Interconnection networks of degree three obtained by pruning two-dimensional tori

Iain A. Stewart

***Abstract*—We study an interconnection network that we call $3Torus(m,n)$ obtained by pruning the $4m \times 4n$ torus (of links) so that the resulting network is regular of degree 3. We show that $3Torus(m,n)$ retains many of the useful properties of tori (although, of course, there is a price to be paid due to the reduction in links). In particular: we show that $3Torus(m,n)$ is node-symmetric; we establish closed-form expressions on the the length of a shortest path joining any two nodes of the network; we calculate the diameter precisely; we obtain an upper bound on the average inter-node distance; we develop an optimal distributed routing algorithm; we prove that $3Torus(m,n)$ has connectivity 3 and is Hamiltonian; we obtain a precise expression for (an upper bound on) the wide-diameter; and we derive optimal one-to-all broadcast and personalized one-to-all broadcast algorithms under both a one-port and all-port communication model. We also undertake a preliminary performance evaluation of our routing algorithm. In summary, we find that $3Torus(m,n)$ compares very favourably with tori.**

***Index Terms*—interconnection network, torus, degree 3, shortest paths, routing, broadcasting.**

I. Introduction

Interconnection networks are becoming more and more prevalent in computing. Their adoption ranges from the small-scale (in spatial terms), such as in a multi-core processor (e.g., the Tilera TILE 64 multicore processor, with its 64 processor cores), through the medium-scale, such as in a distributed-memory parallel machine or a cluster (e.g., IBM's Blue Gene/Q which can have millions of processor cores), and on to the large-scale, such as in a data centre network (e.g., as used by Google or Amazon and consisting of thousands of servers). The efficiency of any of these computational systems is crucially dependent upon the design and operation of the underlying interconnection network.

No matter how interconnection networks are employed, it is desirable that they possess a variety of specific topological properties, where by a 'topological property' we mean some structural property of the undirected graph that results by abstracting the processing units of the network as vertices or nodes and the inter-unit links of the network as edges. These topological properties impact directly upon key practical aspects of the interconnection network, such as latency, throughput and fault-tolerance, and can be wide-ranging, with the influence of any one of these properties sometimes dependent upon the application to which the host computational system is directed. However, core to almost all performance measures of interconnection networks are the following desirable topological properties. Interconnection networks should:

- be node-symmetric (and, to a lesser degree, link-symmetric) so as to aid: load balancing when designing routing algorithms; parallel programming (the same program can be employed at each node in a distributed-memory parallel machine with the given underlying interconnection network); and theoretical analysis (many different situations requiring analysis can be reduced to a smaller number by applying arguments based on symmetry);
- have low degree so as to lessen hardware implementation costs, associated software complexity and the overheads associated with communication;
- have small diameter and a small average inter-node distance so as to reduce message latency;
- be tolerant of (a limited number of) faulty nodes or links so that their deployment can continue even in the presence of component failures (with this tolerance being in the form of, for example, path redundancy);
- be algebraically concise so that their mathematical descriptions can be utilized in the design and implementation of routing algorithms, flow control and switching methods; and
- possess embeddings of structures such as (Hamiltonian) cycles, paths and trees that are prevalent in parallel programs and so as to aid the implementation of common network operations like one-to-all and all-to-all broadcasting.

We could go on but instead refer the reader to, for example, [1], [14], [15], [22], [24], [42] for more on the theory and application of interconnection networks in a variety of domains. Whilst it is often trivial to build an interconnection network that has some particular property in isolation, designing an interconnection network possessing a range of such properties is extremely difficult (and more often than not impossible); consequently, in practice trade-offs and compromises have to be made.

As regards the choice of interconnection network in practice, the mesh and the torus are probably those that appear most, primarily because of their simplicity allied with relatively good topological properties. However, the mesh suffers from a significant lack of symmetry and a relatively large diameter, and increasingly it is the torus that tends to be more popular (along with its derivations). As technology advances, the dimension of the tori appearing in practice is increasing; for example, the interconnection

I.A. Stewart is with the School of Engineering and Computing Sciences, Durham University, Durham DH1 3LE, UK.
E-mail: see http://www.durham.ac.uk/i.a.stewart

network of IBM's Blue Gene/Q parallel computer [11] is based on a 5-dimensional torus whilst the Tofu interconnection network of Fujitsu's K computer [3] is based on a 6-dimensional torus. Nevertheless, it is 2-dimensional tori that are more common and it is 2-dimensional tori that are our focus here (see, for example, [14] for occurrences of 2-dimensional tori as practical interconnection networks).

Whilst the degree of a 2-dimensional torus is 4, having interconnection networks of degree 3 is often preferable as the lower the node degree, the lower the implementation complexity and cost (for example, fewer wires and ports are required) and the lower the communication overhead (in addition, when an interconnection network in a data centre, say, is composed of commodity switches, sometimes these switches have only 3 ports, or even fewer). Our aim in this paper is to derive an interconnection network that is regular of degree 3 but which is obtained from the 2-dimensional torus by pruning links so that the resulting interconnection network has topological properties that are comparable with those of the 2-dimensional torus. We define the interconnection network $3Torus(m, n)$, where $m, n \geq 1$ and are row and column parameters, that is obtained from the torus with $4m$ rows and $4n$ columns by uniformly pruning selected links so that a network that is regular of degree 3 results. We establish a range of topological properties for $3Torus(m, n)$ concerning, for example, symmetry, the precise lengths of shortest paths, the diameter, connectivity and Hamiltonicity. We also exhibit source routing algorithms and algorithms for one-to-all and personalized one-to-all broadcasting, with these algorithms optimal in both the one-port and all-port models that we study.

We give our basic definitions in Section II, and in Section III we overview a considerable body of research related to existing interconnection networks that are regular of degree 3. In Section IV we prove that the network $3Torus(m, n)$ is node-symmetric, but not link-symmetric (having node-symmetry simplifies our subsequent analysis considerably), and we look at a variety of ways in which it might be constructed. In Section V we establish closed-form expressions for the lengths of the shortest paths between any two nodes of $3Torus(m, n)$, and thus for the diameter of $3Torus(m, n)$. We also obtain an upper bound on the average inter-node distance and we describe an optimal routing algorithm (presented as a source routing algorithm but which can be trivially implemented as a distributed routing algorithm), as well as considering connectivity and Hamiltonicity. In Section VI, we derive optimal one-to-all and personalized one-to-all broadcast algorithms in two models: a one-port model; and an all-port model. We present our evaluation in Section VII, and finally, in Section VIII, we present our conclusions and some directions for further research.

II. Basic definitions

Whilst our interconnection networks are, in fact, undirected graphs, we tend to use the terms 'node' and 'link' rather than 'vertex' and 'edge' in order to emphasise the network context; indeed, we usually refer to a 'graph' as a 'network' for the same reason, although we revert to graph-theoretic terminology when we discuss concepts residing almost exclusively within graph theory. Except where we give explicit definitions of or specific references for such definitions, our terminology and notation is standard and can be found in, for example, [22], [42].

The torus $Torus(m, n)$ has node set $\{(i, j) : 0 \leq i < m, 0 \leq j < n\}$ and link set:

$$\{((i, j), (i, j')) : j' = j \pm 1\} \cup \{((i, j), (i', j)) : i' = i \pm 1\},$$

with addition modulo m or n, as appropriate. Tori are abundant in the study and application of interconnection networks, and their properties in this regard have been extensively investigated (see, for example, [14], [15], [22], [42]).

In order to obtain the interconnection networks relevant to this paper, we prune tori by judiciously removing selected links. The interconnection network $3Torus(m, n)$ has node set $\{(i, j) : 0 \leq i < 4m, 0 \leq j < 4n\}$ (and so it shares its node set with $Torus(4m, 4n)$) and its link set is:

$$\begin{aligned}
&\{((i, j), (i, j')) : j' = j \pm 1\} \\
&\quad \cup \ \{((i, j), (i + 1, j)) : \ i \equiv 0 \pmod 2, \\
&\qquad\qquad\qquad\qquad\qquad\quad j \equiv 0, 1 \pmod 4\} \\
&\quad \cup \ \{((i, j), (i - 1, j)) : \ i \equiv 1 \pmod 2, \\
&\qquad\qquad\qquad\qquad\qquad\quad j \equiv 0, 1 \pmod 4\} \\
&\quad \cup \ \{((i, j), (i - 1, j)) : \ i \equiv 0 \pmod 2, \\
&\qquad\qquad\qquad\qquad\qquad\quad j \equiv 2, 3 \pmod 4\} \\
&\quad \cup \ \{((i, j), (i + 1, j)) : \ i \equiv 1 \pmod 2, \\
&\qquad\qquad\qquad\qquad\qquad\quad j \equiv 2, 3 \pmod 4\},
\end{aligned}$$

with addition modulo $4m$ or $4n$, as appropriate. The interconnection network $3Torus(4, 5)$ can be visualized in Fig. 1 (as a pruned version of $Torus(16, 20)$ and with the node $(0, 0)$ in the centre of the diagram).

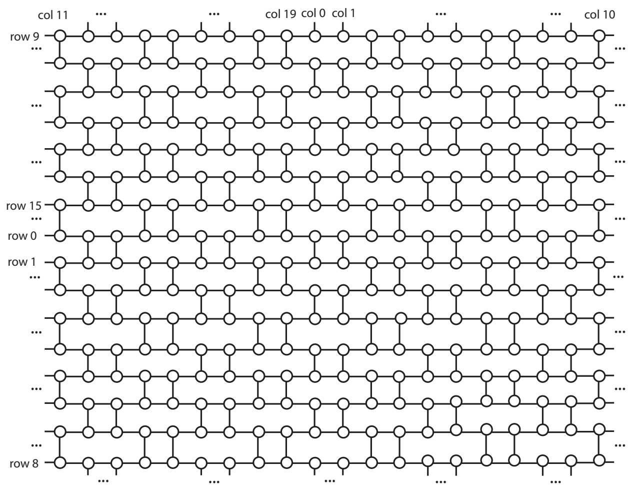

Fig. 1. $3Torus(4, 5)$ (a pruned version of $Torus(16, 20)$).

