Optimal Broadcasting in Toroidal Networks

Izidor Jerebic, Roman Trobec
Department for Computer Networks and Digital Communications
Jožef Stefan Institute, Jamova 39
61 111 Ljubljana, Slovenia

Abstract

In this paper we study routing algorithms for broadcasting (sending data from one to all other processors) in multiprocessors, whose interconnection networks have toroidal structure. A toroidal network is an n-dimensional rectangular mesh with additional edge-to-edge connections. Mathematically speaking, these networks are undirected graphs obtained by cartesian product of cycles.

We propose criteria for optimality of algorithms for broadcasting. On the basis of these criteria we analyse two routing algorithms, one known and the other proposed by authors. [1]

1 Introduction

Toroidal interconnection networks are becoming the standard interconnection scheme for multiprocessors [1, 2, 3, 4]. Figure 1 shows a two-dimensional toroidal network.

An extensive comparative analysis of the latency in such networks was done by Dally in [4]. An optimal nondeterministic routing policy was presented by Badr and Podar [5]. A deadlock-free routing algorithm was invented by Dally and Seitz [6] and later generalized by Linder and Harden [7]. The exact lower bound for load in such networks was given by Heydemann et al. [8].

To use the toroidal networks efficiently, one needs good routing algorithms. A routing algorithm is said to be optimal, if it distributes the communication load equally among all communication channels.

Several concurrent algorithms require data from one processor to be sent to all other processors. Such data transfer is called *one-to-all broadcast*, if all processors receive the same data. If each processor is to receive a unique piece of data, the data transfer is called *one-to-all personalized broadcast*. Concurrent data transfers, where each processor sends data to all other processors, are called *all-to-all broadcast* and *all-to-all personalized broadcast*, respectively. Algorithms for optimal broadcasting in hypercubes were analysed by Johnsson and Ho in [9]. Their article also introduced the notation for broadcasting which we adopted. Routing and broadcasting in hexagonal meshes were investigated by Chen et al. in [10].

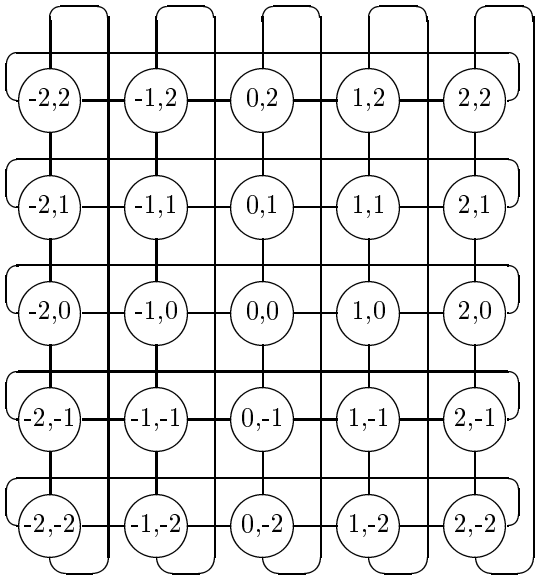

Figure 1: Two-dimensional toroidal network T_5^2.

In this paper, we investigate algorithms for broadcasting in toroidal networks. We develop criteria for the optimality of these algorithms. Using these criteria, we analyse two routing algorithms, one already known and the other proposed by the authors. The proposed algorithm behaves better than the old one and should therefore be used for broadcasting in toroidal networks.

The outline of the paper is as follows. In the next section, notations and definitions used throughout the paper are introduced. Criteria for the optimality of routing algorithms are presented in the section 3. In the section 4 the analysis of two routing algorithms is given. The last section contains a short summary of the results presented in this paper.

Proofs are in most cases omitted because of the limit imposed on the number of pages.

[1] In Proceedings of the International Coference COMPEURO'92, Hague, IEEE Comp. Soc. Press, May 1992, pp. 671–676

2 Definitions

An *interconnection network* is an undirected graph $IN = (V, E)$, where vertices V represent processors, and edges E represent communication channels between processors. We assume that each edge represents two unidirectional channels. Consequently, bidirectional communication, without any impact of the message flow in one direction on the communication in other direction, is possible.

If the interconnection network is toroidal, we can represent vertices of the graph with integer vectors:

$$V \subset \mathcal{Z} \times \mathcal{Z} \times \ldots \times \mathcal{Z},$$

where $\mathcal{Z}$ denotes the set of all whole numbers. We refer to the vector representation of vertices as *points*.

