)$ generated by $S \subseteq V$ i. e. the smallest subgroup of Γ which contains S.

Definition 6 *Let $\Gamma = (V, S)$ be a group and $S \subseteq V(G) - \{1\}$ such that $S^{-1} = S$. The Cayley graph $G(\Gamma, S)$ is* strongly hierarchical *if S is a* minimal generator *for G, i. e. if $< S >_G = G$ but $< S - \{s, s^{-1}\} >_G$ is a proper subgroup of G for every $s \in S$.*

For further generalizations of this concept, we refer to [17].

Among Cayley DHTs, HyperCup (hypercubes) and Chord (ring graphs) can be proven to be hierarchical [17]. In Figure 8 we show a 4-dimensional hypercube which can be decomposed into two 3-dimensional hypercubes. A Chord ring graph with n-bit key space can be viewed as two Chord ring graphs with $(n-1)$-bit key space connected together by links between adjacent vertices.

Hierarchical Cayley graphs "often allow inductive proofs by decomposing (stripping) the graph into smaller members of the same family, thus are scalable in the sense that they recursively consist of copies of smaller Cayley graphs of the same variety" [17]. For the DHT design, hierarchy can strongly affect the node organization and aggregation, which is closely associated with two important DHT system features: fault tolerance (i.e. network resilience) and proximity (i.e. network latency). Most hierarchical Cayley DHTs, except for a very particular family, are optimally fault tolerant as their connectivity is equal to their degree [2]. Furthermore, in hierarchical Cayley DHTs, there might

3 Constant degree is associated with maintainability of DHTs, it implies that joining/leaving/failing nodes only influence a constant number of other nodes

4 By notational convention for undirected graphs, wx denotes the set $\{w, x\}$, so $\varphi(w)\varphi(x) = yz$ does not mean that $(\varphi(w), \varphi(x)) = (y, z)$ but that $\{\varphi(w), \varphi(x)\} = \{y, z\}$

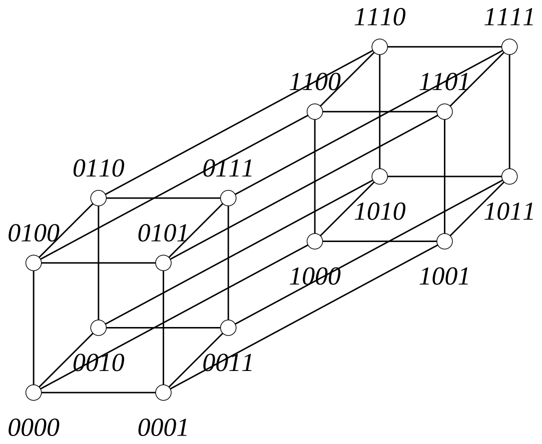

Figure 8. 4**-dimensional hypercube**

exist easy solutions to dynamically organize nodes (or node aggregations) to ensure proximity of DHTs. It is worth noting that so far the use of hierarchical Cayley graphs have not yet been intensively investigated for DHT design. Some promising hierarchical Cayley graphs not yet used are star graphs and pancake networks [5], which have smaller network diameter than hypercubes of the same degree.

4.3. Connectivity and Fault Tolerance

Definition 7 *A graph G is* disconnected *if it contains two vertices x, y such that there is no x, y-path in G. The* connectivity $\kappa(G)$ *of a finite (nonempty) graph is the minimum cardinality of a set X of vertices such that $G - X$ is disconnected or has less than two vertices.*

Similarly, the *edge–connectivity* $\lambda(G)$ of a graph G can be defined by varying over all subsets X of edges of G.

A graph is called d*-regular* if every vertex has degree d. For example, every vertex transitive graph is regular. Clearly, d is an upper bound for the connectivity of a d-regular graph. Let us call a d-regular graph G *optimally fault tolerant* if its connectivity equals d.For example, complete graphs are optimally fault tolerant, and so are hypercubes and hyperskewbes (as one can prove by induction on the dimension, using the recursive characterizations). For edge transitive graphs, we have even the following.

Theorem 4 *[21, 20, 35] (cf. [7]) Every connected edge transitive graph is optimally fault tolerant.*

In general, connected Cayley graphs are not optimally fault tolerant; the smallest example showing this is the 5-regular circulant graph $G := G(\mathcal{Z}_8, \{\pm 1, \pm 3, 4\})$, as $G - \{0, 2, 4, 6\}$ is disconnected. However, the following theorem on connected vertex transitive graphs shows that connectivity and degree can't differ too much.

