Title
Interconnection Networks with Hypercubic Skeletons

Journal
Journal of Interconnection Networks, 15(01n02)

ISSN
0219-2659

Authors
XIAO, WENJUN
CHEN, WEIDONG
PARHAMI, BEHROOZ

Publication Date
2015-03-01

DOI
10.1142/s0219265915500061

Peer reviewed

Journal of Interconnection Networks
Vol. 15, Nos. 1 & 2 (2015) 1550006 (14 pages)

DOI: 10.1142/S0219265915500061

Interconnection Networks with Hypercubic Skeletons*

WENJUN XIAO
School of Software Engineering,
South China University of Technology, Guangzhou 510006, China

WEIDONG CHEN
Department of Computer Science,
South China Normal University, Guangzhou 510631, China

BEHROOZ PARHAMI†
Department of Electrical and Computer Engineering,
University of California, Santa Barbara, CA 93106-9560, USA
parhami@ece.ucsb.edu

Received 20 June 2015
Accepted 29 June 2015

The hypercubic family of interconnection networks, encompassing the hypercube and its derivatives and variants, has a wide range of applications in parallel processing. Various problems in general complex networks can be addressed by choosing a hypercubic network as a skeleton. In this paper, we provide insight into why hypercubic networks are suitable as network skeletons and discuss a mapping scheme to take advantage of the symmetry of such networks for developing efficient algorithms.

Keywords: Hypercube; network; parallel processing; mapping; virtualization.

1. Introduction

The hypercube and its numerous derivatives and variants, collectively referred to as the hypercubic family of networks, have a wide range of applications as interconnection structures or performance comparison benchmarks.[1–4] In particular, there is a wealth of research about the use of hypercubic architectures as interconnection networks for parallel processing, necessitating studies of their structural, performance, and fault tolerance attributes.[5–9] The references just cited are intended as a small sample of studies, the likes of which are still ongoing, with results published regularly in parallel processing, computer architecture, and some domain-specific venues.

*Project supported by the National Natural Science Foundation of China (Grants 61170313, 61103037, 61370003).
†Corresponding author.

In complex networks, it is common to address various structure-related issues by choosing a hypercubic network as a skeleton.[10–15] It is quite natural to question why hypercubic networks are used as skeletons. In this paper, we provide a theoretical explanation of this choice, following the theory with an investigation of mapping problems.

An N-node general network G can be trivially embedded into the complete graph K_N through an isomorphic mapping. So, given that K_N can be mapped onto a hypercube Q with dilation $\log N$, any N-node network G can be mapped onto a hypercube Q with dilation $\log N$, which implies that a path in Q corresponding to an edge in G has length of at most $\log N$. The hypercube Q is highly symmetric, leading to simple analysis and algorithms. Thus, a highly efficient routing algorithm in Q readily translates to an efficient routing algorithm in the general network G. This is the essence of network virtualization,[16] an active research branch in computer science.

We have found hypercubic architectures useful in our prior work and have used them both for proposing novel networks and for modeling existing ones. In this paper, we further develop the methods used in Refs. 16–22 to provide insight into why hypercubic networks are suitable as network skeletons and discuss a mapping scheme to take advantage of the symmetry of such networks for developing efficient algorithms in various application domains. We also provide concrete evidence for the resultant benefits via application to real networks that are of practical interest.

The paper is organized as follows. Section 2 introduces related work, Section 3 explains in detail the reason why hypercubic networks are chosen as skeleton of networks. Two applications of our theory in mapping and network virtualization are discussed in Section 4. We discuss some related problems and conclude the paper in Section 5. An appendix provides details of 3 example networks.

2. Previous Work

The hypercubic family of interconnection networks has been investigated extensively for several decades.[1–9] In parallel processing, much research has been devoted to the hypercube and its derivatives and variants, because their high symmetry leads to simple analysis and efficient algorithms, making them both theoretically and practically desirable. An n-dimensional hypercube has $N = 2^n$ nodes, a diameter of $n = \log_2 N$, and highly efficient routing schemes (e.g., dimension-order routing) of order $\log N$. Some variants of the hypercube possess similar desirable properties, and have thus found applications in several fields of computer science.[10–15]

Real networks, on the other hand, are generally complicated, with many of them having small-world properties and order-($\log N$) diameters, where N is the network size.[10–13] An effective approach to investigating these networks is via network virtualization,[16] that is, simulating them by using certain networks with simpler structures through a mapping between them. These networks with simple structures generally have good communication performance.

Proper choice of the mapping leads to efficient communications for a general network.[16] Unfortunately, such mapping problems, with constraints between networks, are generally NP-hard. Therefore, for a network under consideration, it is vital to choose a simple network that renders the mapping problem feasible.

