Twisted Torus Topologies for Enhanced Interconnection Networks

José M. Cámara, Miquel Moretó, Enrique Vallejo, Ramón Beivide, *Member*, *IEEE*, José Miguel-Alonso, *Member*, *IEEE Computer Society*, Carmen Martínez, and Javier Navaridas

Abstract—Many current parallel computers are built around a torus interconnection network. Machines from Cray, HP, and IBM, among others, make use of this topology. In terms of topological advantages, square (2D) or cubic (3D) tori would be the topologies of choice. However, for different practical reasons, 2D and 3D tori with different number of nodes per dimension have been used. These mixed-radix topologies are not edge symmetric, which translates into poor performance due to an unbalanced use of network resources. In this work, we analyze twisted 2D and 3D mixed-radix tori that remove the network bottlenecks present in nontwisted ones. Such topologies recover edge symmetry, and consequently, balance the utilization of their links. The distance-related properties of twisted tori together with a full characterization of their bisection bandwidth are described in this paper. A simulation-based performance evaluation has been carried out to assess the network performance under synthetic and trace-driven workloads. The obtained results show noticeable and consistent performance gains (up to an increase of 74 percent in accepted load). In addition, we propose scalable and practicable packet routing mechanisms and wiring layouts for these interconnection systems. The complexity of the architectural proposals is similar to the one exhibited by routing and folding mechanisms in standard tori.

Index Terms—Multiprocessor interconnection, parallel architectures, routing, supercomputers.

1 Introduction

Many parallel computers using direct interconnection networks have been designed and commercialized in last decades. Meshes, tori, and hypercubes have been the most popular topologies. Nowadays, hypercubes have declined in favor of lower degree networks such as two-dimensional and three-dimensional tori and meshes due to its implicit difficulty to scale. Different machines, such as the Alpha 21364-based HP GS1280 [10] and the Cray X1E vector computer [9], use 2D tori. Others, such as the Cray T3D and T3E [23], which preceded the Cray XT3 [7] and XT4 [8], use 3D tori. The IBM BlueGene family of massively parallel computers is a remarkable example of mixed radix 3D tori as the largest configurations of the /L and /P models use $64 \times 32 \times 32$ and $72 \times 32 \times 32$ nodes, respectively, [1], [13].

Usually, 2D tori arrange their N nodes in a square Mesh with $\sqrt{N}$ nodes per side. Sides are connected pairwise by means of $2\sqrt{N}$ wraparound links. Above, more or less, a thousand nodes parallel computers should use 3D topologies as suggested in [2], being a cubic 3D torus of side $\sqrt[3]{N}$ the most desirable solution. However, the number of nodes per dimension might be different, leading to rectangular and prismatic topologies for two and three dimensions, respectively. These topologies, denoted by mixed-radix networks in [11], are often built for practical reasons such as packaging, modularity, cost, and scalability. For instance, the HP GS1280 employs a 2D rectangular torus network [10] and the IBM BlueGene a 3D prismatic one [1]. Mixed-radix tori have two important drawbacks: first, they are not edge symmetric, and second, the distance-related network parameters (diameter and average distance) are far from the optimum values of square and cubic topologies. The edge asymmetry leads to a lack of balance in utilization of resources, and for many traffic patterns, the load on the longer dimensions is higher than the load on the shorter ones [5]. Hence, links in longer dimensions become network bottlenecks. In addition, maximum and average packet delays are relatively long as they depend on the poor values of diameter and average distance exhibited by these networks.

In order to avoid or, at least, mitigate these problems, we analyze in this work alternative mixed-radix 2D and 3D torus topologies that twist their wraparound links of one or two network dimensions. In this way, network symmetry is partially or totally recovered. As we will see, the performance measured on these twisted tori is significantly higher than the one exhibited by their standard mixed-radix counterparts. We concentrate in this paper on aspect ratios 2:1 and 2:1:1 although our results could be extended to other cases. The selected ratios have been previously used by manufacturers [6], [10], allowing the number of nodes to be a power of 2, which is sometimes a desirable property. Even more, these ratios represent the simplest way to upgrade a square or cubic network, doubling their number of nodes by only rearranging some wraparound links.

- *J.M. Cámara is with the Department of Electromechanical Engineering, University of Burgos, Avda. Cantabria s/n, 09006, Burgos, Spain. E-mail: checam@ubu.es.*
- *M. Moretó is with the Department of Computer Architecture, Universitat Politècnica de Catalunya, Jordi Girona, 1-3, Office D6-113, Campus Nord, 08034, Barcelona, Spain. E-mail: mmoreto@ac.upc.edu.*
- *E. Vallejo, R. Beivide, and C. Martínez are with the Departamento de Electrónica y Computadores, Universidad de Cantabria, Avenida de los Castros s/n, 39005 Santander, Spain. E-mail: enrique@atc.unican.es, {ramon.beivide, carmen.martinez}@unican.es.*
- *J. Miguel-Alonso and J. Navaridas are with the Department of Computer Architecture and Technology, University of the Basque Country, P. Manuel de Lardizabal 1, 20018 Donostia-San Sebastian, Spain. E-mail: {j.miguel, javier.navaridas}@ehu.es.*

Manuscript received 14 July 2008; revised 30 Apr. 2009; accepted 2 Sept. 2009; published online 26 Jan. 2010.
Recommended for acceptance by A. Pietracaprina.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-07-0265.
Digital Object Identifier no. 10.1109/TPDS.2010.30.

The main contributions of the paper can be summarized as follows:

1. The proposal of twisted 2D and 3D tori as alternative topologies to mitigate the performance flaws on existing rectangular and prismatic tori.
2. A detailed analysis of their topological properties.
3. A simple routing mechanism for the proposed topologies.
4. A performance evaluation, both theoretical and by means of network simulations, showing that performance increases up to 74 percent.
5. A detailed analysis of the injection rate limit under uniform traffic based on its relation with the network bisection bandwidth.
6. A layout for the proposed networks that keeps link length under a bounded value, eliminating long peripheral wires and facilitating their implementations.

The rest of the paper is organized as follows: Section 2 is about related work. Section 3 analyzes the topological properties of 2D and 3D mixed-radix twisted tori. Section 4 describes routing. Section 5 analyzes the bisection bandwidth and establishes bounds on network throughput under random traffic. Section 6 describes the simulation tools employed and Section 7 provides performance metrics. Section 8 is about network folding and wiring, and finally, Section 9 summarizes the contributions of this research and discusses some future work.

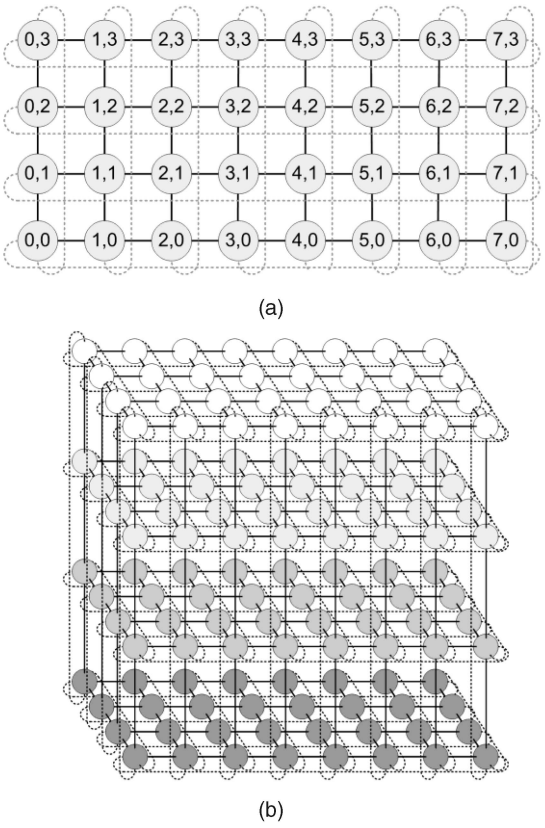

Fig. 1. Regular tori of two and three dimensions, with $a = 4$. (a) RT of size 8×4. (b) PT of size $8 \times 4 \times 4$.

2 Related Work

The idea of twisting 2D square tori in one of its two dimensions for obtaining architectural benefits is not new. A first approach appeared in the ILLIAC IV [4]. This torus-based parallel computer employed a twist of one unit in the wraparound links of dimension X. Such a twist was introduced in order to embed a Hamiltonian ring into the topology, which facilitated some control and data movement operations. Doubly twisted tori were introduced by Sequin in [22] looking for optimal mappings of binary trees onto processor arrays.

Other square meshes with twisted wraparound links were proposed in [3] to reduce maximum and average distance among their nodes. However, the inclusion of a twist in a square torus generates little improvement in terms of network throughput and delay, [3]. Additionally, the twist breaks the natural full symmetry of a square network. Alternatively, the inclusion of two twists (one per dimension) provides better performance, but has collateral drawbacks, such as the presence of edge-asymmetry and the lack of optimal adaptive routing algorithms.

Not too much work has been done in order to improve the performance of rectangular tori by twisting one of its two dimensions. The results introduced in [3] appear to be the first attempt in this direction. More recently, in [10], it was shown that a 4×2-node HP GS1280 computer can offer some extra performance when introducing a "shuffle" in the wraparound links of the longest dimension. Similar topologies were proposed in [26] as a component of hierarchical networks.

The study of 3D prismatic networks is an important extension of the 2D rectangular case. Nevertheless, up to our knowledge, no significant work has been published for optimizing network performance in prismatic 3D tori by twisting one or two of their network dimensions.