Note that there are alternative ways in which we might have defined $3Torus(m,n)$. Consider all the cycles of length 4 within $3Torus(m,n)$. If we condense each of these cycles into a single node, as is depicted in Fig. 2, then we obtain a network that can be described as a tessellation of the toroidal surface so that the faces are 'diamonds'. In more detail, this 'diamond' network has node set $\{(i,j) : 0 \leq i < 2m, 0 \leq j < 2n, i+j \equiv 0 \pmod 2\}$ and link set:

$$\{((i,j),(i',j')) \quad : \quad (i',j') = (i,j) + (\epsilon,\delta), \text{ with } \epsilon,\delta \in \{+1,-1\}\}$$

(with all addition componentwise and modulo $2m$ or $2n$, as appropriate). Consequently, we could have started with this 'diamond' network and obtained $3Torus(m,n)$ by amending the 'diamond' network so as to replace each node with a cycle of length 4, similarly to when the cube-connected cycles network is obtained from an n-dimensional hypercube by replacing each node with a cycle of length n (see, for example, [42]). Note that viewing $3Torus(m,n)$ in this way yields an alternative indexing for the nodes, with a node determined by its coordinates (i,j) in the 'diamond' network and a 'tag' of N, S, E or W (denoting 'north', 'south', 'east' or 'west').

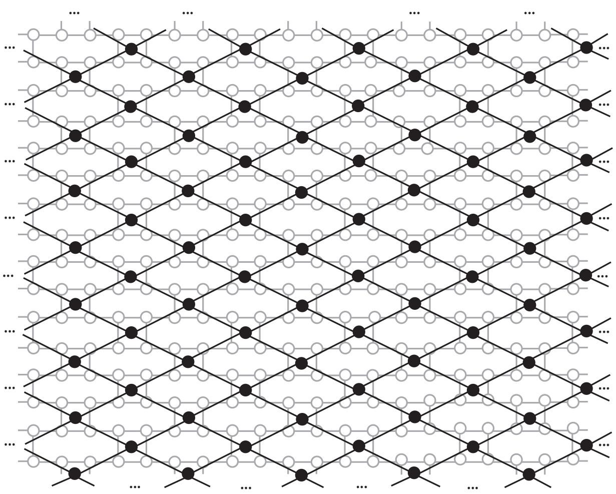

Fig. 2. Condensing 4-cycles in $3Torus(4,5)$ to get a 'diamond' network.

As a matter of fact, the 'diamond' network pictured in Fig. 2 is actually one of the two (isomorphic) connected components of the Kronecker product of a cycle of length 16 and a cycle of length 10. The *Kronecker product* of the graph $G_1 = (V_1,E_1)$ and the graph $G_2 = (V_2,E_2)$ was first defined in [39] and has vertex set $V_1 \times V_2$ and edge set $\{((u_1,u_2),(v_2,v_2)) : (u_1,v_1) \in E_1 \text{ and } (u_2,v_2) \in E_2\}$. In general (when constructing $3Torus(m,n)$ as we did $3Torus(4,5)$ in Fig. 2), the starting 'diamond' network is one of the two connected components of the Kronecker product of C_{4m} and C_{2n}. Such a 'diamond' network can also be realised as a *two-dimensional circulant* or an *L-network* (see [10] for definitions and further details).

Yet another alternative view of $3Torus(m,n)$ is as the network obtained by pruning a *hexagonal torus* $H_{2m,2n}$ (see, for example, [43] for a definition of this network) and then undertaking a 'cube-connected cycles' construction, as above. We can visualize the construction of $3Torus(4,5)$ in this way in Fig. 3 (where the dashed edges are the pruned edges). Finally, we remark that our network $3Torus(m,n)$ has actually arisen as a Brane Tiling in mathematical physics; more precisely, as the *Hirzebruch zero brane tiling* (see Fig. 10 of [20], for example). The upshot is that although $3Torus(m,n)$ arises in this paper as an attempt to obtain a degree-3 interconnection network that broadly retains many of the property of the torus, it clearly has a more wide-ranging significance.

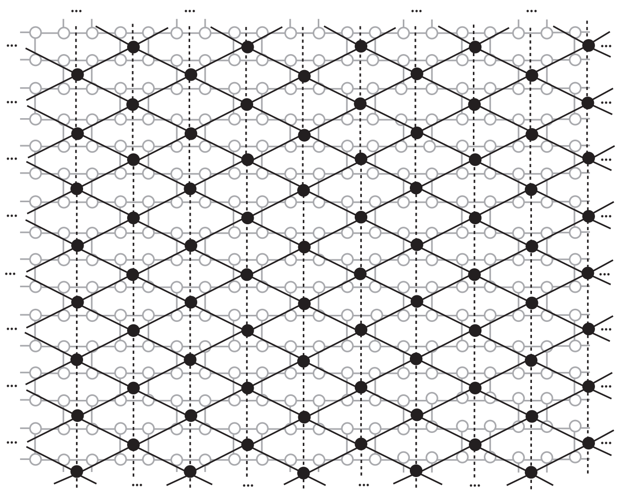

Fig. 3. Constructing $3Torus(4,5)$ from $H_{2m,2n}$.

III. Related work

As we have stated, our intention is to study networks of degree 3 with a view to ascertaining their effectiveness for deployment as interconnection networks. There already exists a significant body of research as regards the study of (the systematic construction of) interconnection networks of degree 3,

Parhami, Kwai and Xiao have undertaken a detailed and systematic consideration of pruning techniques so as to yield interconnection networks that are regular of degree 3. Their thesis has been that if pruning is undertaken with care then the pruned networks inherit many desirable properties from the parent networks. They have especially focussed on Cayley graphs (and the associated algebra) in order to yield pruned networks that inherit various properties, relating to symmetry and inter-node distances, for example, from their parent networks. In [33] Parhami and Kwai applied pruning techniques to 2- and 3-dimensional tori (building on earlier ad hoc instances of research on sub-networks of tori; see [33] for further details of these ad hoc considerations). The resulting networks were the honeycomb networks and the diamond networks, which they proved are Cayley graphs. They also derived results relating to the diameter, the average inter-node distance and the layout of these networks. Incomplete (or pruned) k-ary n-cubes were derived by Parhami and Kwai in [34] and properties relating to symmetry, shortest paths, connectivity and Hamiltonicity were established. Certain pruned 3-dimensional tori were also studied by Xiao and Parhami in [40], and in [41] Xiao and Parhami established general algebraic constructions (based on commutative groups) to develop pruning techniques, which were used to improve known results relating to honeycomb networks and to diamond networks. The algebraic approach of Parhami, Kwai and Xiao was subsequently continued: in [35] where Rahman, Jiang, Masud and Horiguchi applied pruning techniques to the hierarchical torus network and studied properties relating to shortest paths, average inter-node distance, bisection width and VLSI layout area; and in [9] where an algebraic construction related to group semidirect products was developed and used to provide a generalization of earlier pruning schemes. Beyond the concerted research effort described above, there have also been various more isolated considerations of interconnection networks of degree 3, e.g., [6], [13], [25], [37], [44].

Regular graphs of degree 3 (that is, cubic graphs) have also been studied mathematically and in combinatorial chemistry with respect to some of the properties that happen to be of interest in interconnection network design (note that these graphs were not studied as potential interconnection networks *per se*). These studies include, for example, [16], [17], [26], [29]–[31]. Also, the *Foster Census* [8], [12] was an enumeration of symmetric connected cubic edge-transitive graphs of order up to 768. Interestingly, in [27] Hamiltonian cubic graphs from the Foster census were used as interconnection networks for computational clusters undertaking efficient parallel molecular dynamics simulations.

Finally, there is an extensive literature on other low-degree interconnection networks but where the degree is (at least) 4. Noteworthy amongst this literature is the work in [10] and the references therein relating to manipulations of tori but where the tori are not pruned and consequently always have degree 4. The interconnection networks studied in [10] include tori, twisted and doubly twisted tori, toroidal diagonal meshes, chordal rings, and circulant graphs. Other recent considerations of low-degree interconnection networks can be found in [4], [18], where hexagonal mesh networks, Gaussian networks and Eisenstein-Jacobi networks are studied (and the degree of the networks considered tend to be 4 or 6), and in [36] where Spidergon-Donuts of degree 5 are studied.

IV. Symmetry

We begin by proving that the network $3Torus(m,n)$ is node-symmetric. Recall that a network $G = (V, E)$ is *node-symmetric* if given any two nodes u and v, there exists an *automorphism* mapping u to v; that is, a bijection $f : V \to V$ such that if $(u, v) \in E$ then $(f(u), f(v)) \in E$. We do this by constructing a set of basic automorphisms of $3Torus(m,n)$ that can be composed to yield a required automorphism.

Consider the following (node-) maps of $3Torus(m,n)$: $\alpha : (i,j) \mapsto (i, j+4)$; $\beta : (i,j) \mapsto (i+2, j)$; $\gamma : (i,j) \mapsto (i+1, j+2)$; $\varphi : (i,j) \mapsto (i', j)$, where $i + i' \equiv 1$ (mod $4m$); and $\psi : (i,j) \mapsto (i, j'')$, where $j + j'' \equiv 1$ (mod $4n$) (all values and their additions are modulo $4m$ or modulo $4n$ as appropriate). All of these maps are clearly automorphisms: α, β and γ can be thought of as 'translations' horizontally, vertically and diagonally, respectively; and φ and ψ as 'reflections' in a horizontal line between row 0 and row 1 and in a vertical line between column 0 and column 1, respectively. By composing these automorphisms we can clearly map any node to any other node. For example, in order to map $(0,0)$ to $(3,5)$ in $3Torus(4,5)$, we apply the automorphisms: φ (to take $(0,0)$ to $(1,0)$); ψ (to take $(1,0)$ to $(1,1)$); and γ^2 (to take $(1,1)$ to $(3,5)$).

However, $3Torus(m,n)$ is not link-symmetric. Recall that a network is *link-symmetric* if there exists an automorphism mapping any given link to any other given link. To see this, the link $((0,0),(1,0))$ of $3Torus(m,n)$ lies in a cycle of length 4 whereas the link $((1,1),(1,2))$ does not.

Thus, we have proven the following.

Theorem 1: The interconnection network $3Torus(m,n)$ is node-symmetric but not link-symmetric.

We remark here that the network $3Torus(m,n)$ does possess a stronger property than node-symmetry in that it is, in fact, a *Cayley graph* (and consequently its node-symmetry follows immediately). However, a proof of this fact involves combinatorial group theory and will be presented elsewhere.

V. Paths, routing and connectivity

In this section, we look at some other aspects of $3Torus(m,n)$ in relation to its adoption as an interconnection network: the lengths of shortest paths between nodes; the average inter-node distance; its diameter; routing algorithms; its connectivity; its wide-diameter; and its Hamiltonicity.

