Understanding Turn Models for Adaptive Routing: the Modular Approach

Edoardo Fusella, Alessandro Cilardo

Department of Electrical Engineering and Information Technologies, University of Naples Federico II
via Claudio 21, 80125 Napoli, Italy, Email: {edoardo.fusella, acilardo}@unina.it

Abstract—Routing algorithms were extensively studied first in multi-computer systems, then in multi- and many-core architectures. Among the commonly used routing techniques, the turn model seems the most promising solution when targeting adaptiveness. Based on the turn model, several alternative approaches with different turn prohibition schemes were proposed. This paper gives a new theoretical background for designing deadlock-free partially adaptive logic-based distributed routing algorithms that are based on the turn model. Two properties are presented, including a necessary and sufficient condition to prove that a routing algorithm is deadlock-free as long as turn restrictions follow a modular distribution. Existing approaches can be considered a subset of the solution space identified by this work. Finally, we propose a novel routing algorithm exhibiting encouraging performance improvements over state-of-the-art approaches.

I. INTRODUCTION

The network-on-chip (NoC) paradigm is nowadays considered the main solution to alleviate the communication scalability problems that arise in multi- and many-core systems. NoCs are mainly based on regular tile-based structures with two dimensional (2D) mesh/torus topologies [1]. The modularity and regularity of these topologies make them suitable for a scalable interconnect fabric.

Given an underlying topology, the routing algorithm determines the routes taken by messages to reach their destinations. It plays a major role affecting performance, power consumption and reliability of the whole system. Routing has been extensively studied and several alternative approaches have been proposed. Minimal logic-based distributed routing algorithms are desirable due to their advantages in terms of performance, energy efficiency, scalability and implementation cost [1]. One example is the dimension-ordered routing (DOR) where packets are allowed to move in a single dimension at a time. This approach is very popular since it is easy to implement and guarantees decent performance. However, it follows a deterministic behavior and messages are forced to take the same path between every source-destination pair. On the contrary, *adaptive routing* allows different minimal paths between a source node and a destination node. In case of *fully adaptive* routing, all minimal paths can be used to route a message, while *partially adaptive* routing exploits a limited set of all minimal paths. In general, adaptiveness can provide better performance and fault tolerance by allowing alternative paths but requires complex techniques to avoid deadlock situations.
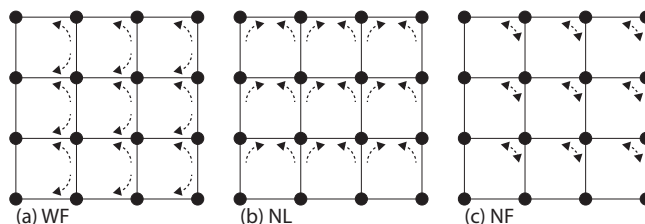

Fig. 1. Prohibited turns of WF (a), NL (b), NF (c) routing algorithms. The dashed arrows indicate routing restrictions.

Deadlocks are tackled with two different strategies, i.e. virtual channels [2] or turn model [3]. Virtual channels are more suitable for fully adaptive routing algorithms and involve a non-negligible implementation cost due to the complex control logic and large buffer space required. On the contrary, deadlock-free partially adaptive routing algorithms can be designed with techniques of turn prohibitions, i.e. messages are not allowed to take some turns in the network.

A turn consists in a 90-degree change in the traveling direction. 2D meshes are characterized by eight types of turns, i.e. *ES*, *SW*, *WN*, *NW*, *WS*, *SE*, *EN*, *NE*. For instance, a *NE* turn involves a change of direction from North to East. Turns can be classified into clockwise, i.e. NE, ES, SW, WN, and counter-clockwise, i.e. NW, WS, SE, EN. According to were to place turn prohibitions, several algorithms based on the turn model were proposed [3], [4], [5], [6]. The turn model, that is based on turn prohibitions, requires routing algorithms to prohibit enough turns to avoid possible cycles in the channel dependency graph (CDG) [3], thus ensuring deadlock freedom [7].

