All-to-All Personalized Communication on Multistage Interconnection Networks

Annalisa Massini
Dipartimento di Scienze dell'Informazione, Università di Roma la Sapienza
Via Salaria 113, 00198 Roma, Italy
e-mail: massini@dsi.uniroma1.it

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). We prove that the set of admissible permutations for a MIN can be partitioned in Latin Squares. Routing permutations belonging to a Latin Square of the partition provides the all-to-all personalized communication. From the method of partitioning the set of admissible permutations, we derive a procedure to realize the complete exchange with optimal time complexity,* $O(N)$*. The implementation proposed, compared with other ones in literature, does not need either pre-computation or memory allocation to record the Latin Square, because an explicit construction of the Latin Square is not required.*

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. In all-to-all communication every processor in a processor group sends a message to all other processors in the group. In particular, in all-to-all personalized communication every 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 [1, 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. Algorithms with time complexity $O(N^{\frac{3}{2}})$ and $O(N^{\frac{k+1}{k}})$ for 2-dimensional and k-dimensional meshes/tori respectively [6, 7, 8, 10, 11], have been proposed. In [5], 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. 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 a multiprocessor system (see Fig. 1), 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 [2, 3]. $\log N$ 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

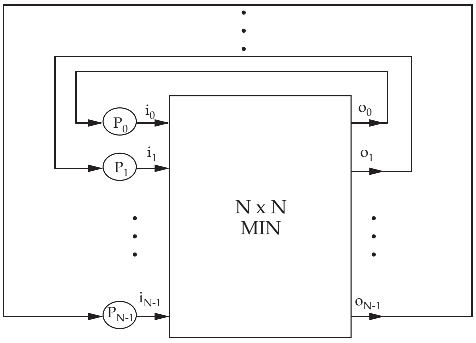

Figure 1: Communications among N processors using a MIN of size N.

realize all the $N!$ possible permutations, but only a subset of them (admissible permutations, 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.

MINs have been already considered for the all-to-all personalized communication among N processors in [12]. By routing a set of N permutations forming a Latin Square, the complete exchange is realized in optimal time $O(N)$. The permutations belonging to the Latin Square are obtained by means of an off-line algorithm run at the time the network is built. Then, the method proposed in [12] provides only one particular Latin Square for each network size, and the requirement to realize the complete exchange is to keep in memory a matrix of size $N \times N$ containing the destination tags for the N permutations.

In this work we prove that the set of admissible permutations for a MIN can be partitioned in subsets that are Latin Squares, that is we provide a method to obtain all the possible Latin Squares. Then we propose a realization of all-to-all personalized communication, that can utilize any Latin Square of the partition and is suitable for any size MIN. 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 $\log N$ stage MINs and Latin Squares

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, $0 < j < N-1$ 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 $j = N-1$ is connected with a pair of outputs. Each node of the MIN can be set to straight or cross. A MIN of size N can realize $2^{\frac{N}{2}\log N} = N^{\frac{N}{2}}$ permutations, called *admissible* permutations for the MIN (since it consists of $\frac{N}{2}\log N$ nodes), each corresponding to one of the $2^{\frac{N}{2}\log N}$ possible *network configurations* of the MIN, defined by the switch setting of its node. Let us associate to each node a bit which value is 0 if the node is set to straight and 1 if the node is set to cross. Then a given network configuration can be represented as a matrix $M = (m_{h,k})$, $h = 0, \ldots, \frac{N}{2} - 1$, $k = 0, \ldots, \log N - 1$, which entries $m_{h,k}$ belong to set $\{0, 1\}$.

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

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 at output i (and not the destination tag as often used), see Fig. 2.

The realization of the N permutations belonging to a Latin Square by means of a MIN provide the all-to-all personalized communi-

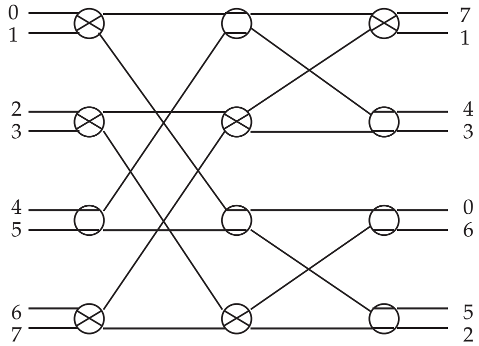

Figure 2: Example of permutationrepresented by means of input tags, on a Butterfly of size $N = 8$.

