

# A NEW DEADLOCK-FREE FAULT-TOLERANT ROUTING ALGORITHM FOR NOC INTERCONNECTIONS

*Slaviša Jovanović, Camel Tanougast, Serge Weber*

*Christophe Bobda*

Laboratoire d'instrumentation électronique de Nancy - LIEN  
Université Henri Poincaré  
Vandœuvre lès Nancy, France  
email: slavisa.jovanovic@lien.uhp-nancy.fr

Universität Potsdam  
Institut für Informatik  
Potsdam, Germany  
bobda@cs.uni-potsdam.de

## ABSTRACT

In this paper, we present a new deadlock-free fault-tolerant adaptive routing algorithm for the 2D mesh NoC interconnections. The main contribution of this routing algorithm is that it allows both, routing of messages in the networks incorporating the regions not necessarily rectangular, and routing to all nodes which are not completely blocked by faulty nodes. The proposed routing algorithm is based on a modified *turn model* and well known XY algorithm. We detail the basic principle of this routing algorithm, prove its deadlock freeness, its feasibility and efficiency through the simulation results.

*Index Terms* - Network-on-chip (NoCs), Fault - tolerance, Reconfigurable systems, Adaptive - Routing, Wormhole - Routing

## 1. INTRODUCTION

The growing chip complexity and the need to integrate more and more components on the chip (processors, DSP cores or memories) imposes on the designers to change their traditional design concepts. Thus, in the new SoC designs the network-centric approaches called Network-on-chips (NoCs) replace progressively shared arbitrated bus [1, 2]. The NoCs are characterized by high level of parallelism, high performances and scalability. Moreover, the NoCs support modularity and IP reuse via standard interfaces and present a natural architectural choice for multi-core processor systems.

Communication in the NoCs between the source and destination components takes place via data packets. The paths that the data packets take in the network depend on two basic NoC parameters: topology and routing algorithm. Several network topologies have been proposed in literature. The frequently used topologies are mesh, torus, hypercube and butterfly [2, 3, 4]. Depending on the degree of adaptiveness, we distinguish three categories of the routing algorithms: non-adaptive, partially-adaptive and fully-adaptive routing algorithms. The non adaptive routing algorithms or often called *deterministic* routing algorithms route the data packet via only one path between the source and destination components. This form of routing has been popular due to its simple deadlock avoidance algorithm, resulting in a simple router implementation. In the partially-adaptive routing algorithms, between the source and destination components, a set of available paths can be used for the packet routing but not for all packets. It happens that some

data packets can use only one path from the source to destination as in the non-adaptive routing algorithms. The partially routing algorithms produce a routing with limited adaptivity and establish a balance between performances and router complexity. In the fully adaptive algorithms, the communication between the components takes place via any available path for all packets. The fully adaptive routing algorithms make the potential for the better use of all available network resources to the detriment of the router complexity.

Efficient routing of messages within networks is essential in order to achieve good performances for applications running on them. However, faulty nodes and especially regions (area in the networks having a size larger than a tile) make the communication in the networks more difficult, even impossible for existing routing algorithms. This is the main reason why a new class of fault-tolerant algorithms has been introduced. The most of fault-tolerant routing algorithms could be used with modifications in the networks incorporating regions. Moreover, with the advent of the reconfigurable technology (i.e. FPGA-based reconfigurable systems), the regions can be placed not only statically at a compile time, but also dynamically at a run-time [5, 6]. To handle the routing of messages in these mixed networks what is needed is an adaptive routing algorithm allowing getting around faulty nodes or regions (placed statically or dynamically).

In this paper, we describe a new routing algorithm for 2D mesh NoCs topologies which allows at the same time routing of packets in presence of faulty nodes and regions not necessarily rectangular. The regions can be created statically at a compile time or placed dynamically at a run-time. This is the reason why this routing algorithm can be used also for the dynamically reconfigurable NoCs. The proposed routing algorithm is deadlock free and based on a well known XY and turn model routing schemes. In the case of faulty nodes or regions, the routing algorithm helps the data packets to get around it.

This paper is organized as follows. Section 2 gives an overview on existing routing algorithms. Section 3 details the proposed routing algorithm and proves its deadlock freeness. In Section 4 the proposed routing algorithm is evaluated through simulations. Future work and some conclusions are both given in Section 5.

