Scientia Iranica D (2016) 23(6), 2891–2897

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
www.scientiairanica.com

Invited Paper

Families of communication architectures for data centers and parallel processing derived by switching network dilation

B. Parhami*

Department of Electrical & Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA.

Received 17 April 2016; accepted 17 June 2016

KEYWORDS
Communication;
Graph theory;
Interconnection network;
Parallel processing;
Routing algorithm;
Symmetric network.

Abstract. Network dilation is a way of offering system families, at a range of sizes and computational powers, which share an underlying communication architecture and routing algorithm. We consider indirect networks that connect processing nodes via intermediate switch nodes. In the simplest such indirect networks, there is a switching network of some regular topology, where each switch is connected to d other switches and to exactly one processing node. A variant, which we adopt here because it is more robust in the sense of not losing any processing capability to single-switch failures, is the use of 2-port processing nodes that connect to two neighboring switches. This alternate architecture also has the advantage of increasing the number of processing nodes from n to $(d/2)n$ with a factor-of-2 increase in internode distances. A k-dilated version of the latter architecture replaces each processing node with a path network (linear array) of length k, thus growing the network size to $k(d/2)n$ and also further increasing internode distances. In this paper, we study topological and performance attributes of such dilated network architectures, proving general theorems about worst-case and average internode distances and deriving the routing algorithm from that of the underlying switch network.

1. Introduction

The list of proposed interconnection architectures in parallel and distributed systems is rather extensive, as noted by Duato et al. [1], Haddadi et al. [2], Parhami [3], and Xu [4]. Liszka et al. [5] have observed that comparing such networks with respect to their suitability for a particular application domain is quite challenging, given the multitude of static attributes (diameter, average distance, bisection width, VLSI layout area) and dynamic properties (routing algorithms, deadlock prevention, traffic balance, fault tolerance) that must be taken into consideration. Thus, introduction of new interconnection networks, while enriching the repertoire of parallel computer designers, also adds to the selection difficulty.

Interconnection networks can be direct (switches and routers are built into processing nodes) or indirect (a separate switch network connects the processing nodes), with the latter type being more readily scalable and thus preferred in modern parallel computing implementations. The optimal interconnection network depends on the volume of data exchange, expected interchange patterns, and, of course, system size. Given that parallel processors are built in a range of sizes constituting system families, it is unreasonable to expect each member of the family to have a separately optimized network that is incompatible with those of other members. Among other difficulties, such an

*. *Tel.: +1 805 893 3211; Fax: +1 805 893 3262*
E-mail address: parhami@ece.ucsb.edu

approach will lead to the need of different routing and computational algorithms for each member of the system family.

In this paper, we propose a scheme based on path dilation for expanding the system size, in terms of the number of processing nodes, within a broad range, while keeping the same underlying switching-node structure and switch network architecture. This approach allows all members of a computer family to share their communication scheme within a range of system sizes and computational capabilities. After presenting the general method in Section 2, we prove some theoretical results about static and dynamic network attributes in Sections 3 and 4. Variations and extensions are considered in Section 5. The paper is concluded in Section 6.

2. Switch-network architectures and dilated networks

We focus on indirect networks consisting of a separate switch network and a rule for connecting the switches to processing nodes. We use the switch-network architectures depicted in Figure 1 as the bases for our running examples. The network in Figure 1(a) is an 8-node hypercube built of degree-3 switches. The hypercube (Hayes and Mudge [6]) is a well-studied network and has many interesting theoretical and practical attributes that can make routing and other aspects of application development fairly simple. Our second network, shown in Figure 1(b), is derived from K_4, the 4-node complete graph, via the replacement of each node with a cycle of length 3 and assigning each of the original links of that node to one of the nodes in the cycle. This is the same idea that is used in deriving the cube-connected-cycles architecture of Preparata and Vuillemin [7] from the hypercube, but applied to K_4.

The most straightforward way of connecting processing nodes to switches is to dedicate one of the ports of each switch for this purpose, using it to link the switch to an associated processing node, as depicted in Figure 2. With this scheme, an n-node switch network will produce an n-processor parallel system. If the switch network basis has node degree d, diameter D, and average internode distance Δ, the corresponding parameters for the parallel system will be $d + 1$ (for switches, given that processing nodes are all degree-1), $D + 2$, and $\Delta + 2$. Similarly, if the switch network is κ-connected, the resulting architecture will have the same robustness attribute.