A. Shortest paths, the diameter and average distances

We begin by investigating the lengths of the shortest paths joining any two nodes in the network $3Torus(m,n)$. By Theorem 1, we may assume that our source node is the node $(0,0)$ and that we wish to find the lengths of the shortest paths from node $(0,0)$ to every other node. As illustrations, the situation for the networks $3Torus(4,5)$ and $3Torus(4,4)$ can be depicted in Fig. 4 and Fig. 5, respectively, and for the upper- and lower-right 'quadrants' of the network $3Torus(4,8)$ in Fig. 6. In these figures, the white nodes are such that the length of the shortest path from node $(0,0)$ (the black node) is identical to the length of a shortest path in $Torus(16,20)$, $Torus(16,16)$ or $Torus(16,32)$ (as appropriate), otherwise the integer labelling a node is the amount by which the length of a shortest path from $(0,0)$ in the particular network $3Torus(4,5)$, $3Torus(4,4)$ or $3Torus(4,8)$ differs from the length of a shortest path from $(0,0)$ to the node in the corresponding torus. A useful fact to bear in mind when calculating these labels is that if in $3Torus(m,n)$ the link (u,v) is such that node u lies on a shortest path from $(0,0)$ to node v in $Torus(4m,4n)$ and $+i$ labels u then $+i$ must also label v, where i is some integer.

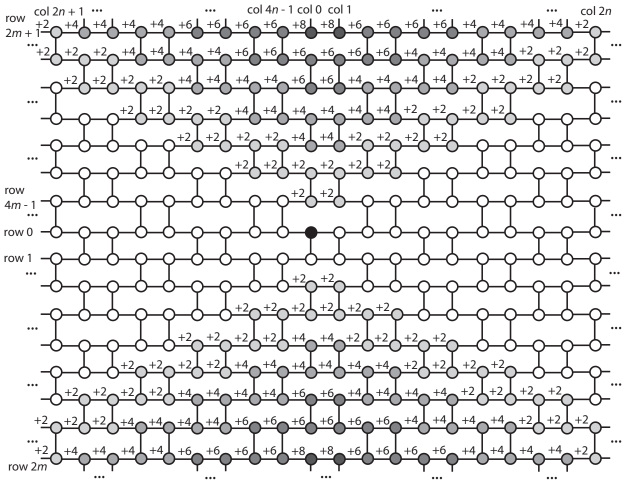

Fig. 4. Comparative lengths of shortest paths in $3Torus(4,5)$.

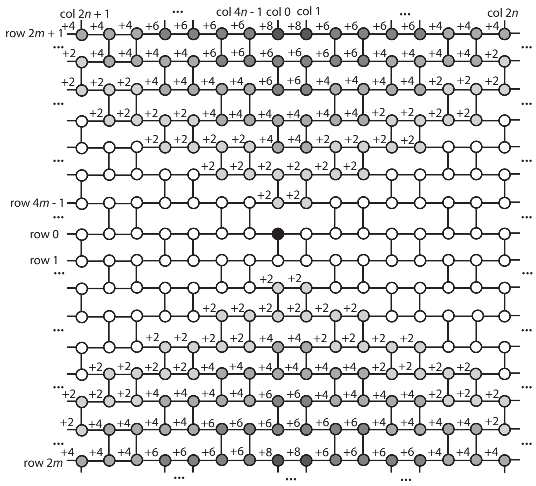

Fig. 5. Comparative lengths of shortest paths in $3Torus(4,4)$.

No matter which network $3Torus(m,n)$ we are working with, the general pattern and construction of white and labelled nodes is similar (with variations only being down to the relative differences between m and n). This construction should be obvious from Figs. 4–6: we start at node $(0,0)$, with the labellings of nodes that are 'nearby' as depicted in these figures, and 'work outwards' until we reach the boundaries. Note that in $3Torus(m,n)$ and $3Torus(m+a,n+b)$, where $a>0$ and $b>0$, any node (i,j) in $3Torus(m,n)$ is labelled identically to node:

- (i,j) in $3Torus(m+a,n+b)$, if $0 \leq i \leq 2m$ and $0 \leq j \leq 2n$;
- $(i,j+4b)$ in $3Torus(m+a,n+b)$, if $0 \leq i \leq 2m$ and $2n+1 \leq j \leq 4n-1$;

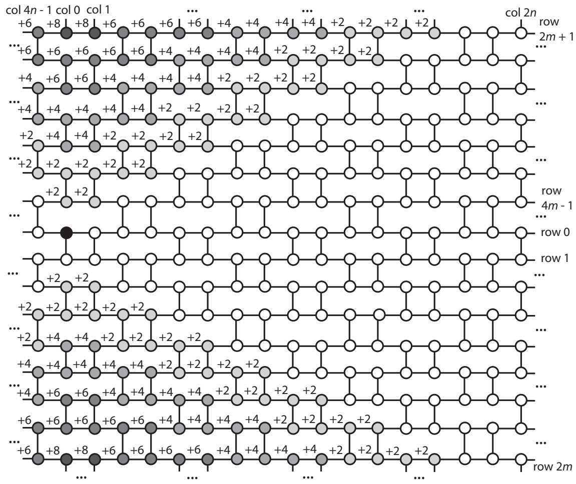

Fig. 6. Comparative lengths of shortest paths in $3Torus(4,8)$.

- $(i+4a,j)$ in $3Torus(m+a,n+b)$, if $2m+1 \leq i \leq 4m-1$ and $0 \leq j \leq 2n$; and
- $(i+4a,j+4b)$ in $3Torus(m+a,n+b)$, if $2m+1 \leq i \leq 4m-1$ and $2n+1 \leq j \leq 4n-1$.

Given the transparency of the pattern and the uniformity of its construction, we do not mention the general case any further but simply observe that closed-form expressions for the lengths of shortest paths from node $(0,0)$ to the other nodes of $3Torus(m,n)$ are as follows:

- if node (i,j) is such that $0 \leq i \leq 2m$ and $0 \leq j \leq 2n$ (so the node lies in the lower-right quadrant) then the length of a shortest path from $(0,0)$ to (i,j) is
$$i+j+\max\left\{2\left\lfloor\frac{i-\lfloor\frac{j}{2}\rfloor}{2}\right\rfloor,0\right\}$$
- if node (i,j) is such that $2m+1 \leq i \leq 4m-1$ and $0 \leq j \leq 2n-1$ (so the node lies in the upper-right quadrant) then the length of a shortest path from $(0,0)$ to (i,j) is
$$(4m-i)+j+\max\left\{2\left\lfloor\frac{(4m+1-i)-\lfloor\frac{j}{2}\rfloor}{2}\right\rfloor,0\right\}$$
- if node (i,j) is such that $0 \leq i \leq 2m$ and $2n+1 \leq j \leq 4n-1$ (so the node lies in the lower-left quadrant) then the length of a shortest path from $(0,0)$ to (i,j) is
$$i+(4n-j)+\max\left\{2\left\lfloor\frac{i-\lfloor\frac{4n-1-j}{2}\rfloor}{2}\right\rfloor,0\right\}$$
- if node (i,j) is such that $2m+1 \leq i \leq 4m-1$ and $2n+1 \leq j \leq 4n-1$ (so the node lies in the upper-left quadrant) then the length of a shortest path from $(0,0)$ to (i,j) is
$$(4m-i)+(4n-j) \\ +\max\left\{2\left\lfloor\frac{(4m-i)-\lfloor\frac{4n-1-j}{2}\rfloor}{2}\right\rfloor,0\right\}.$$

We are now in a position to establish the diameter $\Delta(3Torus(m,n))$ of $3Torus(m,n)$ (we denote the diameter of any network G by $\Delta(G)$). Consider any network $3Torus(m,n)$ annotated according to our shortest path labelling (as in Figs. 4–6). Choose any column. The node on row $2m$ is the node of this column for which the shortest path from $(0,0)$ is longest. According to this labeling, the lengths of a shortest path from $(0,0)$ to $(2m,0),(2m,1),\ldots,(2m,4m-3),(2m,4m-2)$, assuming that $2n<4m-2$, are:

$$\begin{aligned}&4m,4m+1,4m,4m+1,4m+2,4m+3,\\&\quad 4m+2,4m+3,4m+4,4m+5,\\&\quad 4m+4,4m+5,4m+6,4m+7,\\&\quad \ldots\\&\quad 6m-4,6m-3,6m-2,6m-1,\\&\quad 6m-2,\end{aligned}$$

and if $2n\geq 4m-2$ then the lengths of a shortest path from $(0,0)$ to $(2m,4m-1),(2m,4m),\ldots,(2m,2n)$ (within the lower-right quadrant) are:

$$6m-1,6m,6m+1,6m+2,\ldots,2m+2n.$$

Consequently, a node on row $2m$ that is furthest from node $(0,0)$ in $3Torus(m,n)$ (from amongst all nodes on row $2m$) is as follows:

- if $n<2m$ is odd then the node $(2m,2n-1)$ is furthest from node $(0,0)$ and its distance from $(0,0)$ is $4m+n$;
- if $n<2m$ is even then the node $(2m,2n)$ is furthest from node $(0,0)$ and its distance from $(0,0)$ is $4m+n$;
- if $n\geq 2m$ then the node $(2m,2n)$ is furthest from node $(0,0)$ and its distance from $(0,0)$ is $2m+2n$.

Hence, we have proven the following result.

Theorem 2: The network $3Torus(m,n)$ has diameter $4m+n$, if $n<2m$, and diameter $2m+2n$, if $n\geq 2m$.

Whilst the diameter gives a worst-case bound on the length of the path taken by a message according to an optimal routing algorithm, sometimes a more representative estimation is required. The *average inter-node distance* of a network is the average length of a shortest path joining any two nodes in the network.

Suppose that $n=2m-1$. As can be seen in Fig. 6, this situation is the cusp when there is at least one column of nodes, none of which has a label. Let us calculate the sum of all the labels (we then use this sum to obtain an estimate on how much the average inter-node distance in $3Torus(m,2m-1)$ exceeds the average inter-node distance in $Torus(4m,8m-4)$).

