Efficient Routing in Data Center with Underlying Cayley Graph*

Miguel Camelo, Dimitri Papadimitriou, Lluís Fàbrega, and Pere Vilà

Abstract. Nowadays data centers are becoming huge facilities with hundreds of thousands of nodes, connected through a network. The design of such interconnection networks involves finding graph models that have good topological properties and that allow the use of efficient routing algorithms. Cayley Graphs, a kind of graphs that represents an algebraic group, meet these properties and therefore have been proposed as a model for these networks. In this paper we present a routing algorithm based on Shortlex Automatic Structure, which can be used on any interconnection network with an underlying Cayley Graph (of some finite group). We show that our proposal computes the shortest path between any two vertices with low time and space complexity in comparison with traditional routing algorithms.

1 Introduction

The growing demand for cloud computing services is leading to an increasing deployment of large-scale Data Center (DC) as its underlying infrastructure [1]. A fundamental component of such DCs is the interconnection network that provides communication among the different components of the physical infrastructure. The design of such networks has the goal of finding graph models that i) have good properties topological (e.g. high connectivity, small degree, etc.) to ensure good

Miguel Camelo · Lluís Fàbrega · Pere Vilà
IIiA, Universitat de Girona, Spain
e-mail: {miguel.camelo,lluis.fabrega,pere.vila}@udg.edu

Dimitri Papadimitriou
Alcatel-Lucent Bell, Belgium
e-mail: dimitri.papadimitriou@alcatel-lucent.com

* This research work is partially funded by the European Commission (EULER project, FP7 258307), the Spanish Government (RoGER project, TEC 2012-32336), the Generalitat of Catalonia (CSI project, SGR-1202), and the SUR of the Generalitat of Catalonia and the European Social Fund (PhD grant FI-DGR 2011).

P. Contucci et al. (eds.), *Complex Networks V*, Studies in Computational Intelligence 549,
DOI: 10.1007/978-3-319-05401-8_18,

performance in terms of throughput, delay, robustness, etc., and that ii) allow to have routing algorithms with both low time complexity (time to take routing decision) and low space complexity (memory resources to build the routing table) with respect to the number of nodes of the network [2]. Cayley Graphs (CGs) [3], a kind of graphs that represents an algebraic group, meet these 2 properties and therefore have been proposed as a model of DCs interconnection networks [4, 5, 6].

Concerning topological properties, CGs have high symmetry, hierarchical structure, recursive construction, high connectivity and fault tolerance, among others [2]. The definition of the CG implies that the vertices are elements of some group but it does not imply any specific group. This flexibility allows to get a graph that meets the desired requirements on diameter, vertex degree, number of nodes, etc [7]. Moreover, it has been demonstrated that CGs can also be used as models of deterministic small world networks [8]. With respect to routing algorithms, the traditional Dijkstra or Bellman-Ford routing algorithms can be used in any kind of graph but requiring large amount of memory and/or with slow convergence time for large graphs [4]. Unlike them, there are routing algorithms for specific type of graphs that take advantage of their particular topological characteristics, reducing their time and space complexity. This is the case of the routing algorithms for network topologies based on hypercubes [9], butterflies [10] and star graphs [11], among others, which actually are CGs of some specific groups.

A routing algorithm for two specific classes of Cayley-based networks, the star and pancake graphs, is presented in [13]. Since these graphs have a representation by permutations, they propose a routing algorithm based on permutation sort. However, this approach does not ensure a shortest path routing. K. Tang and B. Arden prove in [14] that all finite CGs can be represented by generalized chordal rings (GCR) and then propose an iterative routing algorithm based on table look-up. The space complexity of such algorithm is $O(n^2)$ and its time complexity O(D), where n and D are the size and diameter of the network, respectively. Wang and Tang [15] propose a topology-based routing for Xmesh with CGs as the underlying topology. They prove that the average path lengths between nodes is smaller and the averaged power consumed is less than the original Xmesh. They use a CG from the Borel Subgroup, which is also known as Borel Cayley Graph (BCG), as underlying topology. Their routing algorithm computes off-line a shortest path routing table from the node *Id* to all other nodes, and then they use this table to create the routing table for the rest of nodes based on the vertex transitivity property of CG. This algorithm is is bounded by $O(\log_4 n)$ and its space complexity is given by $O(n^2 D)$.

