FLOW ORIENTED ROUTING FOR NOCS

Everton A. Carara, Fernando G. Moraes

PUCRS – FACIN – Av. Ipiranga 6681 – Porto Alegre – 90619-900 - Brazil
everton.carara@pucrs, fernando.moraes@pucrs.br

ABSTRACT

Several NoC routing schemes proposals targeting overall performance optimization are available in the literature. However, such proposals do not differentiate the application flows. The goal here is to demonstrate that adaptive routing algorithms can be used in flows with temporal constraints, enabling an enhanced degree of path exploration. The main contribution of this work is to expose the routing algorithm at the IP level. Results show gains in latency, throughput and jitter for hotspot scenarios, with minimal area overhead.

I. INTRODUCTION

Current SoCs integrate several applications with different performance requirements. Typically, such applications are composed by several tasks distributed over the processing elements, which communicates through an interconnection infra-structure. Since the interconnection infrastructure is shared by the processing elements, the application flows can collide, creating inter-application interferences. The ability of a system to distinguish application flows, grouping them in different traffic classes, provides the isolation (total or partial) among the different classes. Therefore, it is possible to reduce the inter-applications interference, achieving the required time constraints (e.g. soft QoS) and increasing the system composability [1].

NoC-based SoCs provide natively some degree of flow isolation due to the several independent links, which connect the routers. However, flow collision cannot be avoided when common paths are taken, but it can be reduced using alternative paths. Alternative paths can be provided by adaptive routing algorithms, allowing a better traffic distribution over the network. Taking an alternative non-minimal path can slightly increases the packet latency, but avoids a blocked link that could result in a greater latency overhead.

Several works propose routing schemes to achieve the advantages of deterministic/adaptive and minimal/non-minimal routing algorithms in the same NoC, such as [2]-[9]. In common, such works try to avoid congested regions using alternative paths provided by some adaptive routing algorithm. In all reviewed works, there is no flow differentiation. The main goal is to improve overall performance. Another common feature is the local congestion view, considering the surrounding routers. The efficiency of all methods is questionable, due to their *reduced design space exploration*.

The main goal of this paper is to differentiate flows using a flow oriented routing. In the proposed approach, flows with temporal constraints, such as soft QoS, use adaptive routing, creating more opportunities for *efficient path exploration*. Flows without temporal constraints, such as BE flows, use a deterministic version of the same adaptive routing. Such differentiation conducts to a flow oriented routing scheme that enables flow differentiation. The proposal is, in principle, independent of the selected adaptive routing scheme and NoC, and exposes the routing scheme to the IP level

II. FLOW ORIENTED ROUTING

Flow oriented routing is a feature that can be added to any adaptive routing algorithm. The basic condition is that there exists a deterministic version of the selected adaptive algorithm. It can be proved that such a version always exist, by fixing a single path between each source/target pair from all paths usable by the adaptive algorithm.

At the IP level, the routing version (deterministic/adaptive) is set in run-time for each sent packet. The *routing engine* inside the router is responsible for execute the routing version set by the IP. This information is included in each packet header, similar to the source routing approach. The proposed scheme allows the NoC to *simultaneously* provide adaptive and deterministic routing. As adaptive routing offers alternative paths, it can be applied to high priority flows while low priority flows are routed deterministically.

An important problem in adaptive routing schemes is the output port selection metric. The common policy is to use the neighborhood congestion level as decision metric. Such metric does not ensure a congestion-free path, since this is local information and may lead packets to not locally visible congested areas. Even the Neighbors-on-Path scheme [9], which can detect congestion beyond adjacent neighborhood, does not ensure a congestion-free path. To reduce the area overhead, and keep the implementation as simple as possible, this work does not adopt local congestion detection. When more than one output port is available, the selected port is the one which leads to the shortest path to the target (non-minimal routing algorithm is adopted). If all available output ports lead to shortest paths, the first free port is selected. The area overhead of the proposed scheme is very small (less than 1%), while the presented related works have an area overhead around 5%.