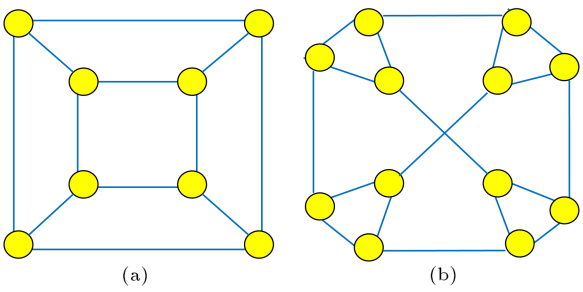

Figure 1. Two switch networks for our running examples in this paper: (a) Eight-node hypercube or 3-cube; and (b) Twelve-node K_4-connected-cycles.

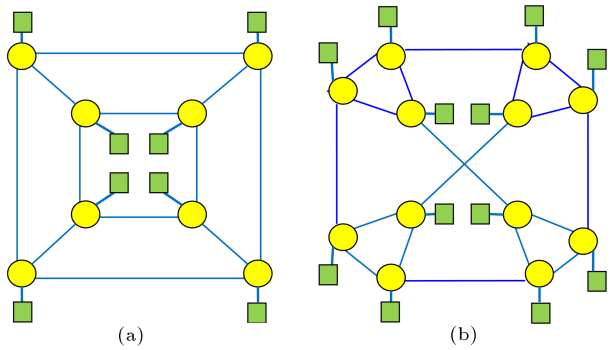

Figure 2. Simplest parallel architecture based on our example switch networks: (a) Eight-node hypercube or 3-cube; and (b) Twelve-node K_4-connected-cycles.

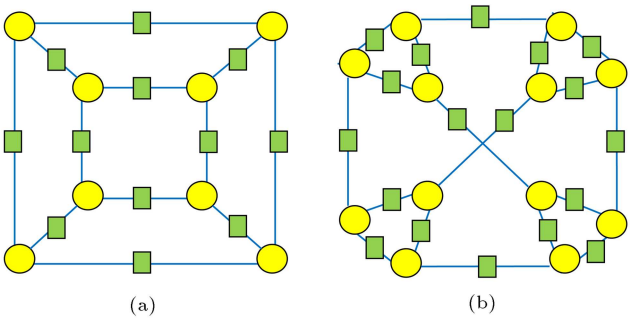

Figure 3. Alternative parallel architecture with 2-port processing nodes: (a) Eight-node hypercube or 3-cube; and (b) twelve-node K_4-connected-cycles.

A second scheme, which may be preferred due to its robustness, in the sense of a single switch node failure not isolating a node, is depicted in Figure 3, where $(d/2)n$ processors connect switch nodes that were linked directly in the base architectures of Figure 1. In this scheme, switch nodes remain of degree d, but the interprocessor diameter and average distance become $\leq 2D + 2$ and $\leq 2\Delta + 2$, respectively. We will take the architectures of Figure 3 as our bases and use the technique of dilation to expand the size, deriving the relevant system parameters in the process.

Dilation means that we string k processors, instead of placing just 1, on the link between two switches. We call the resulting interconnection scheme k-dilated, where the base case of no dilation (or, more accurately, 1-dilation) is represented by Figure 3. Two examples, 3-dialated hypercube-based architecture and 2-dilated K_4-connected-cycles, are shown in Figure 4.

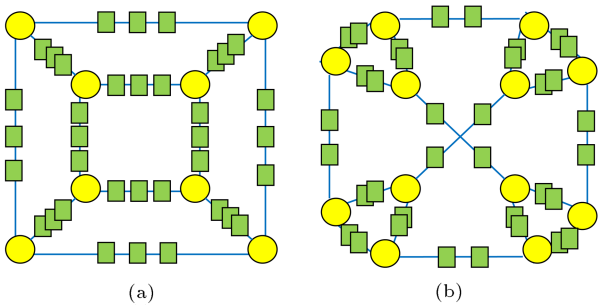

Figure 4. Examples of 3-dilated and 2-dilated interconnection architectures: (a) Eight-node hypercube or 3-cube; and (b) twelve-node K_4-connected-cycles.

Regardless of the value of the dilation parameter k, the switch structure and switch network architecture remain the same, thus allowing the same network to be used for systems of varying sizes. Congestion in the switches is, of course, an issue of concern that we will address later.

