Adaptive routing on the Recursive Diagonal Torus

A. Funahashi[1] and T.Hanawa[1] and T.Kudoh[2] and H. A mano[1]

[1] Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223 Japan
[2] Real World Computing Partnership, 1-6-1 Takezono, Tsukuba, Ibaraki, 305 Japan

Abstract. Recursive Diagonal Torus, or RDT consisting of recursively structured tori is an interconnection network for massively parallel computers. By adding remote links to the diagonal directions of the torus network recursively, the diameter can be reduced within $log_2 N$ with smaller number of links than that of hypercube.
For an interconnection network for massively parallel computers, a routing algorithm which can bypass a faulty or congested node are essential. Although the conventional vector routing is a simple and near-optimal method, it can only use a deterministic path. In this paper, adaptive routing algorithms on RDT are proposed and discussed. The first algorithm is based on Duato's necessary and sufficient condition. With this method virtual channels are effectively used while paths with redundant routing steps are prohibited. Another algorithm based on the turn model is proposed. By prohibiting certain turns on RDT, it permits paths with additional hops. Both algorithms are proved to be deadlock free.

1 Introduction

Communication network is one of the critical components of a highly parallel multicomputer. Recently, multicomputers providing more than a thousand computation nodes are commercially available, and efforts have been exerted to implement Massively Parallel Computers (MPCs) with tens of thousands nodes. In these systems, the connection topology often dominates the system performance.

Instead of hypercube used in first-generation multicomputers, most recent machines take the 2-D or 3-D mesh (torus) network[1][2][3]. Although the diameter of a mesh network is large ($O(\sqrt{M})$ or $O(\sqrt[3]{M})$ for M nodes), it only requires four or six links per node unlike the hypercube which requires $log_2 M$ links per node.

However, in an MPC with more than ten thousands nodes, the large diameter of the mesh network is intolerable. To address this problem, we proposed a novel extension of mesh network called Recursive Diagonal Torus (RDT) [4], which consists of recursively structured mesh (torus) connection. It supports a smaller diameter and degree than that of the hypercube if the number of nodes is 1000-10000. Through the computer simulation, the bandwidth and latency are much improved compared with 2-D/3-D tori [4]. The router chip providing the vector routing algorithm with multicasting was implemented for a massively parallel machine JUMP-1[5].

In this paper, deadlock-free adaptive routing algorithms on RDT are proposed. In Section 2, the structure of RDT and the vector routing algorithm are briefly introduced. An adaptive routing using minimal paths based on Duato's method is proposed in Section 3. More flexible routing algorithm based on the turn model is also proposed in Section 4.

2 Interconnection Network: RDT

Recursive Diagonal Torus (RDT) is a novel class of networks which consists of recursively structured mesh (torus) connections of meshes with different sizes in the diagonal directions[4][6].

When four links are added between node (x, y) and nodes $(x \pm n, y \pm n)$ (n: *cardinal number*) respectively, additional links result in a new torus-like network. New torus-like network is formed at an angle of 45 degrees to the original torus, and the grid size is $\sqrt{2}$n times that of the original torus. We call this new torus-like network the rank-1 torus. On the rank-1 torus, we can form another torus-like network (rank-2 torus) by providing additional links in the same manner. Figure 1 shows rank-1 and rank-2 tori when n is 2. The RDT consists of such recursively formed tori.

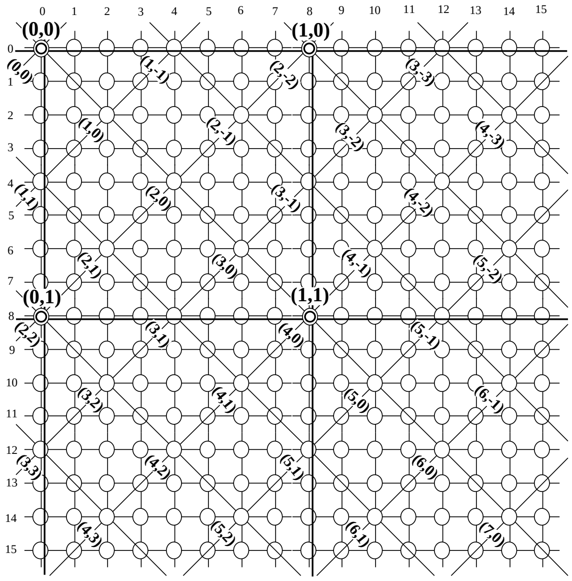


Fig. 1. Upper rank tori

RDT(n,R,m) can be defined as a class of networks in which each node has links to form base (rank-0) torus and m upper tori (the maximum rank is R) with cardinal number n. Note that, each node can select different rank of upper tori from others.

