A Novel Non-minimal/Minimal Turn Model for Highly Adaptive Routing for 2D and 3D NoCs

Manoj Kumar, Student Member, IEEE, Vijay Laxmi, Member, IEEE, Manoj Singh Gaur, Member, IEEE, Masoud Daneshtalab, Member, IEEE, Seok-Bum Ko, Member, IEEE, and Mark Zwolinski, Member, IEEE,

Abstract—One of the current issues in NoC routing is the use of an acyclic Channel Dependency Graph (CDG) for ensuring freedom from deadlock. This requirement forces certain routing turns to be prohibited, thus reducing the degree of adaptiveness. In this paper, we propose a novel turn model for 2D and 3D mesh topologies which allows cycles in the CDG, provided the Extended Channel Dependency Graph (ECDG) remains acyclic. The end result is that the proposed turn model reduces the number of restrictions on routing turns and hence is able to provide path diversity through additional minimal and non-minimal routes between the source and destination. The average performance gain of the proposed method is up to 78% across all selected benchmarks compared with XY and 11% compared with other adaptive algorithms, for a 7 × 7 mesh.

Keywords—Networks on Chip, routing, turn models, deadlock freedom, non-minimal paths.

I. INTRODUCTION AND BACKGROUND

The Network-on-Chip (NoC) model has been accepted commercially as the communication paradigm for Systems-on-Chip (SoC), instead of dedicated wires or shared buses. Industrial examples include Tilera’s TILE-Gx72, TILE64™ [1] processors and Intel’s terascale processor[2]. The overall NoC performance depends on many parameters such as topology, flow control, routing methods, quality-of-service and switching methods. In all cases, routing algorithms will have a major impact on the network performance and the area. The higher the degree of adaptiveness (path diversity) of the routing algorithm, the lower the probability for a packet to enter a faulty or congested area. The focus of this research is to enhance the performance of the routing algorithm by increasing the degree of adaptiveness (allowing additional minimal paths) and using fewer virtual channels (VCs) for freedom from deadlock.

A turn model produces partially adaptive routing [3]–[6] schemes that restrict some routing turns to achieve deadlock freedom, but thus resulting in low adaptiveness. Deadlock-free fully adaptive routing schemes require additional VCs to achieve a high degree of adaptiveness. Several minimal/non-minimal fully adaptive routing algorithms have been proposed in [7]–[13] using additional VCs.

The routing schemes introduced in [7], [8] produce an equivalent fully adaptive and minimal routing algorithm called double-y for a 2D mesh. It requires one and two VCs in the X and Y dimensions, respectively. In [9], the authors introduced a maximally adaptive double-y (Mad-y) routing algorithm which is an improvement over double-y network based schemes [7], [8] and allows more routing turns by making better use of VCs to increase adaptivity.

Minimal routing methods guarantee the shortest route between the source and destination, but it would be imprudent to neglect the performance improvements achievable by non-minimal routing schemes. For example, if all output channels corresponding to minimal paths are congested (or faulty), routing the packets along a non-minimal route may be a good (or the only) alternative. Minimal routing algorithms also suffer from a low degree of adaptiveness, even if they accurately detect the state of congestion. Ebrahimi et al. [12], [13] proposed non-minimal routing schemes for a 2D mesh. These provide better adaptability than [9] with the same VCs. However, these impose some unnecessary restrictions on routing turns, which could be removed to increase path diversity. Thus, in this paper, we present a novel turn model for a Congestion-aware and Highly Adaptive Routing method (CHARM) for 2D and 3D meshes. CHARM offers a high degree of adaptiveness by removing some restrictions on routing turns and permitting cycles in the CDG.

The rest of this paper is organized as follow. In Section II, we discuss our proposed work, CHARM. Section III presents the results evaluation. Finally, Section IV concludes the paper.

II. PROPOSED MODEL

Since the proposed model is based on the Mad-y and LEAR turn models, we first briefly describe them.

A. Mad-y [9] and LEAR [12] Turn Models

Figure 1(a) illustrates the routing turns for the Mad-y and LEAR routing algorithms. In order to achieve deadlock freedom, Mad-y imposes the following constraints on routing turns:

1) It prohibits four 90-degree turns (N2-W, S2-W, E-N1 and E-S1) as shown in Fig. 1(a)(i) and 1(a)(ii).
2) It also prohibits two 0-degree turns (S2-S1 and N2-N1) as shown in Fig. 1(a)(iii).

