Available online at www.sciencedirect.com

SCIENCE DIRECT®

Discrete Applied Mathematics 128 (2003) 435–446

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

All-to-all personalized communication on multistage interconnection networks☆

Annalisa Massini

Dipartimento di Informatica, Università di Roma la Sapienza, Via Salaria 113, 00198 Rome, Italy

Received 29 March 2001; received in revised form 12 June 2002; accepted 24 June 2002

Abstract

In parallel/distributed computing systems, the all-to-all personalized communication (or complete exchange) is required in numerous applications of parallel processing. In this paper, we consider this problem for $\log N$ stage Multistage Interconnection Networks (MINs). It is proved that the set of admissible permutations for a MIN can be partitioned in Latin Squares. Since routing permutations belonging to a Latin Square provides the all-to-all personalized communication, a method to realize the complete exchange with time complexity $O(N)$, that is optimal, can be derived. This method, compared with other ones in literature, does not necessitate of neither pre-computation nor memory allocation to record the Latin Square, because an explicit construction of it is not required; furthermore it is applicable to any $\log N$ stage multistage networks since it is independent of the topology.

© 2003 Elsevier Science B.V. All rights reserved.

Keywords: Multistage interconnection networks; All-to-all personalized communication; Latin Squares

1. Introduction

In a parallel/distributed computing system, processors often need to communicate with each other. According to the number of processors involved, communication can be one-to-one (unicast), one-to-many (multicast), one-to-all (broadcast), and all-to-all.

☆ A preliminary version was published in Proceedings of the International Conference of Communications in Computing (CIC'2000), Las Vegas, Nevada, June 2000.

E-mail address: massini@dsi.uniroma1.it (A. Massini).

0166-218X/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.

PII: S0166-218X(02)00504-8

In all-to-all communication every processor in a processor group sends a message to all other processors in the group. According to type of message to be sent all-to-all communications can be classified in all-to-all broadcast and all-to-all personalized. In all-to-all broadcast every processor sends the same message to all other processors, in all-to-all personalized communication each processor sends a distinct message to every other processor. The all-to-all personalized communication (or complete exchange) is a relevant communication pattern and it plays an important role in many applications such as matrix transposition, fast Fourier transform (FFT) and distributed table lookup.

All-to-all personalized communication problem has been extensively studied for many networks topologies. Many results have been reported for meshes [2,4,6,9] and tori [10,11,8], that are network models with a simple and regular topology, a bounded node degree and present a good scalability. All-to-all personalized communication algorithms with time complexity $O(N^{3/2})$ on 2D meshes and tori and algorithms with time complexity $O(N^{(k+1)/k})$ for k-dimensional meshes and tori, see for example [7], have been proposed. An optimal complete exchange algorithm for an N-node hypercube with $O(N \log N)$ and $O(N)$ time complexity for one-port model and all-port node, respectively, is given in [5]. A drawback of using high-dimensional hypercubes is the unbounded node degree, a feature that implies a poor scalability.

In this paper, we consider as interconnecting scheme for multiprocessor systems multistage interconnection networks (MINs) of size N (with N inputs and N outputs) consisting of $\log N$ stages, each composed of $N/2$ nodes (2×2 switching elements). Examples of topologies for $\log N$ stage MINs are Omega, Flip, Baseline and Reverse Baseline, Butterfly and Reverse Butterfly that are all topologically equivalent [1,3]. $\operatorname{Log} N$ stage MINs are banyan, that is a unique path exists between any input and any output in the network, and present attractive advantages such as efficient routing algorithms, partitionability, small number of switching elements. These MINs are not rearrangeable, that is cannot realize all the $N!$ possible permutations, but only a subset of them, then are suitable for a specialized use, as in the case of the all-to-all personalized communication problem, for which a full permutation capability is not required.