## 2. RELATED WORK

An example of the non-adaptive routing algorithms is the XY algorithm. In this algorithm, the data packet is routed firstly along the X-dimension and then along the Y-dimension until the packet reaches its final destination. The path between the source and destination used by this algorithm is one of the shortest ones but always the same. The XY algorithm cannot handle with faulty nodes and regions. If a faulty node (region) is placed on the path between the source and destination, the packet will remain blocked in one of the switches.

Examples of the partially-adaptive routing algorithms are the *turn model* type [7, 8]. In the algorithms based on the *turn model*, in order to avoid deadlock situations, some turns are prohibited. This is the main reason why partially-adaptive algorithms cannot route packets along every shortest path. Fully adaptive routing algorithms allow routing of the data packets along any of the shortest paths. Unfortunately, these algorithms achieve their adaptiveness and deadlock freeness at the expense of adding physical or virtual channels. The fully adaptive routing algorithms provide routing adaptability allowing only avoidance of congested links [9, 10]. Neither partially nor fully adaptive routing algorithms can handle with faulty nodes or regions without being modified.

In literature, several fault-tolerant routing algorithms have been presented. These routing algorithms can be adapted for the routing in the networks with regions. A fault-tolerant partially adaptive routing algorithm based on the negative-first algorithm is presented in [11]. This routing algorithm is made for  $n$ -dimensionally meshes and can tolerate up to  $(n - 1)$  faults. It does not use virtual channels. The major drawback of this routing algorithm is that it cannot always route the packets around multiple faulty nodes. Another fault-tolerant partially adaptive routing algorithm based on *odd-even turn model* is presented in [12]. This algorithm routes the data packets around the faulty blocks, the rectangular structures which are composed of the faulty and *unsafe* nodes. All data packets are routed around these faulty blocks. The non-faulty unsafe nodes in faulty blocks are not reachable. The similar routing approaches using virtual channels can be found in [13], where the faulty nodes are combined in *f-chains* and *f-rings* around which the data packets are routed. A similar fault-tolerant approach that uses *f-chains* and *f-rings* for packet routing around faulty regions is presented in [14]. Other examples of routing algorithms using virtual channels can be found in [15, 16]. The major drawback of the approaches using virtual channels is the additional logic for switching between virtual channels. The switches using virtual channels are more susceptible to faults. More logic blocks cause a bigger probability of fault making in circuits. Another drawback is that the fault of a physical link leads to a disuse of all non-faulty virtual channels that belong to that physical channel.

A fault-tolerant fully adaptive routing algorithm, called *the force-directed wormhole routing - FDWR*, uses routing tables for data packets routing from the source to destination [17]. Each node has the routing table which contains all routing information towards all other nodes in the network. The routing tables are built up in the *initialization phase*. When it receives a header flit, a node transfers it to its neighbors, which send back a *confirmation* if in the routing table exists an available path to destination indicated in the header flit. Thus, the node decides which path the packet will

take. In order to distribute the packets uniformly, all nodes have a *penalty table* where one can find measure values for each outgoing port. That way, the overloaded links are avoided. The main drawback of this approach is that the packets sometimes do not arrive at a good order and need reordering at the destination node. Moreover, the deadlock-free property for this routing algorithm is not proven.

Another class of routing algorithms treating the traffic distribution over the network as a main criterion can be found in literature. All these algorithms use *stress values* to indicate the status of adjacent nodes. A *contention-aware* routing scheme uses a local routing decision, at each node's level [18]. This routing scheme is based on the stress values which indicate the traffic loads of the adjacent nodes. These values are propagated between all neighboring nodes. Thus, the data packets are steered to less congested nodes. The similar but improved routing approach called *look-ahead contention* is presented in [19]. In this approach, to reach the destination node, a packet chooses between a *profitable route* and *misroute*. The profitable routes give the shortest path to the destination node, whereas the misroutes avoid the contention situations. The only information which is provided by the used dedicated wires is the direct neighboring occupancy, which means that each node looks one node ahead for investigating the status of its neighboring nodes. These routing algorithms distribute uniformly the network traffic but they cannot handle the faulty nodes or regions.