Theorem 5 *[21, 20, 35] (cf. [7]) The connectivity of a connected vertex transitive d-regular graph is at most d and at least $\frac{2}{3}(d + 1)$.*

In particular, for $d \in \{2, 3, 4\}$, every d-regular connected vertex transitive graph is d-connected and, thus, optimally fault tolerant. For $d = 5$, this statement is wrong even for Cayley graphs as seen in the previous example, but for $d = 6$ it's true "again": Every 6-regular vertex transitive graph is 6-connected. This follows easily from the main result in [21] which implies that *every triangle free connected vertex transitive graph is optimally fault tolerant*. Even more generally, every vertex transitive graph without four pairwise adjacent vertices is optimally fault tolerant [20]. This gives alternative proofs of the optimal fault tolerance of the hypercubes and hyperskewbes and of d-dimensional tori $T_{n_1,\ldots,n_d}$ with $n_i \geq 4$ for all $i \in \{1, \ldots, d\}$.

The graph $G(\mathcal{Z}_8, \{\pm 1, \pm 3, 4\})$ indicates that it might be already a problem to characterize the optimally fault tolerant circulants. It has been solved by Boesch and Tindell [6] as follows. Suppose that G is an arbitrary d-regular circulant $G = G(\mathcal{Z}_n, \{\pm a_1, \ldots, \pm a_k\})$ with $0 < a_1 < \ldots < a_k < (p + 1)/2$. Then G is optimally fault tolerant if and only if for every proper divisor m of n, the number of distinct positive residues $\mod m$ of the numbers $a_1, \ldots, a_k$ is at least the minimum of $m - 1$ and dm/p.

Edge connectivity is less interesting from the point of view of optimal fault tolerance, as every d-regular vertex transitive graph has edge connectivity equal to d [20, 35] (cf. [7]). Hierarchical Cayley graphs as in Definition 6 and as in [17] or [2] are also known to be optimally fault tolerant unless they belong to a particular family of graphs whose d-regular members still have connectivity $d - 1$. For the technical details, we refer the reader to [17] or [2].

As an alternative, we propose a slightly different hierarchy concept as well. For a set $\mathcal{H}$ of finite graphs let $\mathcal{G}(\mathcal{H})$ be the smallest superset of $\mathcal{H}$ which is closed under taking the cartesian product. For example, every graph in $\mathcal{G}(\{K_1\})$ is isomorphic to some hypercube, and every hypercube is isomorphic to some graph in $\mathcal{G}(\{K_1\})$. In this sense, $\mathcal{G}(\{K_1\})$ "is" the class of hypercubes, and if $\mathcal{C}$ consists of cycles of every length plus K_2 then $\mathcal{G}(C)$ "is" the class of all tori. Certain properties of the members of $\mathcal{H}$ are inherited to those of $\mathcal{G}(\mathcal{H})$, as they are stable under taking products. If $\mathcal{H}$ is a set of Cayley graphs then so is $\mathcal{G}(\mathcal{H})$. If we take $\mathcal{H}$ as an arbitrary set of triangle free vertex transitive graphs, then every graph in $\mathcal{G}(\mathcal{H})$ will also have these properties and will be optimally fault tolerant by the results of [21].

Among Cayley DHTs, HyperCup (hierarchical Cayley graphs), Chord (hierarchical Cayley graphs), Cycloid (3-regular Cayley graphs) and Viceroy (4-regular Cayley graphs) are optimally fault tolerant based on their static DHT topology perspective. CAN (d-regular Cayley graphs) can also be proven optimally fault tolerant based on its dynamic DHT algorithm features such as multiple realities and multiple dimensions [27]. For non-DHTs it is much harder to prove optimal fault tolerance. However, as fault tolerance is one of the principal design concerns of DHTs, most non–Cayley DHTs have included various techniques in their dynamic DHT algorithms to pursue possibly higher fault tolerance, although optimality cannot guaranteed. One possible such technique is to force each node to maintain a successor list in dynamic DHT algorithms, like in Koorde [12].

For DHTs whose static DHT topologies are optimal fault tolerant, it is much easier to also ensure this in the dynamic algorithm design for sparsely populated DHT identifier spaces, or frequently leaving / failing nodes. Possible techniques include the successor list in Chord [33] or the state-machine approach based replication in Viceroy [22].