Given N processors, a $\log N$ stage MIN can be used for communication among processors as shown in Fig. 1. This network model has been already considered by Yang and Wang in [13,14,12,15]. Yang and Wang developed a generic method for decomposing all-to-all personalized exchange patterns into some permutations which are realizable in multistage networks and present an optimal algorithm, ATAPE, with $O(N)$ time complexity. Namely the all-to-all personalized communication is realized by routing a set of N permutations forming a Latin Square. The algorithm ATAPE exploits the decomposition of an admissible permutation in basic permutations (interstage permutations and stage permutations). Although Baseline, Omega and indirect binary-cube networks are equivalent, in paper [13] each network is discussed separately, since the algorithm depends on interstage permutations, that is on the network topology. The Latin Square is generated by means of an off-line algorithm run at the time the network is built. Then this realization is specific for the size of the network and provides only one particular Latin Square. The requirement to realize the complete exchange with this method is to keep in memory a matrix of size $N \times N$ containing the destination tags for the N permutations.

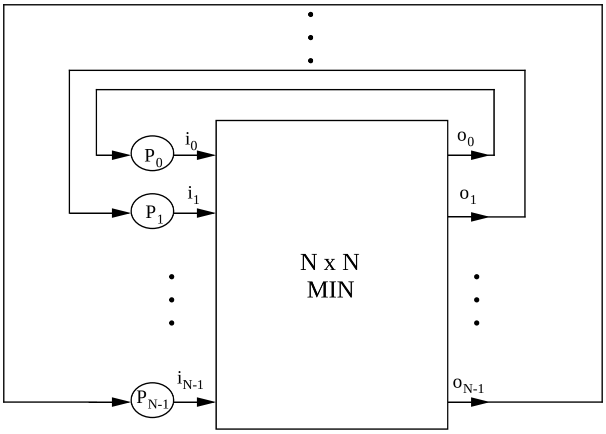

Fig. 1. Communications among N processors using a MIN of size N.

In this work, we prove in a constructive way that the set of admissible permutations for a MIN can be partitioned in sets that are Latin Squares, that is, we provide a method to obtain all the possible Latin Squares. Furthermore, starting from any network configuration we can use this method to generate all the N permutations that form a Latin Square. Then, we propose a realization of all-to-all personalized communication, suitable for any size MIN, that can utilize any Latin Square of the partition. The proposed procedure does not need a pre-computation and does not require the recording of the matrix of permutations to be realized, because an explicit computation of permutations belonging to a Latin Square is not necessary.

2. Preliminary definitions

A $\log N$ *stage* MIN (in the following simply MIN) has N inputs and N outputs and consists of $n = \log N$ stages of $N/2$ nodes that are 2×2 switches. Each node belonging to stage j, $0 < j < \log N - 1$ is connected with two nodes of stage $j - 1$ and two nodes of stage $j + 1$, according to a rule depending on the network topology. Each node in stage $j = 0$ is connected with a pair of inputs and each node in stage $j = \log N - 1$ is connected with a pair of outputs. In Fig. 2 a butterfly of size $N = 8$ is depicted.

Each node of the MIN can assume the *state* straight or cross as shown in Fig. 3.

The *network configuration* of a MIN is defined by the switch setting of its nodes, that is by specifying the state of each node of the network. A given network configuration can be represented as a matrix $M = (m_{h,k})$, $h = 0, \ldots, N/2 - 1$, $k = 0, \ldots, \log N - 1$, in which the entry $m_{h,k}$ represents the hth node in stage k. Entry $m_{h,k}$ belongs to set $\{0, 1\}$ and its value is 0 if the node is set to straight and 1 if the node is set to cross.

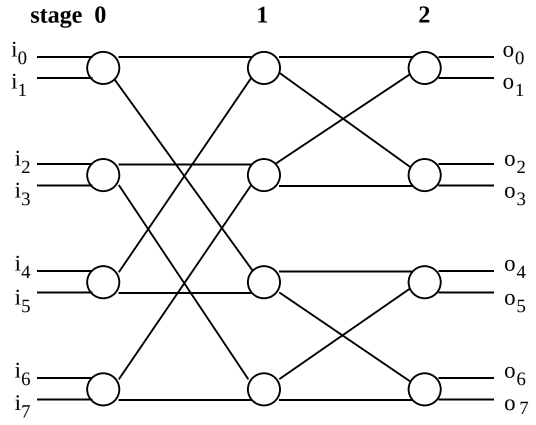

Fig. 2. A Butterfly of size $N = 8$.