The RDT in which every node has links to form all possible upper tori is called the perfect RDT (PRDT(n,R)) where n is the cardinal number (usually, 2) and R is the maximum rank. Although PRDT is unrealistic due to its large degree (4(R+1)), it is important as the basis for establishing routing algorithm, broadcasting/multicasting, and other message transfer algorithms.

For a system with thousand of nodes, the RDT whose degree is 8 and the maximum rank of upper tori is 4, that is, RDT(2,4,1) is suitable. In the RDT, each node can select different rank tori from others. Thus, the structure of the RDT(2,4,1) also varies with the rank of tori which are assigned to each node. This assignment is called the *torus assignment*. Various torus assignment strategies can be selected considering the traffic of the network.

2.1 The vector routing

The vector routing is an assignment independent of routing algorithm which represents the route of a message with a combination of unit vectors each of which corresponds to each rank of tori.

On the torus structure, a vector from a source node to the destination node is represented with a vector $\mathbf{A} = a\mathbf{x_0} + b\mathbf{y_0}$ where $\mathbf{x_0}$ and $\mathbf{y_0}$ are unit vectors of the base (rank-0) torus. The goal of the routing algorithm is to represent the vector $\mathbf{A}$ with a combination of vectors each of which corresponds to a unit vector of each rank of torus.

First, the direction of the unit vector corresponding to each rank torus must be defined. Here, the direction of the unit vector for each rank torus is changed clockwise at an angle of 45 degrees. That is, the unit vectors of rank-(i+1) torus $\mathbf{x}_{i+1}, \mathbf{y}_{i+1}$ can be represented with the unit vectors of rank-i $\mathbf{x_i}, \mathbf{y_i}$ as follows:

$$\mathbf{x}_{i+1} = n\mathbf{x_i} + n\mathbf{y_i} \tag{1}$$

$$\mathbf{y}_{i+1} = -n\mathbf{x_i} + n\mathbf{y_i} \tag{2}$$

First, the target vector $a\mathbf{x_0} + b\mathbf{y_0}$ is represented with a combination of $\mathbf{x_1}$,$\mathbf{y_1}$,$\mathbf{x_0}$ and $\mathbf{y_0}$ as follows:

$$a\mathbf{x_0} + b\mathbf{y_0} = g\mathbf{x_1} + f\mathbf{y_1} + j\mathbf{x_0} + k\mathbf{y_0} \tag{3}$$

Here, we select maximum g and f in order to use the upper torus as possible. From equations (1) and (2), maximum integers for g and f are represented as follows:

$$g = \frac{a+b}{2n}, f = -\frac{a-b}{2n}$$

In order to minimize j and k corresponding to the remaining unit vectors of the rank-0 torus (thus, the required message transfers using the rank-0 torus), the integer divisor used here is rounded to the nearest whole number (If the remainder is greater than n, increment the divisor).

Thus, j and k are represented with g and f:

$$a\mathbf{x_0} + b\mathbf{y_0} = g(n\mathbf{x_0} + n\mathbf{y_0}) + f(-n\mathbf{x_0} + n\mathbf{y_0}) + j\mathbf{x_0} + k\mathbf{y_0}$$

$$a = ng - nf + j, b = ng + nf + k$$

$$j = a - ng + nf, k = b - ng - nf$$

.

Then, $g\mathbf{x_1} + f\mathbf{y_1}$ are represented with a combination of vector $\mathbf{x_2}$, $\mathbf{y_2}$, $\mathbf{x_1}$, and $\mathbf{y_1}$ in the same manner. By iterating this process to the maximum rank, vectors for message routing are obtained.

The routing vectors for each rank are obtained in the array $vector[rank]$.