3. Hypercubic Skeletons in Networks

In this section, we provide a theoretical explanation of why hypercubic networks are often chosen as skeletons in networks. For this purpose, we need some terminology from algebra, beginning with the definitions of Cayley directed graphs, Cayley (undirected) graphs, and coset graphs.[4]

Definition 1. (Cayley directed graph). Assume that G is a finite group with identity element e, and let S be a generating set of G with $e \notin S$. The Cayley directed graph defined on G and S is denoted by $\Gamma = Cay(G, S)$ with $V(\Gamma) = G$ and $E(\Gamma) = \{(g, gs) | g \in G,\ s \in S\}$. $\square$

Definition 2. (Cayley undirected graph, or simply Cayley graph). Assume that G is a finite group with identity element e, and let S be a generating set of G with $e \notin S$, such that $g^{-1} \in S$ iff $g \in S$. The Cayley undirected graph defined on G and S is denoted by $\Gamma = Cay(G, S)$ with $V(\Gamma) = G$ and $E(\Gamma) = \{(g, gs) | g \in G,\ s \in S\}$. $\square$

Definition 3. (Coset graph). Given a Cayley directed graph $\Gamma = Cay(G, S)$ and a subgroup $K \leq G$, the right coset graph defined on G, K, and S is a directed graph, denoted by $\Delta = Cos(G, K, S)$, such that the node set is the right coset G/K, that is, $V(\Delta) = G/K$, and for any g, $g' \in G$, $(Kg,\ Kg^{'}) \in E(\Delta)$ iff there exist k, $k' \in K$ and $s \in S$ satisfying $kgs = k'g'$. $\square$

Hypercube, cube-connected cycles, and butterfly networks are well-known examples of Cayley graphs, while de Bruijn and shuffle-exchange networks belong to the class of coset graphs.[1–4] For instance, a k-dimensional n-ary hyper-torus is the Cayley graph $\Gamma = Cay(G, S)$ defined on group G and its subset S, where $G = Z_n^k$ and $S = \{(\pm 1, 0, \ldots, 0),\ (0, \pm 1, \ldots, 0), \ldots, (0, 0, \ldots, \pm 1)\}$. It is easily seen that the basic k-cube is a k-dimensional binary hyper-torus. A k-dimensional n-ary de Bruijn graph is the coset directed graph $\Delta = Cos(G, K, S)$ defined on group G, subgroup K, and subset S, with $G = Z_n^k Z_k$, $K = Z_k$, and $S = \{(0, 0, \ldots, 0; 1),$ $(1, 0, \ldots, 0; 1\}$. A k-dimensional n-ary cube-connected cycles network is the Cayley directed graph $\Gamma = Cay(G, S)$ defined on group G and its subset S, where $G = Z_n^k Z_k$ and $S = \{(0, 0, \ldots, 0; 1),\ (1, 0, \ldots, 0; 0)\}$. Undirected versions of k-dimensional n-ary de Bruijn and cube-connected cycles graphs can be similarly defined.

The Cayley and coset graphs above can be viewed as hypercube variants, with similarly favorable topological and performance properties. For example, de Bruijn and cube-connected cycles graphs are highly symmetric, have small diameters, and support efficient communication. Moreover, these graphs have relatively small node degrees, which is beneficial for many network applications.

We now introduce a special mapping between two networks, called generalized homomorphism, which is a vital concept in the following discussion, and has a wide range of applications.[16]

Generalized Homomorphism: Given graph $\Sigma = (V(\Sigma), E(\Sigma))$ and $\Gamma = (V(\Gamma), E(\Gamma))$, a mapping f from Σ to Γ is a *generalized homomorphism* if f maps any edge (a, b) in Σ onto a path from $f(a)$ to $f(b)$ in Γ. If the path from $f(a)$ to $f(b)$ has length at most n, then mapping f is said to have dilation n. □

Consider a general network G with N nodes. The facts that G can be embedded into the complete graph K_N through an isomorphism and graph K_N can be mapped onto a hypercube Q with dilation $\log N$ lead to the following important result.

Theorem 1. Any N-node graph G can be mapped onto a hypercube Q with dilation $\log N$.

Proof. First, consider the N-node complete graph K_N. The fact that K_N has all the edges of G leads to G being embeddable into K_N with an isomorphic mapping. We next show that K_N can be mapped onto a hypercube with dilation $\log N$. Let $2^{n-1} < N \leq 2^n$. Define a generalized homomorphism f from K_N to an n-dimensional hypercube Q such that for any edge e in K_N, $f(e)$ is a shortest path in Q. For example, provided that $N = 2^n$, the edge $(0, N-1)$ in K_N is mapped onto a path $f((0, N-1)) = 00\ldots0 \to 0\ldots01 \to 0\ldots11 \to \ldots \to 1\ldots11$ in Q, where $1\ldots11$ is the n-bit binary representation of $N-1$. Noting that the mapping f has a dilation of no more than n completes the proof. □

It is well-known that many real networks possess small-world properties, implying that they have diameters of order $\log N$. Thus, Theorem 1 suggests that it is quite likely to find a routing algorithm of order $\log^2 N$ for such networks. Next, we elaborate on the latter property and its practical importance by means of two examples.

4. Two Example Applications

In the following we illustrate the application of Theorem 1 to obtaining efficient routing algorithms in general networks. Our two representative examples have been chosen from the classes of complex networks and peer-to-peer networks.