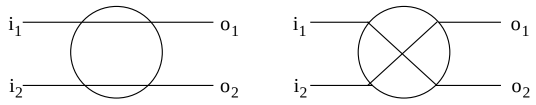

Fig. 3. Set of function of a node (2 × 2 switch).

Permutations realizable by a MIN are called *admissible* permutations. A MIN of size N can realize $2^{(N/2)\log N} = N^{N/2}$ permutations, each corresponding to one of the $2^{(N/2)\log N}$ possible network configurations (since a MIN consists of $(N/2)\log N$ nodes).

A *Latin Square* is defined as an $N \times N$ matrix $A = (a_{i,j})$, $i, j = 0, \ldots, N-1$, where entries $a_{i,j}$ belong to set $\{0, 1, \ldots, N-1\}$ and no two entries in a row or a column have the same value. In particular, for all i and j, $0 \leqslant i,\ j \leqslant N-1$, the entries of each row i in the matrix $a_{i,0}, a_{i,1}, \ldots, a_{i,N-1}$ form a permutation

$$\begin{pmatrix} 0 & 1 & \ldots & N-1 \\ a_{i,0} & a_{i,1} & \ldots & a_{i,N-1} \end{pmatrix}$$

and the entries of each column j in the matrix $a_{0,j},\ a_{1,j}, \ldots, a_{N-1,j}$ form a permutation.

$$\begin{pmatrix} 0 & 1 & \ldots & N-1 \\ a_{0,j} & a_{1,j} & \ldots & a_{N-1,j} \end{pmatrix}$$

In this work, a column of A represents a permutation p realized by the MIN and the elements $a_{i,j}$ of column j, $i = 0, \ldots, N-1$, represent input tags of information arrived

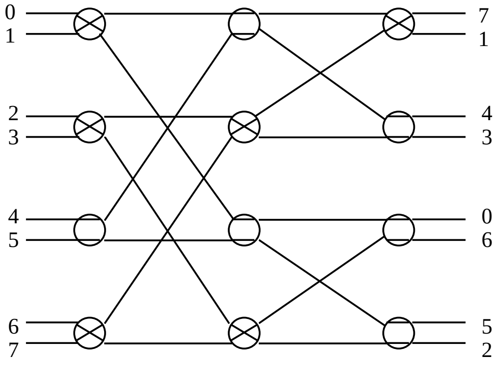

Fig. 4. Example of permutation represented by means of input tags, on a Butterfly of size $N = 8$.

at output i (and not the destination tag as often used), see Fig. 4. Note that, given a permutation, defined in terms of input tags

$$\begin{pmatrix} 0 & 1 & \dots & N-1 \\ p(0) & p(1) & \dots & p(N-1) \end{pmatrix}$$

it is possible to obtain the representation with destination tags by simply computing the composition $p(p(i))$ (and in the same way it is possible to obtain the input tag representation from the destination tag one).

The realization of the N permutations belonging to a Latin Square by means of a MIN provides the all-to-all personalized communication; the lower bound on the maximum communication delay is given by the following lemma [13]:

Lemma 1. *The maximum communication delay of all-to-all personalized communication in a* $\log N$ *stage MIN of size* N *is at least* $\Omega(N + \log N)$.

In fact each message must pass through $\log N$ stages from the source processor to the destination processor and each processor must receive one message from all other $N - 1$ processors.

In the following section, we prove that the set of admissible permutations P for a MIN can be partitioned in $\frac{2^{(N/2)\log N}}{N} = N^{(N/2)-1}$ sets P^l, $l = 0, \dots, N^{(N/2)-1} - 1$, each consisting of N permutations in such a way permutations belonging to a set P^l form a *Latin Square*, as illustrated in the example below for a MIN with $N = 4$.

Example 1. Two ways of partitioning admissible permutations for a Butterfly of size $N = 4$ are shown in the following tables. The binary matrix representing the network configuration that produces each permutation, is specified. Each column of tables is a Latin Square.