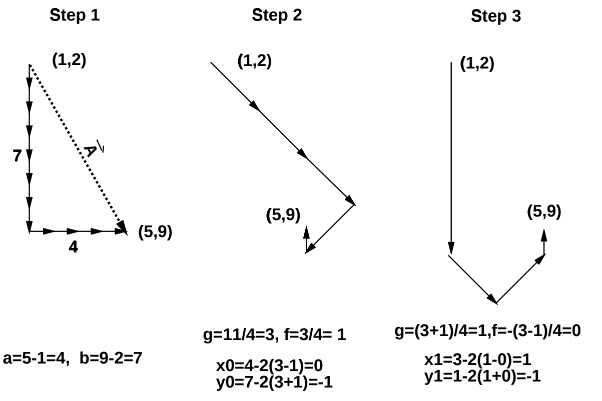


Fig. 2. An example of the vector conversion

Figure 2 shows an example of a vector from (1,2) to (5,9) converted into a combination of unit vectors of rank-0, rank-1, and rank-2.

3 Adaptive routing with minimal paths

Adaptive routing is a technique to select the route of packet dynamically. When a packet encounters a faulty or congested node, the packet can select another bypassing route. The vector routing is useful for a basis of an adaptive routing, since as alternative routes can be easily obtained by changing the order of vectors.However, we must not forget that an adaptive routing has a possibility of deadlock. There are a lot of researches on deadlock free adaptive routing techniques[7]. These techniques are classified into two methods: using only minimal paths, and using alternative paths with additional routing steps. The former method does not require extra routings while the latter can use alternative routes more flexibly. First, deadlock free adaptive routings with minimal routes are proposed for the RDT based on Duato's protocol. Then, another algorithm which permits redundant routing steps is proposed based on the turn model.

3.1 Duato's protocol in the k-ary n-cube

Duato states a general theorem defining a criterion for deadlock freedom and then uses the theorem to propose a fully adaptive, profitable, progressive protocol[8], called Duato's protocol (DP). The theorem states that by separating virtual channels on a link into restricted and unrestricted partitions, a fully adaptive routing can be performed and yet be deadlock-free. This is not restricted to a particular topology or routing algorithm. Cyclic dependencies between channels are allowed, provided that there exists a connected channel subset free of cyclic dependencies.

Simple description of Duato's protocol is as follows.

a. Provide that every packet can always find a path toward its destination whose channels are not involved in cyclic dependencies(escape path).
b. Guarantee that every packet can send to any destination node using escape path and the other path which cyclic dependency is broken by escape path.

By selecting these two routes a. and b. adaptively, deadlock can be prevented. Duato applied this method to the k-ary n-cube[9].

3.2 Applying Duato's protocol on PRDT

Here, we apply this routing algorithm for PRDT.

Definition 1. : Duato's protocol on PRDT

1. Provide an escape path C_1 on a torus of PRDT as well as the case for the k-ary n-cube.
2. Next, the order of rank usage is restricted. Let X_i and Y_i be channel of each dimension in the rank i torus. Use the channel in the X first and descending order of the rank. That is, for PRDT(2,4), the channel is used in the following order
   $X_3 \rightarrow Y_3 \rightarrow X_2 \rightarrow Y_2 \rightarrow X_1 \rightarrow Y_1$
   We refer this escape path C_1'.
3. Add a new virtual channel $C_F(Fully\ \ adaptive)$ which is used for the fully adaptive routing. There are two algorithms: D-A and D-B.
   Algorithm: D-A
   Provide the virtual channel C_F directly for the escape channel C_1'. In C_F, each direction of $+X$ and $+Y$ in odd rank and even rank must be the same direction. In the vector routing, the unit vector for each rank torus

is changed clockwise at an angle of 45 degree as represented in function(1) and function(2), the unit vector for odd rank torus must be same direction with the unit vector for rank 0 torus $(\mathbf{x}_0, \mathbf{y}_0)$ and the unit vector for even rank torus must be the same with the one for rank 1 torus$(\mathbf{x}_1, \mathbf{y}_1)$.

Algorithm: D-B

Provide the virtual channel C_{Fn} not for C_1' but for C_1 in each rank. C_{Fn} channels can cross dimensions in any order following a minimal path, but must cross ranks in descending order.