Consider the sum of the labels of nodes in column 2 and rows 3-$2m$. This sum is $4(1+2+\ldots+(m-1))=2m(m-1)$, and this is true for the sum of labels of nodes in column 3 and rows 3-$2m$. Consider the sum of the labels of nodes in column 4 and rows 4-$2m$. This sum is $2m(m-1)-2(m-1)=2(m-1)^2$, and this is true for the sum of labels of nodes in column 5 and rows 4-$2m$. Proceeding in this way yields that the sum of the labels of nodes in columns 1-$2n$ and rows 1-$2m$ is

$$\begin{aligned}&2m^2+4m(m-1)+4(m-1)^2\\&\quad+4(m-1)(m-2)+4(m-2)^2+\\&\quad\ldots\\&\quad+4\cdot 2(2-1)+4\cdot 1^2\\&\quad<8\sum_{i=1}^{m}i^2=\frac{4}{3}(2m^3+3m^2+m).\end{aligned}$$

Consequently, by symmetry of $3Torus(m,2m-1)$ (see, for example, Fig. 6), we have that the average inter-node distance in $3Torus(m,2m-1)$ exceeds that of $Torus(4m,8m-4)$ by

$$\begin{aligned}\frac{1}{3(2m-1)}(2m^2+3m+1)&=\frac{m+2}{3}+\frac{1}{2m-1}\\&<\frac{m+4}{3}.\end{aligned}$$

We do not feel that it is particularly worthwhile pursuing a similar comparison for when $n<2m-1$ and for when $n>2m-1$ beyond remarking that in the former case the difference gets bigger as n gets smaller, and in the latter case the difference gets smaller as n gets bigger. We shall return to this comparison later on in our evaluation in Section VII.

B. Routing algorithms

We now turn to routing algorithms in $3Torus(m,n)$. By Theorem 1, we may assume that we have a message at node $(0,0)$ and that we wish to route this message to node (u,v). If the source node is some other node, (x,y) say, and the destination node is (u,v) then we simply: obtain the mapping, Γ say, from Section IV that maps (x,y) to $(0,0)$; apply this mapping Γ to (u,v) so as to obtain (u',v'), say; obtain the path from the algorithm below that results when we wish to route from $(0,0)$ to (u',v'); and obtain our required path by applying the inverse mapping Γ^{-1} to each node of this path.

We begin by assuming that $0\leq u\leq 2m$ and $0\leq v\leq 2n$ (that is, that the node (u,v) lies in the lower-right quadrant). Consider the following algorithm $route(u,v)$.

```
route(u,v):
i = j = 0
while i < u and j < v do
   if i + j ≡ 1 or 2 (mod 3) then
      j = j + 1
   else
      i = i + 1
   output (i,j)
if i = u then
   while j < v do
      j = j + 1
      output (i,j)
else
   while i ≠ u or j ≠ v do
      if (i ≡ 0 (mod 2), j ≡ 0,1 (mod 4))
      or (i ≡ 1 (mod 2), j ≡ 2,3 (mod 4))
```

```
then
   i = i + 1
else
   if j ≡ 1 (mod 2) then
      j = j + 1
   else
      j = j − 1
output (i, j)
```

The path obtained by executing the algorithm $route(7,5)$ in $3Torus(4,8)$ is depicted in Fig. 7. Given our analysis above, it is clear that the algorithm $route(u,v)$ is optimal; that is, in general always results in a shortest path from $(0,0)$ to (u,v).

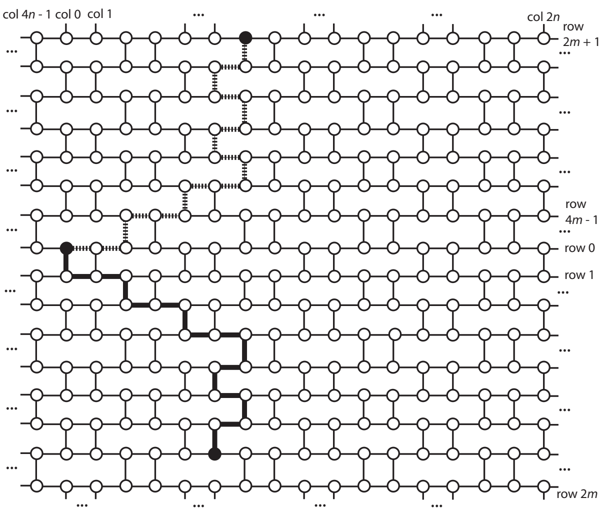

Fig. 7. Shortest paths in $3Torus(4,8)$ from $(0,0)$ to $(7,5)$ and $(9,6)$.

If our target node (u,v) lies in some other quadrant of $3Torus(m,n)$ then the basic principles behind the algorithm $route(u,v)$ easily yield an optimal routing algorithm (the resulting path for the node $(9,6)$ in $3Torus(4,8)$ is also depicted in Fig. 7). Also, it is trivial to implement any of our routing algorithms so that they are distributed routing algorithms (that is, the next link to traverse is calculated by the node at which the message currently resides). Note that the discussion immediately prior to the statement of our routing algorithm, where we use the mapping Γ to obtain a generic routing algorithm, can trivially be incorporated into $route(u,v)$ so that appropriate offsets are first calculated and then universally applied throughout. Thus, we have the following.

Theorem 3: The network $3Torus(m,n)$ has an optimal source routing algorithm so that the time taken to output a shortest path is linear in the length of this path. Moreover, this algorithm can be implemented as a distributed routing algorithm so that the time taken at each node to calculate the next link to traverse is constant.

C. Connectivity and Hamiltonicity

We end this section by looking at some useful structural properties of $3Torus(m,n)$ in the context of interconnection networks, namely its connectivity, its wide-diameter and its Hamiltonicity.

Suppose that $n \geq 3$. Let the node (u,v) lie in the lower-right quadrant. We shall construct 3 node-disjoint paths from $(0,0)$ to (u,v). Our first path is the path ρ_1 constructed by $route(u,v)$. There are two possibilities: the path ρ_1 has length different to the length of a shortest path from $(0,0)$ to (u,v) in $Torus(4m,4n)$; or these lengths are the same.

Suppose that it is the former possibility (which is the more complex; as regards the latter possibility, we proceed similarly to below and so we omit the details).

- Our second path ρ_2 starts at $(0,0)$ so that the next node is $(0,1)$. Thereafter, it runs 'parallel' to ρ_1 before 'zig-zagging' down and approaching (u,v) from the right. This second path can be visualized in Fig. 8 as the dotted path leaving node $(0,0)$ to the right (the first path is depicted in solid bold). We do not detail the actual path ρ_2 but note that there are 4 essential cases, depending upon the value of v (mod 4). It is easy to verify that in each case the length of the path ρ_2 is at most the length of ρ_1 plus 8.
- Our third path ρ_3 starts at $(0,0)$ so that the next three nodes are $(0,4n-1),(0,4n-2),(0,4n-3)$. Thereafter, it runs 'parallel' to ρ_1 before 'zig-zagging' down and approaching (u,v) from the left. This third path can be visualized in Fig. 8 as the dotted path leaving node $(0,0)$ to the left. We do not detail the actual path ρ_3 but note that again there are 4 essential cases, depending upon the value of v (mod 4). It is easy to verify that in each case the length of the path ρ_3 is at most the length of ρ_1 plus 8.

What results is 3 mutually node-disjoint paths where the length of any of them is at most the length of a shortest path from $(0,0)$ to (u,v) plus 8.

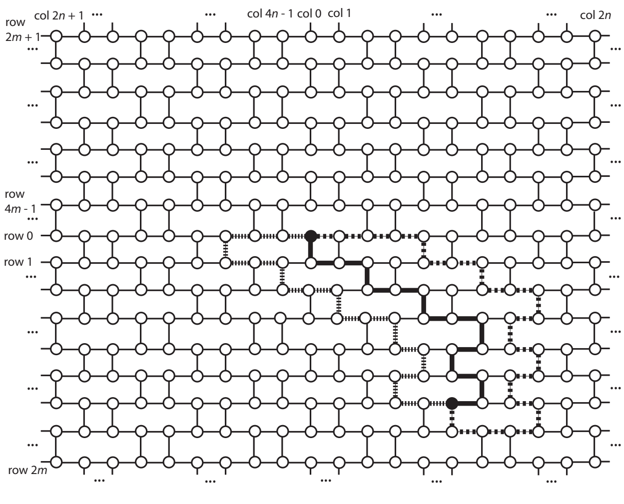

Fig. 8. Three paths in $3Torus(4,5)$ from $(0,0)$ to $(6,5)$.

Now suppose that the node (u,v) lies in any other quadrant. By proceeding exactly as we have done above, we can easily construct 3 mutually node-disjoint paths from

$(0,0)$ to (u,v) where the length of any of them is at most the length of a shortest path from $(0,0)$ to (u,v) plus 8.

When $n=2$, we require a slightly different construction. Rather than repeat the above analysis, we refer the reader to the illustration in Fig. 9 where we depict 3 node-disjoint paths from $(0,0)$ to $(7,3)$. We can adapt the strategy in this figure to any destination node in the lower-right quadrant (and so to any destination node in $3Torus(m,2)$). Moreover, we looking at each distinctive case in turn, as we did above, it is easy to verify that in $3Torus(m,2)$ we can construct 3 mutually node-disjoint paths from $(0,0)$ to (u,v) where the length of any of them is at most the length of a shortest path from $(0,0)$ to (u,v) plus 8.

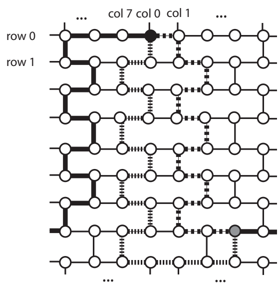

Fig. 9. Three paths in $3Torus(m,2)$ from $(0,0)$ to $(7,3)$.

The *wide-diameter* δ_w of a graph G of connectivity γ is the smallest integer such that for any pair of distinct vertices u and v, there are γ mutually node-disjoint paths from u to v such that each of these paths has length at most δ_w. We have thus proven the following result.

Theorem 4: When $n \geq 2$, the network $3Torus(m,n)$ has wide-diameter at most $\Delta(3Torus(m,n)) + 8$ and connectivity 3.

Note that when $n=1$, the conclusions of Theorem 4 do not hold. In order to see this note that no matter which two consecutive rows we consider, there are only 2 links from one of these rows to the other. Hence, if we have 3 paths from $(0,0)$ to some destination node in the lower-right quadrant, say, at least one of these paths must 'wrap around' via a link joining a node in row $2m+1$ and a node in row $2m$.