4.1. *Routing in complex networks*

Many real networks are small-world networks, whose diameters are of order $\log N$. Moreover, many of these real networks also are scale-free.[17,18] Before proceeding with a discussion of routing, we describe several concepts related to such networks.

Small-world, scale-free networks: An N-node network is a small-world network if its diameter is of order $\log N$ and its clustering coefficients are positive. An N-node network is scale-free if its node degrees follow the distribution $P(k) = ck^{-\gamma}$

for $1 \leq k < N$, where c and γ are constants. For many scale-free networks, the degree exponent γ is at least 2 and the number of degree-1 nodes is nonzero. □

Let $1 = k_1 < k_2 < \cdots < k_l$ be the degree sequence of a network and n_i be the number of degree-i nodes. Regarding the degree sequence of a scale-free network, we have the following result.[18,19]

Theorem 2. For an N-node scale-free network Γ with the degree exponent $\gamma \geq 2$, if $n_1 \neq 0$, then n_1 is of order N, and the degree sequence length l is of order $\log N$.

Theorem 2 is a very useful result. Consider a scale-free, small-world network Γ. The diameter and degree sequence length l of the network being $O(\log N)$, and our ability to estimate the number n_k of degree-k nodes from $n_k \approx n_1 k^{-\gamma}$,[17,18] allow us to derive an efficient routing algorithm as follows:

- For any $k_i > 3$, the subgraph induced by all the nodes of degree k_i in Γ is simulated by a k_i-dimensional hypercube or another hypercubic network.
- The network Γ itself is simulated by the Cartesian product of hypercubic graphs involved in the simulation above, along with all degree-1, degree-2, and degree-3 nodes.
- To route from source node u to destination node v, we build two paths. One path begins at u, proceeds through nodes with increasing degrees along the degree sequence, and arrives at a node with maximum degree. The second path, beginning at v and ending at a node of maximum degree, follows a similar course. Combining these two paths, we obtain a path of length $O(\log^2 N)$ from u to v.

Figure 1 illustrates a route from node u to node v via an example, with B_i denoting the set of degree-i nodes. Beginning at u, the route proceeds through nodes with increasingly higher degrees, eventually arriving at node w in B_l, the set of max-degree nodes. Then, the route continues from w along the node degree sequence in a reverse direction to finally reach v. If for every $i = 4, 5, \ldots, l$, the routing in the subgraph induced by B_i is efficient, say its path lengths are of order $\log N$, then there exists an efficient routing algorithm in the whole network Γ that is of order $\log^2 N$.

To illustrate the applicability of our method to real-world networks, we have provided an appendix with the graphs for "elegans" (a biological network, data from http://www.cosin.org/extra/data), "abiword" (a software network, http://www.tc.cornell.edu/_myers/Data/SoftwareGraphs/index.html), and "ncstrlwg2" (a network representing collaboration by scientists, data supplied by E. J. Newman), along with associated data on the number of nodes and the number of intra-cluster edges for each "cluster," a name commonly given to the subgraph induced by all the nodes having the same degree. These graphs are not scale-free in the strict sense of the term, but have degree sequence lengths on the order $(\log N)^c$ for a reasonably small constant c.

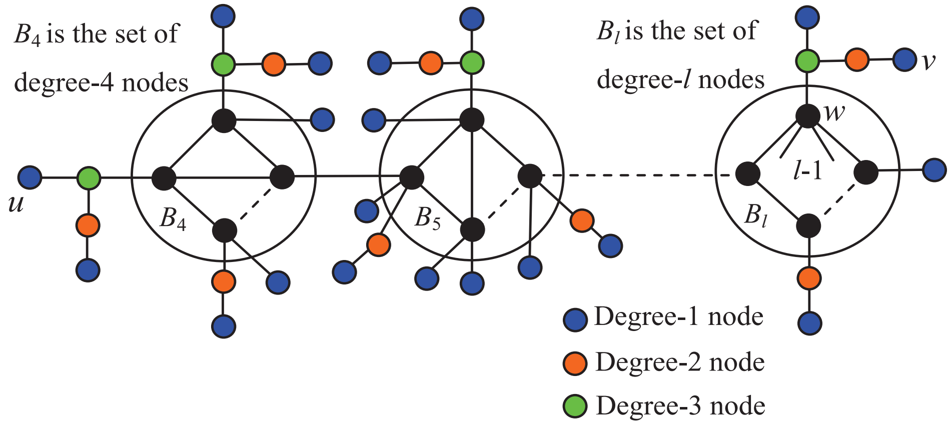

Fig. 1. Illustrating the routing in a scale-free small world network.

4.2. *Routing in peer-to-peer networks*

Peer-to-peer networks, often entailing the construction of a virtual topology or overlay network on a physical network, have been studied extensively over the past decade.[14–16] Setting up a homomorphism between an overlay network Σ and a hypercubic network Γ allows the structural simplicity and high routing efficiency of Γ to produce simple and efficient routing algorithms for Σ. Let us illustrate our method through certain variants of the cube-connected cycles network.