Partition 1

0213	00 00	0231	00 01	0312	00 10	0321	00 11
2031	01 01	2013	01 00	3021	01 11	3012	01 10
1302	10 10	1320	10 11	1203	10 00	1230	10 01
3120	11 11	3102	11 10	2130	11 01	2103	11 00

Partition 2

0213	00 00	0231	00 01	0312	00 10	0321	00 11
2031	01 01	2013	01 00	3021	01 11	3012	01 10
1320	10 11	1302	10 10	1230	10 01	1203	10 00
3102	11 10	3120	11 11	2103	11 00	2130	11 01

3. Canonical partition of admissible permutations in Latin Squares

In this section, we describe a method to obtain a partition of set P of admissible permutations for a MIN in sets P^l, $l = 0,\dots,N^{(N/2)-1} - 1$, and we prove that the sets obtained are Latin Squares. We call the partition obtained with this method *canonical*. In the example above the canonical partition is *Partition 1*.

The method provides a way to generate all the sets P^l, $l = 0,\dots,N^{(N/2)-1} - 1$, sequentially with respect to l, starting from $l = 0$.

Let S be the set of all possible binary matrices of size $N/2 \times \log N$, and let S^l be the set of binary matrices that produce permutations belonging to the set P^l. The method gives the N binary matrices providing the network configurations that produce the N permutations belonging to a set P^l, that is for each $l = 0,\dots,N^{(N/2)-1} - 1$ provides the set S^l.

Note that any permutation corresponds to one and only one network configuration, then there is a one-to-one mapping between elements of P (sets P^l) and elements of S (sets S^l) and between elements of P^l (permutations) and sets S^l (matrices), then it is equivalent to refer to P^l or S^l.

3.1. LS construction method

Once the index l of the set S^l to be built is chosen, one of the N binary matrices belonging to S^l is implicitly fixed, namely the matrix which sequence of rows $r_0, r_1, \dots, r_{(N/2)-1}$ provides the binary representation of l. Since the matrix size is $(N/2) \times \log N$ and $l = 0,\dots, \frac{2^{(N/2)\log N}}{N} - 1 = N^{(N/2)-1} - 1$, this matrix has its first row consisting of 0's. Let us denote it $M^{l,0}$, and let $M^{l,x}$, $x=1,\dots,N-1$, be the other matrices belonging to S^l.

To generate all the sets S^l, we start from $l=0$ and for each value of l we determine the matrix $M^{l,0}$ from the binary representation of l, then we construct the matrices belonging to S^l performing the *XOR phase* described below, by using $M^{l,0}$.

XOR phase: The XOR phase consists of $N-1$ steps, *XOR steps*, each of which produces a matrix $M^{l,x}$, $x=1,\dots,N-1$. Let $x_{\log N-1}\dots x_1x_0$ be the binary representation of x.

XOR step x: Each row i of $M^{l,x}$, $r_i^{l,x}$, is obtained from row i of $M^{l,0}$, $r_i^{l,0}$, as

$$r_i^{l,x}=r_i^{l,0}\ \text{XOR}\ x_{\log N-1}\dots x_1x_0$$

or, equivalently, each entry $m_{i,j}^{l,x}$ of matrix $M^{l,x}$ is obtained from entry $m_{i,j}^{l,0}$ of matrix $M^{l,0}$ as

$$m_{i,j}^{l,x}=m_{i,j}^{l,0}\ \text{XOR}\ x_{\log N-1-j}.$$

The XOR operation performed in the XOR step implies that column j of matrix $M^{l,0}$ is flipped if bit $x_{\log N-1-j}$ of the binary representation of x is 1.

By applying the *XOR step* for all possible value of x from 1 up to $N-1$, to a given matrix $M^{l,0}$, the XOR phase is performed and all matrices belonging to the set S^l are generated. By executing the *XOR phase* for $l=0$ up to $N^{(N/2)-1}-1$ all the sets S^l partitioning S are obtained.

Note that, since the first row of $M^{l,0}$ consists of all 0s, the first row of $M^{l,x}$ provide the binary representation of x. For this reason and for the meaning of l in $M^{l,0}$, we call this partition *canonical.*

Example 2. Two elements of the partition obtained with the LS construction method in the case $N=8$ are shown in the tables on the following page. These tables show P^l and S^l for $l=18$ and $l=235$ respectively.

Lemma 2. *The LS construction method provides a partition of P as* $P=\{P^l \mid l=0,\dots,N^{(N/2)-1}-1\}$ *by partitioning the set S of binary matrices of size* $N/2\times\log N$ *as* $S=\{S^l \mid l=0,\dots,N^{(N/2)-1}-1\}$.