For now, we note that the interprocessor diameter and average distance will increase by a factor that is a function of k (more on the relationship later). Thus, this scheme cannot be used for growing the system size indefinitely, as the performance penalty may become unacceptable for very large number $k(d/2)n$ of processing nodes. However, we can envisage growth within a predetermined range for which the architecture has been assessed and fine-tuned. We note in passing here that despite k-dilated paths, the performance penalty is expected to be much less than a factor of k. One reason is that the processors relaying messages make no routing decisions. A processing node takes away a message that is addressed to it and simply forwards all other messages. Pipelined routing through processors leads to very high performance. A second mitigating factor is that the network's aggregate bandwidth grows with an increase in k.

3. Static attributes of dilated networks

As already stated, node degrees do not change with the dilation parameter k. Switch nodes remain of degree d and processor nodes of degree 2, regardless of the network size or total number $k(d/2)n$ of processors. A trivial upper bound on the network diameter is $(k+1)(D_s+1)$, where D_s is the switch network diameter. This is because the length of each of the D_s hops between switch nodes is multiplied by $k+1$ and the initial distance to the first switch node plus distance from the final switch node to the destination processor is no more than $k+1$. We will prove a tighter upper bound shortly. Similarly, $(k+1)(\Delta_s+1/2)$ is a trivial upper bound on the average internode distance, where Δ_s is the average distance of the switch network.

Theorem 1. The diameter of a k-dilated network based on a diameter-D_s switch network is bounded as $(k+1)D_s \le D \le (k+1)D_s + k$. Both bounds are tight in the sense of equality being possible on both sides for suitably chosen networks.

Proof. Referring to Figure 5, a shortest path from source/beginning node B first goes to one of the two switch nodes U_B or V_B; then, it follows a dilated path along the shortest path of the switch network to U_E or V_E on the two ends of the edge containing the destination/end node E, and from there to E. We first note that, in the worst case, all four $U_B \to U_E$, $U_B \to V_E$, $V_B \to U_E$, and $V_B \to V_E$ paths are diametral in the switch network. This is seen in the example of Figure 1(b), where each of the two nodes on the upper left is 3 hops away from each of the two nodes on the lower right. In this case, the shortest path from B to E goes through the closer of U_B and V_B to B and the closer of U_E and V_E to E. For k even, the total distance then becomes $(k+1)D_S + k$ and for k odd, it becomes $(k+1)D_S + k - 1$. In the general case, when $U_B \to U_E$, say, is diametral but the other three paths ($U_B \to V_E$, $V_B \to U_E$, $V_B \to V_E$), may or may not be diametral, we note that $U_B \to V_E$ and $V_B \to U_E$ will be no more than 1 hop shorter and $V_B \to V_E$ is no more than 2 hops shorter. In fact, the hypercube example of Figure 1(a) demonstrates that the $V_B \to V_E$ path may also be diametral, even when $U_B \to V_E$ and $V_B \to U_E$ are not. When the latter two paths are one hop shorter, the shortest distance from B to E corresponds to one of the following 4 paths:

$$P_1 : |B \to U_B \to U_E \to E| = |B \to U_B| + (k+1)D_s + |U_E \to E| = x + (k+1)D_s + y,$$

$$P_2 : |B \to U_B \to V_E \to E| = |B \to U_B| + (k+1)(D_s - 1) + |V_E \to E| = x + (k+1)D_s - y,$$

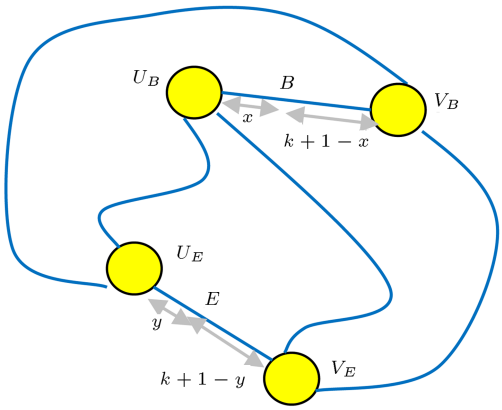

Figure 5. Establishing bounds on the diameter of a dilated network.