The LEAR turn model has the same turn constraints as Mad-y for 90-degree and 0-degree turns (Fig. 1(a)(i), 1(a)(ii), and 1(a)(iii)). In addition, the LEAR model allows one 180-degree turns, as shown in Fig. 1(a)(iv), whereas the Mad-y prohibits all 180-degree turns. The deadlock freedom of both Mad-y and LEAR is proved using Daly’s work [14].

B. 2D-CHARM

Deadlock-freedom is essential for every routing algorithm. Most routing algorithms [13], [15] including Mad-y [9] and
LEAR [12] are proved deadlock-free by using an acyclic CDG [14]. This requirement forces certain routing turns to be prohibited, thus, the routing function cannot generate all qualified turns to forward packets. The proposed method imposes substantially fewer restrictions on routing turns (especially 90-degree) by allowing cycles in the CDG provided that the ECDG is acyclic [16]. Thus, it provides additional minimal and non-minimal paths between the source and destination compared to Mad-y and LEAR.

Figure 1(b) shows the turn model representation of 2D-CHARM. A packet is permitted to use the first virtual channel at any time, as shown in Figure 1(b)(i). It can use a second virtual channel only if it has already been routed in the negative direction of the X dimension (west), as shown in Figure 1(b)(ii). It is allowed to take two 180-degree turns from west to east (W-E) and south to north (S2-N2), as shown in Figure 1(b)(iv), but only if it has completed routing in the west and south directions, respectively. In short, in order to avoid deadlocks, 2D-CHARM imposes the following constraints on routing turns:

1) It prohibits two 90-degree turns (S2-W and N2-W).
2) It allows 0-degree turns (S1-S1, N1-N1, S2-S2 and N2-N2) as shown in Figure 1(b)(iii). It allows 0-degree turns (S1-S2, N1-N2, S2-S1 and N2-N1), as shown in Figure 1(b)(iii), with some restrictions. It allows these restricted turns only when a packet does not need to be forwarded further west.
3) It permits some 180-degree turns, as shown in Fig. 1(b)(iv).

C. 3D-CHARM

Now, we extend 2D-CHARM in three dimensions. 3D-CHARM deploys one, two and two VCs in the X (+X: east, −X: west), Y (+Y: north, −Y: south) and Z (+Z: up, −Z: down) dimensions, respectively. Figures 1(b) and 1(c) show turn model representations for the XY (XZ) plane and YZ plane, respectively. A packet is allowed to use VC1 of any dimension at any instant of time, as shown in Figures 1(b)(i), 1(c)(i) and 1(c)(ii). VC2 can be used by a packet only if the packet has already completed routing in negative directions (west and south) of all lower dimensions (X and Y). It is allowed to take a 180-degree turn from west to east, south to north and down to up only if it has completed routing in west, south and down directions, respectively.

For the XY-plane, 3D-CHARM imposes the same constraints as in 2D-CHARM (Section II-B). Constraints for the XZ-plane can be deduced in a similar fashion to the XY-plane. For the YZ-plane, 3D-CHARM imposes the following constraints on routing turns:

1) It prohibits four 90-degree turns (U2-S1, U2-S2, D2-S1 and D2-S2), as shown in Figures 1(c)(iii) and 1(c)(iv).
2) It allows 0-degree turns (U1-U1, D1-D1, U2-U2 and D2-D2), as shown in Figure 1(c)(v). 3D-CHARM allows 0-degree turns (U1-U2, U2-U1, D1-D2 and D2-D1), as shown in Figure 1(c)(v), with some restrictions. It allows these restricted turns only when a packet does not need to be forwarded further west or south. Similarly, we can find allowed and restricted 0-degree turns for north and south.
3) It permits some 180-degree turns, as shown in Figure 1(c)(vi).

D. Deadlock and Livelock Freedom

With deterministic routing, packets can be routed over a single output channel at each node. Thus, it is mandatory to remove all cyclic dependencies between network channels in order to achieve deadlock freedom. In adaptive routing, packets often have several options for routing at each node. Thus, it is not mandatory to eliminate all cyclic dependencies between channels, provided that every packet can be forwarded on a route whose channels are not involved in cyclic dependencies. The channels involved in these acyclic routes are considered as escape channels from deadlocks (cycles).