It has been 25 years since the *turn model* [3] was proposed by Glass and Ni. They demonstrated that deadlocks can be avoided by prohibiting a single clockwise turn and a single counter-clockwise turn in each cycle. On this assumption, they proposed three different routing algorithms: West-First (WF), North-Last (NL), Negative-First (NF). In the West-First routing, all turns to the West (NW and SW) are not allowed and, hence, packets that must reach the West must start travelling in that direction. Similarly, in the North-Last routing, all turns when travelling North (NW and NE) are not allowed and, hence, the North is the last direction that may be taken. Finally, in the Negative-First, the NW and ES turns are prohibited. Fig. 1 depicts the prohibited turns.

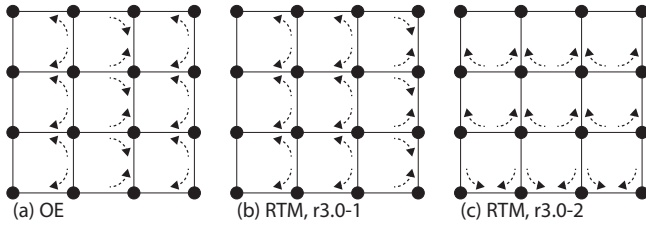The *Odd-Even turn model* (OE) was proposed in [4]. In

Fig. 2. Prohibited turns of OE (a), RTM r3.0-1 (b), RTM r3.0-2 (c) routing algorithms. The dashed arrows indicate routing restrictions.

order to provide a fairer distribution of prohibited turns, EN and ES turns are not allowed on even columns and NW and SW turns are not allowed on odd columns (Figure 2(a)). In this way, nodes located on adjacent columns have different prohibited turns. This helps reduce the load unbalancing of previously proposed approaches.

Finally, more recently, the *repetitive turn model* [6] (RTM) was introduced. The authors pointed out the importance of repetitively distributing the prohibited turns in order to implement the routing algorithm in a distributed logic-based way. Then, they investigated the design space of routing algorithms that have prohibited turns repetitively distributed along the rows or columns. Finally, they presented two routing algorithms, named r3.0-1 and r3.0-2, that outperform the OE routing algorithm. In the r3.0-1 routing, NW and SW turns are not allowed on the ith column if $i \bmod 3 \neq 0$, while ES and EN turns are not allowed on the ith column if $i \bmod 3 = 0$. Similarly, in the r3.0-2 routing, WN and EN turns are not allowed on the ith row if $i \bmod 3 \neq 0$, while SE and SW turns are not allowed on the ith row if $i \bmod 3 = 0$. Figs. 2(b) and 2(c) depict the prohibited turns of the two algorithms.

These algorithms have in common that they are obtained through purely empirical approaches, as the choice of the forbidden turns as well as their distribution come from the intuition and expertise of a few researchers. The novel contribution of this paper is two-fold. First, the problem domain is described and formalized. We thus provide a new theoretical background for designing deadlock-free partially adaptive routing algorithms based on the turn model. We present two properties that help pruning the solution space and designing new routing algorithms. Finally, we also propose a novel routing algorithm that exhibits better performance for some traffic patterns compared to alternative approaches.

II. PROPOSED MODEL AND ROUTING ALGORITHM

In an $m \times n$ mesh, a node can be uniquely identified by its position in the 2D grid. In the following we will use (x_i, y_i) with $0 \leq x_i \leq m - 1$ and $0 \leq y_i \leq n - 1$ to indicate the ith node of the network. Two nodes i and j are neighbors if and only if $x_i = x_j$ and $y_i = y_j \pm 1$ or $y_i = y_j$ and $x_i = x_j \pm 1$. Neighboring nodes are connected by means of two unidirectional channels. Channels are denoted according to their traveling directions. For instance, a channel connecting nodes i and j is called a *north* channel if $y_i = y_j - 1$. In

the same way, we have *east*, *south* and *west* channels. The nodes that have the same coordinates in the first dimension are located on the same column. In the same way, the nodes that have the same coordinates in the second dimension are located on the same row.