Finally, we note that $3Torus(m,n)$ is Hamiltonian, for $n \geq 1$ and $m \geq 1$. To see this, note that the nodes on every row form a cycle and these cycles can be iteratively 'joined' by removing a pair of links and including an additional pair of links (where the 4 links involved form a cycle).

Theorem 5: The network $3Torus(m,n)$ is Hamiltonian.

VI. BROADCASTING

In this section, we consider different aspects of broadcasting in $3Torus(m,n)$. We look at: *one-to-all broadcasts*, where one node sends the same message to every other node; *personalized one-to-all broadcasts* (also called *one-to-all scatters* or *single-node scatters*), where one node sends different messages to every other node; and the 'reverse' operation to a personalized one-to-all broadcast, namely a *gather*, where every node sends a message to a particular node (personalized one-to-all broadcasts and gathers feature heavily in data centre networks in the form of *map* and *reduce* operations; see, for example, [28]).

Our presentation will be algorithmic and with respect to two different distributed-memory models of computation: a one-port model; and an all-port model. In both models, each node of our interconnection network represents a processor and the network is: *synchronous*, in that each processor at each node operates according to the ticks of a global clock (which determine the time-steps of the global computation), with message-passing undertaken on a clock tick and local computation undertaken between-times; and *full-duplex*, in that neighbouring nodes can send messages to each other at the same time. We assume that the size of any message is such that the message can be delivered between two neighbouring nodes in one time step and that any amount of local computation can be undertaken between ticks of the global clock. Our *one-port model* (as in, for example, [2], [7], [21]) is such that on any tick of the global clock, any processor can send at most one message (to one of its neighbours) and receive at most one message (from one of its neighbours). This one-port model roughly corresponds to having store-and-forward switching so that network contention is impossible (if a message arrives at some node then it can always be forwarded on at the next tick of the global clock; that is, queues do not build up at nodes). Our *all-port model* (as in, for example, [7], [32], [38]) is less stringent and such that on any tick of the global clock, any processor can send a message to any number of its neighbours and also receive a message from any number of its neighbours. It is possible within our all-port model for network contention to occur, for we might have two simultaneously incoming messages that need to be next sent down the same output link. We are only interested in contention-free algorithms; that is, where such contention does not arise. Finally, our algorithms are *distributed* in that any decision as to where to route a message from some node is undertaken by the node at which the message resides (that is, the complete route of the message has not been pre-calculated by the source node and included within the message).

A. A spanning tree

We shall begin by constructing a minimum-depth spanning tree in $3Torus(m,n)$. We subsequently use this spanning tree to undertake various broadcasts. However, we need to be aware that not only must we prove the existence of such a tree but we must also construct algorithms that stipulate what each node does on any tick of the global clock (in both of our models). In order to facilitate these algorithmic descriptions in our one-port model, we construct a d-*labelled tree*, which is a spanning tree rooted at our source node with the property that:

- a positive integer labels every link of this spanning tree;
- the integers used to label the links incident with any given node of this tree are distinct;

- the integer used to label the link joining some (non-root) node to its parent in this tree is less than any of the integers used to label any link from this node to one of its children; and
- the integers used as labels come from the set $\{1, 2, \ldots, d\}$, for some d, with every integer from this set appearing as a label at least once.

Given some d-labelled tree, we perform a one-to-all broadcast in our one-port model, for example, as follows. The message ϵ starts at the root node and on the ith tick of the global clock, where $i \in \{1, 2, \ldots, d\}$, any node u joined to a child v via a link labelled i sends the message ϵ, currently at u, to v. A simple induction yields that if we have a d-labelled spanning tree then the resulting algorithm is well-defined; that is, is such that when a node u wishes to send the message ϵ, this message does indeed reside at u.

Theorem 6: Define $d = \Delta(3Torus(m, n))$ and let u be any node of $3Torus(m, n)$. There exists a d-labelled spanning tree T_u of depth d that is rooted at u so that for every node v of $3Torus(m, n)$, the length of the path from u to v in T_u is equal to the length of a shortest path from u to v in $3Torus(m, n)$.

Proof: By Theorem 1, we may assume that $u = (0, 0)$. There are two case: when $2m \leq n$; and when $2m > n$.

<u>Case (*i*)</u>: $2m \leq n$.

We begin by defining 4 paths, each of which starts at node $(0, 0)$. These paths can be visualized as in Fig. 10 for $3Torus(2, 4)$. Note that we work with respect to our visualization of $3Torus(m, n)$ as having the source node $(0, 0)$ at the centre and delimitted by columns $2n$ and $2n+1$ to the right and left, respectively, and rows $2m$ and $2m+1$ to the bottom and top, respectively (of course, there are wrap-around links).

- Path p_1 is built by iterating the following construction $2m$ times: move one link right, one link down and one link right. Hence, we obtain a path from $(0, 0)$ to $(2m, 4m)$.
- We begin building the path p_2 by iterating the following construction $2m - 1$ times: move one link down and two links to the left. We then extend the resulting path from $(0, 0)$ to $(2m-1, 4n-2(2m-1))$ with the links $((2m-1, 4n-2(2m-1)), (2m, 4n-2(2m-1))$ and $((2m, 4n-2(2m-1)), (2m, 4n-2(2m-1)-1)$ to obtain a path p_2 from $(0, 0)$ to $(2m, 4n-4m+1)$.
- We begin building the path p_3 by iterating the following construction $2m - 1$ times: move two links left and one link up. We then extend this path from $(0, 0)$ to $(2m + 1, 4n - 2(2m - 1))$ with the link $((2m+1, 4n-2(2m-1)), (2m+1, 4n-2(2m-1)-1)$ to obtain a path p_2 from $(0, 0)$ to $(2m+1, 4n-4m+1)$.
- We begin building the path p_4 as $(0, 0), (0, 1), (0, 2)$ and then iterate the following construction $2m - 1$ times: move one link right, one link up and one link right. Thus, we obtain a path p_4 from $(0, 0)$ to $(2m+1, 4m)$ (note that p_1 and p_4 share a link).

Path p_1 has its links labelled by starting at 1 and thereafter incrementing each label by 1; path p_2 has its links

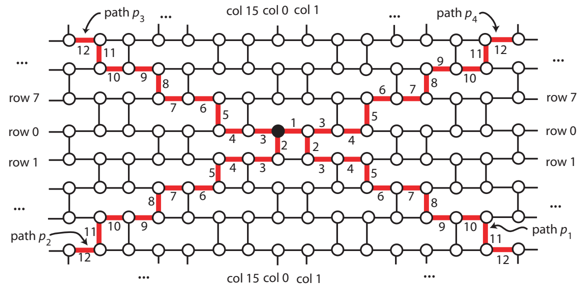

Fig. 10. The skeleton of our spanning tree in $3Torus(2, 4)$.

labelled by starting at 2 and thereafter incrementing each label by 1; path p_3 has has its links labelled by starting at 3 and thereafter incrementing each label by 1; and path p_4 has its links labelled by starting at 1 so that the next label is 3 and thereafter incrementing each label by 1. Note that so far the maximum label used is $6m$. Paths p_1, p_2, p_3 and p_4 partition $3Torus(m, n)$ into 4 obvious 'geographic' zones. Call these zones 'top', 'bottom', 'left' and 'right'.

Given our skeleton tree T formed by the paths p_1, p_2, p_3 and p_4, we extend T to a spanning tree as follows. Consider the path p_1. We build a path from every right-most node on each row (apart from row 0) along the row until we reach column $2n$ (these are paths in the right zone). We label the links of these paths incrementally starting from the integer one greater than any currently labelled link incident with the start node of each path. Our construction can be visualized in $3Torus(2, 4)$ as in Fig. 11. We build paths analogously in the left zone. Finally, in the bottom zone, we include all pendant links 'hanging down' from nodes incident with two row-links in either the path p_1 or p_2, and then we zig-zag downwards to row $2m$, just as we did earlier. Links on these paths are labelled incrementally as were the links in the left and right zones. We proceed analogously in the top zone. These paths in $3Torus(2, 4)$ can also be visualized in Fig. 11.

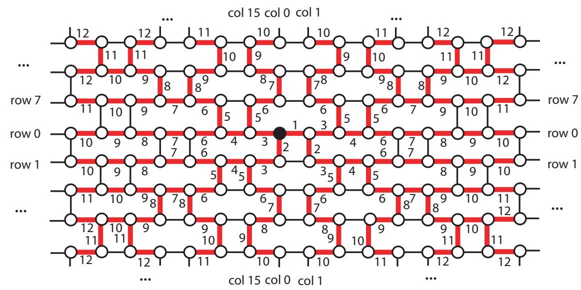

Fig. 11. Our spanning tree in $3Torus(2, 4)$.

Our analysis from the previous section yields that our spanning tree is as required.

<u>Case (*ii*)</u>: $2m > n$.

We proceed analogously to as in Case (*i*) except that when any of our paths p_1, p_2, p_3 and p_4 reach column $2m$ we halt their construction. The skeleton tree is extended to a spanning tree in exactly the same way as in Case (*i*) and has the required properties. ■

B. One-to-all broadcasts

Our spanning tree from Theorem 6 (rooted at u) can clearly by constructed in a distributed fashion; that is, upon receipt of some message in which is contained the name of the message's source node and the name of its destination node, any (transit) node can easily calculate which of its neighbours it is incident with within this tree. In order to see this, consider the skeleton tree T. The role of a node in this tree can be understood by simple numeric calculations relating to the node's name. Any node's name clearly determines which zone it lies in. Also, for example, a node lies on path p_1 if it lies in the bottom and right zones and its name is of the form $(i, 2i-1)$, $(i, 2i)$ or $(i, 2i+1)$; in the first case, there are no pendant links incident with the node, in the second case there is a 'downward' pendant link and in the third case there is a 'horizontal' pendant link. Similarly, all other nodes in the bottom and right zones can determine their two neighbours in the resulting spanning tree solely from their own names. Finally, all nodes can determine the labels of their incident links by a simple numeric calculation using their names. Consequently, we immediately obtain the following result from Theorem 6.

Corollary 7: There exists a distributed one-to-all broadcast algorithm for $3Torus(m, n)$ that can be implemented under our one-port model so as to take $\Delta(3Torus(m, n))$ communication rounds. This algorithm is optimal in terms of the number of communication rounds.

Note that optimality in Theorem 6 follows from the fact that the diameter is a lower bound on the number of communication rounds.