## 3. ROUTING ALGORITHM

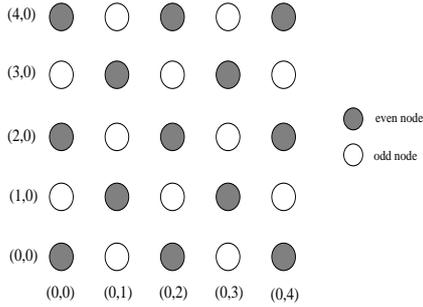
We propose a partially adaptive deadlock-free routing algorithm without using the virtual channels. The main originality of this routing algorithm is that it allows both, routing of messages in the networks incorporating the regions not necessarily rectangular, and routing to all nodes which are not completely blocked by faulty nodes. Our routing scheme is based on the turn model and the XY routing algorithm and allows handling of faulty nodes and regions placed statically or dynamically on the chip. We name this routing approach the *Module Proximity Algorithm - MPA* where module refers to either faulty nodes or regions. This routing algorithm is designed for 2D mesh topology and uses wormhole routing as a switching technique.

### 3.1. The MPA turn models

In a  $n \times n$  2D mesh, a node  $K$  is identified by a two-element vector  $(k_x, k_y)$ ,  $0 \leq k_x, k_y \leq n - 1$ , where  $k_x$  and  $k_y$  are respectively the coordinates of dimension  $X$  and  $Y$ . Each node has five incoming and outgoing ports to which it can send and from which it can receive the packets. Four pairs of ports towards other adjacent nodes are named respectively West, North, East and South and the fifth one presents the local connection port.

**Definition 1.** In a 2D mesh, a node is called an *even* (respectively, *odd*) node if the sum of its coordinates ( $X$  and  $Y$  dimension) is an even (respectively, odd) number.

The regular placement of even and odd nodes in the network is presented in Figure 1. It can be seen that each node of one type is surrounded by elements of other type, even nodes by odds and vice-versa.



**Fig. 1.** Regular placement of even and odd nodes in a network

We distinguish two types of functions of each node: an *activated* and *deactivated* mode. We define the *activated* and *deactivated* area of a network as follows:

**Definition 2.** The *activated area* of a network is the minimal rectangular area which envelops all faulty nodes or regions in the network.

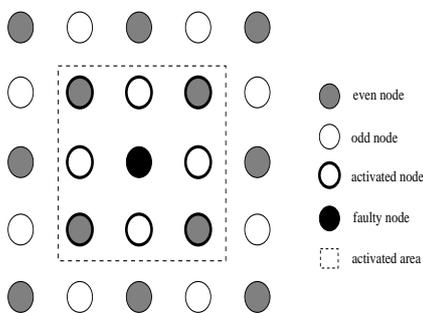
If a network does not have faulty nodes or regions, there is no activated area. Otherwise, a network can contain only one activated area. All nodes belonging to the activated area are **activated**. The activated area cannot have at the same time an odd (even) node at its upper right-hand corner and an even (odd) node at its lower left-hand corner.

**Definition 3.** The *deactivated area* of a network is the rest of the network which does not belong to the activated area.

If a network does not have an activated area (no faulty nodes or regions), the whole network is deactivated. All nodes belonging to the deactivated area are **deactivated**. Figure 2 depicts a network with an activated area formed around a faulty node. In this case, only the nodes wrapping the faulty node become activated. The nodes belonging to the rest of the network do not change their mode, that means that they remain deactivated.

The deactivated node routes a data packet firstly along the X-dimension and then along the Y-dimension until the packet reaches its final destination. If a packet, before reaching the final destination, reaches the activated area's boundary, new rules of routing are applied.

The activated nodes do not obey to the same set of rules as the deactivated nodes. These rules are presented in the following:



**Fig. 2.** Example of an activated area

```

Procedure used to form activated area
repeat
  scanning of entire network
  if (there is faulty node or region)
    wrap the faulty node or region with activated
    nodes and update local routing tables
    if (there is already an activated area)
      merge the new activated area with the old one
      and update routing tables
    end if
  end if
  else
    keep all nodes deactivated
  end if
end repeat

```

**Fig. 3.** Procedure used to form activated area

**Rule 1.** The activated odd node cannot route a packet from North to East side and vice-versa.

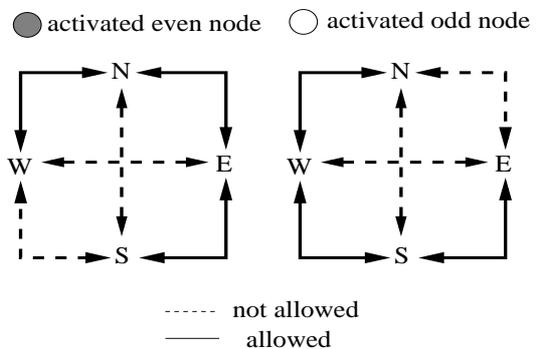
**Rule 2.** The activated even node cannot route a packet from South to West side and vice-versa.