Below we will present two properties that are essential for the design of partially-adaptive routing algorithms that are based on the turn model. In particular, the second property gives a necessary and sufficient condition to prove that a routing algorithm is deadlock-free as long as certain requirements are meet. Note that all the following proofs only consider the clockwise deadlock condition, since the proof in the case of counterclockwise deadlock is equivalent.

Definition 1. *A turn restriction is called* upper/lower *if it breaks cycles that are located above/under the node where it is placed. A turn restriction is called* right/left *if it breaks cycles that are located on the right/left of the node where it is placed.*

EN, SE, SW, WN are upper turn restrictions, while NE, NW, ES, WS are lower turn restrictions. NE, SE, WN, WS are right turn restrictions, while NW, EN, ES, SW are left turn restrictions.

Property 1. *In a deadlock-free routing algorithm based on the turn model, if a node prohibits a right (left) lower (upper) turn, the nodes on the same row must prohibit right (left) turns and the nodes on the same column must prohibit lower (upper) turns.*

Proof of Property. *Let (x_0, y_0) and (x_0, y_1) be two generic nodes of the network that are located on the same column and are characterized by the same turn restriction. Without loss of generality, we assume that they both prohibit a right lower turn. Let (x_1, y_0) and (x_1, y_1), with $x_1 < x_0$, be two generic nodes located on the same rows of (x_0, y_0) and (x_0, y_1). We proceed by contradiction. Assume that (x_1, y_0) and (x_1, y_1) prohibit left turns. Now, consider the set of channels connecting nodes (x_1, y_0), (x_1, y_1), (x_0, y_1), and (x_0, y_0) in a clockwise direction. There is no turn restriction avoiding a set of packets to be deadlocked clockwise and, hence, (x_1, y_0) and (x_1, y_1) must prohibit right turns. In the same way, consider the set of channels connecting in a clockwise direction nodes (x_0, y_0), (x_0, y_2), (x_2, y_2), and (x_2, y_0), with $y_2 > y_0$ and (x_2, y_0) prohibiting the same turn of (x_0, y_0). In case of nodes (x_0, y_2) and (x_2, y_2) prohibiting upper turns, there is no turn restriction avoiding a set of packets to be deadlocked clockwise and, hence, nodes (x_0, y_2) and (x_2, y_2) must prohibit a lower turn. The property is proved.*

There are some important consequences that arise from this property:
1) At most two turn restrictions can be enforced in the clockwise and counter-clockwise direction.
2) It is not possible to have lower right and upper left turn restrictions at the same time. Similarly, it is not possible to have lower left and upper right turn restrictions at the same time.

Property 2. *A routing algorithm exploiting the turn model is*

deadlock-free iff turn restrictions follow a modular distribution, i.e. $\exists\, n \in \mathbb{N}$ such that every two nodes i and j prohibit the same turns if $x_i \equiv x_j \bmod n$ or $y_i \equiv y_j \bmod n$.