3 Topological Properties of Mixed-Radix Tori

This Section analyzes the main distance properties of mixed-radix tori. It is organized into four sections. The first one deals with definitions, the second considers standard 2D and 3D mixed-radix tori, and the third and fourth ones present 2D and 3D twisted tori, respectively.

3.1 Topological Distances and Link Utilization

Rectangular (2D) and prismatic (3D) standard and twisted tori are considered in this work. Such networks lay out their links into orthogonal dimensions. We will consider coordinates (X, Y) in the plane and (X, Y, Z) in the space, as usual. For practical reason, we restrict our attention to $2a \times a$ Rectangular Tori (RT) and $2a \times a \times a$ Prismatic Tori (PT). Standard RT and PT are shown in Figs. 1a and 1b, for $a = 4$. Their twisted versions can be seen in Figs. 2, 3, and 4.

We define next the main topological network parameters. The diameter k is the length of the longest minimum path. The average distance $\bar{k}$ is the average length of all minimum paths. The average distance of the networks considered in this paper can be split into the average distances on each of its dimensions, i.e., $\bar{k} = \bar{k}_x + \bar{k}_y$

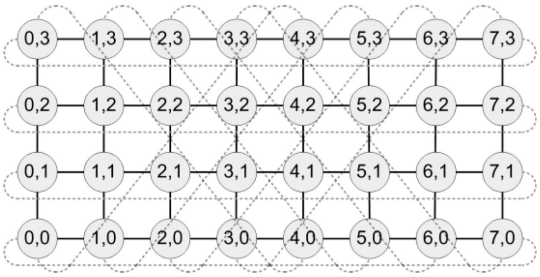

Fig. 2. RTT of size 8×4 (a = 4).

for 2D topologies and $\bar{k} = \bar{k}_x + \bar{k}_y + \bar{k}_z$ for 3D topologies. We denote $\bar{k}_{\max}$ as the maximum of the average distances per dimension, i.e., $\bar{k}_{\max} = \max(\bar{k}_x, \bar{k}_y, \bar{k}_z)$. Finally, Link Utilization, LU, is the average usage of the network links under uniform random traffic at maximum network load; LU = 1 means that all the network links are used every network cycle to convey data. This can be only the case of completely symmetric networks in which their average distance is equally distributed among dimensions. Link Utilization relates with average distance per dimension, according to the following Proposition:

Proposition 1. *In D-dimensional mixed-radix tori, Link Utilization can be computed as follows:*

$$LU = \frac{1}{D}\sum_{i=1}^{D}\frac{\bar{k}_i}{\bar{k}_{\max}} = \frac{1}{D}\frac{\bar{k}_i}{\bar{k}_{\max}}. \quad (1)$$

Proof. Consider first 2D standard $x \times y$ networks, with $x > y$. Note that the number of links per dimension is identical (xy in dimension X and yx in dimension Y) but dimension X is x/y times longer than dimension Y. Hence, the average distance of the path traversed by a random packet in dimension X will be x/y times longer than that of dimension Y. When dimension X uses its links at full capacity, links of dimension Y are used, at most, at a y/x (or $\bar{k}_y/\bar{k}_x$) rate. Consequently, the longest dimension constitutes a network bottleneck that limits the use of the shortest one. The same reasoning applies to higher dimensional standard and twisted networks. Hence, the link utilization, LU, will be the sum of the normalized average distances per dimension, averaged among the network dimensions. Twisted networks obey the same rules but in this case, average distances per dimension are not related with the length of each dimension. □

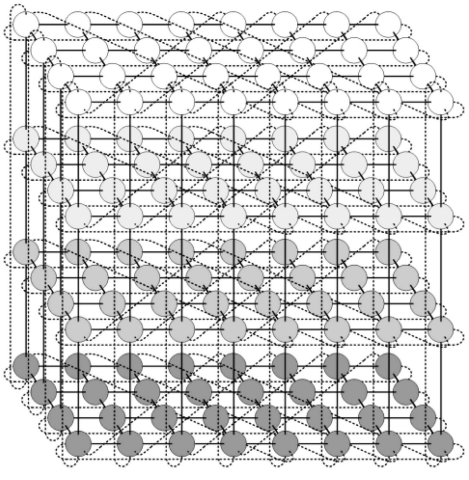

Fig. 3. 3D PTT of $8 \times 4 \times 4$ nodes.

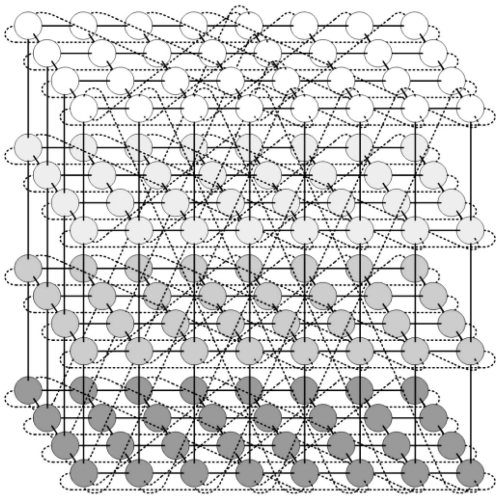

Fig. 4. 3D PDTT of $8 \times 4 \times 4$ nodes.

3.2 Mixed-Radix Standard Tori

RT networks with $2a \times a$ nodes are the Cartesian graph product of two rings of $2a$ and a nodes and PT networks with $2a \times a \times a$ nodes are the Cartesian graph product of a ring of $2a$ nodes and two more rings of a nodes, [14]. By product associativity, a $2a \times a \times a$ PT can also be seen as the Cartesian product of $2a \times a$ RTs times a ring of a nodes. In such graph products, overall distances can be computed as the sum of their distances per dimension. Therefore, to study the distance properties of RTs and PTs, we first consider such properties in an n-node ring. We consider n to be even; having n odd would only slightly modify the resulting values.

Remark 2. The diameter and the average distance of a ring of n nodes are $k = \frac{n}{2}$ and $\bar{k} = \frac{n}{4}$, respectively.

Since a $2a \times a$ RT is the product of two rings of $2a$ and a nodes, respectively, we can straightforwardly infer the following remark (see [14, Lemma 1.37]):

Remark 3. The diameter k, average distance $\bar{k}$, and average link utilization LU of an RT with $2a \times a$ nodes are:

$$k = \frac{2a}{2} + \frac{a}{2} = \frac{3a}{2}; \quad \bar{k} = \frac{2a}{4} + \frac{a}{4} = \frac{3a}{4}; \quad LU = \frac{3a/4}{2 \cdot 2a/4} = \frac{3}{4}. \quad (2)$$

For 3D tori, the following result can be easily obtained since a PT is the outcome of the composition of three sets of orthogonal rings:

Remark 4. The diameter k, the average distance $\bar{k}$, and the average link utilization LU of a PT with $2a \times a \times a$

nodes are:

$$k = \frac{2a}{2} + \frac{a}{2} + \frac{a}{2} = 2a; \quad \bar{k} = \frac{2a}{4} + \frac{a}{4} + \frac{a}{4} = a;$$

$$LU = \frac{a}{3 \cdot 2a/4} = \frac{2}{3}. \tag{3}$$

3.3 Rectangular Twisted Tori

Definition 5. *A $2a \times a$ RTT consists of $2a^2$ nodes arranged in a rectangle with labels (x, y) such that $0 \leq x \leq 2a-1$ and $0 \leq y \leq a-1$. All the inner links in the rectangle form an orthogonal 2D mesh, that is, any node (x, y) such that $0 < x < 2a-1$ and $0 < y < a-1$ is adjacent to the four nodes $(\mathrm{x}+1, \mathrm{y}), (\mathrm{x}-1, \mathrm{y}), (\mathrm{x}, \mathrm{y}+1), (\mathrm{x}, \mathrm{y}-1)$. The wraparound links are defined as:*

- *$(x, 0)$ is adjacent to $(x + a, a - 1)\ \forall x : 0 \leq x \leq a - 1$.*
- *$(x, 0)$ is adjacent to $(x - a, a - 1)\ \forall x : 0 \leq x \leq a - 1\ \ a \leq x \leq 2a - 1$.*
- *$(0, y)$ is adjacent to $(2a - 1, y)\ \forall x : 0 \leq x \leq a - 1\ \ 0 \leq y \leq a - 1$.*
- *(a, y) is adjacent to $(2a - 1, y - a)\ \forall x : 0 \leq x \leq a - 1\ \ 0 \leq y \leq a - 1$.*

Fig. 2 depicts an RTT for the case $a = 4$. Note that RTTs only differ from RTs in the twist of a columns on the vertical wraparound links. We describe next the distance properties of $2a \times a$ RTT networks by referring to its distance distribution $\Omega_{\mathrm{a}}(d)$, or the number of nodes at distance d from a given node.