In our discussion we assume the symmetry of the network and denote the set of vertices by T_K^n or shortly T, when it is not necessary to know the size of a network:

$$T_K^n = \underbrace{\mathcal{Z}_K \times \mathcal{Z}_K \times \ldots \times \mathcal{Z}_K}_{n},$$

where $\mathcal{Z}_K$ denotes the set of K consecutive whole numbers, starting at $-\lfloor \frac{K-1}{2} \rfloor$. For simplicity we assume K is an odd integer greater than 2.

Neighbourhood relation

We represented vertices of the graph as n-dimensional integer vectors. What about neighbourhood?

First, we define the addition $\oplus$ and subtraction $\ominus$ as operations in a cyclical group of a size K, where K is the size of the set to which both operands belong. This means that $\lceil \frac{K-1}{2} \rceil \oplus 1 = -\lfloor \frac{K-1}{2} \rfloor$ and $-\lfloor \frac{K-1}{2} \rfloor \ominus 1 = \lceil \frac{K-1}{2} \rceil$.

For the norm of a vector we will use the vector 1-norm:

$$\|A\| = \sum_{i=1}^{n} |a_i|.$$

Now we can express the neighbourhood relation $\sim$ between two vertices, represented with vectors A and B:

$$A \sim B \iff \|A \ominus B\| = 1.$$

2.1 Communication

We assume that processors are able to communicate concurrently on all their connections with neighbours. Transferring a message to a single neighbour takes the same amount of time as transferring m equally sized messages to m different neighbours. In the analysis this time is taken to be the unity time interval. The action of transferring the data is called a *communication step*.

A message from the point A comes to the point B over intermediate points X_i. The ordered sequence P of the points is called a *path* from the point A to the point B:

$$P(A, B) = \langle X_1, X_2, \ldots, X_k \rangle \iff$$

$$(X_i \sim X_{i+1}) \wedge (X_1 = A) \wedge (X_k = B).$$

When a message from A destined for B comes to the intermediate point X_i, it has to be sent to the next point X_{i+1} on the path, except if the point X_i is the destination point B. The decision, to which neighbour this message will be sent, is called a *routing algorithm* R:

$$R : \langle A, B, X_i \rangle \mapsto X_{i+1}.$$

We assume that paths between all points in the network are defined, so messages can be sent from any point to any other point.

We furthermore assume a non-adaptive routing algorithm, what means that there is only one possible path between any two points A and B. All messages from the point A take the same route to the point B.

The *length* L of a path P is the number of components minus one:

$$L(P(A, B)) = k - 1.$$

A unidirectional communication connection from a point X to a neighbouring point Y is called a *channel* and is denoted by $c_{X,Y}$.

A path from A to B can be equivalently expressed as an ordered sequence of channels instead of points:

$$P(A, B) = \langle c_{X_1,X_2}, c_{X_2,X_3}, \ldots, c_{X_{k-1},X_k} \rangle.$$

If a path is expressed with channels, the length of the path is the number of the channels in the path.

The *distance* d between points A and B is the length of a shortest path between these two points. It can be expressed as:

$$d = \sum_{i=1}^{n} |a_i \ominus b_i|.$$

This is in fact the manhattan distance in n dimensions.

Homogeneity

Two paths $P_1 = (X_1, X_2, \ldots, X_p)$ and $P_2 = (Y_1, Y_2, \ldots, Y_q)$ are *congruent* ($P_1 \longleftrightarrow P_2$) iff components of the paths differ by a constant and the numbers of components in the paths are equal:

$$P_1 \longleftrightarrow P_2 \iff (p = q) \wedge$$

$$(\exists D : X_i = Y_i \oplus D, 1 \leq i \leq p).$$

The routing algorithm R is *homogeneous* iff for a path from any point A to any other point B from a network T holds, that it is congruent to the path from the point $A \ominus B$ to the point $\vec{0} = (0, 0, \ldots, 0)$:

$$\forall A, B \in T : P(A, B) \longleftrightarrow P(A \ominus B, \vec{0}).$$

2.2 Communication graph

The *input communication graph* $G_A^{(i)}$ of a routing algorithm R in a toroidal network T, where A is a point in T, is defined as follows:

- graph $G_A^{(i)}$ is an undirected rooted tree with the root A,
- the set of vertices is equal to the set of all points in T:
$$G_A^{(i)} = (T, E_A^{(i)}),$$
- points X and Y in the graph $G_A^{(i)}$ are connected iff there exists a path $P(Z, A)$, defined by the routing algorithm R, which contains a channel between points X and Y:
$$(X, Y) \in E_A^{(i)} \iff$$
$$\exists Z : c_{X,Y} \in P(Z, A) \vee c_{Y,X} \in P(Z, A).$$