A distributed and fault-tolerant routing algorithm for BCG is presented in [16]. This two-phase algorithm uses two types of routing tables according to link failures: (1) a Static Routing Table (SRT) (computed using [14]) and (2) a Dynamic Routing Table (DRT). The first phase performs routing through the shortest path according to the Static Routing Table. If there is a link failure making the shortest path unavailable, DRT is updated and other shortest paths will be used. In the case that all shortest paths are disconnected, the phase 2 exploits the path length information in the SRT to search additional routes besides the shortest paths. Finally, authors in [5] present a routing algorithm on a special class of CGs used as underlying graph for a wireless data center. A two-level routing algorithm is proposed to send messages between 1) servers in the same rack and 2) servers in different racks. This algorithm is a geographical routing that exploits the uniform structure of the underlying topology. The identification of each server is defined by composition of three values: the coordinates of the rack, the story that contains the server within the rack and the index of the server in the story. In addition, each server uses three routing tables to forward package from source to destination using a shortest path route.

Note that the works presented above could be grouped into the one or more of the following routing algorithm categories: a) those ones that are designed to specific CGs, b) general purpose ones with high space/time complexity and c) low complexity ones that do not ensure shortest paths. In contrast with them, in this paper we present a low space and time complexity routing algorithm for any interconnection network where its underlying graph is a CG of some finite group. The input of the algorithm can be either the group presentation $G = \langle S|R \rangle$, where S and R are the generators and relators of G /citewmagnus, or the permutation (or matrix) representation of the group. The proposed algorithm is based on the fact that finite groups are Automatics and have an Shortlex Automatic Structure (SAS) [12]. These structures solve the shortest path problem in CGs of finite groups by solving the equivalent Minimum Word Problem, which is NP-Hard [18], in quadratic time with respect to the length of the equivalent path written as a sequence of group generators.

The paper is organized as follows. In Section 2 we present the theoretical background about group theory, geometrical group theory and automatic structures. In Section 3 we describe our proposed shortest path algorithm, its time and space complexity and an example of the application of our routing algorithm to a 3-cube graph. Conclusions and future work are presented in Section 4.

2 Preliminaries

In this section, we establish terminology, notation, and background material about group theory and Automatic Groups. For more definitions and results on combinatorial group theory we refer the reader to [3], and for groups and graphs to [20].

Let G be a finite group. The identity of the group G is denoted by Id and the group operation is the multiplication. Let $S = \{s_1, \ldots, s_n\}$ be a set of elements in a group G. We say that S generates G if every element of G can be expressed as a product of elements from S and their inverses. A group G is finitely generated if it has a finite generating set. A word is a sequence $w = (s_1 s_2 \ldots)$, where $s_i \in S \cup S^{-1}$, for all i. We say that w is freely reduced if it does not contain any sub-word $s_i s_i^{-1}$. We say that a group is a free group with basis S, represented by $F(S)$, if S is a set of generators for the group and no freely reduced word $w \in F(S)$ represents the identity.

Definition 1. *Let G be a group with generating set S. The Cayley Graph $\Gamma(G,S)$ of G with respect to S is the graph with vertex set $V(\Gamma(G,S)) = \{g \mid g \in G\}$ and edge set $E(\Gamma(G,S)) = \{(g,gs) \mid s \in S, g \in G\}$.*

The group G acts on $\Gamma(G,S)$ by multiplication on the left: the element $g \in G$ defines a map $\phi_g : h \to gh$ that maps a vertex $h \in \Gamma(G,S)$ to the vertex gh, while it brings adjacent vertices to adjacent vertices, preserving edges. The graph $\Gamma(G,S)$ is directed but it also can be considered undirected if we take an inverse-closed generating set, i.e. $S = S^{-1}$. If $\Gamma(G,S)$ has not auto-loops, then $Id \notin S$. If $\Gamma(G,S)$ has no loops and no multiple edges, then we say that $\Gamma(G,S)$ is reduced, and then we say that it is simple if in addition is undirected. Finally, for any finite presentation of a group in terms of generators and relators, there exists an associated Cayley Graph, i.e. the geometry and structure of the $\Gamma(G,S)$ is directly related with a group presentation and specifically with its generator set [3].