Proposition 6. *The distance distribution of an RTT with $2a \times a$ nodes is:*

$$\Omega_{\mathrm{a}}(0) = 1;\ \forall d : 0 < d < a\ \ \Omega_{\mathrm{a}}(d) = 4d;\ \ \Omega_{\mathrm{a}}(a) = 2a - 1. \tag{4}$$

This distance distribution was proved in [16]. In that paper, a general family of networks denoted by *Gaussian networks* was presented. RTTs are particular members of this family that are optimum in terms of diameter and average distance, [16]. In addition, the network twist provides them with edge symmetry. Edge symmetry means that all edges incident to a node have an identical view of the rest of the network. Therefore, all the links entering a certain node must belong to orthogonal rings of the same size. This symmetry can be intuitively observed by considering the horizontal and vertical rings that compose the RTT network in Fig. 2. Both rings, in dimensions X and Y, have the same size ($2a$ nodes), whereas the rings in the original RT have different sizes ($2a$ nodes and a nodes, respectively). Then, using Proposition 6 and the network edge symmetry property, we can state the following result:

Corollary 7. *The diameter k, the average distance $\bar{k}$, and the link utilization LU of an RTT with $2a \times a$ nodes are:*

$$k = a; \quad \bar{k} = \frac{\sum_{i=1}^{a} i \cdot \Omega_a(i)}{2a^2} = \frac{4a^2 - 1}{3 \cdot 2a^2} \cdot a \approx \frac{2a}{3}; LU = 1. \tag{5}$$

Therefore, a twist of amplitude a to the vertical $2a$ links of an RT leads to a significant performance improvement. The diameter and average distance of the original RT are 50 and 12.5 percent higher, respectively. The link utilization is 33 percent higher in the RTT. The optimum link utilization of RTTs coming from their edge symmetry implies that average distances among dimensions are equalized, $a/3$ each.

3.4 Three-Dimensional Topologies: Prismatic Twisted and Doubly Twisted Tori

This resource imbalance exhibited by rectangular 2D networks also arises in noncubic, or prismatic, 3D Tori. We employ two different approaches to build 3D networks derived from RTTs. The simplest one with a single twist, denoted by Prismatic Twisted Torus (PTT), can be seen as a stack of RTTs. The second one twists its wraparound links on both Y and Z dimensions, resulting in a Prismatic Doubly Twisted Torus (PDTT).

Definition 8. *A $2a \times a \times a$ PTT consists of $2a^3$ nodes labeled as triplets (x, y, z) such that $0 \leq x \leq 2a-1$ and $0 \leq y, z \leq a-1$. Any node (x, y, z) is adjacent to:*

- *(x', y', z), where (x, y) and (x', y') must be adjacent in a $2a \times a$ RTT, and*
- *$(x, y, z \pm 1 \pmod a)$.*

Fig. 3 depicts a PTT of $8 \times 4 \times 4$ nodes. Note that, by definition, the PTT is built as the Cartesian graph product of a $2a \times a$ RTT and a ring of a nodes. Therefore, the expressions for the distance-related parameters of a PTT can be computed as stated in the following Proposition:

Proposition 9. *The diameter k, the average distance $\bar{k}$, and the average link utilization LU of a PTT with $2a \times a \times a$ nodes are:*

$$k = a + \frac{a}{2} = \frac{3a}{2}; \quad \bar{k} = \left(\frac{a}{3} + \frac{a}{3}\right) + \frac{a}{4} = \frac{11a}{12}; \quad LU = \frac{11a/12}{3 \cdot a/3} = \frac{11}{12}. \tag{6}$$

From a topological point of view, the diameter in the original PT is 33.3 percent higher than that of the PTT, and the average distance is 9.1 percent higher. Furthermore, PTTs provide average distance equalization on X and Y dimensions, leading to a significant improvement on link utilization.

Definition 10. *A $2a \times a \times a$ PDTT consists of $2a^3$ nodes labeled as triplets (x, y, z) such that $0 \leq x \leq 2a-1$ and $0 \leq y, z \leq a-1$. Any node (x, y, z) is adjacent to:*

- *(x_1, y_1, z), where (x, y) and (x_1, y_1) must be adjacent in a $2a \times a$ RTT;*
- *(x, y_2, z_2), where (y, z) and (y_2, z_2) must be adjacent in a $2a \times a$ RTT; and*
- *(x_3, y, z_3), where (x, z) and (x_3, z_3) must be adjacent in a $2a \times a$ RTT.*

Intuitively, this network can be interpreted as a collection of a RTTs in the XY and XZ planes, respectively. Fig. 4 shows horizontal and vertical cuts of a PDTT of $8 \times 4 \times 4$ nodes. Note that a PDTT is built by applying twists to Y and Z wraparound links of a PT, which provides edge symmetry. Although diameter of PDTTs can be inferred using combinatorial techniques, the complexity

to compute its average distance invited us to use an empirical procedure.

Remark 11. We have used a breadth-first search (BFS) algorithm to find the shortest path between nodes in PDTTs. We have obtained that the diameter k, the average distance $\bar{k}$, and the average link utilization LU of a PDTT with $2a \times a \times a$ nodes are:

$$k = \frac{3a}{2}; \quad \bar{k} \approx \frac{7a}{8}; \quad LU = 1. \tag{7}$$

Hence, the diameter in the original PT is 33.3 percent higher than in PDTT, while the average distance is 14.28 percent higher. Finally, X, Y, and Z-dimensional average distances are fully equalized.

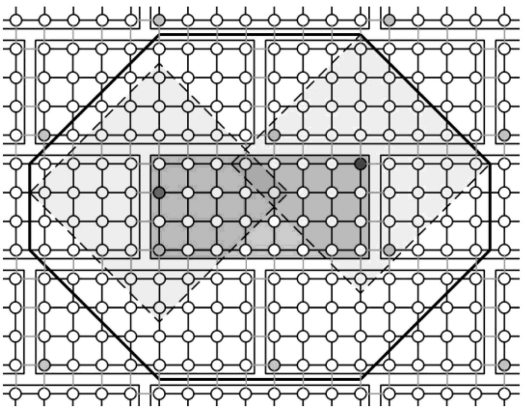

Fig. 5. Possible paths departing from an 8×4 RTT. The original tile (corresponding to nodes 1-32) is shadowed and centered.

4 Network Routing

A minimal routing algorithm to convey packets in rectangular twisted tori is considered next. There are multiple implementations of routing record generator units for RTT networks. Algorithm 1 presents a function to compute a routing header that records the shortest path between any pair of nodes. Such an algorithm is based only on simple operations over node coordinates: sums, subtractions, and comparisons. In addition, all the operations can be done in parallel, minimizing the time required for obtaining the routing record.

Algorithm 1. Routing Record Generator for RTT

input: a : Parameter of the $2a \times a$ RTT.
(s_x, s_y): Source node.
(d_x, d_y): Destination node.
output: $(\Delta X, \Delta Y)$: Routing_record.
begin
DO IN PARALLEL:
begin
$\Delta x_0 := d_x - s_x$; $\Delta x_1 := d_x - s_x - a$;
$\Delta y_0 := d_y - s_y$; $\Delta y_1 := d_y - s_y - a$;
$\Delta x_2 := d_x - s_x + a$; $\Delta x_3 := d_x - s_x + 2a$;
$\Delta y_2 := d_y - s_y - a$; $\Delta y_3 := d_y - s_y$;
$\Delta x_4 := d_x - s_x - 2a$; $\Delta x_5 := d_x - s_x + a$;
$\Delta y_4 := d_y - s_y$; $\Delta y_5 := d_y - s_y + a$;
$\Delta x_6 := d_x - s_x - a$;
$\Delta y_6 := d_y - s_y + a$;
end
$(\Delta X, \Delta Y) := (\Delta x_i, \Delta y_i)$ such that $|\Delta x_i| + |\Delta y_i|$ is minimum.
End

For a generic routing operation, a source node interface will generate a routing record $(\Delta X, \Delta Y)$ heading the packet. ΔX represents the number of links (hops) that the packet must traverse along the axis of the first coordinate, and ΔY along the second coordinate's axis. Their signs indicate East/West and North/South directions. Routers will process the header information in the same way as in a standard torus, decrementing the corresponding field header before sending the packet to the selected neighbor. A packet with $\Delta X = 0$ and $\Delta Y = 0$ has reached its destination and will be consumed. In our experiments, we will employ an optimal fully adaptive routing mechanism built on this basis, [19].

Proposition 12. *The routing in Algorithm I is optimal for RTTs.*

Proof. As these kinds of regular graphs can be fully represented by plane tessellations, the correctness of this routing mechanism can be geometrically proved, [25]. A tile of area $2a \times a$ that tessellates the plane characterizes the graph. Its origin node (left lower corner nodes, highlighted in each rectangle of Fig. 5 for an RTT of 8×4 nodes) defines each tile. Points in the same position of different tiles represent the same network node. As the diameter of the network is a, any source node reaches any other destination node with no more than a jumps, as shown in Fig. 5 with a black border comprising the original tile (shadowed tile in the center of the figure) and its six immediate neighbor tiles. The shaded diamonds in Fig. 5 show the possible destinations for two given source nodes (0, 2) and (7, 3). □

The next part of this geometric proof is based on showing, in terms of a, that the eight sides of the black border only comprise the seven tiles whose left lower corners are the nodes $\{(0,0),(a,a),(-a,a),(-2a,0),(2a,0),(-a,-a),(a,-a)\}$. This implies that the minimal path within any pair of nodes can be found in that tile set. Following this scheme, our routing algorithm computes all the possible paths from a source node to the destination node in each one of the seven possible tiles. After computing these paths' lengths in parallel, the algorithm returns the routing record of the path with minimum length.

With respect to 3D networks, an optimal routing algorithm for PTTs is very simple as we can independently compute the routing for the selected ring of the Z dimension and for the RTT in the selected XY plane. PDTT routing can be obtained using a lightly more complex three-dimensional combinatorial procedure. In this case, with the actual implementation, 23 cases have to be checked in parallel. Notwithstanding, this procedure can admit optimizations.

5 Network Throughput under Uniform Traffic

In this Section, we analyze the performance of mixed-radix networks with respect to their Bisection Bandwidth (B_B), which represents an upper bound in the management of random uniform traffic. We also relate B_B with the network topological properties.

