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

Changtao Qu[1], Wolfgang Nejdl[1] and Matthias Kriesell[2]

[1] L3S and University of Hannover
Expo Plaza 1, D–30539 Hannover, Germany
{qu, nejdl}@l3s.de

[2] Institute of Mathematics (A), 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. While obviously dynamic DHT algorithms which have to approximate these topologies for dynamically changing sets of peers play a very important role for DHT networks, important insights can be gained by clearly focussing on the static DHT topology as well. In this paper we analyze and classify current DHTs in terms of their static topologies based on the Cayley graph group-theoretic model and show that most DHT proposals use Cayley graphs as static DHT topologies, thus taking advantage of several important Cayley graph properties such as vertex/edge symmetry, decomposability and optimal fault tolerance. Using these insights, Cayley DHT design can directly leverage algebraic design methods to generate high-performance DHTs adopting Cayley graph based static DHT topologies, extended with suitable dynamic DHT algorithms.

1 DHTs and Static DHT Topologies

Two important characteristics of distributed hash tables (DHTs) are network degree and network diameter. As DHTs are maintained 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), and CAN (tori).

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

2 Cayley DHTs — A Group-Theoretic Model for Analyzing DHTs

2.1 Groups and Cayley Graphs

Cayley graphs were 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, as every vertex transitive interconnection network can be represented as the quotient of two Cayley graphs [2]. 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.

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) $1 \cdot a = a$ 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 $a \cdot 1 = a$ 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.

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*.

2.2 Non-constant Degree Cayley DHTs

HyperCup [3] 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. [4, 5]) 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*.

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 joint copies of the previously constructed graphs and add edges between pairs of corresponding vertices.

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 $K_2 \times \cdots \times K_2$ (d factors K_2).

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 can 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, e.g. permutation groups: For Γ, we take the subgroup of the permutation group $S_6 := (\{f : \{1, \ldots, 6\} \longrightarrow \{1, \ldots, 6\} : f$ 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.)

Chord [6] 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. For the topology of the ideal d-bit Chord key space, we simply take the Cayley graph $R_d := G((\mathcal{Z}_{2^d}, +), \{\pm 2^k : k \in \{0, \ldots, d-1\}\})$.

2.3 Constant Degree Cayley DHTs

Cycloid [4] 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) [7], defined as:

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 $(i = j \wedge x = s_i \cdot y)\}$.

$CGCC(\Gamma, S)$ is 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*. [8] 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 node with a cycle of length n.

Viceroy [9] 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 [9]. 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 [9]). 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* $\boldsymbol{B}_2^d$ is a graph with vertices $V = V(\boldsymbol{B}_2^d) = \mathcal{Z}_{d-1} \times \mathcal{Z}_2^d$ 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 $1 \pmod 2$in the ith component of v. The d-*dimensional binary wrapped butterfly* B_2^d is the underlying graph of the digraph $\boldsymbol{B}_2^d$, where there is a (single) edge ab whenever there is an edge (a, b) or an edge (b, a) in $\boldsymbol{B}_2^d$. As we can see, B_2^d is 4-regular for $d \geq 3$.

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 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 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 more details see [8]).

CAN [10] 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 $i \in \{1, \ldots, d\}$. They lose then, however, the possibility to consider the d-dimensional hypercube as a special torus, namely as $T_{2,\ldots,2}$ (d indices 2).

2.4 Non-Cayley DHTs

P-Grid [11] Among non-Cayley DHTs, to the best of our knowledge, only P-Grid [11] still retains most of the advantages of Cayley networks. P-Grid uses prefix based routing, and can be considered as a randomized approximation of hypercube. The routing network has a binary trie abstraction, with peers residing only at the leaf nodes. Each peer is thus responsible for all data items with the prefix corresponding to the peer's path in the trie. For routing, peers need to maintain routing information for the complimentary prefix for each of the intermediate nodes in its path. However, routing choice can be made for any peer belonging to the complimentary paths, and P-Grid exploits these options in order to randomly choose routing peer(s), which in turn provides query-forwarding load-balancing and by choosing more than one routing options, resilience. Additionally, the choices can be made based on proximity considerations, and though routing is randomized, since it is to complimentary key-space partitions, P-Grid routes have the added flexibility to be either bidirectional or unidirectional.

Pastry/Tapestry [12] [13] 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 an approximation of hypercubes. As analyzed in [5], 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.

3 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 provide a good insight into Cayley DHT design.

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 $xy \in E(G)$.