4.4. Hamiltonicity and Cyclic Routing

Definition 8 *A path or cycle which visits every vertex in a graph G exactly once is called a* hamiltonian path *or* hamiltonian cycle*, respectively.*

Hamiltonicity has been received much attention of theorists in this context, as the following conjecture is still open.

Conjecture 1 *Every 2-connected Cayley graph has a hamiltonian cycle.*

The question of *hamiltonian cycles and paths* in Cayley graphs has a long history [9]. All aforementioned static DHT topologies of Cayley DHTs such as hypercubes, ring graphs, butterfly, cube-connected cycles, and d-dimensional tori have been proven to be hamiltonian. In addition, the non–Cayley DHT de Bruijn topology has also been proven hamiltonian [28].

Hamiltonicity is important for DHT design because it enables DHTs to embed a ring structure so as to implement ring based routing in dynamic DHT algorithms. Ring based routing, characterized by the particular organization of the DHT identifier space and ensuring the DHT fault tolerance in a dynamic P2P environment by means of maintaining successor/predecessor relationships between nodes, is used by almost all DHT proposals. Gummadi et al. [16] also observe that the ring structure "allows the greatest flexibility and hence achieves the best resilience and proximity performance of DHTs". Although in terms of our analytical framework, we do not fully agree with Gummadi et al. on the conclusion that ring graphs are the best static DHT topologies[5], we agree that at least one hamiltonian path/cycle should exist in static DHT topologies in order to ease the dynamic DHT algorithm design. From the static DHT topology perspective, all aforementioned DHTs are hamiltonian except for Pastry/Tapestry (Plaxton trees). Even Pastry/Tapestry, though, maintain a ring structure through their dynamic DHT algorithm [29, 36].

5. Discussion

So far as we can see, some desirable DHT system features are inconsistent with each other, which means that tradeoffs must be considered when deciding on a static DHT topology.

As a general conclusion, we have shown that Cayley DHTs have clear advantages over non–Cayley DHT designs. Static topologies of non-Cayley DHTs cannot naturally ensure many desirable DHT features such as communication load balancing, optimal fault tolerance, etc., which leads to additional overhead and complexity in their dynamic DHT algorithms to achieve these features. Cayley DHTs can naturally guarantee those, and result in simpler dynamic DHT algorithm design.

Cayley DHTs cover both non-constant degree DHTs and constant degree ones, so in each case we can start from Cayley graphs as underlying topology for DHT design. The few good non-Cayley graphs such as de Bruijn are closely related to Cayley graphs, and can also be "Cayley-lized" so as to be taken as better static DHT topologies for future DHT design, as already investigated for VLSI layout design ([14] uses "Cayley-lized" de Bruijn graphs).

As for constant versus non-constant degree Cayley DHTs, constant-degree ones have the main advantage that their "maintainability" (regarding leaving / failing nodes) is independent of the size of the network. In a dynamic P2P environment, maintainability of nodes might be preferrable to other desirable DHT system features such as communication load balancing and fault tolerance, since the loss of other DHT system features can often be compensated through some additional techniques in the dynamic DHT algorithm design, whereas maintainability is almost uniquely determined by the static DHT topology.

When designing constant degree Cayley DHTs, we view

5 Our conclusion is much more general than theirs, as the ring graph can be viewed as a variant of k-ary d-cubes.

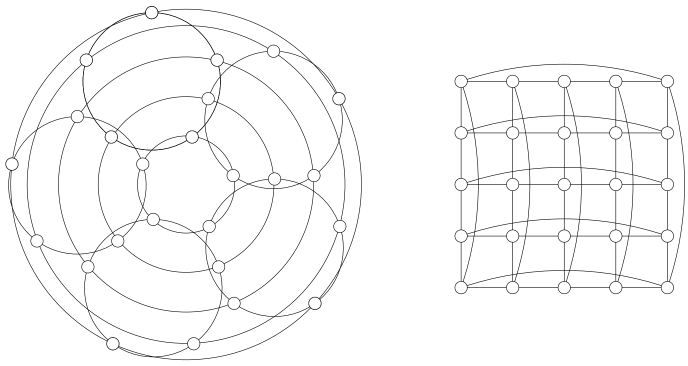

Figure 9. Two embeddings of $C_5 \times C_5$

cube connected cycles as an especially promising family of static DHT topologies in terms of our analytical framework, taking into account the simplicity they enable for dynamic DHT algorithm design in comparison to for example butterfly graphs. This conclusion can be extended to a generalized type of constant degree Cayley graphs: Cayley Graph Connected Cycles (CGCC), as we have discussed in Section 3.3. We therefore expect that different variants of CGCC will heavily influence the design mainstream for future constant degree Cayley DHTs.