Proof of Property. (⇒) *Let (x_0, y_0) be a node of the network. Without loss of generality, we assume that it prohibits a lower right turn. Due to Property 1, all the nodes (x_0, y_i) $\forall y_i$ located on the column x_0 must prohibit lower turns, and all the nodes (x_i, y_0) $\forall x_i$ located on the row y_0 must prohibit right turns. As a consequence, considering the clockwise direction, nodes (x_0, y_i) can prohibit NE or ES turns, while nodes (x_i, y_0) can prohibit NE or WN turns. The case of all the nodes (x_i, y_0) and (x_0, y_i) prohibiting NE turns is trivial, since, according to Property 1, we will have all nodes of the network prohibiting the same turn. Differently, we assume that some nodes located on row y_0 prohibit WN turns (similar considerations hold true in the case of some nodes located on the column x_0 prohibiting ES turns). Due to the first consequence of Property 1, ES turn restrictions are not possible and hence all the nodes (x_0, y_i) prohibit NE turns that are lower right turns. Now, consider a generic node (x_1, y_1). Since node (x_0, y_1) prohibits the NE turn, that is a right turn, (x_1, y_1) must prohibit a right turn too. Now, if node (x_1, y_0) prohibits an upper turn, then node (x_1, y_1) must prohibit a right upper turn, otherwise a right lower turn. Note that in this way, we have that two generic nodes (x_i, y_i) and (x_i, y_j), $\forall x_i, y_i, y_j$ prohibit the same turns. This means that also nodes (x_i, y_i) and $(x_i, y_i + 1)$, $\forall x_i, y_i$ prohibit the same turns. As a consequence, the nodes located on the same columns prohibit the same turns. Hence, there is a modular distribution and we prove the property.*

(⇐) Consider that there exists a positive integer $n \in \mathbb{N}$ such that every two nodes i and j prohibit the same turns if $x_i \equiv x_j \bmod n$ (similar considerations hold true in the case of $y_i \equiv y_j \bmod n$). This means that all nodes located on the same column prohibit the same turns. We prove the property by contradiction. Assume that there exists a set of packets $p_0, p_1, ..., p_{n-1}$, where p_{i+1} waits for p_i with $0 \le i \le n-2$. A deadlock arises if p_0 waits for p_{n-1}, as a circular dependence is generated through the packets. The waiting path must include both row and column channels. Now, consider the rightmost and leftmost columns. Two cases need to be considered. In case 1, the nodes on the rightmost column prohibit a left turn and hence the dependency cycle is broken in the rightmost column. In case 2, the nodes on the rightmost column prohibit a right turn. According to Property 1, the nodes on the leftmost column must prohibit a right turn too and, hence, the dependency cycle is broken in the leftmost column. Hence, the property is proved.

Property 2 gives a necessary and sufficient condition to choose a turn prohibitions distribution scheme ensuring freedom from deadlocks. As a consequence, all routing algorithms based on the turn model, including existing approaches, are part of the solution space identified by this work. For instance, consider the routing algorithms surveyed in Section I. The case of the routing algorithms WF, NL and NF is trivial, since all nodes of the network prohibit the same turns. Differently, in the OE algorithm, a generic node (x_i, y_i) prohibits ES and EN turns if $x_i \bmod 2 = 0$ and SW and NW turns if $x_i \bmod 2 = 1$.

Similar considerations apply for RTM r3.0-1 and RTM r3.0-2 routing algorithms. In the RTM r3.0-1 algorithm, a generic node (x_i, y_i) prohibits ES and EN turns if $x_i \bmod 3 = 0$, otherwise SW and NW turns. Similarly, in the RTM r3.0-2 algorithm, a generic node (x_i, y_i) prohibits SW and SE turns if $y_i \bmod 3 = 0$, otherwise WN and EN turns. Note that, Property 2 partitions the nodes of the network in n class of nodes, with nodes in the same class providing the same turn restrictions. Each node belongs to a class for clockwise turn restrictions and to another class for counter-clockwise turn restrictions, potentially enlarging the solution space.

Novel routing algorithms can be found in two steps: 1) find a positive integer value $n \in \mathbb{N}$ that is the dividend of the modulo operation. 2) specify which turn prohibitions are associated with each class of nodes. In case of a $n \times n$ NoC, there are $2^{n+2} - 4$ different turn prohibitions distribution schemes for both the clockwise and counter-clockwise directions. This is because according to the first consequence of Property 1, a routing algorithm can prohibit at most two turns in the clockwise direction and two turns in the counter-clockwise direction. As a consequence, there are $2^n - 1$ different class of nodes and four pairs of turn restrictions.