cation. The lower bound on the maximum communication delay is given by the following lemma [12]:

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

Infact 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: a) the set of admissible permutations, P, for a MIN can be partitioned in $\frac{2^{\frac{N}{2}\log N}}{N} = N^{\frac{N}{2}-1}$ sets, P^l, $l = 0, \ldots, N^{\frac{N}{2}-1} - 1$, each consisting of N permutations; b) permutations belonging to a set P^l form a *Latin Square.*

Example 1 In the following tables two ways of partitioning admissible permutations for a Butterfly of size $N = 4$ are shown. Each row of tables is a Latin Square; for each permutation the binary matrix, representing the network configuration producing it, is specified.

Partition 1

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

Partition 2

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

3 Canonical partition of admissible permutations in Latin Squares

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

With this method we obtain the N binary matrices providing the network configurations that produce the N permutations belonging to a set P^l. To this end, we indicate with S the set of all possible binary matrices of size $\frac{N}{2} \times \log N$, and with S^l the set of binary matrices that produce permutations belonging to the set P^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 and elements of S and between sets P^l and sets S^l, then it is equivalent to refer to P^l or S^l.

3.1 LS Construction Method

When the index l of the set S^l to be built is fixed, even one of the binary matrices belonging to S^l is implicitly fixed, and it is the matrix which sequence of rows $r_0, r_1, \ldots, r_{\frac{N}{2}-1}$ provide the binary representation of l. Since a matrix in S^l has $\frac{N}{2}$ rows and $\log N$ columns, and $l = 0, \ldots, \frac{2^{\frac{N}{2}\log N}}{N} - 1 = N^{\frac{N}{2}-1} - 1$, then the matrix fixed in S^l has its first row consisting of 0s. Let it be $M^{l,0}$.

XOR phase: Let $M^{l,x}$, $x = 1, \ldots, N-1$, be the other matrices belonging to S^l and let $x_{\log N-1} \ldots x_1 x_0$ be the binary representation of x. The XOR phase consists of $N-1$ steps, each of which produces a matrix $M^{l,x}$, $x = 1, \ldots, N-1$.

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

$$r_i^{l,x} = r_i^{l,0} \ \text{XOR} \ x_{\log N-1} \ldots x_1 x_0$$

or, equivalently, 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

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

This XOR operation performed in the XOR step x 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. 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.*

Hence, by applying the **XOR step** sequentially 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.

Example 2 Two elements of the partition obtained with the LS Construction Method in the case $N = 8$. Both P^l and S^l are shown in the following tables.

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

Lemma 2 *The LS Construction Method provides a partition of* P *as* $P = \{P^l | l = 0, \ldots, N^{\frac{N}{2}-1} - 1\}$*, by partitioning the set* S *of binary matrices of size* $\frac{N}{2} \times \log N$ *as* $S = \{S^l | l = 0, \ldots, N^{\frac{N}{2}-1} - 1\}$

Proof. Sets S^l, $l = 0, \ldots, N^{\frac{N}{2}-1} - 1$ are generated sequentially starting from $l = 0$. By definition, matrix $M^{l,0}$, which row sequence $r_0 r_1 \ldots r_{\frac{N}{2}-1}$ provides the binary representation of l, belongs to S^l. The remaining $N-1$

elements $M^{l,x}$, $x = 1, \ldots, N-1$, of S^l are generated ordinately starting from $x = 1$. It is guaranteed that, by varing l from 0 up to $N^{\frac{N}{2}-1} - 1$ and x from 1 up to $N-1$, all the possible binary matrices of size $\frac{N}{2} \times \log N$ are generated.

It is obvious that if $l_1 \neq l_2$, $0 \leq l_1, l_2 \leq N^{\frac{N}{2}-1} - 1$, then $M^{l_1,0} \neq M^{l_2,0}$.

To generate $M^{l,x} \in S^l$, $x = 1, \ldots, N-1$, 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.

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 consenquently of P, is obtained. Q.E.D.

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 $l = 0, \ldots, N^{\frac{N}{2}-1} - 1$.*

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, $i = 0, \ldots, N-1$, 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 $a_{i,h} \in \{0, \ldots, N-1\}$, $h = 1, \ldots, N$, appears only once. Due to the banyan property of $\log N$ stage MINs, an information reaches its destination by means of a unique path given by the sequence of nodes crossed and their state (straight or cross). Since matrices $M^{l,x}$, $x = 0, \ldots, N-1$, belonging to the set S^l are all different, then N different paths arriving to output i are defined, that is N different starting inputs are used to reach output i. Then any row is a permutation.