**Rule 3.** All activated nodes by default cannot route a packet from North to South and from East to West.

**Rule 4.** All activated nodes by default cannot route a packet from South to North and from West to East.

The above presented rules are all applied by default in the activated area. Some rules may not be applied depending on positions of faulty nodes or regions in activated area of the network. This is especially applied for the rules 3 and 4. By placing a faulty node next to an activated node, some circular waits that were feasible before placing become not possible any more and thus it relaxes certain routing rules. The rules below explain some of these situations:

**Rule 5.** If an activated even node at the position  $(x, y)$  is a faulty node, the odd nodes at positions  $(x + 1, y)$  and  $(x, y + 1)$  do not apply the rule 3.



**Fig. 4.** Prohibited turns and directions in activated nodes by default

**Rule 6.** If an activated odd node at the position  $(x, y)$  is a faulty node, the activated even nodes at positions  $(x-1, y)$  and  $(x, y-1)$  do not apply the rule 4.

**Rule 7.** If an activated even node at the position  $(x, y)$  has at the positions  $(x-1, y+1), (x+1, y+1)$  or  $((x, y+1)$  and  $(x-2, y))$  a faulty node, it can route a packet from East to West.

Similarly, if an activated odd node at the position  $(x, y)$  has at the positions  $(x-1, y-1), (x+1, y-1)$  or  $((x, y-1)$  and  $(x+2, y+1))$  a faulty node, it can route a packet from West to East.

**Rule 8.** If an activated even node at the position  $(x, y)$  has at the positions  $(x+1, y+1), (x+1, y-1)$  or  $((x+1, y)$  and  $(x+1, y+2))$  a faulty node, it can route a packet from North to South.

Similarly, if an activated odd node at the position  $(x, y)$  has at the positions  $(x-1, y+1), (x-1, y-1)$  or  $((x-1, y)$  and  $(x, y-2))$  a faulty node, it can route a packet from South to North.

The network is scanned regularly, as it is explained in Figure 3. If a new faulty node or region is detected, the activated node's routing tables are updated with the new information about positions of new faulty nodes or regions. The information contained in these routing tables help activated nodes to make routing decisions. For example, the activated area with the minimal number of active nodes which form a sort of a ring around faulty nodes does not route the packets the same way as the activated area having more activated nodes. In the last case, the packet does not use only the ring of activated nodes to get around the faulty nodes or regions, it can use other activated nodes to route its packets.

### 3.2. Routing in a network using the MPA routing scheme

Routing in the network using the MPA routing algorithm is presented in Figure 5. When a message, before reaching its final destination, reaches the activated area, its routing is modified from the standard XY-routing. When a message comes from the X-direction, it can leave the activated area either in the X-direction from the opposite side of arrival or in the Y-direction from the activated node having the same X-dimension as its destination. When a message comes from the Y-direction, it can leave the activated area only in the Y-direction at the opposite side of arrival from the activated node having the same X-dimension as its destination.

Routing in the activated area depends on the number of faulty nodes or regions. In the case of the rectangular region (or a group of faults forming a rectangular shape) the activated nodes form a sort of a ring around the region and the routing in the activated area is reduced to route along the formed ring.

Figure 6 depicts some situations of routing in a  $8 \times 8$  network using the MPA algorithm in the case of 3 faulty nodes. The formed activated area in this case has a size of a  $4 \times 4$  network which is determined by the number of faulty nodes (in this case 3). Figure 6 illustrates some cases of the routing of messages arriving from the X and Y direction. One of the particularities of the MPA routing algorithm is that it allows routing of messages to all activated nodes in the activated area, if they are not cut or completely enveloped by faulty nodes or regions.