Based on the above considerations, it is possible to find new routing algorithms, potentially outperforming previously proposed approaches. In that respect, we analyzed all routing algorithms considering a maximum dividend of the modulo operation equal to 4. Please, note that this is very preliminary and is here only to show that the proposed model enhances the solution space found by previously proposed works and, hence, gives the opportunity to find routing algorithms that outperform existing ones. Second, all routing algorithms are evaluated according to their *routing pressures* [8], a new metric useful to measure routing performance in a fast and effective way. Last, we choose the routing algorithm whose average pressure is closer to the ideal pressure, i.e. the minimum value among the pressures of all the different algorithms. The prohibited turns of the proposed routing algorithm are shown in Figure 3 for the case of an 7×7 mesh topology. Basically, in the proposed algorithm, a generic node (x_i, y_i) prohibits SW and NW turns if $x_i \bmod 3 = 0$, ES and NW turns if $x_i \bmod 3 = 1$ and SW and EN turns if $x_i \bmod 3 = 2$.
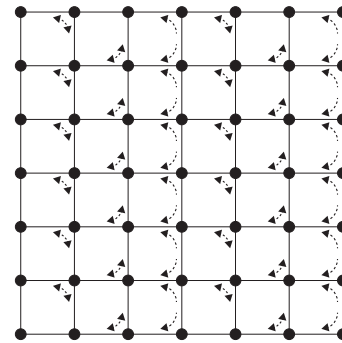

Fig. 3. Prohibited turns of the proposed routing algorithm. The dashed arrows indicate routing restrictions.
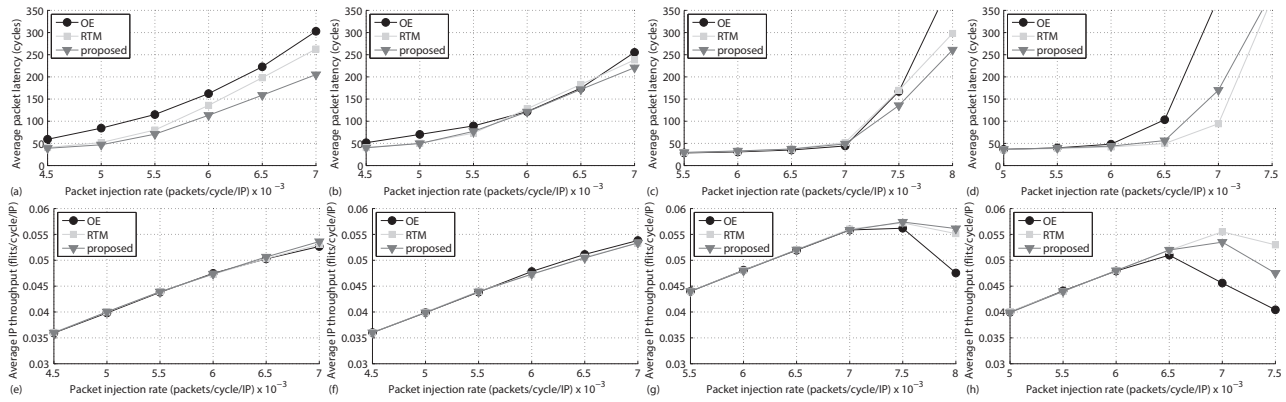
Fig. 4. (a) (b) (c) (d) Latency and (e) (f) (g) (h) throughput variation under (a) (e) transpose1 (b) (f) transpose2 (c) (g) shuffle and (d) (h) uniform traffic scenarios.