Hence matrix A^l is a Latin Square. Q.E.D.

From Lemma 2 Lemma 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.*

The following theorem provides a way to obtain a Latin Square starting from any of its element.

Theorem 2 *Given any binary matrix of size $\frac{N}{2} \times \log N$, the set S^l to which belongs to can be obtained by applying to it the* **XOR phase** *of the LS Construction Method.*

Proof. The binary representation of index l is provided by the XOR between the sequence of $r_0 r_1 \ldots r_{\frac{N}{2}-1}$ of rows of the given matrix and the sequence $r_0 r_0 \ldots r_0$, where r_0 appears $\frac{N}{2}$ times, performed bitwise. 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. Q.E.D.

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 is to compute and record the N permutations belonging to 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 that 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 betweeen 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;

- 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 j-th bit of the binary representation associated with the information;

- when the 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 pass 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.

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	$O(N^{\frac{1}{2}})$	$O(N^2)$	$O(N^{\frac{3}{2}})$
3D mesh/torus	6	$O(N^{\frac{1}{3}})$	$O(N^3)$	$O(N^{\frac{4}{3}})$
MIN	4	$\log N$	$O(N \log N)$	$O(N)$

Table 1: Comparison of different network models.

In Table 1 (see also [12]) 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.

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 network procedure. 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 in a suitable way 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 [12], does not necessitate of either precomputation or memory allocation for a precomputed Latin Square, because an explicit construction of it is not required. Furthermore, algorithms described in [12] 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.

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.

References

[1] S. H. Bokhari and H. Berryman, "Complete Exchange on a Circuit Switched Mesh", *Proc. Scalable High Performance Computing Conf.*, 300-306, 1992.

[2] J. C. Bermond, J. M. Fourneau and A. Jean-Marie, "Equivalence of Multistage Interconnection Networks", *Inform. Proc. Letters*, 26, 45-50, 1987.

[3] T. Calamoneri and A. Massini, "Efficiently Checking the Equivalence of Multistage Interconnection Networks", *Proc. Parallel and Distributed Computing and Systems (PDCS'99)*, 23-30, 1999.

[4] S. Gupta, S. Hawkinson and B. Baxter, "A Binary Interleaved Algorithm for Complete Exchange on a Mesh Architecture", Technical Report, Intel Corporation, 1994.

[5] S .L. Johnsson and C. T. Ho, "Optimum Broadcasting and Personalized Communication in Hypercubes", *IEEE Trans. Computers*, C38, 1249-1268, 1989.

[6] D. S. Scott, "Efficient All-to-All Communication Pattern in Hypercube and Mesh Topologies", *Proc. IEEE Distributed Memory Conf.*, 398-403, 1991.

[7] Y. J. Suh and K. G. Shin, "Efficient All-to-All Personalized Exchange in Multidimensional Torus Networks", *Proc. 1998 International Conference on Parallel Processing*, 468-475, 1998.

[8] Y. J. Suh and S. Yalmanchili, "All-to-All Communication with Minimum Start-up Costs in 2D/3D Tori and Meshes", *IEEE Trans. Parallel and Distributed Systems*, 9, 442-458, 1998.

[9] N. S. Sundar, D. N. Jayasimha, D. K. Panda and P. Sadayappan, "Complete Exchange in 2D Meshes", *Proc. Scalable High Performance Computing Conf.*, 406-413, 1994.

[10] Y. –C. Tseng and S. Gupta, "All-to-All Personalized Communication in Wormhole-Routed Torus", *IEEE Trans. Parallel and Distributed Systems*, 7, 498-505, 1996.

[11] Y. –C. Tseng, T. –H. Lin, S. Gupta and D. K. Panda, "Bandwidth-Optimal Complete Exchange on Wormhole-Routed 2D/3D Torus Networks: a Diagonal-Propagation Approach", *IEEE Trans. Parallel and Distributed Systems*, 8, 380-396, 1997.

[12] Y. Yang and J. Wang, "All-to-All Personalized Exchange in Banyan Networks", *Proc. Parallel and Distributed Computing and Systems (PDCS'99)*, 78-86, 1999.