Let $K = Z_q$ $(q \geq 4)$ be an order-q cyclic group, $N = Z_2^q$ be the direct product of order-2 basic group, and $G = Z_2 \nabla Z_q$ be the half-direct product of N and K. The identity element $(0^q, 0)$ and the existence of an inverse for every element in G make G a group. For each vertex identifier (c, r) in G, we call $c \in Z_2^q$ the group identifier and $r \in Z_q$ the region identifier. In order to obtain a Cayley graph $\Gamma = Cay(G, S)$, we specify a subset S of group G that contains two parts, that is, $S = S_g \cup S_r$, where S_g is used to connect different groups and S_r is used to connect different regions.

$$S_g = \{(0^q, \pm 1)\}$$

$$S_r = \{(10^{q-1}, 0),\ (010^{q-2},\ 0),\ (0010^{q-3},\ 0),\ (110^{q-2},\ 0)\}$$

Figure 2 shows the connections between node $(0^q, 0)$ and its neighbors, with solid lines used to depict edges between node $(0^q, 0)$ and its neighbors and broken lines representing edges among the neighbors. Figure 3 shows an example Γ with $q = 4$.

The main parameters of an overlay network include the number of nodes (network size), diameter, node degree, and clustering coefficients. The mean node degree of a graph is the average of the number of neighbors for all nodes and the diameter is the maximum length among all shortest paths. We can derive any node (c, r) from the identity element and the subset q where S is the generating set of G. Thus, Γ is a connected graph. Since the node degree of a Cayley graph is equal to the order of S, the network Γ of Figure 3 is a degree-6 regular graph.

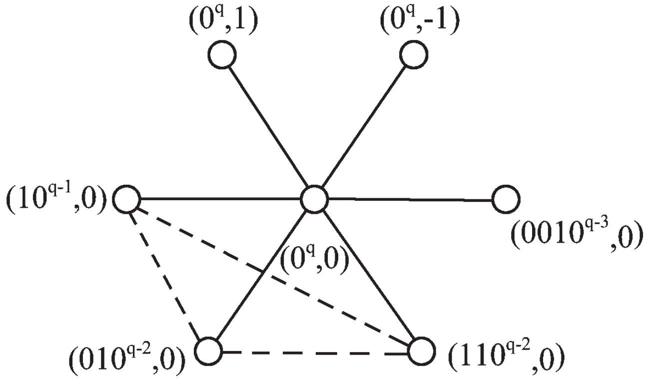

Fig. 2. Connections between node (0^q, 0) and its neighbors.

Based on all neighbors of a node, we obtain a routing algorithm in a static network topology, which is described in Algorithm 1.

For example, by Algorithm 1, the path from source node (0000, 0) to destination node (1111, 3) is: $(0000, 0) \to (1100, 0) \to (1100, 1) \to (1100, 2) \to (1111, 2) \to (1111, 3)$. In general, the routing path from (c_i, r_i) to (c_j, r_j) consists of two segments. First, c_i is changed into c_j in at most $2q$ hops. Then, the region identifier r_i is changed into r_j in at most $q/2$ hops. Therefore, the resulting path goes through at most $5q/2$ hops, which implies the diameter of Γ is at most $5q/2$. For a given q, the number of nodes in Γ is $N = q2^q$, leading to the conclusion that the diameter of Γ is $O(\log N)$. Recall that if the connectivity of a d-regular graph is d, then the graph is maximally fault-tolerant. We know that any 6-regular graph that is vertex-transitive is maximally-fault tolerant.[1–4] Thus, graph Γ is maximally fault-tolerant.

By the foregoing discussion, graph Γ has a small node degree, a simple routing algorithm, and is both maximally fault-tolerant and small-world. Next, we construct a virtual topology or overlay network Σ based on Γ.

A vertex identifier in Σ is an ordered pair (c, r), where $c \in \{x_1x_2 \ldots x_{l*}^{q-l} | x_i \in Z_2, 0 \leq l \leq q\}$ (i.e., a string of 0s and 1s of length l, padded with $_*$s to make the total length equal to q) is the group identifier and $r \in Z_q$ a region identifier in Γ; the parameter q is chosen according to the size of the overlay network. Moreover, key values that need to be stored are mapped onto a 2-element group by using several hash functions, where the bit-width m of group identifiers (generally $m > q$) is a constant chosen such that $q2^m$ is large enough to provide unique identifiers for all needed objects.

The dynamic nature of a P2P overlay networks, with nodes often added or deleted, must be taken into account in any routing algorithm. The deletion of a node can be viewed as the result of merging operation of some nodes in the network Γ. According to the naming rule of identifier space, we can reduce the node-to-node routing problem and the location resource problem to the same problem by omitting the last $m - l$ bits in group identifiers.