Similarly we define the *output communication graph* $G_A^{(o)}$:

- graph $G_A^{(o)}$ is an undirected rooted tree with the root A,
- the set of vertices is equal to the set of all points in T:
$$G_A^{(o)} = (T, E_A^{(o)}),$$
- points X and Y in the graph $G_A^{(o)}$ are connected iff there exists a path $P(A, Z)$, defined by the routing algorithm R, which contains a channel between points X and Y:
$$(X, Y) \in E_A^{(o)} \iff$$
$$\exists Z : c_{X,Y} \in P(A, Z) \vee c_{Y,X} \in P(A, Z).$$

It should be noted, that in general case communication graphs are not trees. But if the routing algorithm considers only current position and final destination of a message as the basis for its decision, its communication graphs are trees. Since all routing algorithms known to authors work this way (it is the only efficient way to build a routing software or hardware), we assume that the routing algorithm is such that the resulting graphs are trees. In addition, if the routing algorithm R is homogeneous, all its input and output communication graphs are isomorphic.

From now on, if we omit the superscript denoting input or output communication graph, we mean input communication graph. All statements using the notation G_A should be understood as they were using $G_A^{(i)}$.

Communication graph and routing

Communication graphs are defined with one-to-one communication paths. We can use these graphs to define one-to-all communication in the following way.

Broadcast from A to all other points proceeds with the help of the communication graf G_A: if an intermediate point X receives a broadcast message, it sends the message to all the neighbours, whose parent in the graph G_A is X.

In the communication one-to-one we also employ communication graphs: if an intermediate point X receives a message, destined for B, it sends the message further to the neighbour Y, which is the parent of the point X in the graph G_B.

In this way, we have a single function to compute for one-to-one, one-to-all, and all-to-all communication: a parent in a communication graph. If this function is simple, we have efficient and fast routing algorithms for every communication problem.

3 Optimality criteria

In this section we are going to establish optimality criteria for one-to-all and all-to-all broadcasting and personalized broadcasting. The routing algorithms are judged according to the number of communication steps necessary for the last processor to receive its data.

We assume all algorithms use only the shortest paths.

3.1 One-to-all communication

The following two optimality criteria are widely known and accepted.

Proposition 1 A routing algorithm R is optimal for one-to-all broadcasting, when it is using only the shortest paths between two points. □

It is said, that the channel $c_{X,Y}$ is in the dimension i, if the difference $X \ominus Y$ has its non-negative component in the dimension i.

Let $S_i(G_A)$ be the i-th complete subtree in the communication graph G_A induced by the routing algorithm R, rooted in the direct child of the root A, where $-n \leq i \leq +n$. Subscripts are chosen according to the dimension and direction of the channel, connecting the root of the subtree with the root of the communication graph.

Definition 1 Define $\Delta(R)$ as the maximal difference between the number of points in $S_i(G_X)$ and the number of points in $S_j(G_X)$ for all points X in the network and all pairs (i, j). □

Proposition 2 A routing algorithm R is optimal for one-to-all personalized broadcasting, iff $\Delta(R) \leq 1$. □

3.2 All-to-all communication

At all-to-all broadcast there is a problem with overloaded channels, because each processor, when receives one message, in most cases sends more messages to its neighbours. The communication will be the fastest if the load is equally distributed among channels. In this way we use all of the network's bandwidth. This requirement is similar to the one at the one-to-all personalized broadcast, except that in this case messages "come into existence" at all stages of the communication and are received from all directions.

From now on, we assume that the routing algorithm used for all-to-all broadcasting is a homogeneous routing algorithm and that broadcasting proceeds as described in section 2, using a communication graph .

Observe closely, what is going on in the point $\vec{0}$ during the communication. Since the network and routing algorithm are homogeneous, the results can be generalized to any other point in the network.

What can we say about the load on the output channels, that is a consequence of the arrival of the messages from certain distance? The connection between this load and the communication graph is given by the following two lemmas.

Lemma 1 If the two paths $P_1 = \langle X_1, X_2, \ldots, X_k \rangle$ and $P_2 = \langle Y_1, Y_2, \ldots, Y_k \rangle$ are congruent, the channels between succesive components of the paths are in the same dimension and in the same direction:

$$P_1 \longleftrightarrow P_2 \Longrightarrow$$

$$X_j - X_{j+1} = Y_j - Y_{j+1}, 1 \leq j < k.$$

□

The points in the communication graph $G_{\vec{0}}$ at different levels (distances from the root, which is at the level 0) are connected with their parents and children via channels (each edge in the communication graph is interpreted as a channel).