□

Figure 3 illustrates the fully adaptive virtual channel C_F in Algorithm D-A. Since C_F is directly assigned to the escape path C_1', the C_F itself must be a minimal routing. This means that a packet must not use the opposite direction which used in the past.

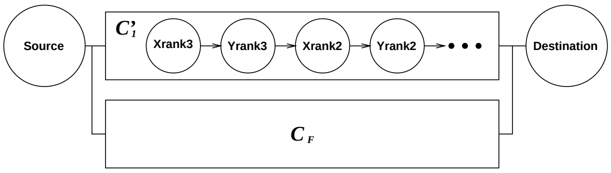


Fig. 3. Channels using Algorithm D-A

On the other hand, the fully adaptive path is assigned to the escape path C_1 of each rank in Algorithm D-B(Figure4). Therefore, there is no restriction for using unit vector, while the order of ranks is restricted.

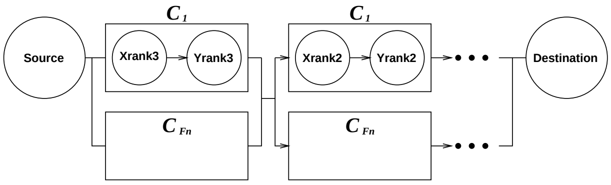


Fig. 4. Channels using algorithm D-B

Figure5 illustrates the possible path and impossible path for algorithm D-A and D-B. The path (b) which uses rank 2 before rank 3 is allowed in the algorithm D-A while it is prohibited in the algorithm D-B, since the rank is not be used in the descending order. On the contrary, path (c) in which the unit vectors of rank 1 and rank 3 are directed opposite to each other is prohibited in the algorithm D-A but allowed in the algorithm D-B.

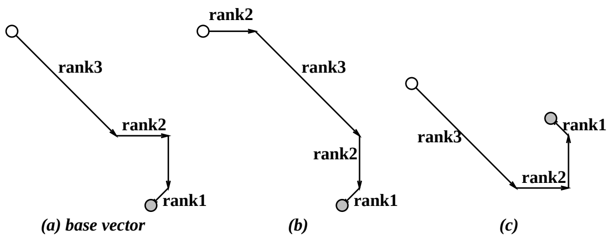


Fig. 5. Examples of vectors in algorithm D-A and D-B

Theorem 2. *Algorithm D-A is deadlock-free.* □

Proof Since the order of the rank is the same as that of the e-cube routing[10], the escape path $C'1$ is deadlock free. In C_F, the opposite direction which used in the past is prohibited, and so C_F is a minimal path. From Duato's theorem[9], Algorithm D-A is deadlock-free. □

Theorem 3. *Algorithm D-B is deadlock-free.* □

Proof C_1 is the same escape path used in Duato's protocol, and is deadlock free. C_{Fn} is a minimal path in each torus. From Duato's theorem[9], Algorithm D-B is deadlock-free in each rank of torus. Since the order of used rank is the same as the e-cube routing, C_1 nor C_{Fn} in any rank does not cause a cycle each other. Therefore, Algorithm D-B is deadlock-free.□

4 Adaptive routing based on the turn model

Although Duato's protocol is powerful approach for bypassing the congestion, only minimal paths can be used. For selecting paths with additional steps, another adaptive routing based on the turn model[11] is proposed here.

4.1 Turn model for Two-Dimensional Meshes

Deadlock in the wormhole routing is caused by message packets waiting for each other in a cycle. The turn model proposed by Glass is a method which prevents deadlock by prohibiting certain turns.

For two-dimensional meshes, Figure6(a) shows the possible turns and simple cycles. Deadlock can be prevented by prohibiting only one turn from each cycle,

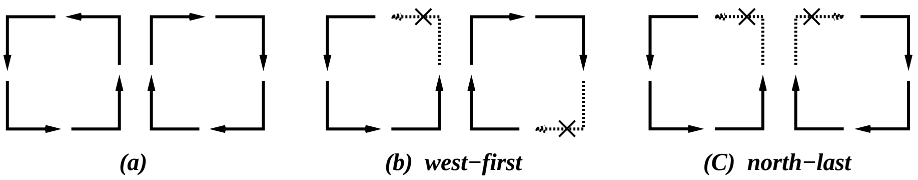


Fig. 6. The turn model for two-dimensional meshes.

as shown in Figure6(b),(c). These routing algorithms are called the west-first routing algorithm and north-last routing algorithm, respectively. Although this model is for a simple mesh network without cyclic links, it is easily used in the torus by introducing extra channels like the e-cube routing.