### 3.3. Deadlock freeness

Deadlock in wormhole routing is caused by packets waiting on each other in a cycle. This happens in a network if the circular

#### Procedure: MPA Routing

```
/*Message  $m_g$  is sent from source S to D; C is current
node; SW, SE, NW and NE are respectively south-west,
south-east, north-west and north-east corner of the activated
area; X(corner) and Y(corner) are X and Y dimension of a
corner; X(D) and Y(D) are X and Y dimension of a destination*/
```

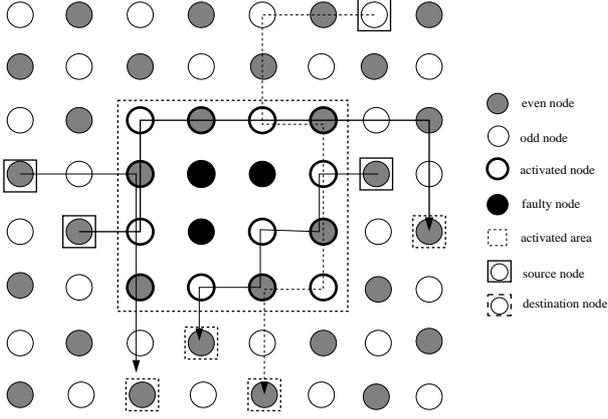
```
if (C is deactivated node)
  XY routing
else
  Activated area routing
end if
/*Activated area routing*/
if ( $m_g$  from X-direction )
  if (D is in activated area)
    route to the D following the rules
  else
    if ( $Y(D) > Y(NE)$  and
       $X(NW) \leq X(D) \leq X(NE)$ )
      route to the  $(X(D), Y(NW))$  and send the
       $m_g$  to  $(X(D), Y(NW)+1)$ ;
    else if ( $Y(D) < Y(SE)$  and
       $X(NW) \leq X(D) \leq X(NE)$ )
      route to the  $(X(D), Y(SW))$  and send the
       $m_g$  to  $(X(D), Y(SW)+1)$ ;
    else
      route to the first activated node at
      the opposite boundary of arrival and send
      the  $m_g$  to a  $X(NE+1)$  or  $X(NW-1)$ 
      deactivated node;
    end if
  else /*  $m_g$  from Y-direction */
    if (D is in activated area)
      route to the D following the rules
    else
      route to the  $(X(D), Y(NW))$  or  $(X(D), Y(SW))$  and
      send the  $m_g$  to  $(X(D), Y(NW)+1)$ ;
    end if end if
```

**Fig. 5.** MPA routing algorithm

waits occur. To avoid deadlock situations, these circular waits must be prevented. In the MPA algorithm, the deadlock freeness is assured by prohibiting certain turns at certain positions defined by the rules 1-8. Figure 7 presents channel dependency graph of a  $4 \times 4$  activated area. It can be seen that there are no circular waits because the above presented rules prohibit enough turns to avoid all deadlock situations.

**Theorem 1.** *The MPA routing algorithm is deadlock free.*

**Proof:** According to [20], the routing algorithm is deadlock free if the corresponding channel dependency graph does not have circular waits. Thus, we prove that the MPA routing algorithm is deadlock free.



**Fig. 6.** Routing in a network using MPA routing scheme in the case of 3 faulty nodes

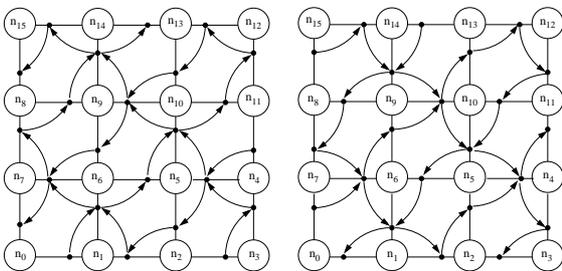
#### 4. SIMULATION RESULTS

We evaluated the MPA routing algorithm by using simulation models. For this purpose we developed a NoC simulator in VHDL supporting both regular as well as irregular mesh topologies.

The simulation model is a  $10 \times 10$  NoC using wormhole as a switching technique with a packet size of 10 flits. Every router has two flit input and one flit output buffer. The minimal link delay is three cycles/flit and the maximum link bandwidth is 1 flit/cycle (1 packet/10 cycles). The average latency was used as a parameter to study the performances of the NoC platform. Performance values were collected over 10,000 packets, after a warm-up session of 2,000 packets.