III. Evaluation

In this section, we evaluate the proposed routing algorithm in terms of performance, i.e. latency and throughput. For our experimental evaluation, we relied on Noxim, an open source SystemC NoC simulator [9]. The proposed routing algorithm, shown in Figure 3, was implemented in SystemC and embedded in Noxim. We consider a 16×16 mesh topology. All packets have the same size of eight flits and routers are configured to have a buffer depth of four flits. The commonly used round-robin policy is adopted to select requesting inputs in switch allocation stage and output ports are choosen from the *outputSet* set according to a random selection strategy. Each simulation runs $50,000$ cycles after $2,500$ cycles of warm-up period in order to allow the system to stabilize. The simulation for each configuration is iterated a number of times to improve accuracy.

We compared the proposed approach with the OE and RTM routing algorithms in terms of latency and throughput. The comparison is done with transpose1, transpose2, shuffle and uniform traffic patterns [9]. Consider an $n \times n$ mesh topology. Nodes (x_i, y_i) communicate with nodes $(n - y_i - 1, n - x_i - 1)$ and (y_i, x_i) in case of respectively transpose1 and transpose2. Differently, nodes with address $a_{n-1}, a_{n-2}, ..., a_1, a_0$ communicate with nodes with address $a_{n-2}, a_{n-3}, ..., a_0, a_{n-1}$ in case of shuffle traffic. In the uniform traffic, destination nodes are chosen with equal probability in the whole network.

The latency and throughput variations under transpose1 and transpose2 traffics are shown in Figure 4(a) and (e) and Figure 4(b) and (f). In case of transpose1, the proposed algorithm decreases on the average the packet delay of more than 55% and 14% compared to OE and RTM routing algorithms. Differently, in case of transpose2, the proposed algorithm and RTM achieve similar performance with a 17% average advantage in terms of latency compared to OE. The simulation results for bit-reversal and uniform traffics are shown in Figs. 4(c) and (g) and 4(d) and (h). As regards shuffle traffic, the three algorithms achieve the same performance with a small advantage compared to OE at higher injection rates. Finally, the proposed algorithm has better performance than

OE under uniform traffic. However, it reaches saturation with a smaller injection rate compared to RTM and, hence, for higher injection rate values, RTM has lower packet latency.

IV. Conclusions

We have presented a novel formal model for designing partially adaptive logic-based distributed routing algorithms that are based on the turn model. This work improves on previously proposed approaches, identifying a superset of the solutions spanned by existing works. Based on the proposed model, we presented a novel routing algorithm that outperforms state-of-the-art approaches for some traffic patterns.

References

[1] A. Cilardo and E. Fusella, "Design automation for application-specific on-chip interconnects: A survey," *Integration, the VLSI Journal*, vol. 52, pp. 102–121, 2016.

[2] W. J. Dally, "Virtual-channel flow control," *IEEE Transactions on Parallel and Distributed systems*, vol. 3, no. 2, pp. 194–205, 1992.

[3] C. J. Glass and L. M. Ni, "The turn model for adaptive routing," *ACM SIGARCH Computer Architecture News*, vol. 20, no. 2, pp. 278–287, 1992.

[4] G.-M. Chiu, "The odd-even turn model for adaptive routing," *IEEE Transactions on parallel and distributed systems*, vol. 11, no. 7, pp. 729–738, 2000.

[5] B. Fu, Y. Han, J. Ma, H. Li, and X. Li, "An abacus turn model for time/space-efficient reconfigurable routing," in *ACM SIGARCH Computer Architecture News*, vol. 39, no. 3. ACM, 2011, pp. 259–270.

[6] M. Tang, X. Lin, and M. Palesi, "The repetitive turn model for adaptive routing," *IEEE Transactions on Computers*, vol. 66, no. 1, pp. 138–146, 2017.

[7] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE transactions on parallel and distributed systems*, vol. 4, no. 12, pp. 1320–1331, 1993.

[8] M. Tang, X. Lin, and M. Palesi, "Routing pressure: A channel-related and traffic-aware metric of routing algorithm," *IEEE transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 891–901, 2015.

[9] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Cycle-accurate network on chip simulation with Noxim," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 27, no. 1, p. 4, 2016.