The term B_B stands for the bandwidth between two equal partitions of a network when a minimum cut is applied. According to [11], in networks with uniform channel bandwidth as the ones we consider here, B_B is proportional to the channel count of the minimum network cut. We will show that, while the standard RT and PT behave in this way, for twisted networks, any optimal routing leads to an effective bisection bandwidth lower than the one computed through the minimum cut. We start with standard RTs, which obey the classical definition of B_B.

5.1 Bisection Bandwidth of Rectangular Tori

For a $2a \times a$ RT as the one depicted in Fig. 1, a minimum cut of the network in two halves is the one that divides it into two adjacent $a \times a$ squares. In Fig. 1, this corresponds to the cut that leaves columns 0-3 on one side and 4-7 on the other. The cut is crossed by $2a$ horizontal links (a internal and a wraparound links). This means that $2a$ phits can traverse these links per cycle in each direction (a phit is the physical data unit transferred by a link). Under random uniform traffic, the $N/2$ nodes in one-half of the network generate packets with probability 0.5 to the other half; therefore, a maximum of $\frac{2a}{N/4} = \frac{8a}{N}$ phits/cycle can be generated by each node. For an RT with $N = 2a^2$, the maximum injection rate per node is $\frac{8a}{2a^2} = \frac{4}{a}$ phits/cycle.

5.2 Bisection Bandwidth of Rectangular Twisted Tori

Now, we consider a $2a \times a$ RTT as the one in Fig. 2. The cut separating columns 3 and 4 is again a minimal cut. However, in this case, it comprises 4a links, twice as much as in the RT network. If we applied the classic BB definition, we would find a BB of $\frac{4a}{N/4} = \frac{16a}{N}$ phits/cycle, establishing an injection limit of $\frac{8}{a}$ phits/cycle per node. This is twice the value obtained for the RT. However, as we will see next, this performance figure is not valid as some wires of the network bisection are used to communicate nodes in the same half of the network.

Fig. 6 shows a 16×8 RTT ($a = 8$), with links omitted for simplicity. The dashed square line comprises one-half of the network and represents the minimal cut. A replica tile on the top right corner is partially shown. Routing from node A to node B (both belonging to the left partition) should choose a path going right and up, crossing the partition boundaries twice (from and then to the original partition). The alternative internal path inside the partition is longer, and consequently, the routing algorithm discards it. Hence, traffic internal to each partition crosses the minimum cut.

We define the *Effective Bisection Bandwidth* (EB_B) as the bisection bandwidth that is really used by traffic going from one partition to the other; EB_B depends on the bisection bandwidth and the amount of packets internal to one partition that cross the boundaries. To quantify this value,

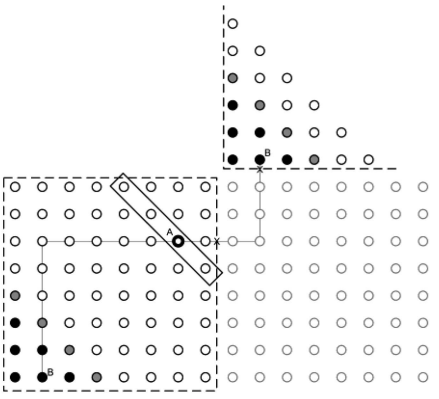

Fig. 6. Path between two nodes in an RTT of size 16×8. Vivid circles on the down left side represent nodes in one-half of the network, while dashed square represents the bisection. Pale circles on the down right side represent nodes in the other half of the network. For the sake of clarity, a fragment of a replica tile is showed in the upright side. Note that the shortest path from A to B crosses the bisection twice.

we initially consider the effect of the replica network, shown on the top right corner of Fig. 6. A message from node A to any of the black nodes in Fig. 6 will find that a destination node in the replica is closer crossing the partition boundaries than using an internal path. The number of black nodes is determined by T_i, which is the *triangular number* for i ($i = 3$ in Fig. 6). The $i + 1$ nodes in gray at distance diameter $k = a$, both in the original and the replica tile, are equally distant from node A. To balance traffic, we assume that internal and external paths are randomly chosen. Finally, the collection of destination nodes being closer in the replica tile than in the original one is the same for any of the $a - 1 - i$ nodes in the same diagonal as A (four nodes in the example Fig. 6 with $i = 3$).

If all nodes in the partition send a packet to nodes inside the partition (uniform traffic), the amount of packets P_1 crossing the boundaries toward the replica tile on the top right corner is given by:

$$P_1 = \sum_{i=0}^{a-2} (a-1-i)T_i + \frac{1}{2}\sum_{i=0}^{a-2} (a-1-i)(i+1). \quad (8)$$

Considering the effect of four replica tiles on the four network corners (not shown in Fig. 6), the amount of packets P that, being internal to one partition, crosses the partition boundaries is:

$$P = 4P_1 = \frac{a^4 - a^2}{6} \approx \frac{a^4}{6}. \quad (9)$$

Thus, if every node sends a packet to any other node in the network, there will be a^4 packets sent from the left partition to the right one (which obviously cross the boundaries), and roughly $2 \cdot \frac{a^4}{6} = \frac{a^4}{3}$ packets which are internal to each partition but crossing twice the bisection boundaries to follow minimal paths. Hence, the traffic crossing the minimum cut is increased from a^4 packets to $\frac{4a^4}{3}$ from which only a^4 are using the bisection to depart from one partition to the other. This means that having $4a$ links in the network bisection, on average, $3a$ links are used for traffic crossing from one partition to the other, while a links are used for traffic internal to the partition.

Hence, for an RTT with $N = 2a^2$, the injection rate per node under random traffic will be limited by $\frac{3a}{N/4} = \frac{12a}{2a^2} = \frac{6}{a}$ phits/cycle/node.

5.3 Injection Rate and Topological Properties

The Effective Bisection Bandwidth metric shows that, under uniform traffic, RTTs improve the maximum injection rate per node with respect to RTs in a 50 percent. This is due to the improvements on two topological properties: average distance and edge symmetry. First, the average distance in the original RT is $\frac{3a}{4}$, while in the RTT, it is $\frac{2a}{3}$, corresponding to a performance improvement of $9/8$. Second, the link utilization in RTs is $3/4$, while in RTTs, it is 1, corresponding to a performance gain of $4/3$. Considering both enhancements, the expected improvement of $\frac{9 \cdot 4}{8 \cdot 3} = \frac{3}{2} = 1.5$ is obtained.

Considering the definitions of link utilization and average distance, it is interesting to make the following derivation:

$$\Delta Inj_{rate} = \Delta\bar{k} \cdot \Delta LU = \frac{\bar{k}_{RT}}{\bar{k}_{RTT}} \cdot \frac{LU_{RTT}}{LU_{RT}} = \frac{\bar{k}_{\max}^{RT}}{\bar{k}_{\max}^{RTT}}. \quad (10)$$

Equation (10) shows that the improvement obtained by the twist is proportional to the reduction of the longest average distance per dimension among all dimensions, which at the end limits performance.

5.4 Bisection Bandwidth and Injection Rate of 3D Networks

Considering 3D networks, the maximum injection rate of a PT is $4/a$ phits/cycle/node. The $\mathrm{EB_B}$ of a PTT can be calculated considering it as a stack of RTTs connecting nodes in Z dimension by rings. This leads to a maximum injection rate of $6/a$ phits/cycle/node that means, as in the 2D case, 1.5 times the maximum PT injection rate. The same result comes from an improvement factor of $12/11$ in average distance reduction and $11/8$ in link efficiency, whose product is again 1.5.

A calculation of the $\mathrm{EB_B}$ for PDTTs, as the one presented in Section 5.2, is quite more laborious. Therefore, we will derive the PDTT maximum injection rate from its topological improvements over a PT, which has been shown to be equivalent to calculate its $\mathrm{EB_B}$. PDTTs present an improvement of $12/7$ (71.43 percent improvement) over the base PT injection rate: a factor of $8/7$ due to the average distance reduction, and a factor of $3/2$ due to the perfect balance in the use of the links. Thus, the maximum injection rate under random traffic is $\frac{48}{7a}$ phits/cycle/node.

6 Experimental Environment

This section is organized into two sections. The first one deals with the simulation platform and performance metrics employed and the second with workloads.

6.1 Simulation Configuration and Performance Metrics

Network evaluations are performed using INSEE, an interconnection network simulator developed by the authors of [21]. It can manage multiple topologies and switching technologies. For this study, the router employed, shown in Fig. 7, is similar to the one implemented in the IBM BlueGene/L: virtual cut-through *switching* strategy, [12], and Bubble flow-control *deadlock avoidance*, [19], with a static virtual channel plus two fully adaptive virtual ones. BlueGene family of supercomputers implements a congestion-control mechanism that prioritizes in-transit traffic against new injections, which is also implemented in our router. In our experiments, packets have a fixed length of 16 phits of 4 bytes each.

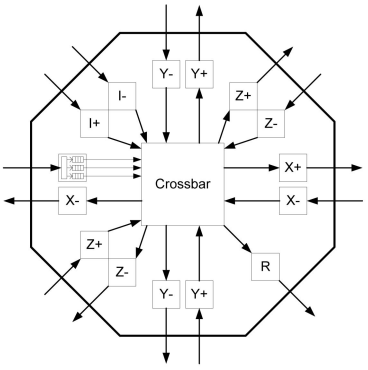

Fig. 7. Model of the simulated routers, with a detailed view of the X+ input port showing the three virtual channels that share this link. The 2D case would not include ports and links for the Z axis.