The fact that our spanning tree in Theorem 6 has depth $\Delta(3Torus(m, n))$ yields the following.

Corollary 8: There exists a distributed one-to-all broadcast algorithm for $3Torus(m, n)$ that can be implemented under our all-port model so as to take $\Delta(3Torus(m, n))$ communication rounds. This algorithm is optimal in terms of the number of communication rounds.

C. Personalized one-to-all broadcasts

Our spanning tree from Theorem 6 can also be used in order to undertake a personalized one-to-all broadcast in $3Torus(m, n)$. It is easy to prove, via a simple induction, that in order to undertake a personalized one-to-all broadcast in our one-port model with *any* spanning tree, it suffices to emit the messages from the source so that if message ϵ, which is intended for node v, is emitted from the source before message ϵ', which is intended for node v', then the length of the path from the source to v is no less than the length of the path from the source to node v' (at any node, each message is always emitted along a link that takes it one link closer to its destination). This simple induction also yields that the number of communication rounds is equal to the number of destination nodes which is $16nm-1$; hence, the resulting algorithm is optimal.

Corollary 9: There exists a distributed personalized one-to-all broadcast algorithm for $3Torus(m, n)$ that can be implemented under our one-port model so as to take $16nm-1$ communication rounds. This algorithm is optimal in terms of the number of communication rounds.

Consider some personalized one-to-all broadcast in our all-port model. Any personalized one-to-all broadcast in our all-port model that injects as many messages as possible from u in any communication round except for perhaps the last communication round only (so, in every round 3 messages would be injected except for the last round when there would be 1 or 2, depending upon whether $16nm-1 \equiv$ 1 or 2, respectively), must necessarily be optimal.

Consider the personalized one-to-all broadcast in our all-port model obtained using our spanning tree from Theorem 6 as follows. Let T_1 be the sub-tree rooted at $(0, 1)$; let T_2 be the sub-tree rooted at $(1, 0)$; and let T_3 be the sub-tree rooted at $(0, 4n-1)$. Our broadcast algorithm might inject messages so as to (simultaneously) simulate a personalized one-to-all broadcast in our one-port model in each of T_1, T_2 and T_3 (of course, a message is injected into the sub-tree in which its destination node resides, is moved one link closer to its destination in each communication round and messages are injected according to a 'farthest-first' philosophy). The number of nodes in T_2, that is, $|T_2|$, is $4mn$; $|T_3|$ is $4mn-1$; and $|T_1|$ is $8mn$. Consequently, the number of communication rounds in this one-to-all broadcast algorithm is $8mn$.

If we can amend the sub-trees T_1, T_2 and T_3 so that the numbers of nodes of any two of these sub-trees differ by at most 1 then (as we argued above) we will obtain an optimal algorithm. This can be done as we now describe. Essentially, we will steal nodes from T_1 and give them to T_2 and T_3. Consider the border between T_2 and T_1, namely the 'gap' between columns 0 and 1 (below row 0). Suppose we wanted to steal a node from T_1 and give it to T_2. We could remove the node $(2m, 1)$ from T_1 and adjoin it to T_2 via the link $((2m, 0), (2m, 1))$. Suppose that we wanted to steal two nodes from T_1. We could, in addition to the amendment just made, remove the node $(2m, 2)$ from T_1 and adjoin it to T_2 via the link $((2m, 1), (2m, 2))$. Suppose that we wanted to steal three nodes from T_1. We could, in addition to the amendments just made, remove the node $(2m-1, 2)$ from T_1 and adjoin it to T_2 via the link $((2m, 2), (2m-1, 2))$. We can clearly proceed similarly, not only with regard to the border between T_1 and T_2 but with regard to the border between T_1 and T_3. Also, if we wanted to work along rows $2m$ and $2m-1$, beyond row $2n$, stealing nodes from T_1, it is easy to see how this can be done by working back from column $2n$ and removing links from row $2m-1$ and including links joining nodes in rows $2m$ and $2m-1$. There are myriad ways in which to steal nodes from one sub-tree and give adjoin to another.

It is not difficult to see that (no matter what the relative values of m and n) we can amend T_1, T_2 and T_3 so that $|T_3| \leq |T_2| \leq |T_1|$ and $|T_1| - |T_3| \in \{0, 1\}$. Given that initially $|T_1| = 8mn$, $|T_2| = 4mn$ and $|T_3| = 4mn-1$, it suffices to steal $\lceil \frac{4mn}{3} \rceil$ nodes from T_1 and give them to T_3, and to steal a further $\lfloor \frac{4mn}{3} \rfloor$ nodes from T_1 and give them to T_2 so as to obtain 'balanced' sub-trees. In consequence, $|T_1| = 8mn - \lceil \frac{4mn}{3} \rceil - \lfloor \frac{4mn}{3} \rfloor$. An amendment of the sub-trees T_1, T_2 and T_3 (in the above fashion) in the case of $3Torus(2,4)$ can be visualized as in Fig. 12, where the roots of the sub-trees are grey nodes, where the dashed lines are the new borders between the amended sub-trees and where $|T_1| = 43$ and $|T_2| = |T_3| = 42$. Moreover, with more precision, we could clearly develop an algorithm that every node of $3Torus(m,n)$ (that is, processor) could apply locally in order to give its amended links in the resulting spanning tree (such an algorithm would be a straightforward, if messy, numeric calculation with parameters m, n and the row and column of the node). Hence, by our discussion above (and including that immediately following Corollary 9), we have the following result.

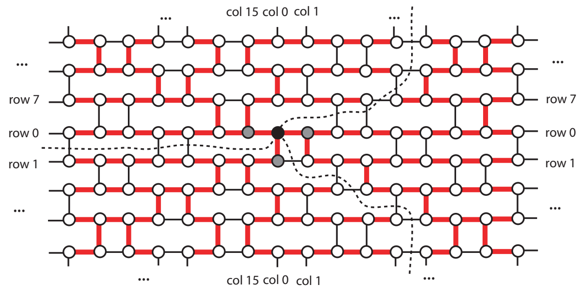

Fig. 12. Our amended sub-trees in $3Torus(2,4)$.

Corollary 10: There exists a distributed personalized one-to-all broadcast algorithm for $3Torus(m,n)$ that can be implemented under our all-port model so as to take $8mn - \lceil \frac{4mn}{3} \rceil - \lfloor \frac{4mn}{3} \rfloor$ communication rounds. This algorithm is optimal in terms of the number of communication rounds.

D. Gathers

Finally, consider a gather in $3Torus(m,n)$. In our one-port model, we can clearly use our spanning-tree from Theorem 6 in order to undertake a gather that is optimal (essentially, we just 'reverse' the personalized one-to-all broadcast from Corollary 9; the resulting algorithm still conforms to our one-port model). As regards our personalized one-to-all broadcast in our all-port model, again by 'reversing' it we obtain a gather, with this algorithm being optimal for the reason detailed immediately after Corollary 9 (this is immediate given that the algorithm implicit in Corollary 10 consists of 3 independent 'one-port algorithms'). Hence, we have the following results.

Corollary 11: There exists a distributed gather algorithm for $3Torus(m,n)$ that can be implemented under our one-port model so as to take $16mn - 1$ communication rounds. This algorithm is optimal in terms of the number of communication rounds.

Corollary 12: There exists a distributed gather algorithm for $3Torus(m,n)$ that can be implemented under our all-port model so as to take $8mn - \lceil \frac{4mn}{3} \rceil - \lfloor \frac{4mn}{3} \rfloor$ communication rounds. This algorithm is optimal in terms of the number of communication rounds.

VII. EVALUATION

Having derived some properties of $3Torus(m,n)$, we now compare $3Torus(m,n)$ with $Torus(4m,4n)$, which has the same number of nodes (note that the primary motivation of the design of $3Torus(m,n)$ is as a 'degree-3' version of a torus). As we shall see, the comparison is generally favourable although (as might be expected) there is a price to be paid by pruning and consequent degree reduction. Our evaluation comes in two parts: first, we compare $3Torus(m,n)$ and $Torus(4m,4n)$ in terms of their structural (graph-theoretic) properties (that are pertinent in their usage as interconnection networks); and second, we undertake a preliminary performance evaluation of routing algorithms in $3Torus(m,n)$ and $Torus(4m,4n)$. Our performance evaluation is but a prelude to a more thorough simulation (as we explain in our conclusions in Section VIII).

A. A structural comparison

First, we note from Theorem 1 that $3Torus(m,n)$ is node-symmetric but not link-symmetric. It is well-known (and easy to see) that $Torus(N,N)$ is both node- and link-symmetric; however, note that $Torus(N_1,N_2)$ is no longer link-symmetric if $N_1 \neq N_2$. Link-symmetry can be important in terms of load balancing. However, the lack of link-symmetry might be ameliorated depending upon the application, for there is still a degree of link-symmetry in $3Torus(m,n)$ in that: there is clearly an automorphism mapping any column-link to any other column-link; and the row-links partition into two disjoint sets, E_4 and E_8, so that there is an automorphism mapping any row-link of E_4 (resp. E_8) to any other row-link of E_4 (resp. E_8) (in group-theoretic terms, there are only three orbits of edges under the action of the automorphism group of $3Torus(m,n)$; the sets E_4 and E_8 are so named so as to reflect the length of a shortest cycle in which the row-link lies).

As regards diameters, when we compare $3Torus(m,n)$ with $Torus(4m,4n)$ we can immediately see from Theorem 2 that $\Delta(3Torus(m,n))$ is identical to $\Delta(Torus(4m,4n)) = 2m + 2n$, if $n \geq 2m$, and greater than $\Delta(Torus(4m,4n))$ by $2m - n$, if $n < 2m$. For the special case where $m = n$ we have that $\Delta(Torus(4m,4m)) = 4m$ and $\Delta(3Torus(m,m)) = 5m$. Consequently, depending on m and n, we might be able to build $3Torus(m,n)$ so that we lose nothing in comparison with the diameter of $Torus(4m,4n)$. Our optimal and easy-to-implement routing algorithm ensures that we can efficiently utilize this state of affairs.

As regards the average inter-node distance, that of $Torus(4m,4n)$ is better than that of $3Torus(m,n)$ (as we might expect). The quality of the average inter-node distance of $3Torus(m,n)$ in comparison with that of $Torus(4m,4n)$ depends upon the relative values of m and n with the larger n is in comparison to m, the better. Just as when we compared the diameters of $Torus(4m,4n)$ and $3Torus(m,n)$, it makes sense to try and increase n with respect to m.