Proof. Sets S^l, $l=0,\dots,N^{(N/2)-1}-1$ are generated starting from $l=0$. By definition, matrix $M^{l,0}$, which row sequence $r_0r_1\dots r_{(N/2)-1}$ provides the binary representation of l, belongs to S^l. The remaining $N-1$ elements $M^{l,x}$, $x=1,\dots,N-1$, of S^l are generated sequentially starting from $x=1$.

By varying l from 0 up to $N^{(N/2)-1}-1$ and x from 1 up to $N-1$, it is guaranteed that all the possible binary matrices of size $N/2\times\log N$ are generated.

It is obvious that if $l_1\neq l_2$, $0\leqslant l_1,l_2\leqslant N^{(N/2)-1}-1$, then $M^{l_1,0}\neq M^{l_2,0}$.

To generate $M^{l,x}\in S^l$, $x=1,\dots,N-1$, the logical operation XOR between all rows of $M^{l,0}$ and the binary representation of x is performed bitwise. It follows that if $x_1\neq x_2$ then $M^{l,x_1}\neq M^{l,x_2}$. Therefore, elements belonging to S^l are all different, that is elements in P^l are all different.

P^{18} and S^{18}

02465713	000 000 010 010
20647531	001 001 011 011
46021357	010 010 000 000
64203175	011 011 001 001
13574602	100 100 110 110
31756420	101 101 111 111
57130246	110 110 100 100
75312064	111 111 101 101

P^{235} and S^{235}

06257134	000 011 101 011
60521743	001 010 100 010
52603471	010 001 111 001
25064317	011 000 110 000
17346025	100 111 001 111
71430652	101 110 000 110
43712560	110 101 011 101
34175206	111 100 010 100

Furthermore, basing on properties of binary representation and logical operation XOR we have that $M^{l_1,x_1} = M^{l_2,x_2}$ if and only if $l_1 = l_2$ and $x_1 = x_2$. Then a matrix $M^{l,x}$ can belong only to one set S^l. Hence, by applying this method a partition of S, and consequently a partition of P, is obtained. □

Lemma 3. *Permutations belonging to set P^l, obtained by means of network configurations given by binary matrices in S^l, form a Latin Square, for any $l = 0,\dots, N^{(N/2)-1} - 1$.*

Proof. The set P^l can be represented as a matrix A^l of size $N \times N$ which columns are the N permutations in P^l. To prove A^l is a Latin Square we have to prove that any row and any column is a permutation.

Columns correspond to permutations by definition.

Row i of A^l, $i=0,\dots,N-1$, represents the sequence of input tags of information arrived on output i for each of the N permutations belonging to P^l; row i is a permutation if any element $a_{i,h}\in\{0,\dots,N-1\}$, $h=1,\dots,N$, appears only once. Due to the banyan property of $\log N$ stage MINs, an information reaches its destination by means of a unique path; a path is given by the sequence of crossed nodes and their state (straight or cross). Due to the fact that matrices $M^{l,x}$, $x=0,\dots,N-1$, belonging to the set S^l are all different and, in particular, corresponding rows of all matrices in a set S^l are different, since they are obtained by means of the application of the XOR step, (i.e. $r_i^{l,x_1}\neq r_i^{l,x_2}$ for any $x_1,x_2\in\{0,\dots,N-1\}$—see proof of Lemma 2) we have that N different paths arriving to output i are defined, that is N different starting inputs are used to reach output i. Then, any row in A^l is a permutation.

Hence matrix A^l is a Latin Square. □

From Lemmas 2 and 3 the following theorem immediately derives:

Theorem 1. *The LS construction method gives a partition of the set P of admissible permutations for a MIN in Latin Squares.*

It is interesting to observe that if the construction of only one Latin Square is required, it is not necessary to derive all the Latin Squares of the partition. This is stated in the following theorem that gives a way to obtain a Latin Square starting from any network configuration.