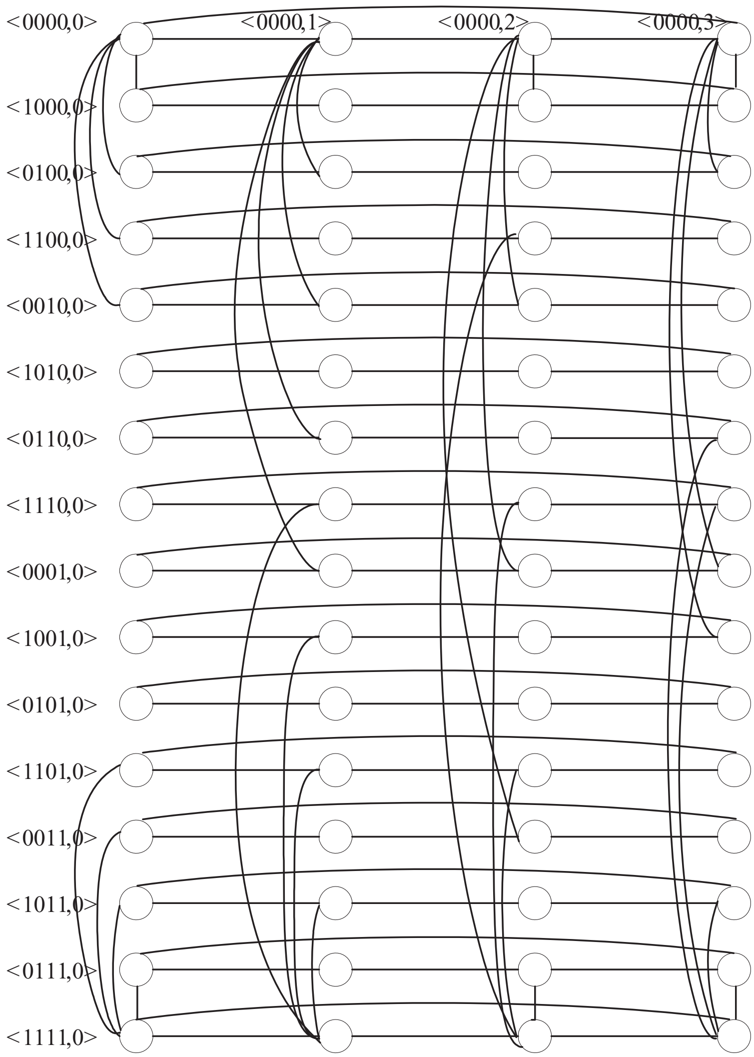

Fig. 3. Graph Γ with $q = 4$.

In the routing algorithm for Σ, two bits in the identifier of the current node are changed in each step, such that the identifier of the node at the next hop is closer to that of the destination node. If by merging identifier of the current node and the identifier of the destination node, the resulting identifier is the same as the identifier of some node in the network, then the routing can be viewed as self-routing. Considering the possibility of faulty nodes, before choosing the next hop, the routing goes back to verify whether the next hop is reachable. If the answer is negative, the routing will choose a normal (non-faulty) neighbor of the current node. In this case, the path will become a bit longer. Algorithm 2 yields the following path from source node $(00_{**}, 0)$ to destination node $(11_{**}, 3)$: $(00_{**}, 0) \rightarrow (11_{**}, 0) \rightarrow (11_{**}, 1) \rightarrow (11_{**}, 2) \rightarrow (11_{**}, 3)$.

Input: The current node $(x_1x_2 \dots x_q, r_x)$ and the destination node $(y_1y_2 \dots y_q, r_y)$
Output: The identifier of the next-hop node on the routing path

Case 1 $(x_1x_2 \dots x_q = y_1y_2 \dots y_q)$:
    if $(r_x = r_y)$ **then**
        The destination has been reached
    else if $(|r_x - r_y| = 1)$ **then**
        Return $(x_1x_2 \dots x_q, r_y \pm 1)$ // + if $r_x = r_y + 1$
Case 2 $(x_1x_2 \dots x_q \neq y_1y_2 \dots y_q)$:
    if $(x_{r_x+1} = y_{r_x+1})$ **then**
        Return $(x_1x_2 \dots x_q, r_x + 1)$
    else
        Return $(x_1x_2 \dots x_{r_x}x_{r_x+1}x_{r_x+2} \dots x_q, r_x)$

Algorithm 1. A routing algorithm in graph Γ.

Input: The current node $(x_1x_2 \dots x_l{*}^{q-l}, r_x)$ and the destination node $(x_1x_2 \dots x_l{*}^{q-l}, r_y)$
Output: The identifier of the next-hop node on the routing path

Case 1 $(x_1x_2 \dots x_l = y_1y_2 \dots y_l)$:
    if $(r_x = r_y)$ **then**
        The destination has been reached
    else
        Return $(x_1x_2 \dots x_l{*}^{q-l}, r_x + 1)$
Case 2 $(x_1x_2 \dots x_q \neq y_1y_2 \dots y_q)$:
    if $(x_{r_x+1} = y_{r_x+1})$ **then**
        Return $(x_1x_2 \dots x_l{*}^{q-l}, r_x + 1)$
    else
        Return $(x_1x_2 \dots x_{r_x}x_{r_x+1}x_{r_x+2} \dots x_l{*}^{q-l}, r_x)$

Algorithm 2. A routing algorithm in graph Σ.