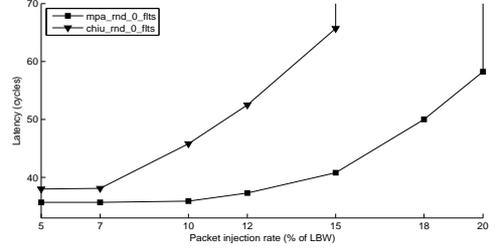
Simulation modules can generate random-traffic and matrix transpose pattern. The random traffic pattern means that the simulation modules send packets randomly to other simulation modules. The matrix transpose pattern means that the  $(i, j)$  node sends to the  $(n - i, n - j)$  node where  $n \times n$  is the size of the network. The time period between the sending time of two packets was uniformly distributed between 5 and 10 cycles.

We simulated two cases: the first one where the network does not have faulty nodes or regions and the second one with 4 faulty nodes in the middle of the network forming an activated area of  $4 \times 4$ . The simulation results for all cases are compared with a *mes-*

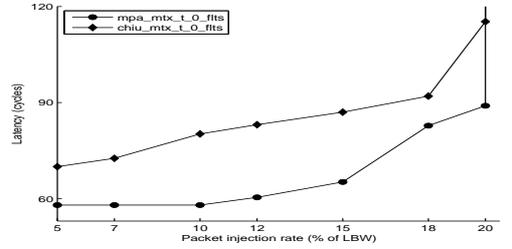


**Fig. 7.** Channel dependency graph of a  $4 \times 4$  activated area without faulty nodes

*sage routing* fault tolerant algorithm [21]. Figures 8 and 9 depict the first simulation case without the faulty nodes under the uniformly distributed random traffic and matrix transpose pattern respectively. These simulation results show that in the absence of faulty nodes the MPA show better results than the *message routing* algorithm.

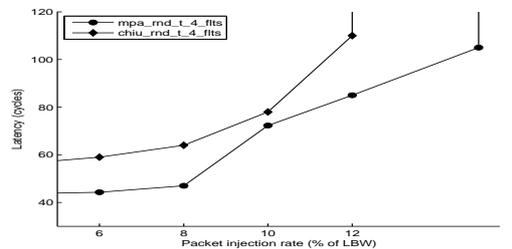


**Fig. 8.** Simulation results for uniformly distributed random traffic pattern in the no faulty node case

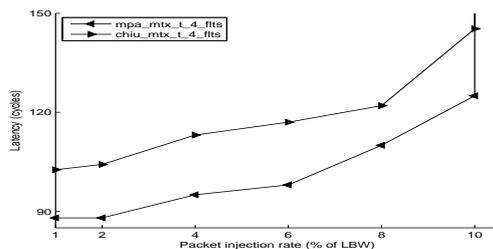


**Fig. 9.** Simulation results for matrix transpose pattern in the no faulty node case

Figures 10 and 11 depict the second simulation case with 4 faulty nodes under the same traffic patterns used for the first simulation case. These figures show that the average latencies for the two algorithms are greater than in the first simulation case. These results are logical and are due to the group of 4 faulty nodes placed in the middle of the network. The maximal injection rate decreases for two algorithms for two communication patterns, but it remains greater for the MPA routing algorithm (about 15% for the random traffic pattern compared to 12% for the *message routing* algorithm).



**Fig. 10.** Simulation results for the uniformly distributed random traffic pattern in the case of 4 faulty nodes



**Fig. 11.** Simulation results for the matrix transpose pattern in the case of 4 faulty nodes

## 5. CONCLUSION

In this paper we proposed a new deadlock-free adaptive routing algorithm for 2D mesh reconfigurable NoCs. The main originality of this routing algorithm is that it allows both, routing of messages in the networks incorporating the regions not necessarily rectangular, and routing to all nodes which are not completely blocked by faulty nodes. Our proposed algorithm allows routing of packets in the networks incorporating faulty nodes and regions. More precisely, in the situations where on packets' path faulty nodes or regions placed statically or dynamically may encounter. We proved the MPA's deadlock freeness and its feasibility through the simulation results. We showed the efficiency of our algorithm in terms of performances in comparison with the *message routing* algorithm. Moreover, the proposed algorithm does not use virtual channels and thus it will allow a low area implementation. In order to provide an homogeneous traffic distribution over the networks and to avoid congested links, an adaption of the proposed algorithm will be considered in the future.