Theorem 2. *Given any binary matrix M of size $N/2\times\log N$, the set S^l to which it belongs to can be obtained by applying to it the XOR phase of the LS construction method.*

Proof. Due to properties of the logical operation XOR, all elements of S^l can be generated by applying the XOR phase to the given binary matrix M. Note that the binary representation of index l is provided by the XOR between the sequence $r_0r_1\dots r_{(N/2)-1}$ of rows of the given matrix M and the sequence $r_0r_0\dots r_0$, where r_0 appears $N/2$ times, performed bitwise. $M^{l,0}$ is obtained when the XOR step with $x=r_0$ (r_0 is the first row of the given matrix M) is performed. Due to properties of the logical operation XOR, all elements of S^l can be generated by applying the XOR phase to the given binary matrix. □

Observe that if an admissible permutation for the MIN is given and we want to construct the Latin Square to which it belongs to, we can easily derive the network configuration that produce that permutation by using a self-routing algorithm and then apply the XOR step for $x=1,\dots,N-1$.

4. Realizing all-to-all personalized communication

The realization of the all-to-all personalized communication on a MIN can be obtained by realizing the N permutations belonging to any of the sets P^l of the partition. Then, it is not necessary to realize a particular Latin Square, that means it is not necessary to compute and record the N permutations belonging to a particular P^l.

In view of Theorem 2 all binary matrices producing permutations of a Latin Square can be generated starting from any given matrix applying to it the XOR phase of the LS construction method. Since a binary matrix represents a network configuration, Theorem 2 can be used to derive an implementation method for the all-to-all personalized communication.

For the sake of homogeneity, we can generalize the XOR phase by performing the XOR step also for $x = 0$, since this operation leaves the binary matrix (network configuration) unchanged.

To generate the N network configurations needed to realize the N permutations of a Latin Square (and implementing the all-to-all personalized communication), the self-routing capability of MINs is not used, but switches are set according to the value obtained from the XOR between a given initial network configuration and the binary representation of numbers $0, 1, \ldots, N-1$, performed sequentially.

All-to-all personalized communication network procedure

- The binary representations of numbers $0, 1, \ldots, N-1$ are sequentially generated and messages starting from every input of the MIN are equipped with the current binary representation.
- When information passes through a node of stage $\log N - 1 - j$, the switch is set to straight or cross according to the value, 0 or 1, respectively, of the XOR between the binary value associated with the switch itself and the jth bit of the binary representation associated with the information.
- When information leaves the switch, it is necessary to reconfigure the switch to its initial value, because for each new binary representation considered, the XOR between it and the value of the initial switch configuration must be computed; then a further application of the XOR with the already used binary representation is needed to reconfigure the switch to its initial value.

The information flux passes through the stages of the network in a synchronous way, then when N messages leave a stage, other new N messages can enter the switches of this stage, that is the N permutations can be realized in pipeline fashion and the procedure proposed for the all-to-all personalized communication problem takes $O(N + \log N) = O(N)$ time, that is optimal.

In Table 1 (see also [13]) the time complexity for all-to-all personalized communication, the node degree, the diameter and the topological complexity (number of nodes) for different network models are shown.

From the table, one can see that MINs and hypercubes achieve the minimum time complexity, but MINs present the advantage to have a bounded node degree that reflects a better scalability of this network model.

Table 1
Comparison of different network models

Network model	Node degree	Diameter	Topol. compl.	Time compl.
Hypercube one-port	$\log N$	$\log N$	$O(N)$	$O(N \log N)$
Hypercube all-port	$\log N$	$\log N$	$O(N)$	$O(N)$
2D mesh/torus	4	$O(N^{(1/2)})$	$O(N^2)$	$O(N^{(3/2)})$
3D mesh/torus	6	$O(N^{(1/3)})$	$O(N^3)$	$O(N^{(4/3)})$
MIN	4	$\log N$	$O(N \log N)$	$O(N)$

5. Conclusions

In this work an optimal procedure for the all-to-all personalized communication problem on $\log N$ stage MINs has been proposed. These MINs are network models suitable for interprocessor communication (if a complete permutation capability is not required), for the short latency time, due to their moderate depth, and for their scalability.

The LS construction method described in Section 3.1 provides a partition of admissible permutations for $\log N$ stage MINs in Latin Squares. Since a Latin Square represents a set of permutations which realization provides the all-to-all personalized communication, from this method we derive a simple procedure to configure the network to obtain this complete exchange. Starting from any network configuration, it is possible to realize the N permutations forming a Latin Square by setting the switches of the MIN by performing the logical XOR between the initial node configuration and the binary representation of numbers from 0 the $N-1$.