The deadlock-freedom of CHARM (2D and 3D) is assured from Duato’s theorem [16], stated as follows:

Theorem 1: (Duato’s Theorem) For a given interconnection network I, a connected and adaptive routing function R is deadlock-free if there exists a routing subfunction R1 ⊆ R, that is connected and has an acyclic ECDG.

Following Duato’s terminology, the route computation function of CHARM is denoted by R and the set of channels used by R is denoted by C. To assure deadlock freedom of CHARM, we first identify the subset of channels C1 ⊆ C, that defines routing subfunction R1 ⊆ R that is connected and has an ECDG with no cycles arising from direct, indirect, direct-cross and indirect-cross dependencies. For CHARM, C1 has all VCs except N1, S1, U1 and D1.

Lemma 1: The routing subfunction R1 is connected.

Proof: The R1 routing subfunction with channel set C1 is an all-but-one-negative-first [3] routing algorithm. Since non-minimal all-but-one-negative-first routing is connected, so R1 is connected. □
**Lemma 2:** The ECDG of channel set $C_1$ with additional channels ($N_1$, $S_1$, $U_1$ and $D_1$) introduced by $R$, does not result in any cyclic dependencies.

**Proof:** There is no direct-cross dependency in the ECDG of $C_1$ as routing function $R$ does not add any new routing options between the channels of $C_1$ directly. However, additional channels introduced by routing function $R$ add new routing options between the channels of $C_1$ indirectly, but do not produce any indirect-cross dependencies. Additional channels introduced by $R$ can cause only indirect dependencies between west (south) channels as a packet can use the west (south) channel and later can use the west (south) channel of a different row and column. But this indirect dependency does not introduce any cycles in the ECDG of $C_1$. The ECDG for $C_1$ has no dependencies from a channel in the north, east, south, up, or down directions to a channel in the west direction, so the west channels are always used before all other channels in $C_1$. Hence, these indirect dependencies introduce new dependencies only between the west VCs and create no cycles using only the west VCs. Similarly the ECDG for $C_1$ has no dependencies from a channel in the north, up, or down directions to a channel in the south direction, so the south channels are always used before other channels (north, up and down) in $C_1$. These indirect dependencies introduce dependencies from west to south or between south VCs. Hence, these indirect dependencies create no cycles using only the south VCs. Since there are no indirect and direct dependencies, which produce cycles in the ECDG, therefore the ECDG of $C_1$ is acyclic.

**Theorem 2:** The proposed routing algorithm is deadlock-free.

**Proof:** It can be concluded from Lemma 1 & Lemma 2 and using Theorem 1 that the proposed routing algorithm is deadlock-free. Non-minimal routing algorithms are susceptible to livelock. The proposed routing algorithm is proved to be livelock-free using the following theorem.

**Theorem 3:** The proposed routing algorithm is livelock-free.

**Proof:** From the discussion in Section II, we notice that whenever a packet is routed in the east direction, it may not be routed back in the west direction. So, in the worst scenario, a packet may reach the west-most column and then start to move towards the destination column. Similarly, whenever a packet is routed in the north direction, it is not allowed to route back to south; and if a packet is routed in the up direction, it is not allowed to route back down. So, in the worst case, a packet may reach the down-most $XY$-plane and then start to move towards the destination node. In each dimension, only one 180-degree turn is allowed. Therefore, after a limited number of hops, the packet reaches its destination node. Thus, the proposed routing algorithm is livelock free.

**III. PERFORMANCE EVALUATION**

We have evaluated the proposed routing method, CHARM (2D and 3D versions), with real and synthetic traffic profiles. Simulation results for a 3D ($6 \times 6 \times 6$) mesh and for a larger 2D ($10 \times 10$) mesh have similar behaviors [3]. Thus, we have shown results for 2D-CHARM.

To evaluate the effectiveness of CHARM, we have implemented three other well-known routing methods, named XY, Mad-y [9], and LEAR [12]. We modified and extended, a cycle-accurate SystemC-based NoC simulator. For all experiments, we consider a $7 \times 7$ mesh. The packet size and input-channel buffer size for each virtual channel are kept constant for all experiments and set to 6 and 8 flits, respectively. The simulator is run for 10000 cycles to discount any start-up transients and the average performance is measured over another 10000 cycles, out of which traffic is generated over 75000 cycles. The congestion threshold is set to 60% of the total buffer size. As the communication performance parameter, we consider latency (delay). It is defined as the time difference (in clock cycles) between the header flit injection from the source router and tail flit reception at the destination router.