Lemma 2 Let the $\rho_{\pm i}(d)$ denote the number of output messages in the point $\vec{0}$, caused by the arrival of broadcast messages from the sources at the distance d, and sent out over the channel in the dimension i and certain direction (positive or negative, according to the sign of subscript). The value of the $\rho_{\pm i}(d)$ is equal to the number of the appearances of a channel in the dimension i and appropriate direction between points at the level d and points at the level $d+1$ in the communication graph $G_{\vec{0}}$. □

It is not easy to count channels between levels of a communication graph. It's easier to count points at the certain level. The following theorem establishes the final value of the $\rho_{\pm i}$.

Theorem 1 The value of $\rho_{\pm i}(d)$ is the number of points at the level $d+1$ in the subtree $S_{\pm i}(G^{(i)}_{\vec{0}})$ of the graph $G^{(i)}_{\vec{0}}$. □

Following our guideline to make the load on channels as equal as possible, we do not need to know the particular values of $\rho_{\pm i}(d)$. It is enough to know the greatest difference between two of them. This difference shows the load distribution over channels.

Definition 2 We define the maximal value of the differences between $\rho_{\pm i}(d)$ and $\rho_{\pm j}(d)$ as $\Lambda(d)$:

$$\Lambda(d) = \max_{i,j} |\rho_{\pm i}(d) - \rho_{\pm j}(d)|.$$

□

Theorem 2 A homogeneous routing algorithm R is optimal for all-to-all broadcasting in a toroidal network T, iff $\Lambda(d) \leq 1$ for all d. □

The analysis of all-to-all personalized broadcast is similar to the analysis of the load in a network in case of one-to-one communication [8]. The discussion and proof for the following theorem is published elsewhere [13].

Theorem 3 A routing algorithm R is optimal for all-to-all personalized broadcast, if it is homogeneous and is using only the shortest paths between two points. □

4 Routing algorithms

In this section we intend to analyse two routing algorithms with the use of criteria, developed in the previous section. The first algorithm is an example of a routing algorithm, which is widely used and popular for its simplicity. As we will see, this simplicity in certain cases causes the algorithm to be non-optimal. Because of this, we propose a new routing algorithm, which is shown to be optimal (considering also the complexity of the algorithm) for all broadcasting problems.

In the analysis we assume that the routing algorithm is homogeneous and therefore we focus on routing towards the point $\vec{0}$ and communication graph $G_{\vec{0}}$.

In the text $\mathrm{sgn}(a)$ denotes the signum function, defined as:

$$\mathrm{sgn}(a) = \begin{cases} +1, & a > 0 \\ 0, & a = 0 \\ -1, & a < 0 \end{cases}$$

4.1 Approach in one dimension first

The routing algorithm, as the title indicates, is decreasing the distance between the message and its final destination in one dimension first, after that in another, and so on until it reaches the destination. The main feature of the routing algorithm is, that the sequence of dimensions is fixed.

Definition 3 (Simple algorithm) Routing from a point X towards $\vec{0}$ proceeds as follows:

- let the point $Y = (y_1, y_2, \ldots, y_n)$ be an intermediate point, where message destined for $\vec{0}$ has just arrived,
- let the r, $1 \leq r \leq n$, be the largest number for which holds $y_r \neq 0$,
- the message is sent to the neighbouring point $Y' = (y_1, y_2, \ldots, y_r - \mathrm{sgn}(y_r), \ldots, y_n)$.

Theorem 4 A routing algorithm R, described by Definition 3, has the value $\Delta(R)$ equal to $\frac{1}{2}(K-1)(K^{n-1}-1)$.

Proof:

In order to prove the theorem, we have to determine the number of points in subtrees S_i of the communication graph $G_{\vec{0}}$.

Denote the neighbours of the point $\vec{0}$ with A_i, $-n \leq i \leq n$:

$$A_i = (a_1, a_2, \ldots, a_n)$$

$$a_k = \begin{cases} 0 & , k \neq i \\ \mathrm{sgn}(i) & , k = i \end{cases}$$

These points are roots of the subtrees S_i.

To find out in which subtree does a point X reside, we have to determine, which of the points A_i is the last point in the path $P(X, \vec{0})$.