The flow oriented routing can be combined with several adaptive routing algorithms, such as odd-even [6], up*/down* [12] and turn model ones [13]. This work suggests as case study the Hamiltonian routing algorithm [11], due to its simplicity to obtain a deterministic version and the possibility to use it for multicasting without incurring in deadlock risk.

A. Hamiltonian Routing Algorithm

A Hamiltonian path for a graph is any path that visits every graph vertex exactly once. In the Hamiltonian routing algorithm, each NoC router receives a label. In a NoC with *N* routers, the label assignment is based on the router position on a Hamiltonian path, where the first router in the path is labeled *0* and the last one is labeled *N-1*. Figure 1 illustrates a possible label assignment to routers based on a Hamiltonian path in a 4x4 mesh NoC.

The labeling process divides the network in two *acyclic* and *disjoint* subnetworks. The *high-links subnetwork* (solid lines) contains all links whose direction is from lower-labeled routers to higher-labeled routers, and the *low-links subnetwork* (dashed lines) contains all links whose direction is from higher-labeled routers to lower-labeled routers.

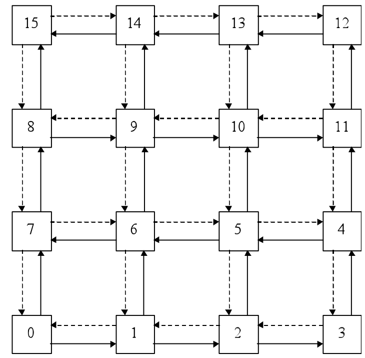

Figure 1 - Example of label assignment based on a Hamiltonian path in a 4x4 mesh.

The *non-minimal partially adaptive* version of the Hamiltonian routing algorithm works as follows. A packet in a router with a label lower than the destination router is forwarded to any higher label neighbor router, which has a label lower or equal to the destination router. Consider, for example, the source router 6 and 14 as the target router. The possible paths taken by the packet are: {6,9,14}, {6,9,10,13,14}, {6,9,10,11,12,13,14}, {6,7,8,9,14}, {6,7,8,9,10,13,14} and {6,7,8,9,10,11,12,13,14}. In a similar way, when a packet is in a router higher than the destination router, it is forwarded to any lower neighbor router, which has a label higher or equal to the destination router. Consider, for example, the source router 13 and 7 as the destination router. The possible paths taken by the packet are: {13,10,9,8,7} and {13,12,11,10,9,8,7}.

To create a *minimal deterministic* version from the partially adaptive Hamiltonian routing algorithm, the forwarding condition can be restricted to "forward to the higher/lower neighbor router, which has a label lower/higher or equal to the destination router" (depending on the source and target labels). In the examples 6→14 and 13→7 the paths are respectively {6,9,14} and {13,10,9,8,7}.

In this paper, the Hamiltonian routing algorithm is used simultaneously in the two presented versions (*i*) non-minimal partially adaptive and (*ii*) minimal deterministic. To allow the routing engine to differentiate packets, one bit of the packet header is defined as the *routing bit*. This routing bit is set by the IP and specifies which version of the routing algorithm should be executed.

III. RESULTS

This section presents the obtained results using the HERMES NoC [10] supporting the Hamiltonian routing algorithm combined with the flow oriented scheme. All simulations were conducted in RTL VHDL, ensuring accuracy to the results.

The evaluated performance figures are *latency*, *throughput* and *jitter*. The transmission of a given flow through the NoC may modify the original flow rate, inducing variable latency values, resulting in missed deadlines at the target IP. Jitter is this instantaneous variation in latency, and must be minimized in applications with QoS constraints.

Two spatial traffic distributions are evaluated: hot-spot and complement. Figure 2 presents the hot-spot traffic distribution. In this experiment, all packets have 50 flits. The dashed line corresponds to the evaluated flow (2→22) and the solid lines are disturbing flows. The path taken by the flows in Figure 2 considers the deterministic version of the Hamiltonian routing algorithm.

The evaluated flow, 2→22, has a fixed injection rate of 30% of the link bandwidth and the disturbing flows have their injection rate varying from 5% to 50%.