A metric on the CG is defined by assigning unit length to each edge and defining the distance between two points to be the minimum length of paths joining them. In this case, the action by left multiplication of G on $\Gamma(G,S)$ is by isometries. Finally, the algebraic structure of the group, which is encoded into its group presentation, permits to define the word length and metric on such group:

Definition 2. *Let $\pi : F(S) \to G$ be a group homomorphism and let $\pi(w)$ be the element of G represented by w under π. The length of g, identified by $l_s(g)$, is the length of the shortest word in the free group $F(S)$ representing g, i.e. $l_s(g) = min\{l_s(w) \mid w \in F(S), \pi(w) = g\}$.*

Definition 3. *Let G be a group with generating set S. The corresponding word metric (i.e. distance function) d_s is the metric on G satisfying $d(Id,s) = d(Id,s^{-1}) = 1$ for all $s \in S$, and $d(g,h) = min\{l_s(w) \mid w \in F(S), \pi(w) = g^{-1}h\}$, for all $g,h \in G$.*

Note that the word metric on a group G is a way to measure the length of the shortest path between any two elements of G on $\Gamma(G,S)$. This metric measures how efficient the difference $g^{-1}h$ can be expressed as a word in the generating set for G. Thus, it is possible to visualize the geometry of a group G by looking at its CG, because the word metric of the group corresponds to the graph metric induced on $\Gamma(G,S)$.

Additionally to the algebraic and geometric structure of the group, there exists a third point of view to work in an efficient way on groups: they can be seen as languages. Let G be a group, A an alphabet and A^* the set of strings (or words) on the alphabet A. By interpreting concatenation as an associative multiplication on G, we define a monoid homomorphism $\pi : A^* \to G$. If w is a string over A, we say that $\pi(w)$ is the element of G represented by w. If the homomorphism is surjective, i.e. $\pi(A)$ generates G as a group, then A is the set of group generators for G. We also define a bijective map $\phi : A \to S$, where S is the set of generators of G, to indicate that each element of the set A represents an element of S. In the rest of the paper, we will use the set S to reference both generators of the group and the alphabet A.

Given any word w, there is an associated edge path in $\Gamma(G,S)$. The path starts at the identity vertex and then traverses edges of $\Gamma(G,S)$ as dictated by w. Conversely, every finite edge path in $\Gamma(G,S)$ describes a word in terms of the generators and their inverses: reading off the labels of edges being traversed, and adding an inverse if they are traveling in the opposite direction of the orientation of the edge. Given this relationship between languages and groups, D. Epstein et. al. in [12] present a complete work about algebraic groups treated by finite state automaton.

Definition 4. *Let G be a group. An automatic structure on G consists of a set S of generators of G, a finite state automaton W over S, and a finite state automaton M_s over (S,S), for $s \in S \cup Id$, satisfying the following conditions: 1) the map $\pi : L(W) \to G$ is surjective and 2) for $s \in S \cup Id$, we have $(w_1, w_2) \in L(M_s)$ if and only if $\pi(w_1)s = \pi(w_2)$ and both w_1 and w_2 are elements of L(W).*

In the definition, W is called the *word acceptor*, M_{Id} the *equality recognizer*, and each M_s, for $s \in S$, a *multiplier automaton* for the automatic structure. An automatic group is one that admits an automatic structure. Note that M_{Id} recognizes equality in G between words in $L(W)$. From a given automatic structure, it is always possible to use M_{Id} to construct another one, such that W accepts a unique word mapping onto each element of G; choosing the lexicographically least among the shortest words that map onto each element as the normal form representative of that element. This W is a word-acceptor with uniqueness.

Definition 5. *Let $\leq_S$ be some total order on the alphabet S, the automatic structure is called shortlex if L(W) consists of the shortlex least representatives of each element $g \in G$; therefore the map $\pi_1 : L(W) \to G$ is bijective and all paths in $\Gamma(G,S)$ according to the words of L(W) are shortest ones. In other words, $L(W) = \{w \in S^* \mid w \leq_S v, \forall w, v \in S^*, w =_G v\}$*

Thus, given a group G with generator set S, a word $w \in S^*$ is called a geodesic if it has minimal length among all strings representing the same element as w. Since the language of all geodesic strings maps finite-to-one onto G, SAS is an automatic structure for G that contains a geodesic representative for each element of G. The package KBMAG [17] implements a procedure for computing SAS for groups with finite presentation.