We can consider a variety of cases where nodes are added to or deleted from the network. The details are omitted. Extensive simulations we have conducted show our method to be feasible and effective.[16,20] The examples provided in this section constitute only two instances among a large number of possible applications. In fact, generalized homomorphism is broadly applicable in wireless networks and other domains, which are worthy of further investigation.

5. Conclusion

From the method described in Section 3 and the two application instances of Section 4, we can see the important role of symmetry and mappings. Network symmetry can lead to structural simplicity and algorithmic efficiency. Real networks are usually complicated, prompting us to develop algorithms for them via mapping to networks with simpler structures. To realize this approach, we face the two problems of network selection and network mapping.

By using a theoretical framework emphasizing network symmetry, our work has shown that hypercubic networks provide good solutions to the first problem. As for the second problem, choosing a desirable network mapping is an NP-hard problem in general. The novelty of our method is in the use of generalized homomorphism for taking advantage of network symmetry. Our method is general, in the sense of being applicable to many problems and application domains. Open problems for future work include load balancing, one-to-many broadcasting, and issues specific to wireless networks.

Appendix

This appendix contains three example small-world graphs, along with their degree distributions and number of intercluster edges. In the tables accompanying the figures, k is node degree, n_k is the number of degree-k nodes, and #ICE is the number of intracluster edges.

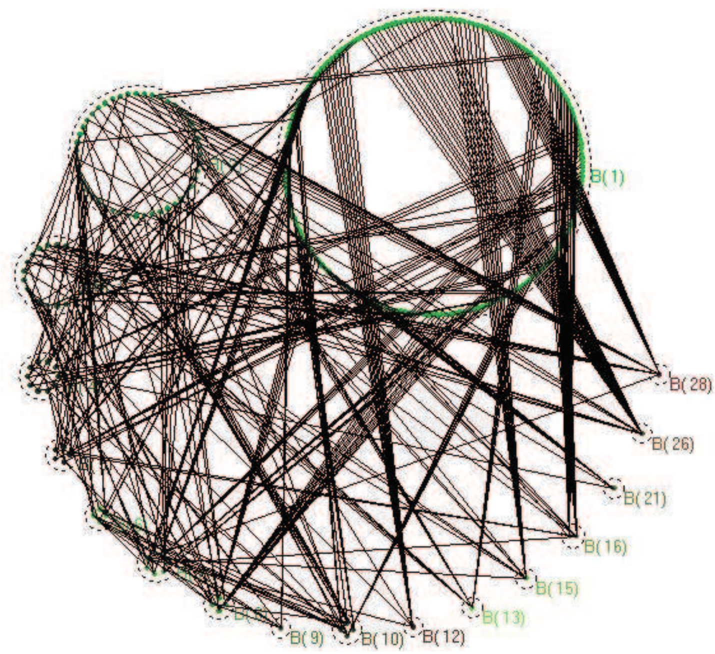

Figure A1. The "elegans" graph, where $B(k)$ is the set of degree-k nodes: the number of degree-1 nodes is 203 and there is a single node of the maximum degree 28. Strictly speaking, the "elegans" graph is not scale-free, but it does come close, given that its degree sequence length $l = 17$ is less than a small power of $\log N = 8.295$.

Table A1. The parameters k, n_k, and #ICE for the "elegans" graph of Figure A1.

k	n_k	#ICE	k	n_k	#ICE	k	n_k	#ICE
1	203	0	7	7	1	15	1	0
2	47	5	8	4	0	16	2	0
3	21	2	9	1	0	21	1	0
4	10	3	10	4	1	26	1	0
5	5	0	12	1	0	28	1	0
6	4	1	13	1	0			