Looking at non-constant degree Cayley DHTs, the most promising family are hypercubes, as they achieve all desirable DHT system features except for the constant degree property. This conclusion can be extended to k-ary n-cube, which can be regarded as a generalization of the d-dimensional hypercube by taking $k = 2$. Formally, the k-ary d-cubes can be defined as in [10]:

Definition 9 *Consider the group* $\Gamma := (\mathcal{Z}_k^d, +)$*, where* $V := \mathcal{Z}_k^d$ *denotes the set of all words of length* d *over the alphabet* $\mathcal{Z}_k$ *and where* $+$ *is the componentwise addition* $\bmod k$*. Let* S *be the set of all* $(k-1) \cdot d$ *words in* V *which have exactly one entry* ± 1 *and all others entries being* 0*. The graph* $H_k^d := G((\mathcal{Z}_k^d, +), S)$ *is the* k-ary d-cube.

By definition, k-ary n-cubes are Cayley graphs. They can be defined recursively as well: Denoting by C_k the cycle of length k, we see that H_k^1 is isomorphic to C_k and H_k^d is isomorphic to $H_k^{d-1} \times C_k$ for $d > 1$, which is in turn isomorphic to $\underbrace{C_k \times \cdots \times C_k}_{d \text{ times}}$.

Figure 9 shows two embeddings of the 5-ary 2-cube, one of which reflects its cyclic symmetries, the other one illustrating that it is a product graph.

Most current Cayley DHTs such as HyperCup, CAN, and Chord use static DHT topologies that are either k-ary d-cubes or isomorphic to k-ary d-cubes such as ring graphs, tori, direct or undirected d-cubes [10]. Even for constant degree Cayley DHTs or non-Cayley DHTs, the static DHT topologies of Cycloid (cube-connected cycles) and Pastry/Tapestry (Plaxton trees) are closely associated with k-ary d-cubes. As we have mentioned, Plaxton trees can be viewed as approximate hypercubes, whereas cube-connected cycles can be viewed as a variant of hypercubes.

6. Related Work

Although the DHT research is still at its infancy, some researchers have already started to investigate the relationships of the various DHT designs. However, there have been relatively few attempts to investigate commonalities and analyze tradeoffs between different DHT design families.

Gummadi et al. [16] investigate some commonly used static DHT topologies including hypercubes, ring graphs, Plaxton trees, and butterflies, and explore how these static DHT topologies affect static resilience (fault tolerance) and proximity routing (total latency of the DHT routing path, and the ability to achieve local convergence). They investigate these issues by analyzing the flexibility of different DHTs, i.e. the algorithmic freedom left after the static topologies has been chosen. Such freedom is exercised in the selection of neighbors and routes.

In comparison to our study, the analysis by Gummadi et al. is not based on a unified analytical framework, so, even though individual static DHT topologies are investigated thoroughly, no common properties of these DHT topologies are concluded. Also their analysis is bound tightly to dynamic DHT algorithms and thus lacks generality for static DHT topology analysis. As a result, Gummadi's conclusion that ring graphs allow the greatest flexibility and hence lead to the best resilience and proximity of DHTs is only part of the truth.

Manku's [23] analysis starts from static DHT topologies, but then heavily involves dynamic DHT algorithms. Manku classifies some commonly used DHT topologies into two categories: deterministic DHT topologies such as hypercubes, torus, and de Bruijn, and randomized DHT topologies such as butterfly. In deterministic DHT topologies, overlay connections of DHTs are a function of the current set of node IDs, whereas in randomized DHT topologies there is conceptually a large set of possible networks for a given set of node IDs, and, at run-time, a specific network is chosen depending upon the random choices made by all participants [23]. While this classification is certainly of value, it cannot serve as an analytical framework for comparing static DHT topologies. Comparatively, our results provide more insight into DHTs with reduced complexity[6].