3 A Greedy Routing Algorithm Using Automatic Structures

Let g and h be two vertices in $\Gamma(G,S)$ represented by the words (or labels) w_g and w_h in the set $S \cup S^{-1}$. If $w_h = w_g s_1 s_2 ... s_t$ with $s_i \in S \cup S^{-1}$, $1 \leq i \leq t$, then $s_1 s_2 \dots s_t$ defines a path from vertex g to h with edges labeled by $s_1 s_2 ... s_t$ in $\Gamma(G,S)$. This is equivalent to finding a path from $g^{-1}h = s_1 s_2 \dots s_t$ to the vertex Id. Notice that given g and h, solving the shortest path problem between any pair of vertices in the CG turns into finding a word $w = s_1 s_2 \dots . s_t$ with minimum length such that $w_g^{-1} w_h = w$. This problem is called the Minimum Word Problem [18], and the SAS is an efficient tool to solve it:.

Theorem 1. *(Theorem 2.3.10, [12]) Let G be an group and $(S, L(W))$ an automatic structure for G. For any word w over S, we can find a word in L(W) representing the same element of G as w, in time proportional to the square of the length of w.*

Note that if the word acceptor has uniqueness, then this structure can solve the Minimum Word Problem by reducing words to their (shortlex) normal forms. Because of finite groups have Automatic Structures (in fact they have a SAS [12]), it is possible to find a SAS for a given group presentation G and to use it as a shortest path computation mechanism using only local information. We assume that an incoming message, whether originating at the vertex or in transit from another vertex, contains the shortlex word that represents the destination vertex.

Our routing algorithm consists of two procedures, the vertex labeling and the message forwarding. The vertex labeling procedure is the following:

- Given a group presentation $G = \langle S|R \rangle$ of the underlying $\Gamma(G,S)$, compute the SAS = $(S, L(W))$ of G. If $\Gamma(G,S)$ is given by either its matrix or permutation representation, then construct the group presentation by using the fundamental cycles of $\Gamma(G,S)$ (e.g. by using [19]).
- Select a random vertex of $\Gamma(G,S)$ and construct a spanning tree $T(\Gamma)$ rooted on it by using the Breadth First Search (BFS) algorithm. In the same process, label all vertices with an integer from 1 (the root vertex) to $| G |= n$, according their order of discovery.
- Use $(S, L(W))$ to enumerate the elements of the group G according to its shortlex ordering (corresponding to a BFS through S^*) and re-label each vertex in the spanning tree with its corresponding shortlex word enumerated before. Note that there exists natural one-to-one mapping between vertices and the elements of the group represented by their shortlex words following their BFS ordering.
- Create a table with d rows in each vertex $g \in T(\Gamma)$, where d is the vertex degree, and keep the label of each vertex at distance 1 from itself.

The message forwarding procedure uses the word metric on G to perform a greedy routing strategy. Given two vertices $g, h \in \Gamma(G,S)$ with labels w_g and w_h, the procedure to find the shortest path between them is the following:

- When a message arrives to g, compare the label of g with the destination label and verify whether they are equal or not.
- If the labels are equal, the destination is reached. Otherwise, send the message to the neighbor p_i of g, where $i \in [1,\ldots,d]$, such that $l_s(w_{p_i}^{-1} w_h)$ is minimum. If there exists more than one neighbor p_i with equal minimum length on $l_s(w_{p_i}^{-1} w_h)$, then the message is sent to the neighbor with shortlex w_{p_i}.