$$P_3: \ |B \to V_B \to U_E \to E| = |B \to V_B| + (k+1)$$

$$(D_s - 1) + |U_E \to E| = -x + (k+1)D_s + y,$$

$$P_4: \ |B \to V_B \to V_E \to E| = |B \to V_B| + (k+1)D_s$$

$$+|V_E \to E| = -x + (k+1)D_s + 2(k+1) - y.$$

Clearly, P_2 is always shorter than P_1 and P_3 is always shorter than P_4. Thus, we need only to consider P_2 and P_3 in determining the shortest path. P_2 is shorter than P_3 when $x < y$ and P_3 is shorter than P_2 when $x > y$. Each of the two paths has its worst-case length of $(k+1)D_s$ for $x = y$. Thus, a lower bound on network diameter is $(k+1)D_s$. The upper bound is achieved for networks, such as Figure 1(b), where all 4 paths considered above are diametral in the switch network and k is even. The lower bound is achieved for all values of k in cases where a diametral path in the switch network becomes non-diametral upon changing the source node to a neighboring node of the original source or the destination node to a neighboring node of the original destination.□

We next focus on the average internode distance. Let the average internode distance of the switch network be Δ_s. The average internode distance of the corresponding k-dilated network will be more than $(k+1)\Delta_s$ by an amount that represents the average length of the initial path from the beginning node B to a switch node, plus the average of the path length from the final switch node to the ending node E.

Theorem 2. The average internode distance of a k-dilated network based on a switch network with average internode hop distance Δ_s is $\Delta = (k+1)\Delta_s + k/2 + 1 + (k \bmod 2)/(2k)$.

Proof. Depending on the evenness or oddness of k, the average lengths of the initial and final path segments will be:

$$[1 + 2 + ... + k/2 + k/2 + ... + 2 + 1]/k$$

$$= (k+2)/4; k \text{ even}$$

$$[1 + 2 + 3 + ... + (k-1)/2 + (k+1)/2 + (k-1)/2$$

$$+... + 3 + 2 + 1]/k = (k+2)/4 + 1/(4k); k \text{ odd}$$

Doubling each sum to account for both ends of the path, we get $\Delta = (k+1)\Delta_s + k/2 + 1$ for even k and $\Delta = (k+1)\Delta_s + k/2 + 1 + 1/(2k)$ for odd k.□

Theorem 3. The bisection (band)width B of a dilated network remains the same as the bisection B_s of the switch network.

Proof. When k is even, any bisection of the switch network leads to a bisection of the dilated network by placing the original edge cut between the two middle processors on each edge. When k is odd, we place the cut alternately on one side or the other side of the center processor on each edge. This leads to perfect halving when the number $e = nd/2$ of edges in the switch network is even. Note that both n and d cannot be odd in a regular network, so $nd/2$ is always an integer. If both k and e are odd, the total number ke of processors is also odd and perfect halving becomes impossible. By alternating the cuts for $e-1$ of the edges as before and cutting the last edge on either side of the center node, we get a partition into $(ek-1)/2$ and $(ek+1)/2$ nodes, forming a bisection (by definition in the case of an odd number of nodes).□

4. Dynamic attributes of dilated networks

The bisection-width equality $B = B_s$ can limit scalability, as random communication patterns grow in their required bandwidth with an increase in k, often superlinearly. However, in application domains where communication tends to be mostly local, it is the average aggregate bandwidth and not the bisection (band)width that is important.

Assuming that the bandwidth of each port is b, regardless of whether the port belongs to a switch or a processing node, the network's aggregate bandwidth (the maximum data volume that can be in transit at any given time) is $(k+1)ndb$, if the parameter b denotes bandwidth in each direction of a full-duplex link. With half-duplex links, the aggregate bandwidth is $(k+1)ndb/2$. This bandwidth is more scalable than B, as it grows linearly with the number $knd/2$ of nodes. The required aggregate bandwidth is often a superlinear function of the number of nodes, but in many practical interconnection networks, such as mesh, torus, and any other fixed-degree network, the linear aggregate bandwidth growth is something we accept and deal with. Any network traffic through the processors of a dilated network can be efficiently pipelined, as messages or flits follow one another in an orderly fashion, with no conflicts ever arising between switch nodes.