Loguinov et al. [18] examine DHTs by analyzing graph-theoretic properties of static DHT topologies. They first propose a new DHT system ODRI (Optimal Diameter Routing Infrastructure), which uses generalized de Bruijn graphs as the static DHT topology, and then make a comparative study of several DHT system features such as routing efficiency, graph expansion and clustering properties from the static DHT topology perspective, taking Chord and CAN as reference systems. As the focus of their research is to propose a new DHT system, they do not conduct a thorough investigation of all static DHT topologies. In terms of our framework, their proposal falls into the category of constant degree non-Cayley DHTs, sharing with Koorde/D2B/Distance Halving all our analysis regarding this category.

Besides, Datar [11] provides an in-depth investigation to butterfly graphs and further proposes a new DHT system called multi-hypercube which uses so-called multi-butterfly as the static DHT topology. Castro et al. [8] make an in-depth comparative study of Pastry, taking Chord and CAN as reference systems. In terms of our framework, these systems can be analyzed just like other DHT proposals.

7. Conclusions and Future Work

Cayley graphs are one of the most important group-theoretic models for the design of parallel interconnection networks, and have been studied for many years. Since any parallel interconnection network can potentially be morphed into a DHT routing protocol [23], associating Cayley graphs with DHTs enables us to directly leverage the research results for interconnection networks for the design of high performance DHTs without the need of starting from scratch. Cayley graphs explicitly enable an algebraic design approach for DHTs, which allows us to start with an arbitrary finite group and construct symmetric DHTs using that group as the algebraic model. An immediate advantage of this approach is the conciseness with which DHTs can be specified by providing only the appropriate group plus a set of generators. This algebraic design approach will also enable us to build new types of structured P2P networks in which data and nodes do not necessarily need to be hashed in order to build content delivery overlay networks, as discussed in [31, 25] for hypercube topologies. Such non-hashed, structured P2P networks allow us to apply semantic web and database technologies for data organization and query processing and implement powerful and expressive distributed information infrastructures which are not implemented easily based on pure DHT designs.

Our analytical framework and its notion of Cayley DHTs provides a unified view of DHTs, which gives us excellent insight for designing and comparing DHT designs. Identifying a DHT design as Cayley DHTs immediately allows us to infer all generic properties for this design, and, through the correspondence of Cayley graph properties to DHT system features, allows us to directly infer the generic DHT features implemented by this design. Furthermore, we can investigate the various tradeoffs between different DHT designs features and use them to guide the design of future DHTs.

Casting and understanding static DHT topologies in a common framework is but the first important step towards principled DHT design. In order to cover all features of a particular design, we also have to explore the general design of dynamic DHT algorithms which can in principle be used to emulate any Cayley graph based static DHT topologies. Such dynamic DHT algorithms need not necessarily be bound to any individual Cayley graph, instead they could be universally applicable to any Cayley graphs, leveraging algebraic design methods in order to build arbitrary Cayley DHTs. Some of these methods and design issues are currently investigated in more detail in our research group.

6 e.g., as we have cleanly separated static DHT topologies from dynamic DHT algorithms, we do not need to analyze the same topology in both the deterministic and randomized category

References

[1] S. B. Akers and B. Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Trans. Comput.*, 38(4):555–566, 1989.

[2] B. Alspach. Cayley graphs with optimal fault tolerance. *IEEE Trans. Comput.*, 41(10):1337–1339, 1992.

[3] F. Annexstein, M. Baumslag, and A. L. Rosenberg. Group action graphs and parallel architectures. *SIAM J. Comput.*, 19(3):544–569, 1990.

[4] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Looking up data in p2p systems. *Communications of the ACM: Special Issue: Technical and social components of Peer-to-Peer computing*, 46(2), 2003.

[5] P. Berthom, A. Ferreira, and S. Perennes. Optimal information dissemination in star and pancake networks. *IEEE Tran. on Parallel and Distrubuted Systems*, 7(12), 1996.

[6] F. Boesch and R. Tindell. Circulants and their connectivities. *J. Graph Theory*, 8(4):487–499, 1984.

[7] F. Boesch and R. Tindell. Connectivity and symmetry in graphs. In *Graphs and applications (Boulder, Colo., 1982)*, Wiley-Intersci. Publ., pages 53–67. Wiley, New York, 1985.

[8] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Exploiting network proximity in distributed hash tables. In *International Workshop on Future Directions in Distributed Computing (FuDiCo)*, Bertinoro, Italy, June 2002.