In our experiments, performance data will be shown in accepted load versus provided load or **throughput**. Provided load is an input parameter, and accepted load is a measured value that indicates how many phits/cycle/node the network is able to successfully deliver. When the network is not saturated, both values are almost identical. However, when some of the routers (or all of them) saturate, actual throughput may be much smaller than applied load. In some cases, we also provide **latency** data in the form of plots of average packet delay versus provided load. We measure the number of cycles between the instant a packet is injected (stored in the input buffer of the source router) until it is consumed at its destination node.

In all the experiments, the internal buffer size has been limited to a practical value. Injection queues have room for, at most, eight packets and transit queues have room for storing up to four packets in each virtual channel.

6.2 Workloads

First, we use independent traffic sources under random uniform traffic (**UN**). In this case, packets are distributed evenly along the whole network so that no persistent bottlenecks are generated. Next, we use random but nonuniform (hot region, **HR**) traffic to check the performance when the network has an unbalanced usage. In HR, 25 percent of the traffic is addressed to the first 12.5 percent of the network (in terms of Cartesian coordinates, i.e., the lowest rows in 2D and planes in 3D) and the remaining 75 percent is uniformly spread along the whole network. In this way, we apply more pressure on those resources used by packets traveling toward the lower rows (planes for 3D). We finish with permutation-based traffic based on [11] (Bit-Complement (**BC**), Bit-Reversal (**BR**), and Perfect Shuffle (**PS**)). In this case, each node sends packets to a given

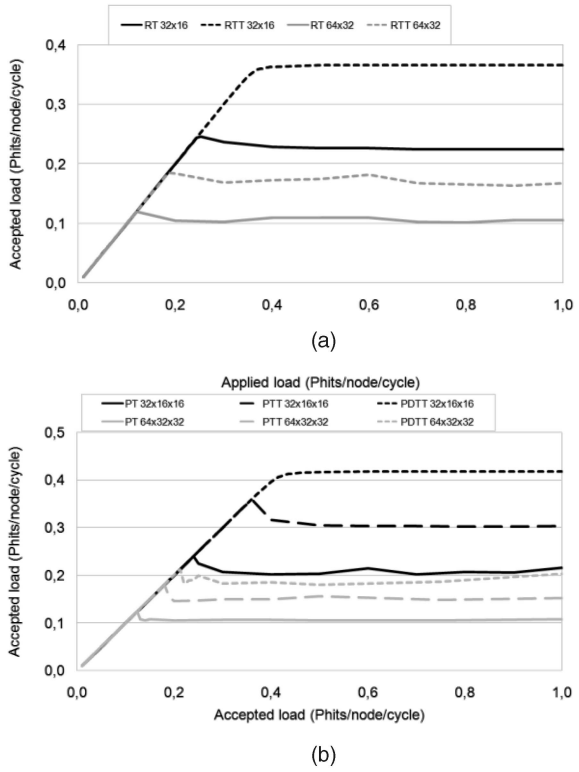

Fig. 8. Throughput for (a) 2D and (b) 3D networks under random uniform traffic.

destination as defined by each of the permutations. For all these workloads, the interinjection interval at each node is random following a Poisson distribution chosen as to modulate the provided load in terms of phits/cycle/node. Other communication pattern of interest included in this study is nearest neighbor communications.

None of the previous workloads considers causality among messages. This assumption, although commonly used in performance studies, is unrealistic because in most current applications, the reception of messages triggers the emission of new ones. Thus, we also evaluate networks with message-passing traces of real applications. These traces have been obtained running a selection of the NAS Parallel Benchmarks (NPBs) MPI applications on an actual multicomputer. Trace-based simulations help us to show that the predicted benefits of mixed-radix twisted tori are also valid for the traffic patterns generated by current applications. We have restricted our study to those benchmarks of the suite that can be mapped onto rectangular networks, and among them, to those that make intensive use of the interconnection network: Conjugate Gradient (**CG**), Integer Sort (**IS**), Fourier Transform (**FT**), LU solver (**LU**), and MultiGrid (**MG**). Limitations on our experimental framework do not allow us to generate traces for applications with more than 128 nodes.

For these workloads, messages are packetized and injected into the network as fast as it can accept them but always obeying causality relationships: if a node is waiting for a message, the simulator stalls its traffic generator until that message arrives [20]. Performance will be measured in terms of the number of cycles required to fully inject and deliver the complete workload.

TABLE 1
Throughput Upper Bound versus Measured Maximum Throughput for 2D and 3D Networks

	32x16		64x32	
	RT	RTT	RT	RTT
Max. Accepted load	0.24548	0.36535	0.11969	0.18497
Theoretical throughput	0.25	0.375	0.125	0.1875

	32x16x16			64x32x32		
	PT	PTT	PDTT	PT	PTT	PDTT
Max. Accepted load	0.24004	0.35943	0.41814	0.1238	0.17996	0.21153
Theoretical throughput	0.25	0.375	0.4286	0.125	0.1875	0.2143

7 Performance Evaluation

This section is organized into five sections, each one dealt with experiments driven by traffic of different nature.

7.1 Random, Uniform Traffic

Figs. 8a and 8b summarize the obtained data for a collection of 2D (32×16 and 64×32) and 3D ($32 \times 16 \times 16$ and $64 \times 32 \times 32$) networks. Injected and accepted loads are identical until reaching the saturation point. Then, even when the simulator tries to inject more traffic, the load actually delivered stabilizes in some cases, and drops to a lower point before stabilizing in others. The reasons for this drop are analyzed in [18].

We expect that the maximum accepted load under random traffic is the one determined by the network effective bisection bandwidth computed in Section 5. Table 1 shows simulated maximum accepted loads versus theoretical upper throughput bounds.

Fig. 9a shows latencies for 64×32 and 32×16 networks, and Fig. 9b does so for $64 \times 32 \times 32$ and $32 \times 16 \times 16$ networks. The average latency grows steadily with injected load until reaching saturation. Beyond that point, it grows boundless. As the average distance is lower in twisted tori, so are their latencies when in a nonsaturated state.

Twisted networks maximize link utilization. Fig. 10 plots the measured link utilization in a 32×16 (a) and a $32 \times 16 \times 16$ (b) networks. For 2D networks, the X channels in the RT are (almost) fully used while the utilization of the Y channels does not reach 50 percent. In the RTT, all channels are close to full utilization. This effect is even more noticeable in 3D networks: the bottleneck at the X channels induces low utilization levels (less than 50 percent) of channels Y and Z. The single twist of the PTT increases link utilization by allowing almost full use of X and Y channels, and higher utilization of Z channels. The PDTT exhibits full channel utilization.

7.2 Random, Hot Region Traffic

The lack of uniformity in this traffic generates an unbalanced utilization of network resources. Figs. 11a and

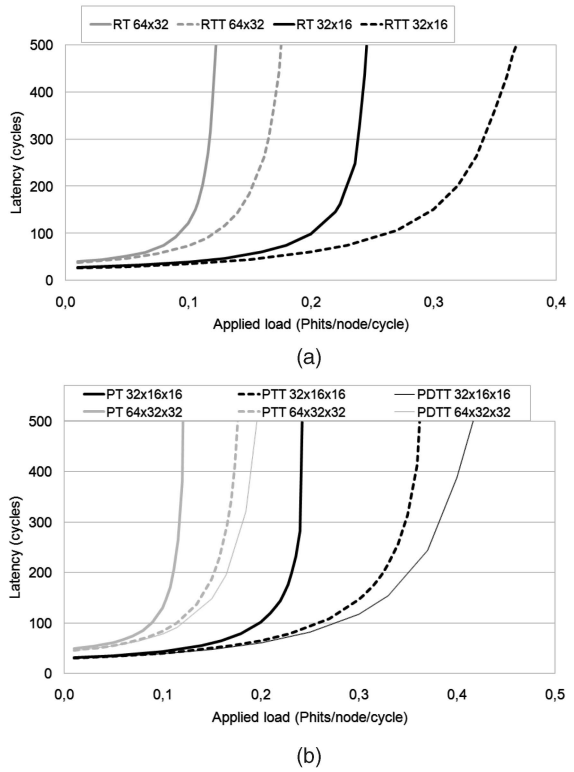

Fig. 9. Average latency under random uniform traffic for (a) 2D and (b) 3D networks.

11b show throughput curves for 2D (32×16 and 64×32) and 3D ($32 \times 16 \times 16$ and $64 \times 32 \times 32$) networks.

In the 2D case, the use of a twist forces the use of the links in the Y dimension for long X displacements, which increases the use of the lowest rows (those saturated by the traffic pattern). For this reason, the obtained results show

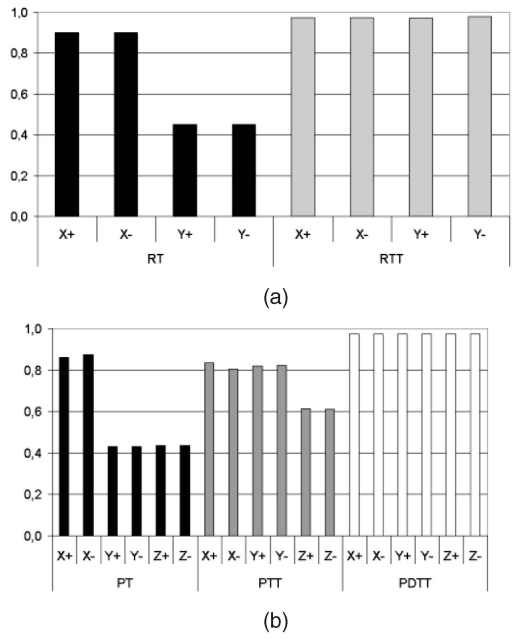

Fig. 10. Average link utilization under random uniform traffic for (a) 2D and (b) 3D networks, with the network saturated.