The space complexity of our algorithm is bounded by $O(dn)$ because each vertex keeps a list of its d neighbors. On the other hand, the time to make a routing decision is bounded by $O(D^2)$, where D is the diameter of $\Gamma(G,S)$. Note that any two vertices $g, h \in \Gamma(G,S)$ with labels w_g and w_h will have $l_s(w_g) \leq D$ and $l_s(w_h) \leq D$. In fact, any resulting word from $w_g^{-1} w_h$ has $l_s(w_g^{-1} w_h) \leq 2D$. Since $(S, L(W))$ can reduce any word w of length $l_s(w)$ in a time proportional to $O(l_s(w)^2)$ (see Theorem 1), any word $v = w_g w_h$ will have a length $l_s(v) \leq 2D$, and then it can be reduced in $O(D^2)$ to a shortlex equivalent word.

The following is an example of the application of our routing algorithm to a 3-cube graph modeling a 8-node interconnection network (see Figure 1a). This graph is isomorphic to the $\Gamma(G,S)$ of the elementary Abelian group of order 2^3 with group presentation equal to $G = \langle S|R \rangle$, where $S = \{a,b,c\}$ and $R = \{a^2, b^2, c^2, aba^{-1}b^{-1}, aca^{-1}c^{-1}, bcb^{-1}c^{-1}\}$. We start with the labeling procedure.

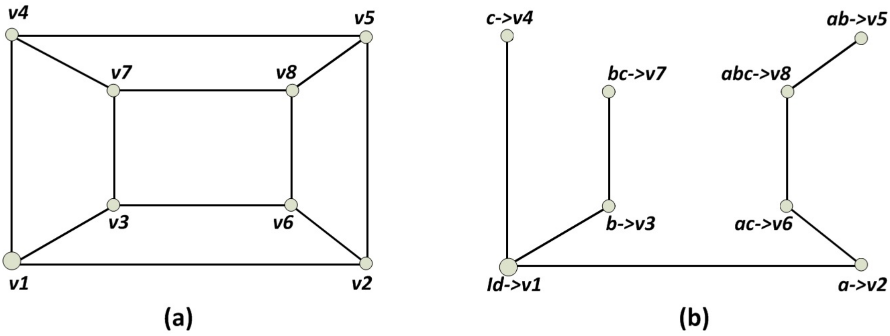

Fig. 1 a) The 3-Cube graph and b) the resulting BFS tree with shortlex labels on vertices

Given a group presentation for $\Gamma(G,S)$, the SAS is computed. Then a random vertex in $\Gamma(G,S)$ (left-bottom corner) is selected and a BFS spanning tree rooted on that vertex is constructed. The resulting tree $T(\Gamma)$ has the following vertices (is discovery order): $\{v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8\}$. At the same time, we use the SAS to enumerate the elements of the group G according to the shortlex ordering: $L(W)=\{Id,a,b,c,ab,ac,bc,abc\}$. Next, the map $\pi : L(W) \to G$ is performed, i.e. the word $w_i \in L(W)$ labels the vertex $v_i \in T(\Gamma)$, for $1 \leq i \leq |G|$ (See Figure 1b). Finally, each vertex creates a table with the labels of its neighbors at distance 1.

Assume now that vertex v_4 sends a message to vertex v_8 (labeled as abd). v_4 uses the labels of its neighbors v_1 and v_7 to compute the length of the shortlex words that represent the following multiplications: $(Id)^{-1}(abc)$ and $(bc)^{-1}(abc)$. As a result of the reduction process using $(S,L(W))$, $(Id)^{-1}(abc)=abc$ and $(bc)^{-1}(abc)=a$. Therefore v_4 sends the message to v_7. Vertex v_7 does the same process with the labels of its neighbors v_6 and v_8. The reduced words that represent $(b)^{-1}(abc)$ and $(abc)^{-1}(abc)$ are ac and Id, respectively. Since Id is the empty word, i.e. the word of length 0, v_7 sends the message to v_8, the final destination. Note that although the labeling process is based on a rooted spanning tree, the algorithm has found the shortest path between v_4 and v_8 in the whole graph and not only in that tree.

4 Conclusions and Future Work

We have proposed a routing algorithm based on SAS for route computation in DC interconnection networks with underlying CG. The input of the algorithm is either a group presentation $G=\langle S|R\rangle$ or the matrix/permutation representation of G. Our routing algorithm is a shortest path one and each node in the network only needs information about its neighbors to take its routing decision. This decision is taken in time proportional to the square of the diameter of $\Gamma(G,S)$. Our proposal uses the fact that finite groups have SAS and these structures can efficiently solve the

Minimum Word Problem in G, which is equivalent to find the shortest path between any pair of vertices of $\Gamma(G,S)$. Moreover, since any finite group is isomorphic to some group of permutations, our algorithm can work on any interconnection network with underlying CG. Finally, although the topologies for DCs are usually very static, it would be important to consider as a future work the network dynamics and to propose fault tolerance mechanism in our algorithm.