Fair comparison of aggregate bandwidth between different networks dictates that we divide the figure derived above by the average internode distance. This is because each message will use up more of the bandwidth when the routing distance is greater. Approximating the average internode distance of Theorem 2 for our dilated network by $(k+1)\Delta_s$, we get the bandwidth scalability ratio of about ndb/Δ_s. If the same number $knd/2$ of nodes are interconnected as a square 2D torus of side length $(knd/2)^{1/2}$, the pertinent bandwidth scalability ratio will be $4(knd/2)b/[(knd/2)^{1/2}/2] =$

$8(knd/2)^{1/2}b$, where the square-bracketed term in the denominator is the average internode distance for a square 2D torus and the initial constant 4 is the node degree of a 2D torus. A hypercube with the same number $knd/2$ of nodes will have a bandwidth scalability ratio of $kndb/2$, when the logarithmic node degree and average internode distance are factored in. Therefore, with appropriate choice of a switch network having an associated Δ_s parameter, a dilated network can perform better than a torus, but likely worse than a hypercube, which has a significantly greater cost and much lower physical scalability. Even a torus, with its degree-4 nodes is likely more expensive than a dilated network of the same size.

The conclusions above must be confirmed with simulation studies on reasonably chosen configurations of dilated and ordinary networks, which we have not yet performed. We have reasons to believe, however, that dilated networks will inherit many desirable performance attributes from their underlying switch networks. For example, the dilation process preserves network edge-Hamiltonicity. Therefore, for adaptive routing algorithms that rely on a vertex-Hamiltonian or edge-Hamiltonian path for deadlock avoidance or recovery, no additional effort will be needed when the chosen switch network is vertex- or edge-Hamiltonian. We are in the process of proving that the switch network being a Cayley graph leads to the dilated network also being a Cayley graph. Cayley graphs possess many desirable symmetry and robustness attributes, and are believed to be Hamiltonian in all cases, so proving this property would be quite helpful in ensuring good performance and reliability.

Issues such as deadlock, adaptive routing (to avoid congested routes or for fault tolerance), and the like are also readily addressed as in the original routing algorithm, with no modification. This ability to use a single routing scheme for a family of networks of various sizes is an important advantage of our dilated networks. Despite the observation by Duato [8] that many commercial parallel computers forego the use of adaptive routing algorithms due to their complexities and subtle implementation challenges, it is important that support mechanisms be made available in the architecture to allow adaptive routing when needed.

With processing nodes of degree 2, the maximum connectivity that we can hope for is 2. This maximum connectivity can be achieved with nearly any switching network. All that is needed is for the two of the four paths depicted in Figure 5 that have different end points to be node-disjoint. This property is offered by many different networks.

A key advantage of dilated networks is the direct transfer of a routing algorithm for the switch network to the overall network, with different values of the dilation parameter k. If the algorithm used is optimal (shortest-path) in the switch network, then the resulting algorithm for the dilated network will also be optimal. The only addition to the switch-network routing algorithm is a decision process to choose in which direction to leave B on the dilated path to get to a switch node (U_B or V_B in Figure 5) and which of the two switch nodes U_E and V_E of Figure 5 to use before arriving at E via the associated dilated path.

A simple decision process to always target the nearest switch node works fine, but is slightly suboptimal in some cases. Because a shorter path within the switch network is always advantageous, regardless of the distance from the potential intermediate destinations U_E and V_E to E, as evident from the paths considered in the proof of Theorem 1, we only need to keep a bit vector at every source processor that indicates the direction to take for each possible destination. Such a table is easily compiled and takes a nominal amount of space. Once at a switch node, the message will be sent on the outgoing link dictated by the routing algorithm of the switch network, and the route from U_E or V_E to E is forced (requires no decision).

5. Variations and extensions

Our work can be extended and generalized in many different directions. For example, one can use half of the switch ports to make a switch network and the other half for connecting switches to strings of processors (Figure 6). Switch node degree doubles as a result, if we want to accommodate the same number of nodes with the same dilation factor, but the network diameter becomes $O(k + D)$, instead of $O(kD)$, which represents a big improvement.