From [23], the wide-diameter of $Torus(4m, 4n)$ is $\Delta(Torus(4m, 4n)) + 1$, if both m and n are greater than 1, and $\Delta(Torus(4m, 4n)) + 2$, if either m or n is equal to 1 and the other component is at least 3. From Theorem 4, the wide-diameter of $3Torus(m, n)$ is $\Delta(3Torus(m, n)) + 8$ which is still relatively good in comparison (in that it is different from the diameter by a small constant value). Of course, the lower degree of $3Torus(m, n)$ in comparison with $Torus(4m, 4n)$ means that we have less connectivity (and so less path diversity and capacity to tolerant faults); nevertheless, the connectivity of $3Torus(m, n)$ is equal to its degree (which is all that we can hope for). As regards Hamiltonicity, by Theorem 5 $3Torus(m, n)$ is Hamiltonian, just like $Torus(4m, 4n)$. However, there cannot exist edge-disjoint Hamiltonian cycles in $3Torus(m, n)$ (as the degree is 3) whereas it is well-known that $Torus(4m, 4n)$ does have edge-disjoint Hamiltonian cycles (see, for example, [5]).

In [2], optimal one-to-all and personalized one-to-all broadcast algorithms for tori were given for the one-port model of computation; in [7], [19] optimal one-to-all broadcast algorithms for tori were given for the all-port model of computation; and in [19] optimal personalized one-to-all broadcast algorithms for tori were given for the all-port model of computation. Given our comments above as regards the relative diameters of $3Torus(m, n)$ and $Torus(4m, 4n)$, along with Corollaries 7, 8, 9 and 10, $3Torus(m, n)$ behaves comparably to $Torus(4m, 4n)$ in terms of one-to-all broadcasts.

B. *Simulating routing algorithms*

Our performance evaluation compares our routing algorithm for $3Torus(m, n)$ with the XY-routing algorithm for $Torus(4m, 4n)$. Recall, the *XY-routing algorithm* in a torus is simply to route along a row until the message reaches a node in the same column as the destination, and then to route along the column, with both the X-route and the Y-route being the shortest from the two possibilities. Given that we have a precise closed form of the length of any route obtained within $3Torus(m, n)$ (via our routing algorithm), we focus on loads arising (or, more precisely, on the distribution of generated paths over each link) in our chosen traffic pattern.

As regards the preliminary nature of our performance evaluation, the primary purpose of the research in this paper is the introduction of the interconnection network $3Torus(m, n)$, together with a structural analysis of its viability as a replacement network for a torus in a 'degree-3' context; that is, a full and proper performance evaluation is beyond our scope and requires additional research as regards more sophisticated (adaptive) routing algorithms and also all-to-all broadcasts (we have more to say in Section VIII). Nevertheless, we are in a position to undertake a preliminary experimental analysis. Of particular interest to us is how the relative sizes of the parameters m and n in $3Torus(m, n)$ impact upon routing performance (in terms of loads arising and the balance of these loads), and of how the routing performance in $3Torus(m, n)$ and $Torus(4m, 4n)$ (its natural counterpart) differ.

We assume the *random traffic pattern* where every node of a network (simultaneously) sends a message to some uniformly randomly chosen node (possibly itself). Our analysis consists of undertaking numerous trials of random traffic messaging (actually, 1000 trials), with respect to our routing algorithms, so as to acquire data as regards the cumulative loads on (that is, number of generated paths using) each individual link in the network. We vary the parameters m and n in $3Torus(m, n)$ and compare the resulting loads with the loads obtained (using XY-routing) in $Torus(4m, 4n)$. Given the structure of $3Torus(m, n)$ and $Torus(4m, 4n)$, we also consider row-links and column-links separately; indeed, for $3Torus(m, n)$ we partition row-links into those that are contained within a cycle of length 4, which we call *row*4*-links*, and those that aren't, which we call *row*8*-links* (*cf.* our comments above on link-symmetry).

From our structural analysis, we have seen that the situation when $n = 2m - 1$ would appear to be a cusp at which the behaviour of $3Torus(m, n)$ changes. One aim of our experiments is to investigate around this threshold. To this end, for a fixed m ranging from 2 to 12, we allow n to vary at intervals of (the integer part of) $\frac{m}{2}$ from $\frac{m}{2}$ through m and $2m$ up to $\frac{7m}{2}$ (which we deem sufficiently large). For each scenario, we collect the load on each link (as described above and with data from 1000 trials) so that we might calculate the average link load and also the standard deviation of the link load, so that we might obtain some appreciation of the balance of loads across all links.

We have included graphs depicting our experimental results for the largest m and n we worked with, namely $m = 12$ with n ranging from 6 up to 42 (the trends when the parameters were smaller are almost identical).

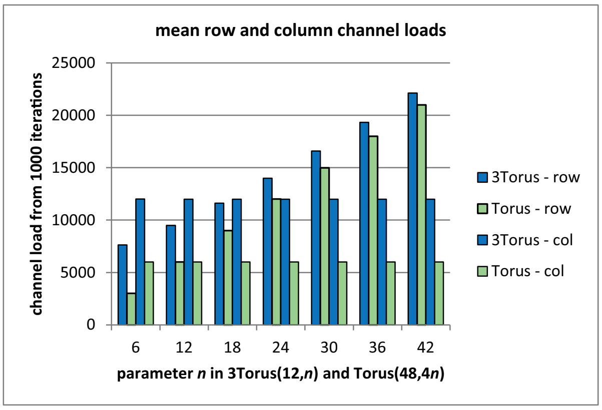

Fig. 13. Mean row and column channel loads when $m = 12$.

We make the following observations on the basis of our experiments.

- The suspected 'cusp' scenario, when $n = 2m - 1$, is indeed such with the mean load for row-links in $3Torus(m, 2m-1)$ being almost identical to the mean load for column-links, as well as the mean load for row-links in $Torus(4m, 8m - 4)$; though the mean

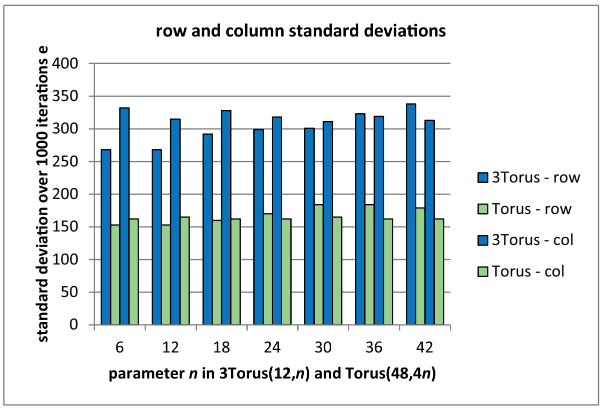

Fig. 14. Standard deviations for row and column channel loads when $m = 12$.

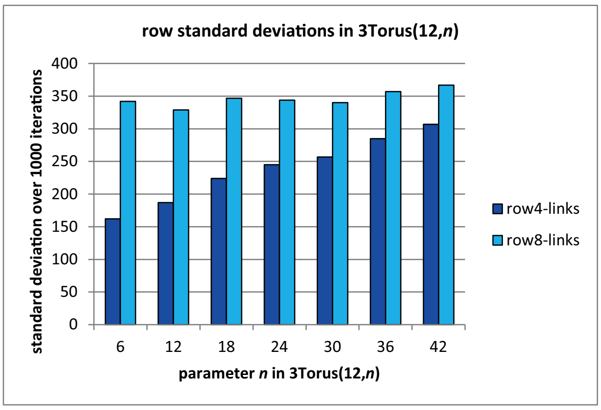

Fig. 16. Standard deviation row channel loads when $m = 12$.

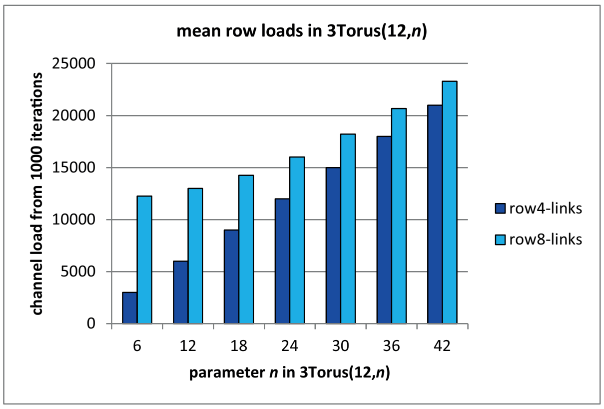

Fig. 15. Mean row channel loads when $m = 12$.

load for column-links in $Torus(4m, 8m-4)$ is around half this value (see Fig. 13). The balance (that is, standard deviation) of row-link and column-link loads is similar in both $3Torus(m, 2m-1)$ with this balance around twice that of link loads in $Torus(4m, 8m-4)$ (see Fig. 14).

- The mean load for column-links in both $3Torus(m, n)$ and $Torus(4m, 4n)$ is pretty much stable with this load in $3Torus(m, n)$ being around twice that in $Torus(m, n)$. However, the mean load for row-links in $3Torus(m, n)$ is only slightly greater than that in $Torus(4m, 4n)$, with both values increasing as n increases (see Fig. 13). The balance of row-link and column-link loads is similar in both $3Torus(m, n)$ and $Torus(4m, 4n)$ with the loads in $Torus(4m, 4n)$ being better balanced (see Fig. 14). However, it is unclear as to whether row-link and column-link balances will remain comparable for larger values of n.
- As regards the two different types of row-link in $3Torus(m, n)$, for some fixed m and for smaller values of n, there is certainly a difference in the relative mean loads as well as their balances (see Figs. 15 and 16). However, as n increases, the loads and the balances converge although they are increasing.

In conclusion (and in the contexts described above), both our structural analysis and our experimental analysis show that the interconnection network $3Torus(m, n)$ can have a good performance in comparison with $Torus(4m, 4n)$ and in some scenarios where the degree of a node is crucial, the potential for $3Torus(m, n)$ as a viable interconnection network has clearly been established.

VIII. Conclusions

We have shown that the network $3Torus(m, n)$ has significant potential as an interconnection network of degree 3 but where properties comparable with those of a torus are retained. Our results form but a preliminary investigation into $3Torus(m, n)$ and we advocate some directions for further research. These directions fall within four strands.