Let j be the smallest number such that $x_j \neq 0$. Since the routing algorithm decrements the distance starting with the highest dimension first, we can conclude, that the last point in the path $P(X, \vec{0})$ is the point $A_{j \cdot \mathrm{sgn}(x_j)}$.

Now we can calculate the number N_i of points in a subtree S_i:

$$N_i = |\{X : \quad X = (0, \ldots, 0, x_{|i|}, x_{|i|+1}, \ldots, x_n)$$

$$\wedge\, \mathrm{sgn}(x_{|i|}) = \mathrm{sgn}(i)\}|.$$

In a closed form, the above expression is:

$$N_i = \frac{K-1}{2} K^{n-|i|}.$$

The value $\Delta(R)$ is obviously the difference between N_1 and N_n:

$$\begin{aligned} \Delta(R) &= N_1 - N_n \\ &= \tfrac{1}{2}(K-1)K^{n-1} - \tfrac{1}{2}(K-1) \\ &= \tfrac{1}{2}(K-1)(K^{n-1}-1). \end{aligned}$$

The proof was done for the sequence of dimensions $(n, n-1, \ldots, 1)$. From the course of the proof follows that the obtained value $\Delta(R)$ is valid for any routing algorithm employing a fixed sequence of dimensions. □

4.2 Diagonal algorithm

As we have seen in the previous subsection, routing algorithms with fixed sequence of dimensions do not show optimal behaviour. The next guess would be decreasing the distance so as to approach the destination in a straight line. This approach is reasonably good, if we can solve the main problem: where to route if the distance in more dimensions is equal? The solutions as "take the first" lead to the same poor performance as the algorithm described in the previous subsection.

This problem can be cast as a problem of routing from diagonal points with all coordinates ± 1 towards $\vec{0}$. We propose an algorithm, which distributes routing at diagonal points equally on all dimensions.

Let the function d define the dimension, in which the message will be forwarded from the diagonal point. It is easy to implement it, provided we have a function b, which maps diagonal points in consecutive natural numbers:

$$d(X) = 1 + (b(X) \bmod n).$$

Define the function b as a mapping from diagonal points to natural numbers from 0 to $2^{n-1} - 1$:

$$b(X) = \sum_{k=1}^{n} \frac{1 + \mathrm{sgn}(x_1) \cdot \mathrm{sgn}(x_k)}{2} 2^{n-k}$$

The function b interprets vectors as binary numbers, where positive and negative signs of the components represent binary digits 0 and 1. In addition, vectors X and $-X$ are mapped to the same number, what makes the routing symmetric. The number of messages, routed in a dimension i in the positive direction, is equal to the number of messages routed in the negative direction in the dimension i.

After the routing algorithm picks out the dimension, in which to route a message from a diagonal point, it has to assure, that this initial distribution of messages will continue up to the point $\vec{0}$. It suffices to apply a modified version of the algorithm from the previous subsection. If a point is not a diagonal point, we choose the highest dimension, which has in the lower dimension nearby distance smaller then the maximal distance in a single dimension. The relation "lower" is cyclical, so if the distance in the dimension n is 2, and the distance in the dimension 1 is 3, and 3 is the maximal distance, then the routing dimension becomes 1.

We can imagine this algorithm as extending the area of zeroes in a distance vector from the zero in the highest dimension towards dimension n in a cyclical manner, i.e. after dimension n starting at dimension 1.

The communication graph $G_{\vec{0}}$ for this routing algorithm is shown in Figure 2.

Definition 4 (Diagonal algorithm) Let the $Y = (y_1, y_2, \ldots, y_n)$ be an intermediate point, where message destined for $\vec{0}$ has arrived. The message is sent forward to the point $Y' = (y_1, \ldots, y_r - \mathrm{sgn}(y_r), \ldots, y_n)$. We have two possibilities, how to determine routing dimension r:

1. The point Y is a diagonal point:

   $$|y_1| = |y_2| = \ldots = |y_n|.$$

   In this case, we calculate the routing dimension with the function b:

   $$r = 1 + (b(Y) \bmod n).$$

2. The point Y has m, $m < n$, maximal coordinates:

   $$|x_{i_1}| = |x_{i_2}| = \ldots = |x_{i_m}| = s, 1 \leq j \leq m,$$

   $$s > |x_k|, k \neq i_j, i \leq j \leq m.$$

   Let p be the index of the highest dimension with non-maximal coordinate:

   $$(|x_p| < s) \wedge (|x_i| < s \Longrightarrow i \leq p).$$

   The routing dimension r is determined by the following expression:

   $$r = 1 + (p \bmod n).$$

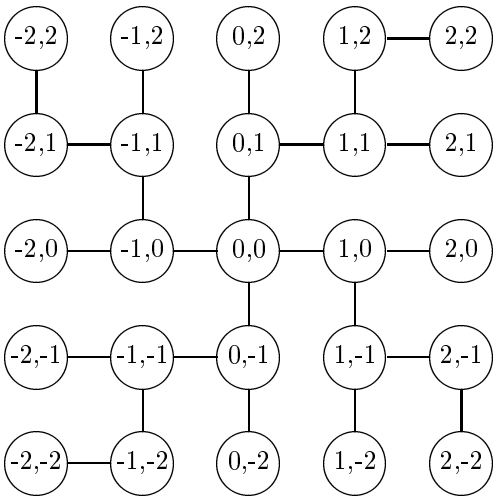

Figure 2: Graph $G_{\vec{0}}$ for algorithm 4

Theorem 5 The routing algorithm R, described by Definition 4, has in a toroidal network T_K^n the value $\Delta(R)$ smaller than $\left(\frac{K-1}{2}\right)^n + \frac{n-2}{2}K^{n-2}$. If n is a power of 2, the first term is dropped, and $\Delta(R)$ is smaller than $\frac{n-2}{2}K^{n-2}$. □

Optimality measure Λ

We did not calculate the measure Λ for the two algorithms, since it can be shown, that Δ and Λ are closely related values, $\Lambda(d)$ being approximately $\frac{\partial \Delta}{\partial K}\big|_{K=d}$.

5 Conclusion

In this paper, we analysed broadcasting in toroidal networks. We have shown, how to apply a single routing function to solve all communication problems with introducing a communication graph.

We also used communication graph as a tool to diskriminate between routing algorithms.

We proposed two measures of optimality: Δ and Λ. From the optimality measures, developed in the paper, it is clear, that the optimal routing algorithm should induce completely balanced communication graphs. Since algorithm from the Definition 3 generates very unbalanced graphs, its performance in broadcasting is poor.

The algorithm from the Definition 4 is shown to exhibit better behaviour, in two dimensions being even optimal. The communication graph is more balanced than predecessor's.

It is very hard to achieve optimality for general n-dimensional networks. Our opinion is, that the optimal routing algorithm would be too complex and time consuming to be implemented in reality. Thus, the proposed algorithm, in spite of its partial optimality, should be sufficient.

References

[1] INMOS ltd., "Transputer," 1989

[2] T.R. Robertazzi, "Toroidal networks," IEEE Comm. Magazine, vol. 26, pp. 45 - 50, June 1988

[3] A.J. Martin, "The TORUS: An exercise in constructing a processing surface," Proc. 2nd Caltech conf. VLSI, pp. 527-537, 1981

[4] R. Suaya and G. Birtwistle, "VLSI and parallel computation," Morgan Kaufman Publishers, Inc., 1990

[5] H.G. Badr and S. Podar, "An optimal shortest-path routing policy for network computers with regular mesh-connected topologies," IEEE Trans. Comp., vol. C-38, pp. 1362 - 1371, Oct. 1989

[6] W.J. Dally and C.L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," IEEE Trans. Comp., vol C-36, pp. 547 - 553, May 1987

[7] D.H. Linder and J.C. Harden, "An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes," IEEE Trans. Comp., vol. C-40, pp. 2 - 12, Jan. 1991

[8] M.C. Heydemann, J.C. Meyer and D. Sotteau, "On forwarding indices of networks," Discr. Appl. Math., vol. 23, pp. 103 - 123, 1989

[9] S.L. Johnsson and C.T. Ho, "Optimum broadcasting and personalized communication in hypercubes," IEEE Trans. Comp., vol. C-38, pp. 1249 - 1267, Sep. 1989

[10] M.S. Chen, K.G. Shin, and D.L. Kandlur, "Addressing, routing, and broadcasting in hexagonal mesh multiprocessors," IEEE Trans. Comp., vol. C-39, pp. 10 - 18, Jan. 1990

[11] R.A. Finkel and M.H. Solomon, "Processor interconnection strategies," IEEE Trans. Comp., vol. C-29, no. 5, pp. 360-370, May 1980

[12] L.D. Wittie, "Communication structures for large networks of microcomputers," IEEE Trans. on Comp., vol C-30, pp. 264-273, Apr. 1981

[13] "Optimal routing in toroidal networks," submitted for publication in Information Processing Letters