**A. Synthetic Traffic Patterns**

In synthetic traffic, we consider both uniform (random) and non-uniform (hotspot and transpose) traffic patterns. With uniform traffic, a node sends several packets to every other node with the same probability. Non-uniform traffic patterns are considered as more realistic traffic profiles, because in most applications, tasks communicate frequently with a small subset of nodes (memory nodes, I/O resources). Under hot-spot traffic patterns, some nodes are appointed as hot-spot nodes, which receive additional traffic to their normal uniform traffic. For simulation, we set nodes 10, 12 and 14 as hot-spot nodes with 0.4 probability of getting additional traffic. With the transpose traffic profile, a node at location $(X, Y)$ only sends packets to another node at location $(n-1-X, n-1-Y)$, for a $n \times n$ 2D mesh. This traffic profile results into heavy traffic flows for the central nodes creating network "hot spots".

At lower traffic loads, all algorithms exhibit similar average latency trends for all synthetic traffic, because of the absence of “hot spots”. But with an increase in the packet injection rate, the network gets congested and all algorithms behave differently. Under uniform traffic, the dimension order routing (XY) outperforms all other adaptive routing methods, as expected and as shown in Fig. 2(a). Because XY incorporates relatively long term and more global information about uniform traffic load characteristics [3], we observe that CHARM performs better than other adaptive routing methods at higher traffic loads.

Figures 2(b) and 2(c) show average latency for non-uniform traffic. Trends for both hot-spot and transpose are similar. It can be observed that deterministic XY has a higher average latency than the three adaptive algorithms. When multiple traffic flows are oriented towards a small subset of “hot spot” nodes, a deterministic XY router will be compelled to forward them towards the same output direction, thus saturating the VC queues. On the other hand, adaptive algorithms can direct packets, directed to the same destination, to different output channels. CHARM achieves smaller average latencies because it can more evenly distribute traffic in a congested network using additional paths (both minimal and non-minimal) than other routing algorithms, by avoiding “hot-spots".
Fig. 2: Performance for (a) Uniform traffic (b) Hot-spot traffic (c) Transpose Traffic (d) Real traffic benchmark (E3S) (e) Average power consumption/router under hot-spot traffic

B. Real Application Traffic

To evaluate the proposed work in a more realistic scenario, we used the E3S benchmark suite. We selected four application suites: automotive/industrial, networking, consumer and office-automation to represent various applications of real-time embedded systems. Each application suite is represented by communication task graphs (CTGs) which give communication patterns and communication volumes among tasks. A random mapping algorithm is used to compute the locations of cores within the NoC to enable honest and intuitive comparisons between routing algorithms. We executed routing algorithms several times using random mapping and the average of the simulation results is used. Figure 2(d) shows average packet latency normalized to XY routing. CHARM provides lower latency than other methods across all four application suites. The average performance gain of CHARM is up to 28% across all selected benchmarks compared with XY and 11% compared with other adaptive algorithms for a 7 × 7 mesh.

C. Power Analysis

The ORION [17] tool is used for power estimation and is integrated with the NoC simulator. It estimates the total power consumption of a router with various sub-components: input buffers, router control logic (arbiter and crossbar) traversal and channels. Figure 2(e) illustrates the average power consumption for hotspot traffic with different traffic loads. XY consumes less power for all traffic loads because it always routes packets through minimal paths. At lower traffic loads, CHARM perform better than other adaptive routing methods as it uses minimal paths due to small “hot spot” creation at lower traffic loads. However, other adaptive methods perform better than CHARM at higher traffic loads.

IV. CONCLUSION

Ayclic CDG for deadlock freedom results in a lower degree of adaptiveness. In this paper, we present a novel and improved turn model, CHARM, for highly adaptive routing for 2D and 3D meshes. CHARM provides a higher degree of adaptiveness by allowing cycles in the CDG. It uses only one additional virtual channel in each Y and Z dimension. It can use minimal or non-minimal routes between source and destination nodes depending on the congestion status to achieve improved performance.

REFERENCES