4.2 The turn model for RDT

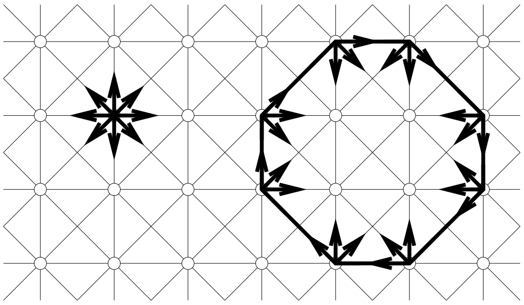


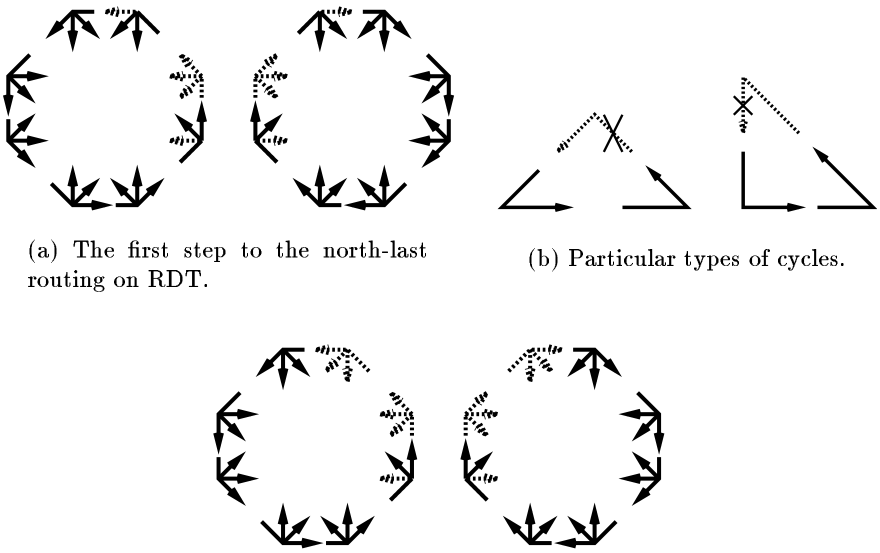
Fig. 7. The possible turns and simple cycles in RDT.

Here, we extend the turn model for two-dimensional meshes of the RDT. The possible turns in RDT are expressed in Figure7. As shown in Figure7, there are eight different directions in the RDT, so there exists sixteen 45-degree turns, sixteen 90-degree turns and sixteen 135-degree turns.

Here, like the north-last routing algorithm for two dimensional mesh, the right top turns and left top turns of cycles are prohibited as shown in Figure8(a).

However, these restrictions are not sufficient for RDT. Cycles without left top turns or right top turns are still possible as shown Figure8(b). In order to break such triangle cycles, dotted turns shown in Figure8(c) must be prohibited. As a result, the following turns are prohibited in RDT.

Definition 4. : North-last routing for PRDT



(a) The first step to the north-last routing on RDT.

(b) Particular types of cycles.

(c) Completed form of the north-last routing on RDT.

Fig. 8. The north-last routing on RDT.

North-last routing for PRDT is a routing in which fourteen turns shown in Figure4.2 are prohibited. A packet transfer through cyclic links is also prohibited. □

As well as the turn model for two dimensional torus, cyclic links can be used by introducing an extra channel for the e-cube routing. Also, this routing can be directly applied for any type of RDT including RDT(2,4,1)/α.

For showing that the proposed north-last routing algorithm for RDT is deadlock free, the channel numbering method by Dally and Seitz[10] is applied. In this method, channels in the direct network is numbered so that every packet is transferred along channels with strictly increasing (or decreasing) numbers. If such a numbering is possible, it shows that there is no cyclic path between buffers in channels.