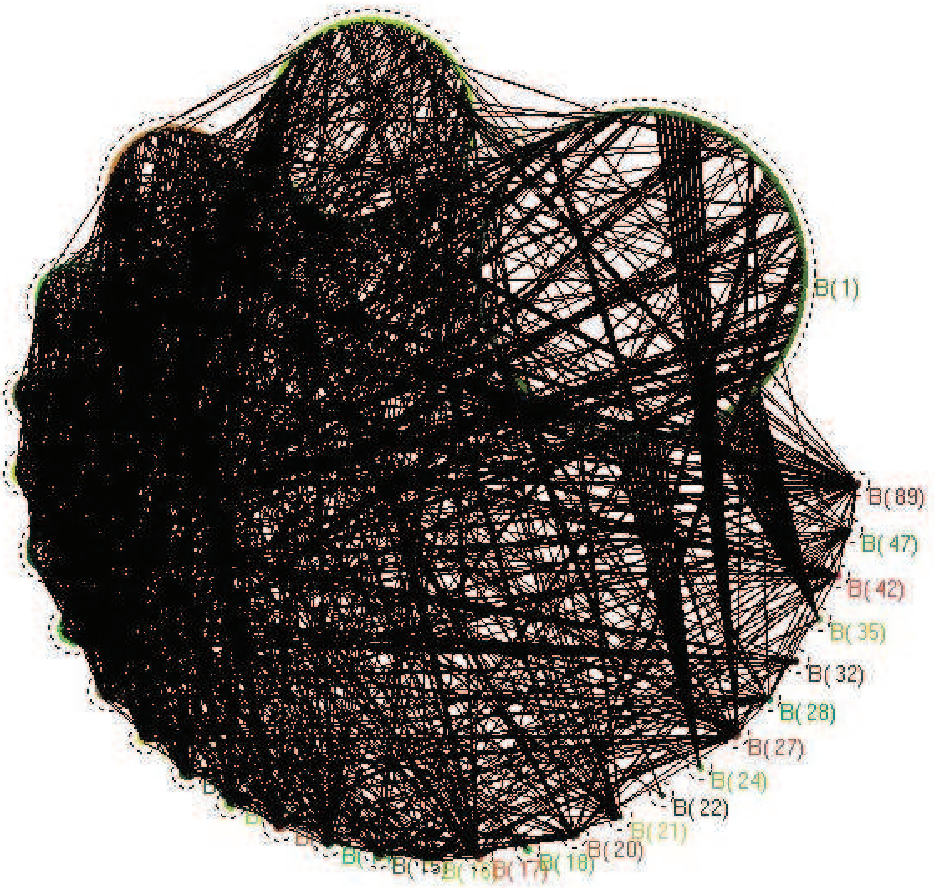

Figure A2. The "abiword" graph, where $B(k)$ is the set of degree-k nodes: the number of degree-1 nodes is 447 and there is a single node of the maximum degree 89. Strictly speaking, the "elegans" graph is not scale-free, but it does come close, given that its degree sequence length $l = 29$ is less than a small power of $\log N = 10.015$.

Table A2. The parameters k, n_k, and #ICE for the "abiword" graph of Figure A2.

k	n_k	#ICE	k	n_k	#ICE	k	n_k	#ICE
1	447	0	11	7	1	22	1	0
2	212	17	12	5	0	24	1	0
3	122	13	13	7	1	27	3	0
4	64	10	14	4	0	28	1	0
5	32	3	15	4	0	32	1	0
6	22	4	16	3	2	35	1	0
7	38	3	17	4	0	42	1	0
8	23	0	18	1	0	47	1	0
9	17	0	20	3	0	89	1	0
10	7	0	21	2	0			

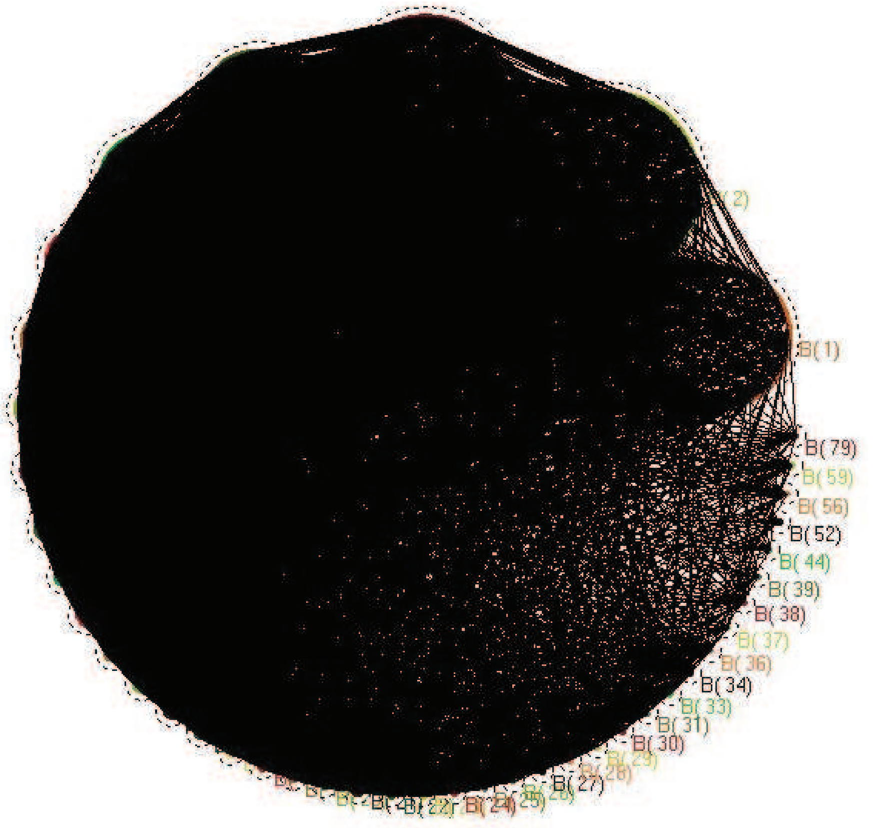

Figure A3. The "ncstrlwg2" graph, where $B(k)$ is the set of degree-k nodes: the number of degree-1 nodes is 894 and there is a single node of the maximum degree 79. Strictly speaking, the "ncstrlwg2" graph is not scale-free, but it does come close, given that its degree sequence length $l = 42$ is less than a small power of $\log N = 12.643$.