## 6. REFERENCES

- [1] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan 2002.
- [2] W. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," *Design Automation Conference, 2001. Proceedings*, pp. 684–689, 2001.
- [3] P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a switch for network on chip applications," *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol. 5, pp. V–217–V–220 vol.5, 25–28 May 2003.
- [4] S. Murali and G. De Micheli, "SUNMAP: a tool for automatic topology selection and generation for NoCs," *Design Automation Conference, 2004. Proceedings. 41st*, pp. 914–919, 2004.
- [5] C. Bobda, A. Ahmadinia, M. Majer, J. Teich, S. Fekete, and J. van der Veen, "Dynoc: A dynamic infrastructure for communication in dynamically reconfigurable devices," *Field Programmable Logic and Applications, 2005. International Conference on*, pp. 153–158, 24–26 Aug. 2005.
- [6] S. Jovanovic, C. Tanougast, S. Weber, and C. Bobda, "CuNoC: A scalable dynamic noc for dynamically reconfigurable fpgas," *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, pp. 753–756, 27–29 Aug. 2007.
- [7] C. Glass and L. Ni, "The turn model for adaptive routing," in *Computer Architecture, 1992. Proceedings., The 19th Annual International Symposium on*, pp. 278–287, 1992.
- [8] G.-M. Chiu, "The odd-even turn model for adaptive routing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 11, no. 7, pp. 729–738, Jul 2000.
- [9] Y. Boura and C. Das, "Efficient fully adaptive wormhole routing in n-dimensional meshes," *Distributed Computing Systems, 1994., Proceedings of the 14th International Conference on*, pp. 589–596, 21–24 Jun 1994.
- [10] A. Chien and J. H. Kim, "Planar-adaptive routing: Low-cost adaptive networks for multiprocessors," *Computer Architecture, 1992. Proceedings., The 19th Annual International Symposium on*, pp. 268–277, 1992.
- [11] Glass and Ni, "Fault-tolerant wormhole routing in meshes," in *FTCS: Fault-Tolerant Computing: 23th Annual International Symposium*. IEEE Computer Society, 1993. [Online]. Available: citeseer.ist.psu.edu/glass93faulttolerant.html
- [12] J. Wu, "A fault-tolerant and deadlock-free routing protocol in 2d meshes based on odd-even turn model," *Computers, IEEE Transactions on*, vol. 52, no. 9, pp. 1154–1169, Sept. 2003.
- [13] R. Boppana and S. Chalasani, "Fault-tolerant wormhole routing algorithms for mesh networks," *Computers, IEEE Transactions on*, vol. 44, no. 7, pp. 848–864, Jul 1995.
- [14] K.-H. Chen and G.-M. Chiu, "Fault-tolerant routing algorithm for meshes without using virtual channels," *J. Inf. Sci. Eng.*, vol. 14, no. 4, pp. 765–783, 1998.
- [15] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. Das, "Exploring fault-tolerant network-on-chip architectures," *Dependable Systems and Networks, 2006. DSN 2006. International Conference on*, pp. 93–104, 2006.
- [16] C.-C. Su and K. Shin, "Adaptive fault-tolerant deadlock-free routing in meshes and hypercubes," *Computers, IEEE Transactions on*, vol. 45, no. 6, pp. 666–683, Jun 1996.
- [17] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, "Fully adaptive fault-tolerant routing algorithm for network-on-chip architectures," *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, pp. 527–534, 29–31 Aug. 2007.
- [18] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," *Design, Automation and Test in Europe Conference and Exhibition, 2003*, pp. 1126–1127, 2003.
- [19] T. T. Ye, L. Benini, and G. D. Micheli, "Packetization and routing analysis of on-chip multiprocessor networks," *J. Syst. Archit.*, vol. 50, no. 2–3, pp. 81–104, 2004.
- [20] W. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on computers*, vol. 100, no. 36, pp. 547–553, 1987.
- [21] K. Chen and G. Chiu, "Fault-tolerant routing algorithm for meshes without using virtual channels," *Journal of Information Science and Engineering*, vol. 14, no. 4, pp. 765–783, 1998.