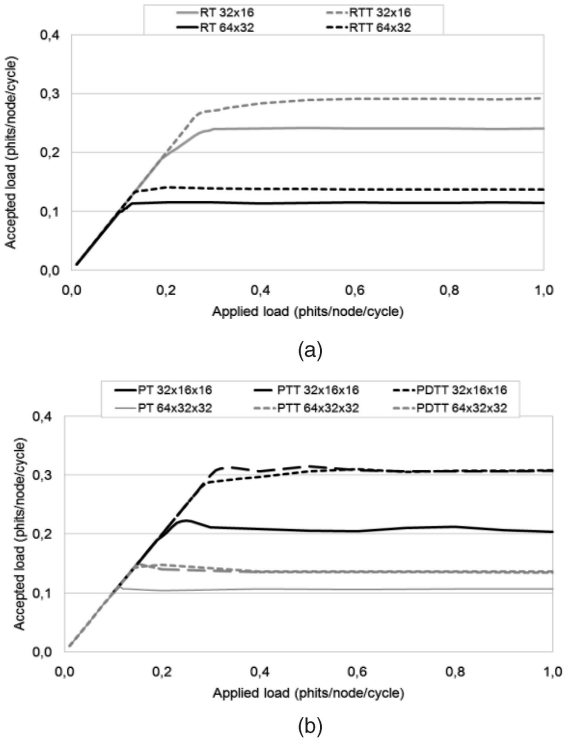

Fig. 11. Throughput under HR traffic for (a) 2D and (b) 3D networks.

that twisted torus outperforms the regular torus but not in the magnitude as when managing uniform traffic.

The introduction of higher pressure in the lowest planes of the 3D networks allows the twisted torus to keep a performance increase over the standard torus similar to the obtained when managing uniform traffic provided that dimension Z is not twisted. When Z dimension became twisted, it is used to shorten paths in the X dimension, and as in the 2D case, the hot region becomes even more saturated due to the extra traffic traveling by Z.

In short, using twists results in a reduction of average distance, and a homogenization of resources for evenly distributed traffic. If the traffic pattern does not make a homogeneous use of the network, it may happen that the topological advantages are not visible.

7.3 Permutation Patterns

Fig. 12a shows performance measurements for the following permutation patterns: Bit-Complement, Bit-Reversal, and Perfect Shuffle [11]. It can be observed that in all cases, the RTT provides a significant improvement with respect to RT, ranging from 24.3 percent (Bit-Complement) to 41.1 percent (Bit-reversal). Fig. 12b shows throughput results on 3D networks under the same traffic patterns. PTT always outperforms PT, with a speedup ranging from 37.1 percent (in Bit-Reversal) to 53.2 percent (in Perfect Shuffle). PDTT improvements over PT are even higher, from 59.7 percent (in Bit-Reversal) to 74.5 percent (in Perfect Shuffle).

7.4 Nearest Neighbor Traffic

Some parallel applications exhibit traffic patterns in which nodes communicate with their nearest neighbors in a torus topology. This can be either due to the inherent symmetry of the application or because of mapping big data matrices

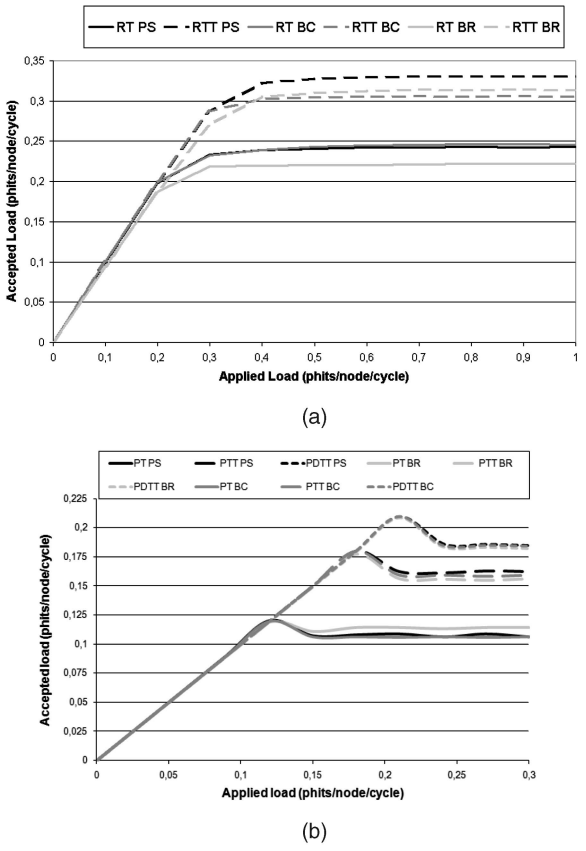

Fig. 12. Throughput for BC, BR, and PS traffic patterns in a (a) 32×16 RT and RTT and (b) $64 \times 32 \times 32$ PT, PTT, and PDTT.

on the network nodes. Twisted tori, contrary to the standard ones, do not provide all the orthogonal wraparound links needed to optimally manage nearest neighbor toroidal communications. We introduce next, through an example, the basis of a simple mapping scheme that removes this apparent drawback in RTTs.

Fig. 13a shows an 8×8 matrix of processes that require toroidal nearest neighbor communications. Fig. 13b shows the proper mapping of such matrix over a 4×8 RTT. The idea lies on mapping the logical Y rings of the application over the twisted physical Y rings of the network. Mapping bigger square matrices only requires a modulo mapping of their columns over the physical twisted Y rings. When mapping processes in this way, all local logical communications are also physically local.

Exceptionally, in some rare cases, small applications with low parallelism may lead a standard torus to behave better than a twisted one under nearest neighbor communication.

7.5 Application Traffic

We consider now a preliminary study of how twisted topologies behave when executing real applications. In most real applications, the way of mapping tasks to nodes has a great impact on performance. The study of application mapping is considered beyond the scope of this paper. Nevertheless, we have tested two simple mapping experiments: consecutive allocation (task i goes to node i) and random, denoted by "_c" and "_r," respectively. For the latter case, graphs show the average of 20 simulation runs.

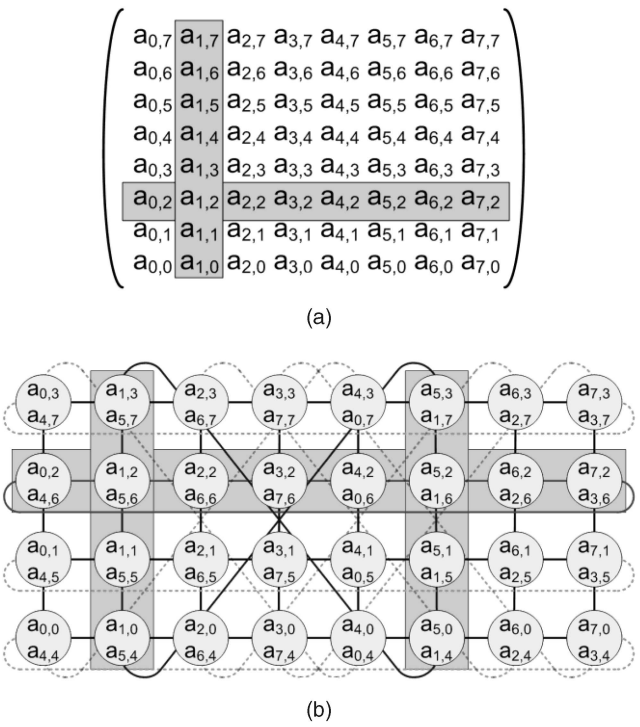

Fig. 13. (a) 8×8 matrix of processes that require nearest neighbor communications and (b) mapping over a 4×8 RTT.

Fig. 14 shows the results obtained when processing the traces of the selected NAS applications.

It can be seen how the RTT performs better than RT for CG (up to 10 percent), FT (up to 18 percent), and IS (up to 20 percent), independently of the placement. Using a consecutive placement in LU, the selection of topology is almost irrelevant. For MG, the twisted alternative is worse than the nontwisted one for consecutive placement but better for random placement. Better mappings could improve these numbers.

8 Wiring Mixed-Radix Twisted Tori

The performance gains exhibited by twisted tori come just from a rearrangement of their wraparound links. Now, we evaluate the technological consequences of the twists on the layout of such networks. We consider first the case of rectangular networks and then the prismatic ones.

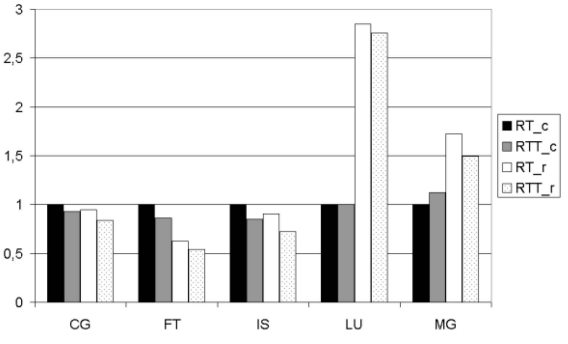

Fig. 14. Total time required to fully inject and deliver the workload with real applications on a 16×8 network.