[9] S. J. Curran and J. A. Gallian. Hamiltonian cycles and paths in Cayley graphs and digraphs—a survey. *Discrete Math.*, 156(1-3):1–18, 1996.

[10] W. J. Dally. *A VLSI Architecture for Concurrent Data Structures*. Hingham, MA: Kluwer, 1987.

[11] M. Datar. Butterflies and peer-to-peer networks. In *10th Annual European Symposium*, Lecture Notes in Computer Science, Rome, Italy, Sept. 2002. Springer.

[12] K. M. F. and R. Karger. Koorde: A simple degree optimal distributed hash table. In *the 2nd International workshop on P2P Systems(IPIPS03)*, Berkeley, CA, USA, Feb. 2003.

[13] P. Fraigniaud and P. Gauron. The content-addressable network d2b. Technical report, CNRS Universie de Paris Sud, Jan. 2003.

[14] E. Ganesan and D. K. Pradhan. The hyper-de bruijn networks: Scalable versatile architecture. *IEEE Transaction on parallel and Distributed System*, 4(9), 1993.

[15] P. Ganesan and G. S. Manku. Optimal routing in chord. In *ACM SIAM Symposium on Distributed Algorithms (SODA)*, New Orleans, USA, Jan. 2004.

[16] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of dht routing geometry on resilience and proximity. In *ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM)*, Karlsruhe, Germany, Aug. 2003.

[17] M. C. Heydemann and B. Ducourthial. Cayley graphs and interconnection networks. *Graph Symmetry, Algebraic Methods and Applications,"NATO ASI C"*, 497:167–226, 1997.

[18] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh. Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience. In *ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM)*, Karlsruhe, Germany, Aug. 2003.

[19] Q. Lv and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *the 16th annual ACM International Conference on Supercomputing*, New York, USA, June 2002.

[20] W. Mader. Über den Zusammenhang symmetrischer Graphen. *Arch. Math. (Basel)*, 21:331–336, 1970.

[21] W. Mader. Eine Eigenschaft der Atome endlicher Graphen. *Arch. Math. (Basel)*, 22:333–336, 1971.

[22] D. Malkhi, M. Naor, and D. Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, Monterey, California, USA, July 2002.

[23] G. S. Manku. Routing networks for distributed hash tables. In *22nd ACM Symposium on Principles of Distributed Computing (PODC 2003)*, Boston, USA, July 2003.

[24] M. Naor and U. Wieder. Novel architectures for p2p applications: the continuous-discrete approach. In *ACM Symposium on Parallel Algorithms and Architectures (SPAA03)*, San Diego, USA, June 2003.

[25] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Lser. Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In *12th Intl. World Wide Web Conference*, Budapest, Hungary, May 2003.

[26] S. R. Oehring, F. Sarkar, S. K. Das, and D. H. Hohndel. Cayley graph connected cycles : A new class of fixed-degree interconnection networks. In *28th Annual Hawaii International Conference on System Sciences*, Hawaii, USA, May 1995.

[27] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Annual Conference of the ACM Special Interest Group on Data Communications (ACM SIGCOMM 2001)*, San Diego, CA, USA, Aug. 2001.

[28] R. A. Rowley and B. Bose. Fault-tolerant ring embedding in de bruijn networks. *IEEE Trans. Comput.*, 42(12):1480–1486, 1993.

[29] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Heidelberg, Germany, Nov. 2001.

[30] G. Sabidussi. Vertex–transitive graphs. *Monatsh. Math.*, 68:426–438, 1964.

[31] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. Hypercup - hypercubes, ontologies and efficient search on p2p networks. In *International Workshop on Agents and Peer-to-Peer Computing*, Bologna, Italy, July 2002.

[32] H. Shen, C. Xu, and G. Chen. Cycloid: A constant-degree and lookup-efficient p2p overlay network. In *International Parallel and Distributed Processing Symposium (IPDPS2004)*, Santa Fe, New Mexico, Apr. 2004.

[33] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Annual Conference of the ACM Special Interest Group on Data Communications (ACM SIGCOMM 2001)*, San Diego, CA, USA, Aug. 2001.

[34] J. Turner. Point-symmetric graphs with a prime number of points. *J. Combinatorial Theory*, 3:136–145, 1967.

[35] M. E. Watkins. Connectivity of transitive graphs. *J. Combinatorial Theory*, 8:23–29, 1970.

[36] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1), 2004.