First, we need to study the important all-to-all and personalized all-to-all broadcasts in $3Torus(m, n)$. We envisage that such a study will be far more straightforward for the one-port model than for the all-port model.

Second, we need to study broadcast algorithms in $3Torus(m, n)$ under alternative switching models, such as wormhole switching (which are distance-insensitive and reflect modern switching methods in many environments).

Third, we need to undertake a thorough, practically-oriented performance evaluation of $3Torus(m, n)$ using proper simulation tools so that we can better understand performance when our networks are deployed under specific and more realistic conditions in which different routing, flow control and switching mechanisms are deployed. We fully recognise that our performance evaluation presented here is somewhat basic; nevertheless, it has provided some additional evidence that $3Torus(m, n)$ might perform well in comparison with tori especially when the parameters m and n are chosen preferentially. Of course, it has long been recognised that structural interconnection network properties such as the ones studied in this paper *do* often translate into good practical performance.

Finally and more speculatively (and after the above directions of additional research have been investigated), we would like to ascertain whether higher dimensional tori might be susceptible to degree reduction using pruning techniques derived from the approach studied here.

References

[1] D. Abts and J. Kim, *High Performance Datacenter Networks*: *Architectures, Algorithms and Opportunities*, Morgan and Claypool (2011).

[2] Y. Ashir, I.A. Stewart and A. Ahmed, “Communication algorithms in k-ary n-cube interconnection networks”, *Information Processing Letters*, vol. 61, no. 1, pp. 43–48, Jan. 1997.

[3] Y. Ajima, T. Inoue, S. Hiramoto and T. Shimizu, “Tofu: Interconnect for the K computer”, *Fujitsu Science and Technical Journal*, vol. 48, no. 3, pp. 280–285, July 2012.

[4] B. Albader, B. Bose and M. Flahive, “Efficient communication algorithms in hexagonal mesh interconnection networks”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 69–77, Jan. 2012.

[5] M.M. Bae and B. Bose, “Edge disjoint Hamiltonian cycles in k-ary n-cubes and hypercubes”, *IEEE Transactions on Computers*, vol. 52, no. 10, pp. 1271–1284, Oct. 2003.

[6] L. Bononi and N. Concer, “Simulation and analysis of network on chip architectures: ring, Spidergon and 2D mesh”, *Proc. Design, Automation and Test in Europe*, 2006.

[7] B. Bose, B. Broeg, Y. Kwon and Y. Ashir, “Lee distance and topological properties of k-ary n-cubes”, *IEEE Transactions on Computers*, vol. 44, no. 8, pp. 1021–1030, Aug. 1995.

[8] I.Z. Bouwer, W.W. Chernoff, B. Monson and Z. Star (eds.), *The Foster census*: *R.M. Foster’s census of connected symmetric trivalent graphs*, Charles Babbage Research Centre (1988).

[9] Z. Cai, W. Xiao, Q. Zhang and X. Liu, “Principle of symmetry for network topology with applications to some networks”, *Journal of Networks*, vol. 5, no. 9, pp. 994–1000, Sept. 2010.

[10] C. Camarero, C. Martínez and R. Beivide, “L-Networks: A topological model for regular two-dimensional interconnection networks”, *IEEE Transactions on Computers*, to appear.

[11] D. Chen, N.A. Eisley, P. Heidelberger, R.M. Senger, Y. Sugawara, S. Kumar, V. Salapura, D.L. Satterfield, B. Steinmacher-Burow and J.J. Parker, “The IBM Blue Gene/Q interconnection network and message unit”, *Proc. of Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, article no. 26, Nov. 2011.

[12] M.D.E. Conder and P. Dobcsányi, “Trivalent symmetric graphs on up to 768 vertices”, *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 40, pp. 41–63, Feb. 2002.

[13] M. Coppola, M.D. Grammatakakis, R. Locatelli, G. Maruccia and L. Pieralisi, *Design of Cost-Efficient Interconnect Processing Units: Spidergon STNoC*, CRC Press (2008).

[14] W.J. Dally and D. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann (2004).

[15] J. Duato, S. Yalamanchili and L.M. Ni, *Interconnection Networks*: *An Engineering Approach*, Morgan Kaufmann (2002).

[16] Y.Q. Feng and J.H. Kwak, “Classifying cubic symmetric graphs of order $8p$ or $8p^2$”, *European Journal of Combinatorics*, vol. 26, no. 7, pp. 1033–1052, Oct. 2005.

[17] Y.Q. Feng and J.H. Kwak, “Cubic symmetric graphs of order twice an odd prime-power”, *Journal of the Australian Mathematical Society*, vol. 81, no. 2, pp. 153–164, Oct. 2006.

[18] M. Flahive and B. Bose, “The topology of Gaussian and Eisenstein-Jacobi interconnection networks”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 8, pp. 1132–1142, Aug. 2010.

[19] P. Fragopoulou and S.G. Akl, “Efficient algorithms for global data communication on the multidimensional torus network”, *Proc. 9th Int. Parallel Processing Symp.*, pp. 324–330, Apr. 1995.

[20] A. Hanany and D. Vegh, “Quivers, tilings, branes and rhombi”, *Journal of High Energy Physics*, 10, 029, Oct. 2007.

[21] C.-H. Hsu and B.-R. Tsai, “Scheduling for atomic broadcast operation in heterogeneous networks with one port model”, *Journal of Supercomputing*, vol. 50, no. 3, pp. 269–288, Apr. 2009.

[22] L.H. Hsu and C.K. Lin, *Graph Theory and Interconnection Networks*, CRC Press (2009).

[23] Y. Ishigami, “The wide-diameter of the n-dimensional toroidal mesh”, *Networks*, vol. 27, no. 4, pp. 7–14, July 1996.

[24] N.E. Jerger and L.-S. Peh, *On-Chip Networks*, Morgan and Claypool (2009).

[25] S.-S. Kao and L.-H. Hsu, “Spider web networks: a family of optimal, fault tolerant, hamiltonian bipartite graphs”, *Applied Mathematics and Computation*, vol. 160, no. 1, pp. 269–282, Jan. 2005.

[26] K. Kutnar, A. Malnič and D. Marušič, “Chirality of toroidal molecular graphs”, *Journal of Chemical Information and Modeling*, vol. 45, no. 6, pp. 1527–1535, Aug. 2005.

[27] K. Kutnar, U. Borštnik, D. Marušič and D. Janeži, “Interconnection networks for parallel molecular dynamics simulation based on hamiltonian cubic symmetric topology”, *Journal of Mathematical Chemistry*, vol. 45, no. 2, pp. 372–385, July 2008.

[28] R. Lämmel, “Google’s MapReduce programming model - Revisited”, *Science of Computer Programming*, vol. 70, no. 1, pp. 1–30, Jan. 2008.

[29] A. Malnič, “Action graphs and coverings”, *Discrete Mathematics*, vol. 244, no. 1–3, pp. 299–322, Feb. 2002.

[30] A. Malnič, R. Nedela and M. Škoviera, “Lifting graph automorphisms by voltage assignments”, *European Journal of Combinatorics*, vol. 21, no. 7, pp. 927–947, Oct. 2000.

[31] D. Marušič and T. Pisanski, “Symmetries of hexagonal molecular graphs on the torus”, *Croatia Chemica Acta*, vol. 73, no. 4, pp. 969–981, Dec. 2000.

[32] C. Martinez, R. Beivide, E. Stafford, M. Moretó and E.M. Gabidulin, “Modeling toroidal networks with the Gaussian integers”, *IEEE Transactions on Computers*, vol. 57, no. 8, pp. 1046–1056, Aug. 2008.

[33] B. Parhami and D.-M. Kwai, “A unified formulation of honeycomb and diamond networks”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 1, pp. 74–80, Jan. 2001.

[34] B. Parhami and D.-M. Kwai, “Incomplete k-ary n-cube and its derivatives”, *Journal of Parallel and Distributed Computing*, vol. 64, no. 2, pp. 1232–1239, Feb. 2004.

[35] M.M.H. Rahman, X. Jiang, M.S.-A. Masud and S. Horiguchi, “Network performance of pruned hierarchical torus network”, in: *Proc. 6th IFIP Int. Conf. on Network and Parallel Computing*, pp. 9–15, Oct. 2009.

[36] F.N. Sibai, “A two-dimensional low-diameter scalable on-chip network for interconnecting thousands of cores”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 2, pp. 193–201, Feb. 2012.

[37] Y.-H. Teng, J.J.M. Tan and L.-H. Hsu, “Honeycomb rectangular disks”, *Parallel Computing*, vol. 31, no. 3–4, pp. 371–388, Feb. 2005.

[38] A. Touzene, K. Day and B. Monien, “Edge-disjoint spanning trees for the generalized butterfly networks and their applications”, *Journal of Parallel and Distributed Computing*, vol. 65, no. 11, pp. 1384–1396, Nov.2005.

[39] P.M. Weichsel, “The Kronecker product of graphs”, *Proceedings of the American Mathematical Society*, vol. 13, no. 1, pp. 47–52, Feb. 1962.

[40] W. Xiao and B. Parhami, “Structural properties of Cayley digraphs with applications to mesh and pruned torus interconnection networks”, *Journal of Computer and System Sciences*, vol. 73, no. 8, pp. 1232–1239, Dec. 2007.

[41] W. Xiao and B. Parhami, “A group construction method with applications to deriving pruned interconnection networks”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 5, pp. 637–643, May 2007.

[42] J. Xu, *Topological Structure and Analysis of Interconnection Networks*, Kluwer (2001).

[43] Z. Zhang, Some properties in hexagonal torus as Cayley graph, *Proc. of Int. Conf. on Intelligent Computing and Information Science*, pp. 422-428, Jan. 2011.

[44] S. Zhou, “A new family of trivalent Cayley networks on wreath product $Z_m \wr S_n$”, *Journal of Systems Science and Complexity*, vol. 19, no. 4, pp. 577–585, Dec. 2006.

Iain A. Stewart received the MA and PhD degrees in mathematics from the University of Oxford in 1983 and the University of London in 1986, respectively. He is currently a Professor in the School of Engineering and Computing Sciences, Durham University, United Kingdom. Previously, he was a Lecturer at the University of Newcastle upon Tyne, a Reader at the University of Wales Swansea, and a Professor at the University of Leicester, United Kingdom. His research interests lie in interconnection networks for parallel and distributed computing, graph theory and algorithms, computational complexity and finite model theory, theoretical aspects of artificial intelligence and group theory.