8.1 Bounded Link Length Layout for 2D RTTs

In tori, the length of the wraparound links grows with the network size. While internal links are supposed to have unitary length, wraparound links grow as $\sqrt{N}$ in square tori. Consequently, real implementations could be negatively affected by this unbalance. The folded torus is a good solution to equalize all the network links by doubling its unitary wire length [11]. The idea is based on applying certain shuffle transformations to rows and columns that interleave the network nodes. Two different shuffle transformations can be considered. Given a row (or column) of n nodes $(0, \ldots, n-1)$, the following transformations map every node location (x, y) onto a different one (x', y) (respectively (x, y')) on the same row (respectively, column). We describe next the two used shuffle transformations when applying to rows:

$$\begin{array}{ll}\text{SHUFFLE A:} & \\ x' = 2x, & \forall x:\ 0 \le x \le n/2, \\ x' = 2n - 2x - 1, & \forall x: n/2 \le x \le n-1, \\ \text{SHUFFLE B:} & \\ x' = 2x + 1, & \forall x:\ 0 \le x \le n/2, \\ x' = 2n - 2x - 2, & \forall x: n/2 \le x \le n-1. \end{array}$$

Analogously to the torus case, we propose a new folding for RTTs, which generates a layout in which link lengths are equalized and bounded by $\sqrt{5}$. The whole folding process is detailed in Algorithm 2. The interested reader can refer to [24] for a formal proof of this algorithm on a similar topology. Fig. 15 shows the resulting *Trellis Folded* layout of an 8×4 RTT. As in the folded torus case, two planes are enough to lay all the links without cutting each other.

Algorithm 2. Trellis Folding Mapping Function

Data: a.	
Step 1: **Initial layout**	Arrange the $2a^2$ nodes in a rows $(0, \ldots, a-1)$ and $2a$ columns $(0, \ldots, 2a-1)$
Step 2: **Row rotation**	For each row i, rotate the row i positions to the right
Step 3: **Column rotation**	For each column i, rotate the column $\left\lfloor \frac{i+1}{2} \right\rfloor$ positions down
Step 4: **Column shuffle**	Apply an A shuffle to even columns and a B shuffle to odd columns.
Step 5: **Row shuffle**	Shuffle all rows according to A shuffle

The previous result is optimal for RTTs as a maximum link length of 2 is impossible to obtain. To see this, consider the case of $2a^2$ nodes arranged in the most compact layout: a $a\sqrt{2} \times a\sqrt{2}$ square. If that were possible, the physical distance among opposite corners would be

$$\sqrt{(a\sqrt{2}-1)^2 + (a\sqrt{2}-1)^2} = 2a\sqrt{1 - \frac{\sqrt{2}}{a} + \frac{2}{a^2}} \approx 2a.$$

The diameter $d = a$ ensures that those nodes are connected by, at most, a links, allowing for a maximum link length of 2 for any a. Any other rectangular layout would present a longer physical distance between two opposite corner nodes, requiring longer links. Being a integer, a square of $2a^2$ nodes is impossible and so is a layout with a maximum link length of 2. Consequently, our folding mechanism leads to an optimal link length of $\sqrt{5}$ for a grid layout.

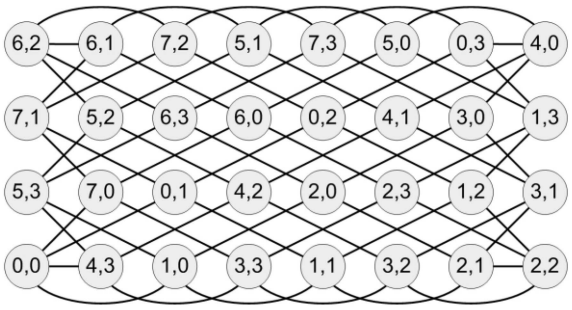

Fig. 15. Trellis folded RTT of size 8×4.

This technique can be also applied in "block mode." As an example, consider a 32×16 RTT, where each 4×4 submesh is a single block. This network can be represented as the basic 8×4 RTT in Fig. 2, but each node becomes in a 4×4 submesh and each link becomes in a group of 4 parallel links joining two such submeshes. After applying the Trellis Folding algorithm shown in Algorithm 2 to the basic 8×4 network, the 32×16 block folded layout is obtained by simply substituting each node by the 4×4 submesh, and each link by the four corresponding parallel links. In general, this method can be applied to any block size.

8.2 Layouts and Cabinet Distributions for 3D PTT and PDTT

Networks for large parallel systems are usually distributed among cabinets. We take, as an example, the system layout of a typical configuration of the BlueGene/L ($64 \times 32 \times 32$ nodes), which is organized as 8×8 cabinets of 1,024 nodes each [6]. Dimension Z evolves inside every pair of cabinets as every cabinet has an $8 \times 8 \times 16$ node configuration. Cabinets are connected in pairs so that every pair contains all of the nodes with different Z for the same (X, Y). Pairs of cabinets are laid on a rectangular array, with a standard folding applied on rows and columns.

Now we analyze PTTs. Considering that a close group of cabinets comprises the whole Z dimension for the (x, y) nodes within it, we can obviate such a dimension. Thus, the folding problem is reduced to the previous bidimensional RTT case. Using the network and cabinet sizes stated above and 8×8 blocks, each pair of cabinets would correspond with a node in a regular 8×4 RTT.

Thus, the Trellis Folding can be directly applied. Fig. 16 shows the layout. Considering the distance between cabinets in the same row or column as the unit length, the maximum link length between cabinets is $2\sqrt{2}$. Vertical labels on cabinets mean x, y, z coordinates of each first node in the block. Note that only links connecting different cabinets have to be modified, thus preserving the internal cabinet connectivity.

In the PDTT case, dimension Z cannot be comprised inside a group of cabinets that share the same (x, y) nodes. Instead, dimension Z is spread between two groups of

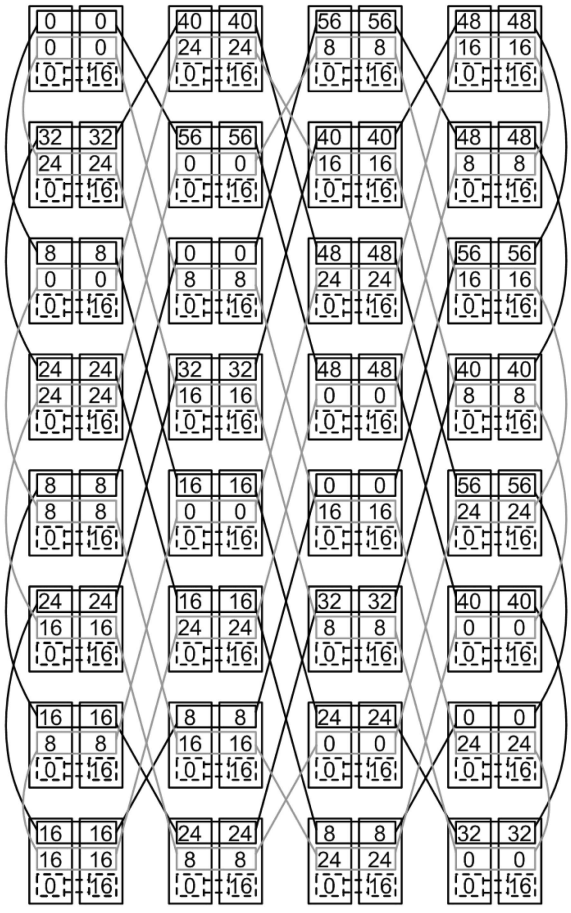

Fig. 16. Trellis folded PTT cabinet distribution.

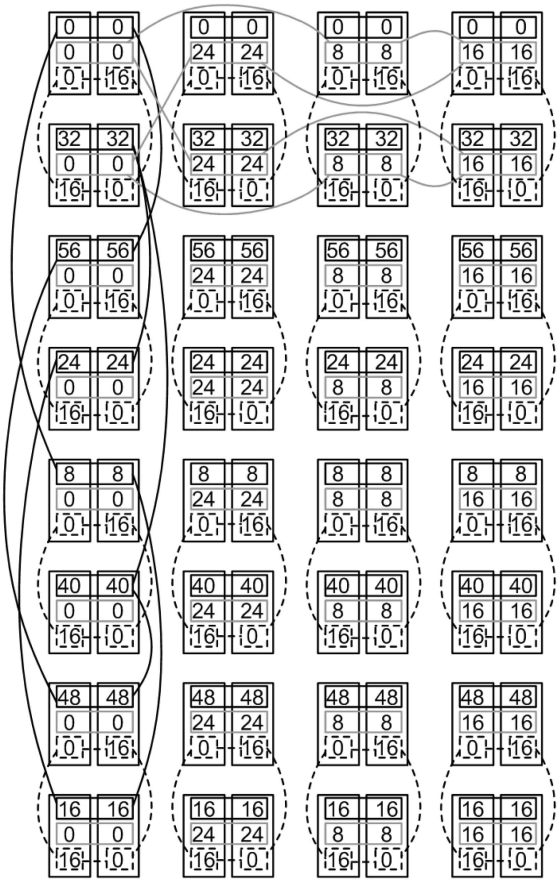

Fig. 17. Folded PDTT cabinet distribution.

cabinets corresponding to different (x, y) nodes. Hence, Z should connect two pairs of cabinets. In this case, we cannot apply the Trellis Folding, as it places these pairs of cabinets in opposite locations. However, another folding technique based on [15] can be applied, which roughly corresponds to applying a shuffle once on Y and twice on X. The resulting layout leaves such pairs of cabinets together, as shown in Fig. 17 (most links are omitted for simplicity), but increases the maximum link length to 4. Note that, as before, only links between cabinets have to be reconnected.

9 Conclusions

Mixed-radix tori present severe communication bottlenecks that negatively affect their performance. These bottlenecks are caused by the asymmetry exhibited by a network that has dimensions of different sizes. In this paper, we have analyzed a class of twisted tori that remove these bottlenecks by equalizing on each dimension the length of the paths traversed by packets.