Table A3. The parameters k, n_k, and #ICE for the "ncstrlwg2" graph of Figure A3.

k	n_k	#ICE	k	n_k	#ICE	k	n_k	#ICE
1	894	0	15	47	43	29	5	3
2	1328	321	16	38	8	30	4	0
3	1090	463	17	30	5	31	4	0
4	741	400	18	21	5	33	5	1
5	554	331	19	25	16	34	3	0
6	362	181	20	12	4	36	2	0
7	266	163	21	37	146	37	1	0
8	203	120	22	18	1	38	2	0
9	159	71	23	14	8	39	1	0
10	170	152	24	8	1	44	1	0
11	81	24	25	3	0	52	2	0
12	100	63	26	5	0	56	1	0
13	67	81	27	5	1	59	1	0
14	79	142	28	6	0	79	1	0

References

1. F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, 1992.
2. B. Parhami, *Introduction to Parallel Processing: Algorithm and Architectures*, Plenum, 1999.
3. S. B. Akers and B. Krishnamurthy, "A Group-Theoretic Model for Symmetric Interconnection Networks," *IEEE Trans. Computes*, Vol. 38, pp. 555-566, 1989.
4. N. Biggs, *Algebraic Graph Theory, Cambridge University Press*, 1993.
5. M. M. Bae and B. Bose, "Edge Disjoint Hamiltonian Cycles in k-Ary n-Cubes and Hypercubes," *IEEE Trans. Computers*, Vol. 52, No. 10, pp. 1271-1284, 2003.
6. J. Chen, I. A. Kanj, and G. Wang, "Hypercube Network Fault Tolerance: A Probabilistic Approach," *J. Interconnection Network*s, Vol. 6, No. 1, pp. 17-34, 2005.
7. C.-H. Tsai, "Cycles Embedding in Hypercubes with Node Failures," *Information Processing Letters*, Vol. 102, No. 6, pp. 242-246, 2007.
8. Y. Wang, H. Guo, and L. Yuan, "A Routing Algorithm for Multicast on Hypercube Multi-Core Architecture," *J. Embedded Computing*, Vol. 4, No. 1, pp. 23-30, 2010.
9. M. Abd-El-Bar and F. Gebali, "Reliability Analysis and Fault Tolerance for Hypercube Multi-Computer Networks," *Information Sciences*, Vol. 276, pp. 295-318, 2014.
10. A.-L. Barabási, and R. Albert, "Emergence of Scaling in Random Networks," *Science*, Vol. 286, pp. 509-512, 1999.
11. D. J. Watts, and S. H. Strogatz, "Collective Dynamics of 'Small-World' Networks," *Nature*, Vol. 393, pp. 440-442, 1998.

12. R. Albert, and A.-L. Barabási, "Statistical Mechanics of Complex Networks," *Reviews of Modern Physics*, Vol. 74, pp. 47-97, 2002.
13. M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review*, Vol. 45, pp. 167-256, 2003.
14. I. Stoica, R. Morris, D. Liben-Nowell, et. al., "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Trans. Networking*, Vol. 11, No. 1, pp. 17-32, 2003.
15. S. Ratnasamy, P. Francis, M. Handley, et. al., "A Scalable Content Addressable Network," *Proc. ACM SIGCOM Annual Conf.*, pp. 161-172, August 2001.
16. W. J. Xiao and W. B. Wang, "On Symmetry for Network Virtual Topology", *J. Networks*, Vol. 5, No. 6, pp. 732-739, 2010.
17. W. J. Xiao and B. Parhami, "Cayley Graphs as Models of Deterministic Small-World Networks," *Information Processing Letters*, Vol. 97, No. 3, pp. 115-117, 2006.
18. W. J. Xiao, W. D. Chen and B. Parhami, "On Necessary Conditions for Scale-Freedom in Complex Networks, with Applications to Computer Communication Systems," *Int'l J. Systems Science*, Vol. 42, No. 6, pp. 951-958, June 2011.
19. W. J. Xiao, S. Z. Jiang and G. R. Chen, "A Small-World Model of Scale-Free Networks: Features and Verifications," *Applied Mechanics and Materials*, Vols. 50-51, pp. 166-170, 2011.
20. W. J. Xiao, M. X He and H. M. Liang, "CayleyCCC: A Robust P2P Overlay Network with Simple Routing and Small-World Features," *J. Networks*, Vol. 6, No. 9, pp. 1247-1253, 2011.
21. J. Wu and Y. S. Wang, "Hypercube-Based Multi-Path Social Feature Routing in Human Contact Networks," *IEEE Trans. Computers*, Vol. 63, No. 2, pp. 383-396, 2014.
22. J. X. Zhou, "Symmetric Property and Reliability of Balanced Hypercube," *IEEE Trans. Computers*, Vol. 64, No. 3, pp. 876-881, 2015