Theorem 5. *The north-last routing for RDT is deadlock-free.* □

Proof Assuming that the size of the base torus of RDT is $m \times n$. Assign two dimensional number of channel from a node (x, y) according to its direction as shown in Figure9, and let the unique number of the channel be $c_x \times m + c_y$.

Since the size of the base torus of RDT is $m \times n$, the range of the possible channel number (c_x, c_y) is represented by the following equations.

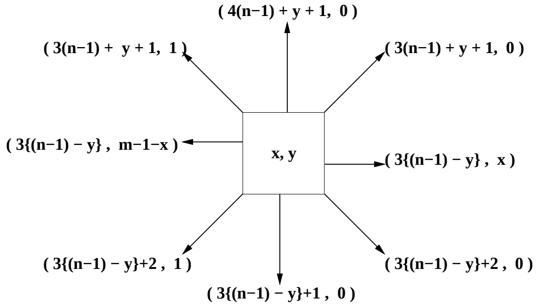


Fig. 9. Numbering of the channels leaving each node (x, y) for the north-last routing algorithm for RDT.

$$0 \le c_x \le 5(n-1)$$
$$0 \le c_y \le m-2$$

In RDT, there are eight possible input directions. As shown in Figure 10, all possible output channel numbers are larger than the number of input channel. In other words, the packet transfer to an output channel whose number is less than input channel is prohibited by the Definition 2 within the range shown in the above equations.

Therefore, channels are used in the increasing order on RDT.□

Figure11 shows an example of routing on the 4×4 RDT. The blocked channels are bypassed with a path consisting of channels in increasing order. This figure also shows that the number of permitted output channel is lager than that of input channel.

5 Conclusion

Two adaptive routing algorithms on RDT are proposed and proved to be deadlock-free. A simulation study which demonstrates the effect of the proposed routing algorithm is required.

References

1. *Paragon XP/S Product Overview.* Intel Corp., 1991.
2. W. J. Dally A. Chien S. Fiske W. Horwat J. Kenn M. Larivee R. Lethin P. Nuth and S. Wills. The J-machine: A Fine-Grain Concurrent Computer. In *IFIP 11th Computer Congress*, pages 1147–1153, August 1989.

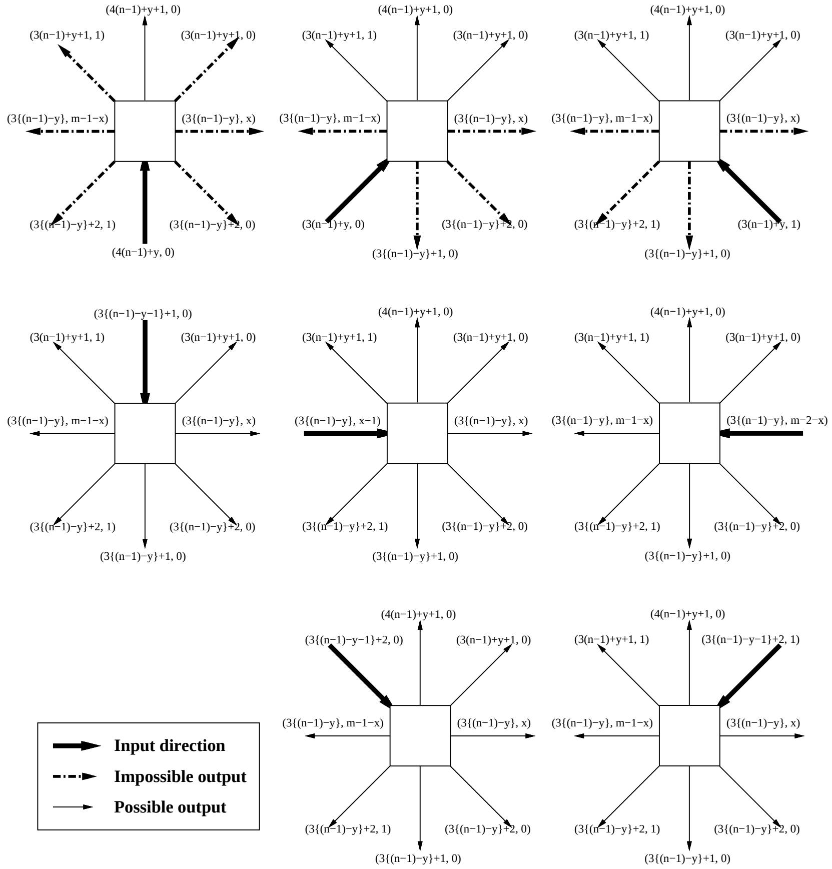


Fig. 10. The possible output channels for each input channel.