Of course, one should pay close attention to the trade-off between cost and performance as well as cost-effectiveness. For example, when we use only half of

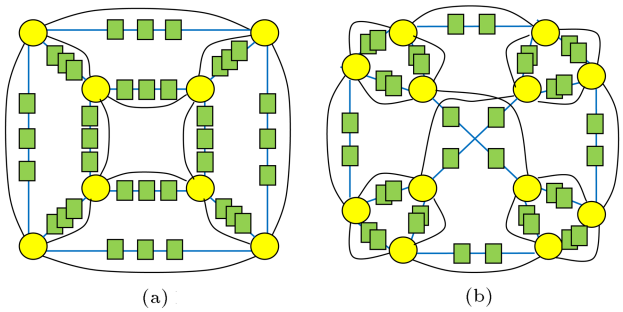

Figure 6. Networks obtained by superimposing a direct network and its dilated version: (a) Eight-node hypercube or 3-cube; and (b) twelve-node K_4-connected-cycles.

the links for placing the processors, we need to place twice as many processors there to end up with the same total network size. The switch network diameter also changes to $D' > D$ when we have half as many links for linking switches together. Therefore, the fair diameter comparison is between $2k + D'$ versus $k + D$. And this is not all! The aggregate network bandwidth and bandwidth scalability factor, and thus expected routing conflicts and delays, will also be different. These are challenging problems to be addressed by future research.

Further extending from our basic configuration discussed here, we can view our network family as comprised of n switch nodes with a total of dn ports, plus various configurations of processor nodes that connect to (a subset of) these ports. In our basic dilated arrangement, switches are not connected to each other, except via strings of processors. In the superimposed architecture of Figure 6, half of the switch ports are connected to each other directly and the other half are linked via strings of processors. Continuing the trend of Figure 6, we can go to the near extreme case where each switch node has only one port available for connection to processing nodes. However, instead of connecting that one port to a single processor, as in the conventional arrangement of Figure 2, we can pool the n free ports together and connect them, for example, to the n free ports of an n^2-node swapped/OTIS [9,10] network. In this way, we have two separate networks linked to each other: a direct processor network and a switch network. Finding the best cross-connection architecture, static parameters, and efficient routing algorithms for such composite networks appears to be a challenging problem.

Still another variation is when exactly two ports of each switch are connected to processing nodes in a dilated configuration. These two ports can be chosen to belong to a Hamiltonian path in the switch network, as depicted in Figure 7. In fact, this particular variation has already been tried by Xiao et al [11] and has been shown to lead to excellent results. These networks are coset graphs and can often be made into Cayley graphs[12], with their desirable properties.

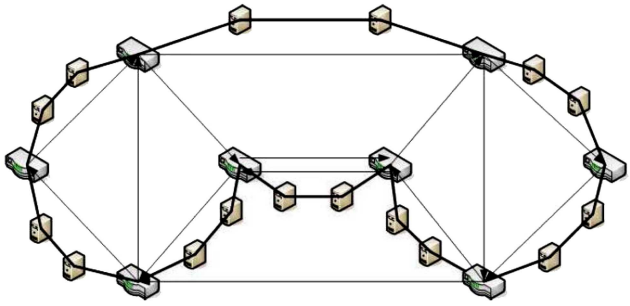

Figure 7. Partially dilated de Bruijn network along one of its Hamiltonian paths [11].

6. Conclusion

The design of interconnection networks for parallel processing has a rich history and many basic choices and attendant variations have emerged over time, as noted by Haddadi et al. [2] and Liszka et al. [5]. In recent years, proposals for data-center networks have emerged, which, while sharing some of the same attributes and performance parameters as interconnection networks for parallel processing, lead to different optimal designs due to the specific communication traffic that they face. In particular, energy efficiency is a major criterion in data centers due to their scale [13,14].

We believe that the criteria for choosing good switch networks that would allow efficient use of the dilation method in building families of parallel computers may turn out to be different from the criteria used for existing direct and indirect parallel-processing networks as well as from data-center networks. Future work must be directed toward identifying network features that bear greater relevance to the overall network efficiency, performance, and robustness, when used with dilation.

We have already discussed some possible directions for further research in Section 4. Designing dilated networks is comprised of the two tasks of choosing a switch network and an associated range $[k_{\min}, k_{\max}]$ of the dilation parameter k. The same overall system sizes can be achieved by a small, dense switch network with large dilations, or by a large, sparse switch network with smaller dilation values. A clear trade-off exists here that must be studied.

As for any other family of designs, the fixed features of the system that are at the heart of the family's compatibility and common capabilities will be optimized for neither the low end nor the high end of the system size. Any reasonable compromise choice will represent an overkill for smaller system sizes and potentially bottleneck-inducing at the higher end. Thus, detailed studies are needed for choosing the architecture and its associated parameters, given a desired system size range $[N_{\min}, N_{\max}]$.