A graph G is called *vertex symmetric* or *vertex transitive* if for arbitrary $x, y \in V(G)$ there exists an automorphism φ of G such that $\varphi(x) = y$. As the automorphism $z \mapsto z \cdot x^{-1} \cdot y$ maps x to y, we obtain the following classical observation.

Theorem 1. 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 a *bubble sort graph* (see [8]).

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.

In addition to vertex transitivity, Cayley graphs may also have another important property, edge transitivity. A graph G is *edge symmetric* or *edge transitive* if for arbitrary edges wx, yz there exists an automorphism φ such that $\varphi(w)\varphi(x) = yz$. Clearly, every edge transitive graph without isolated vertices is vertex transitive, but the converse is not true. For a discussion of the problem of determining the edge transitiv Cayley graphs we refer to [1] and [8].

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. Non–Cayley DHTs are not edge transitive. Edge transitivity results in a unified method to evaluate communication load on edges. In edge transitive Cayley DHTs communication load is uniformly distributed on all edges without points 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 through the design of the routing algorithm.

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. 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 $s \in S$.

Among Cayley DHTs, HyperCup (hypercubes) and Chord (ring graphs) can be proven to be hierarchical [8]. 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" [8]. In 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 [14]. Furthermore, in hierarchical Cayley DHTs, there usually support easy solutions to dynamically organize nodes (or node aggregations) to ensure proximity of DHTs. Hierarchical Cayley graphs have not yet been intensively investigated for DHT design. Two promising hierarchical Cayley graphs not yet utilized in DHT design are star graphs and pancake networks [15], which have smaller network diameter than hypercubes of the same degree.

Connectivity and Fault Tolerance. 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.

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, so are hypercubes (as one can prove by induction on the dimension, using the recursive characterizations). For edge transitive graphs, we have the following.

Theorem 2. [16–18] (cf. [19]) 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 $G := G(\mathcal{Z}_8, \{\pm1, \pm3, 4\})$, 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 3. [16–18] (cf. [19]) 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, i.e. 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 [16] which implies that *every triangle free connected vertex transitive graph is optimally fault tolerant*. More generally, every vertex transitive graph without four pairwise adjacent vertices is optimally fault tolerant [17]. This gives alternative proofs of the optimal fault tolerance of hypercubes 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 $G(\mathcal{Z}_8, \{\pm1, \pm3, 4\})$ indicates that it might be already a problem to characterize the optimally fault tolerant circulants (solved in [20]).

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 [17, 18] (cf. [19]). Hierarchical Cayley graphs as in Definition 3 and as in [8] or [14] 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 [8] or [14].

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 [10]. 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.

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 [6] or the state-machine approach based replication in Viceroy [9].

Hamiltonicity and Cyclic Routing. 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 it is still open whether every 2-connected Cayley graph has a hamiltonian path.

The question of *hamiltonian cycles and paths* in Cayley graphs has a long history [21]. All aforementioned topologies of Cayley DHTs such as hypercubes, ring graphs, butterfly, cube-connected cycles, and d-dimensional tori have been proven to be hamiltonian.

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. [5] observes 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, we agree that an hamiltonian 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), which, however, maintain a ring structure through their dynamic DHT algorithm.

4 Discussion and Related Work

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, naturally supporting desirable DHT features such as communication load balancing and fault tolerance.

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.

Constant-degree Cayley graphs 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, cube connected cycles are 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 2.3. 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 [22]:

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 $\mod k$. 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 $C_k \times \cdots \times C_k$ (d factors).

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 [22]. 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.

Gummadi et al. [5] investigate some commonly used static DHT topologies and explore how these topologies affect static resilience and proximity routing by analyzing the flexibility of different DHTs, i.e. the algorithmic freedom left after the static topologies has been chosen. Manku's [23] analysis starts from static DHT topologies, but then heavily involves dynamic DHT algorithms. His classification for DHT systems (deterministic and randomized) are certainly of value, but cannot serve as an analytical framework for comparing static DHT topologies. Datar [24] provides an in-depth investigation to butterfly graphs and further proposes a new DHT system using multi-butterfles as the static DHT topology. Castro et al. [25] make a comparative study of Pastry, taking Chord and CAN as reference systems.

5 Conclusions

We have discussed DHT topologies in the framework of Cayley Graphs, which is one of the most important group-theoretic models for the design of parallel interconnection networks. Associating Cayley graphs with DHTs enables us to directly leverage the research results for interconnection networks for the DHT design without the need of starting from scratch. Cayley graphs explicitly support an algebraic design approach, which allows us to start with an arbitrary finite group and construct symmetric DHTs using that group as the algebraic model, concisely specifying a DHT topology by providing the appropriate group plus a set of generators. This algebraic design approach also enables 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 [3, 26] 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 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 group.