We have evaluated the performance of twisted tori both analytically and by means of simulation. We have described their main distance-related properties and the relationship between the effective bisection bandwidth and the network performance under uniform traffic. The router model employed for simulations incorporates all the architectural features of current packet routers and resembles the one used in the torus network of the BlueGene/L. The networks have been tested managing both synthetic traffic and workloads from real application traces. In all cases, the twisted topologies showed gains over the standard mixed-radix tori.

The added costs of twisted networks come from both their packet routing mechanisms and their wireability. We have proposed scalable and practicable solutions for both architectural issues. As a conclusion, the proposed topologies appear as an option to improve the overall performance of mixed-radix tori by just rearranging their wraparound links.

Acknowledgments

This work has been supported by the Spanish Ministry of Education and Science (grants TIN2007-68023-C02-01, TIN2007-68023-C02-02, TIN2007-60625, and AP-2005-3318, and CONSOLIDER Project CSD2007-00050), the Basque Government (grant IT-242-07), the European Network of Excellence on High Performance and Embedded Architecture and Compilation (HiPEAC, contract IST-217068), and the SARC European Project (Contract number 27648). Javier Navaridas is supported by a doctoral grant from the UPV/EHU.

References

[1] N.R. Adiga et al. "An Overview of the BlueGene/L Supercomputer," *Proc. ACM/IEEE Conf. Supercomputing (Supercomputing '02) Technical Papers,* Nov. 2002.

[2] A. Agarwal, "Limits on Interconnection Network Performance," *IEEE Trans. Parallel and Distributed Systems,* vol. 2, no. 4, pp. 398-412, Oct. 1991.

[3] R. Beivide, E. Herrada, J.L. Balcazar, and J. Labarta, "Optimized Mesh-Connected Networks for SIMD and MIMD Architectures," *Proc. 14th Ann. Int'l Symp. Computer Architecture,* pp. 163-169, 1987.

[4] W.J. Bouknight, S.A. Denenberg, D.E. McIntyre, J.M. Randall, A.H. Sameh, and D.L. Slotnick, "The Illiac IV System," *Proc. IEEE,* vol. 60, no. 4, pp. 369-388, Apr. 1972.

[5] N.R. Adiga, M.A. Blumrich, D. Chen, P. Coteus, A. Gara, M.E. Giampapa, P. Heidelberger, S. Singh, B.D. Steinmacher-Burow, T. Takken, M. Tsao, and P. Vranas, "Blue Gene/L Torus Interconnection Network," *IBM J. Research and Development,* vol. 49, nos. 2/3, pp. 265-276, 2005.

[6] P. Coteus, H.R. Bickford, T.M. Cipolla, P. Crumley, A. Gara, S. Hall, G.V. Kopcsay, A.P. Lanzetta, L.S. Mok, R.A. Rand, R.A. Swetz, T. Takken, P. La Rocca, C. Marroquin, P.R. Germann, and M.J. Jeanson, "Packaging the Blue Gene/L Supercomputer," *IBM J. Research and Development,* vol. 49, nos. 2/3, pp. 213-248, 2005.

[7] Cray Inc., "Cray XT3 Datasheet," http://www.cray.com/downloads/Cray_XT3_Datasheet.pdf, 2008.

[8] Cray Inc., "Cray XT4 Datasheet," http://www.cray.com/downloads/Cray_XT4_Datasheet.pdf, 2008.

[9] Cray Inc., "Cray X1E Supercomputer," http://www.cray.com/downloads/X1E_datasheet.pdf, 2008.

[10] Z. Cvetanovic, "Performance Analysis of the Alpha 21364- Based HP GS1280 Multiprocessor," *Proc. 30th Ann. Int'l Symp. Computer Architecture,* pp. 218-228, 2003.

[11] W.J. Dally and B. Towles, *Principles and Practices of Interconnection Networks.* Morgan Kaufmann, 2004.

[12] P. Kermani and L. Kleinrock, "Virtual Cut-Through: A New Computer Communication Switching Technique," *Computer Networks,* vol. 3, pp. 267-286, 1979.

[13] IBM, "IBM System Blue Gene Solution," http://www-03.ibm.com/servers/deepcomputing/bluegene.html, 2008.

[14] W. Imrich and S. Klavzar, *Product Graphs: Structure and Recognition.* John Wiley & Sons, Inc., 2000.

[15] F.C.M. Lau and G. Chen, "Optimal Layouts of Midimew Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, no. 9, pp. 954-961, Sept. 1996.

[16] C. Martínez, R. Beivide, E. Stafford, M. Moretó, and E. Gabidulin, "Modeling Toroidal Networks with the Gaussian Integers," *IEEE Trans. Computers,* vol. 57, no. 8, pp. 1046-1056, Aug. 2008.

[17] J. Miguel-Alonso, J.A. Gregorio, V. Puente, F. Vallejo, and R. Beivide, "Load Unbalance in k-ary n-cube Networks," *Proc. Euro-Par Parallel Processing,* pp. 900-907, Springer, 2004.

[18] J. Miguel-Alonso, C. Izu, and J.A. Gregorio, "Improving the Performance of Large Interconnection Networks Using Congestion-Control Mechanisms," *Performance Evaluation,* vol. 65, pp. 203-211, 2008.

[19] V. Puente, C. Izu, J.A. Gregorio, R. Beivide, and F. Vallejo, "Adaptive Bubble Router: A Design to Improve Performance in Torus Networks," *Proc. 28th Int'l Conf. Parallel Computing (ICPP ' 99),* pp. 58-67, Sept. 1999.

[20] J. Miguel-Alonso, J. Navaridas, and F.J. Ridruejo, "Interconnection Network Simulation Using Traces of MPI Applications," *Int'l J. Parallel Programming,* vol. 37, no. 2, pp. 153-174, 2009.

[21] F.J. Ridruejo and J. Miguel-Alonso, "INSEE: An Interconnection Network Simulation and Evaluation Environment," *Proc. Euro-Par Parallel Processing,* pp 1014-1023, Springer, 2005.

[22] C.H. Sequin, "Doubly Twisted Torus Networks for VLSI Processor Arrays," *Proc. Eighth Ann. Int'l Symp. Computer Architecture,* pp. 471-480, 1981.

[23] S.L. Scott and G.M. Thorson, "The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus," *Proc. HOT Interconnects IV Symp.,* 1996.

[24] E. Vallejo, R. Beivide, and C. Martínez, "Practicable Layouts for Optimal Circulant Graphs," *Proc. 13th Euromicro Conf. Parallel, Distributed and Network-Based Processing,* pp. 118-125, Feb. 2005.

[25] C.K. Wong and D. Coppersmith, "A Combinatorial Problem Related to Multimodule Memory Organizations," *J. ACM,* vol. 21, no. 3, pp. 392-402, 1974.

[26] Y. Yang, A. Funahashi, A. Jouraku, H. Nishi, H. Amano, and T. Sueyoshi, "Recursive Diagonal Torus: An Interconnection Network for Massively Parallel Computers," *IEEE Trans. Parallel and Distributed Systems,* vol. 12, no. 7, pp. 701-715, July 2001.

José M. Cámara received the BS degree in telecommunications engineering and the MS degree in electronic engineering from the University of Valladolid, Spain. He received the PhD degree in 2010 at the University of Cantabria. He is an assistant professor at the University of Burgos, Spain. His research interests include interconnection networks in parallel systems and node mapping techniques.

Miquel Moretó received the BS and MS degrees in mathematics and electrical engineering from the Technical University of Catalonia (UPC), Spain, and the PhD degree in 2010 in the Department of Computer Architecture at the same university. His research interests include modeling interconnection networks in parallel systems and studying shared resources in multithreaded architectures.

Enrique Vallejo received the BS and MS degrees in telecommunication engineering (E.E. focused on communications in Spain) from the University of Cantabria in 2005. He received the PhD degree in 2010 at the University of Cantabria. In 2007, he became an assistant professor at the University of Cantabria, where he lectures interconnection networks in the computer science degree. His research interests cover different areas of parallel computing: interconnection networks, kilo-instruction processors, transactional memory, and lock-based synchronization.

Ramón Beivide received the BS and MS degrees in computer science from the Universidad Autonoma de Barcelona in 1981 and 1982, respectively, and the PhD degree in computer science and engineering from the Universidad Politecnica de Catalunya (UPC) in 1985. He was an associate professor at UPC and the Universidad del Pais Vasco. In 1991, he joined the Universidad de Cantabria in Santander, Spain, where he is a full professor of telecommunication engineering and computer science and the dean of the School of Computer Science. His research interests include parallel computers, interconnection networks, memory hierarchies, graph theory, and coding theory. He has published more than 100 technical papers on these topics. He is a member of the IEEE.

José Miguel-Alonso received the PhD degree in computer science from the University of the Basque Country, Gipuzkoa, Spain, in 1996. He is a full professor in the Department of Computer Architecture and Technology at the University of the Basque Country. His research interests include interconnection networks for parallel systems, network (cluster, grid) computing, performance evaluation of parallel and distributed systems, and scheduling for parallel processing. He is a member of the IEEE Computer Society.

Carmen Martínez received the MS degree in mathematics from the University of Cantabria in 2001 and the PhD degree in mathematics from the Department of Mathematics, Statistics and Computers of the same University in 2007. She is currently an assistant professor in the Electronics and Computers Department at the University of Cantabria. Her research interests include error-correcting codes and graph theory.

Javier Navaridas received the MS degree in 2005 and the PhD degree in 2009, both in computer engineering, at the University of the Basque Country UPV/EHU (Gipuzkoa, Spain). His research interests include interconnection networks for parallel and distributed systems, and performance evaluation of parallel architectures, with emphasis on simulation and characterization of application's behavior.