References

1. Duato, J., Yalamanchili, S. and Ni, L., *Interconnection Networks: An Engineering Approach*, Morgan Kaufmann (2002).
2. Haddadi, H., Rio, M., Iannaccone, G., Moore, A. and Mortier, R. "Network topology: Inference,modeling, and generation", *IEEE Communications Surveys & Tutorials*, **10**(2), pp. 48-69 (2008).
3. Parhami, B., *Introduction to Parallel Processing: Algorithms and Architectures*, Plenum (1999).
4. Xu, J., *Topological Structure and Analysis of Interconnection Networks*, Kluwer (2001).

5. Liszka, K.J., Antonio, J.K. and Siegel, H.J. "Problems with comparing interconnection networks: Is an alligator better than an armadillo?", *IEEE Concurrency*, **5**(4), pp. 18-28 (1997).

6. Hayes, J.P. and Mudge, T. "Hypercube supercomputers", *Proc. IEEE*, **77**(12), pp. 1829-1841 (1989).

7. Preparata, F.P. and Vuillemin, J. "The cube-connected cycles: A versatile network for parallel computation", *Commun. ACM*, **24**(5), pp. 300-309 (1981).

8. Duato, J. "Why commercial multicomputers do not use adaptive routing", *IEEE Computer Architecture Technical Committee Newsletter*, pp. 20-22 (1994).

9. Zane, F., Marchand, P., Paturi, R. and Esener, S. "Scalable network architectures using the optical transpose interconnection system (OTIS)", *J. Parallel Distrib. Comput.*, **60**(5), pp. 521-538 (2000).

10. Parhami, B. "Swapped interconnection networks: Topological, performance, and robustness attributes", *J. Parallel Distrib. Comput.*, **65**(11), pp. 1443-1452 (2005).

11. Xiao, W., Liang, H. and Parhami, B. "A class of data-center network models offering symmetry, scalability, and reliability", *Parallel Proc. Lett.*, **22**(4), Article 1250013 (2012).

12. Biggs, N., *Algebraic Graph Theory*, Cambridge University Press, 2nd Ed. (1994).

13. Hamadi, A. and Mhamdi, L. "A survey on architectures and energy efficiency in data center networks", *Computer Commun.*, **40**(1), pp. 1-21 (2014).

14. Bilal, K., Malik, S.U.R., Khalid, O., et al. "A taxonomy and survey on green data center networks", *Future Gen. Computer Syst.*, **36**, pp. 189-208 (2014).

Biography

Behrooz Parhami (PhD in Computer Science from University of California, Los Angeles, 1973) is Professor of Electrical and Computer Engineering, and former Associate Dean for Academic Personnel, College of Engineering, at University of California, Santa Barbara. He has research interests in computer arithmetic, parallel processing, and dependable computing. In his previous position with Sharif (formerly Arya-Mehr) University of Technology in Tehran, Iran (1974-88), he was also involved in educational planning, curriculum development, standardization efforts, technology transfer, and various editorial responsibilities, including a five-year term as Editor of Computer Report, a Persian-language computing periodical. His technical publications include over 280 papers in peer-reviewed journals and international conferences, a Persian-language textbook, and an English/Persian glossary of computing terms. Among his publications are three textbooks on parallel processing (Plenum, 1999), computer arithmetic (Oxford, 2000; 2nd ed. 2010), and computer architecture (Oxford, 2005). Professor Parhami is a Life Fellow of IEEE, a Fellow of IET, a Chartered Fellow of the British Computer Society, a member of the Association for Computing Machinery and American Society for Engineering Education, and a Distinguished Member of the Informatics Society of Iran for which he served as a founding member and President during 1979-84. Professor Parhami has served on the editorial boards of *IEEE Trans. Sustainable Computing* (since 2016), *IEEE Trans. Computers* (2009-14 and 2017), *IEEE Trans. Parallel and Distributed Systems* (2006-10), and *International J. Parallel, Emergent and Distributed Systems* (2006-12). He also chaired IEEE's Iran Section (1977-86), received the IEEE Centennial Medal in 1984, and was honored with a most-cited paper award from *J. Parallel & Distributed Computing* in 2010. His consulting activities cover the design of high-performance digital systems and associated intellectual property issues.