References

1. Cisco System Inc.: Cisco Cloud Computing - Data Center Strategy, Architecture, and Solutions. Point of View White Paper for U.S. Public Sector, 1st edn. (2009)
2. Heydemann, M.-C., Ducourthial, B.: Graph Symmetry. NATO ASI Series, pp. 167–224. Springer Netherlands (1997) ISBN 978-90-481-4885-1
3. Meier, J.: Groups, Graphs and Trees: An Introduction to the Geometry of Infinite Groups. Cambridge University Press (2008) ISBN 978-0521719773
4. Schibell, S.T., Stafford, R.M.: Processor Interconnection Networks from Cayley Graph. Discrete Applied Mathematics 40(3), 333–357 (1992) ISSN 0166-218X
5. Shin, J., Sirer, E., Weatherspoon, H., Kirovski, D.: On the feasibility of completely wireless datacenters. IEEE/ACM Transaction on Networking 21(5), 1666–1679 (2013)
6. Xiao, W., Liang, H., Parhami, B.: A Class Of Data-Center Network Models Offering Symmetry, Scalability, and Reliability. Parallel Processing Letters (2012)
7. Loz, E., Siran, J.: New record graphs in the degree-diameter problem. Australasian Journal of Combinatorics 41, 63–80 (2008)
8. Xiao, W., Parhami, B.: Cayley graphs as models of deterministics small-world networks. Information Processing Letters 97(3), 115–117 (2006)
9. Stamoulis, G.D., Tsitsiklis, J.N.: Efficient routing Scheme for Multiple Broadcasts in Hypercubes. IEEE Trans. on Parallel and Distributed Systems 4(7), 725–739 (1993)
10. Stamoulis, G.D., Tsitsiklis, J.N.: The Efficiency of Greedy Routing in Hypercubes and Butterflies. IEEE Transaction on Communication 42(11), 3051–3061 (1994)
11. Kiasari, A.E., Sarbazi-Azad, H.: Analytic performance comparison of hypercubes and star graphs with implementation constraints. Journal of Computer and System Sciences (2007)
12. Epstein, D., Cannon, J., Holt, D., Levy, S., Paterson, M., Thurson, W.: Word Processing in Groups. Jones and Bartlett Publishers (1992) ISBN 0-86720-244-0
13. Akers, S., Krishnamurthy, B.: A group-theoretic model for symmetric interconnection networks. IEEE Transactions on Computers 38(4), 555–566 (1989)
14. Tang, K., Arden, B.: Vertex-transitivity and routing for Cayley graphs in GCR representations. In: ACM Symposium on Applied Computing, SAC, pp. 1180–1187 (1992)
15. Wang, L., Tang, K.: Topology-Based Routing for Xmesh in Wireless Sensor Networks. In: Powell, S., Shim, J.P. (eds.) Wireless Technology. LNEE, vol. 44, pp. 229–239. Springer, Heidelberg (2009)
16. Ryu, J., Noel, E., Tang, K.: Fault-tolerant Routing on Borel Cayley Graph. In: Next Generation Networking Symposium, IEEE ICC 2012, pp. 2872–2877 (2012)
17. Holt, D.: The Warwick Automatic Groups Software. In: Geometrical and Computational Perspectives on Infinite Groups. Amer. Math. Soc. DIMACS Series, vol. 25, pp. 69–82 (1995)

18. Even, S., Goldreich, O.: The minimal-length generating sequence problem is NP-Hard. Journal of Algorithms 2, 311–313 (1981)
19. Cannon, J.: Construction of defining relators for finite groups. Discrete Math. 5, 105–129 (1973)
20. Magnus, W., Karrass, A., Solitar, D.: Combinatorial Group Theory - Presentation of Groups in Terms of Generators and Relations, 2nd revised edn. Dover Publications (2004)