This method, compared with that presented in [13,14,12,15], does not necessitate of neither pre-computation nor memory allocation for a pre-computed Latin Square, because an explicit construction of it is not required. Furthermore, algorithms described in [13] provide only one Latin Square (corresponding to set P^0 obtained with the LS construction method), whereas the LS construction method gives all the possible Latin Squares obtainable from admissible permutations for a MIN. Finally, each of the equivalent $\log N$ stage banyan MINs requires an “ad hoc” decomposition in basic permutations, since the algorithm to obtain the Latin Square depends on the network topology. On the contrary, the algorithm derived from the LS construction method presented in this work is applicable on any $\log N$ stage banyan MINs.

Finally, as shown in Example 1, the partition of admissible permutations in Latin Squares is not unique, then could be interesting to find other way to obtain partitions of admissible permutations for $\log N$ stage banyan MIN in Latin Squares.

References

[1] J.C. Bermond, J.M. Fourneau, A. Jean-Marie, Equivalence of multistage interconnection networks, Inform. Proc. Lett. 26 (1987) 45–50.

[2] S.H. Bokhari, H. Berryman, Complete exchange on a circuit switched mesh, Proceedings of the Scalable High Performance Computing Conference, Williamsburg, VA, 1992, pp. 300–306.

[3] T. Calamoneri, A. Massini, Efficiently checking the equivalence of multistage interconnection networks, Proceedings of the Parallel and Distributed Computing and Systems (PDCS'99), Cambridge, MA, 1999, pp. 23–30.

[4] S. Gupta, S. Hawkinson, B. Baxter, A binary interleaved algorithm for complete exchange on a mesh architecture, Technical Report, Intel Corporation, 1994.

[5] S.L. Johnsson, C.T. Ho, Optimum broadcasting and personalized communication in hypercubes, IEEE Trans. Comput. 38 (1989) 1249–1268.

[6] D.S. Scott, Efficient all-to-all communication pattern in hypercube and mesh topologies, Proceedings of the IEEE Distributed Memory Conference, Portland, OR, 1991, pp. 398–403.

[7] Y.J. Suh, K.G. Shin, Efficient all-to-all personalized exchange in multidimensional torus networks, Proceedings of the 1998 International Conference on Parallel Processing, Minneapolis, MN, 1998, pp. 468–475.

[8] Y.J. Suh, S. Yalmanchili, All-to-all communication with minimum start-up costs in 2D/3D tori and meshes, IEEE Trans. Parallel Distrib. Systems 9 (1998) 442–458.

[9] N.S. Sundar, D.N. Jayasimha, D.K. Panda, P. Sadayappan, Complete Exchange in 2D Meshes, Proceedings of the Scalable High Performance Computing Conference, Knoxville, TN, 1994, pp. 406–413.

[10] Y.-C. Tseng, S. Gupta, All-to-all personalized communication in wormhole-routed torus, IEEE Trans. Parallel Distrib. Systems 7 (1996) 498–505.

[11] Y.-C. Tseng, T.-H. Lin, S. Gupta, D.K. Panda, Bandwidth-optimal complete exchange on wormhole-routed 2D/3D torus networks: a diagonal-propagation approach, IEEE Trans. Parallel Distrib. Systems 8 (1997) 380–396.

[12] Y. Yang, J. Wang, All-to-all personalized exchange in banyan networks, Proceedings of the Parallel and Distributed Computing and Systems (PDCS'99), Cambridge, MA, 1999, pp. 78–86.

[13] Y. Yang, J. Wang, Optimal all-to-all personalized exchange in self-routable multistage networks, IEEE Trans. Parallel Distrib. Systems 11 (2000) 261–274.

[14] Y. Yang, J. Wang, Optimal all-to-all personalized exchange in multistage networks, Proceedings of the Seventh International Conference on Parallel and Distributed Systems (ICPADS'00), Iwale, Japan, 2000.

[15] Y. Yang, J. Wang, Optimal all-to-all personalized exchange in a class of optical multistage networks, IEEE Trans. Parallel Distrib. Systems 12 (2001) 567–582.