References

1. Akers, S.B., Krishnamurthy, B.: A group-theoretic model for symmetric interconnection networks. IEEE Trans. Comput. **38** (1989) 555–566
2. Sabidussi, G.: Vertex–transitive graphs. Monatsh. Math. **68** (1964) 426–438
3. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: Hypercup - hypercubes, ontologies and efficient search on p2p networks. In: Intl. Workshop on Agents and Peer-to-Peer Computing, Bologna, Italy (2002)
4. Shen, H., Xu, C., Chen, G.: Cycloid: A constant-degree and lookup-efficient p2p overlay network. In: Intl. Parallel and Distributed Processing Symposium, Santa Fe (2004)
5. Gummadi, K., Gummadi, R., Gribble, S., Ratnasamy, S., Shenker, S., Stoica, I.: The impact of dht routing geometry on resilience and proximity. In: ACM Annual Conference of the Special Interest Group on Data Communication, Karlsruhe, Germany (2003)

6. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Annual Conference of the ACM Special Interest Group on Data Communications, San Diego, CA, USA (2001)
7. Oehring, S.R., Sarkar, F., Das, S.K., Hohndel, D.H.: Cayley graph connected cycles : A new class of fixed-degree interconnection networks. In: 28th Annual Hawaii Intl. Conference on System Sciences, Hawaii, USA (1995)
8. Heydemann, M.C., Ducourthial, B.: Cayley graphs and interconnection networks. Graph Symmetry, Algebraic Methods and Applications,"NATO ASI C" **497** (1997) 167–226
9. Malkhi, D., Naor, M., Ratajczak, D.: Viceroy: A scalable and dynamic emulation of the butterfly. In: 21st ACM Symposium on Principles of Distributed Computing (PODC 2002), Monterey, California, USA (2002)
10. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: Annual Conference of the ACM Special Interest Group on Data Communications, San Diego, CA, USA (2001)
11. Aberer, K., Datta, A., Hauswirth, M.: Ch. 21, "Peer-to-Peer-Systems and Applications". In: P-Grid: Dynamics of self-organization in structured P2P systems. Springer LNCS (2004)
12. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: IFIP/ACM Intl. Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany (2001)
13. Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D., Kubiatowicz, J.D.: Tapestry: A resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications **22** (2004)
14. Alspach, B.: Cayley graphs with optimal fault tolerance. IEEE Trans. Comput. **41** (1992) 1337–1339
15. Berthom, P., Ferreira, A., Perennes, S.: Optimal information dissemination in star and pancake networks. IEEE Tran. on Parallel and Distrubuted Systems **7** (1996)
16. Mader, W.: Eine Eigenschaft der Atome endlicher Graphen. Arch. Math. (Basel) **22** (1971) 333–336
17. Mader, W.: Über den Zusammenhang symmetrischer Graphen. Arch. Math. (Basel) **21** (1970) 331–336
18. Watkins, M.E.: Connectivity of transitive graphs. J. Combinatorial Theory **8** (1970) 23–29
19. Boesch, F., Tindell, R.: Connectivity and symmetry in graphs. In: Graphs and applications (Boulder, Colo., 1982). Wiley-Intersci. Publ. Wiley, New York (1985) 53–67
20. Boesch, F., Tindell, R.: Circulants and their connectivities. J. Graph Theory **8** (1984) 487–499
21. Curran, S.J., Gallian, J.A.: Hamiltonian cycles and paths in Cayley graphs and digraphs—a survey. Discrete Math. **156** (1996) 1–18
22. Dally, W.J.: A VLSI Architecture for Concurrent Data Structures. Hingham, MA: Kluwer (1987)
23. Manku, G.S.: Routing networks for distributed hash tables. In: 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003), Boston, USA (2003)
24. Datar, M.: Butterflies and peer-to-peer networks. In: 10th Annual European Symposium. Lecture Notes in Computer Science, Rome, Italy, Springer (2002)
25. Castro, M., Druschel, P., Hu, Y.C., Rowstron, A.: Exploiting network proximity in distributed hash tables. In: Intl. Workshop on Future Directions in Distributed Computing, Bertinoro, Italy (2002)
26. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Lser, A.: Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In: 12th Intl. World Wide Web Conference, Budapest, Hungary (2003)