3. H. Ishihata T. Horie S. Inano T. Shimizu S. Kato and M. Ikesaka. Third Generation Message Passing Computer AP1000. In *International Symposium on Supercomputing*, pages 46–55, November 1991.
4. Y. Yang, H. Amano, H. Shibamura, and T. Sueyoshi. Recursive diagonal torus: An interconnection network for massively parallel computers. *Proceedings of IEEE SPDP*, 1993.
5. H. Nishi, K. Nishimura, K. Anjo, H. Amano, and T. Kudoh. The JUMP-1 router chip: The versatile router for supporting distributed shared memory. *Proceedings of International Phoenix conference on computers and communications*, 1996.
6. Y. Yang and H. Amano. Message Transfer Algorithms on the RDT. *IEICE Transaction on Information and Systems*, 79(2), 1996.
7. L. M. Ni and P. K. McKinley. A Survey of Wormhole Routing Techniques in Direct Networks. *IEEE Transactions on Computers*, February 1993.

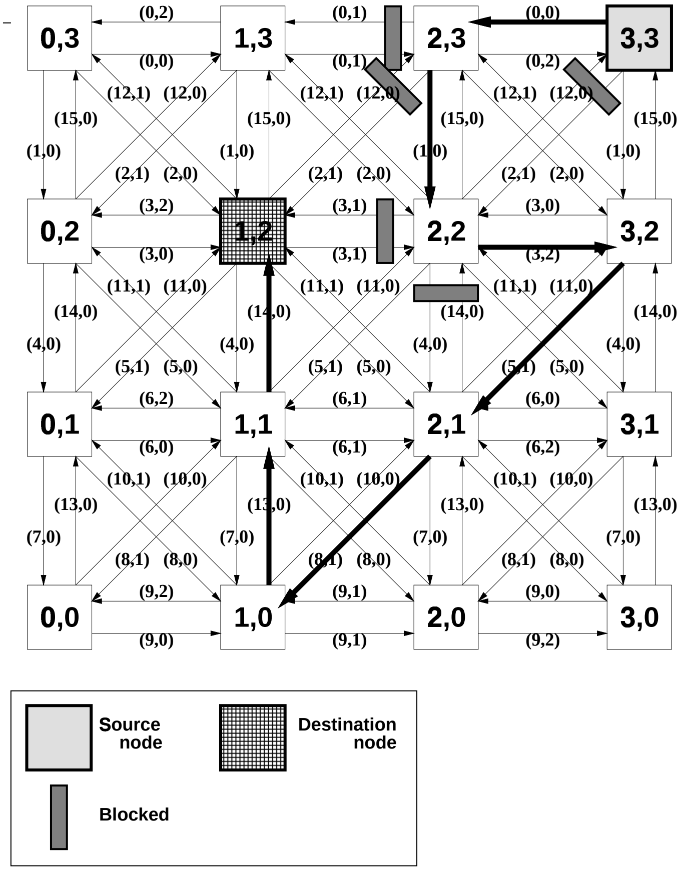


Fig. 11. Example of north-last routing for RDT$(m = 4, n = 4)$.

8. J. Duato. A Necessary And Sufficient Condition For Deadlock-Free Adaptive Routing In Wormhole Networks. *Proceedings of the International Conference on Parallel Processing*, 1:142–149, 1994.
9. J. Duato. A Necessary And Sufficient Condition For Deadlock-Free Adaptive Routing In Wormhole Networks. *IEEE Transaction on Parallel and Distributed Systems*, 6(10), 1995.
10. W. J. Dally and C. L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *IEEE Transactions on Computers*, 36(5):547–553, May 1987.
11. C. J. Glass and L. M. Ni. Maximally Fully Adaptive Routing in 2D Meshes. *Proceedings of ISCA92*, pages 278–287, 1992.

This article was processed using the LaTeX macro package with LLNCS style