The flow 2→22 is evaluated in three routing scenarios: (1) *deterministic* – all flows are routed using the minimal deterministic version of the Hamiltonian algorithm; (2) *adaptive* - all flows are routed using the non-minimal partially adaptive version of the Hamiltonian algorithm; (3) *flow oriented* – only the flow 2→22 is routed using the non-minimal partially adaptive routing whereas the disturbing flows are routed deterministically.

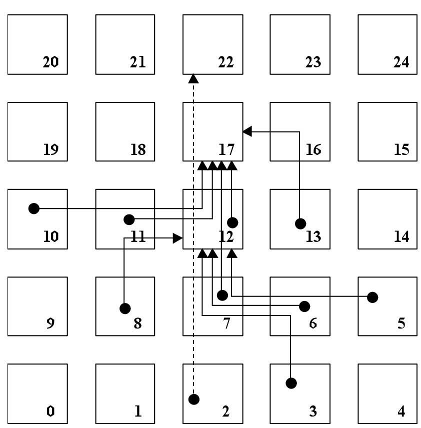

Figure 2 – Hot-Spot traffic distribution.

Figures 3-5 show the obtained results for the flow 2→22. Jitter results were obtained considering a disturbing injection rate equal to 20% of the link bandwidth. For other reasonable rates (below the NoC saturation point), the behavior is similar.

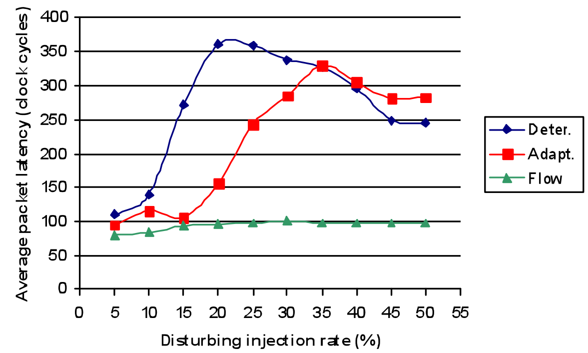

Figure 3 – Average latency for flow 2→22.

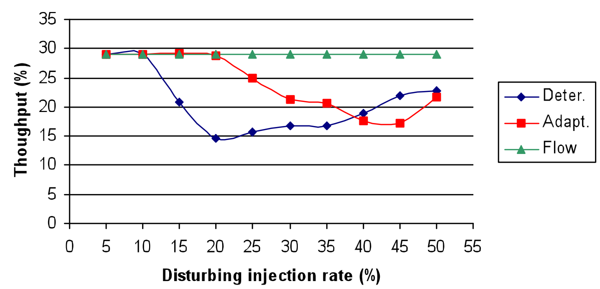

Figure 4 – Average throughput for flow 2→22.

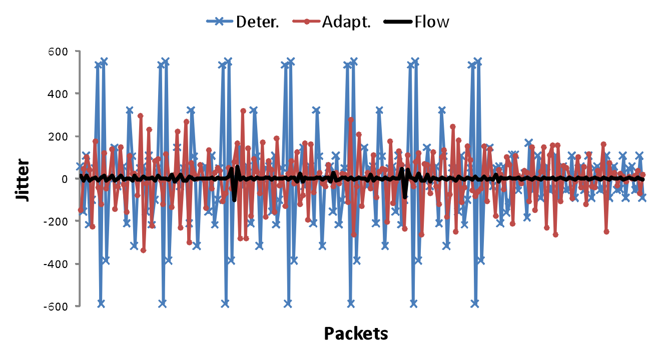

Figure 5 – Jitter for flow 2→22, in clock cycles.

Note the latency and throughput behavior when all flows are routed using the same algorithm (deterministic or adaptive). Adaptive routing performs well for moderately disturbing traffic rates. With highly disturbing traffic rates the benefits of adaptive routing are lost.

The latency and the throughput of flow 2→22 is not affected by the disturbing traffic when using the flow oriented routing scheme, since it is the only one able to explore new paths using adaptive routing. For example, when router 7 North port is not available, flow 2→22 can take the path {2, 7, 8, 11, 18, 21, 22}, which has less traffic. When all flows are routed deterministically, flow 2→22 cannot avoid the hot-spot area and its performance is strongly affected. The same performance penalty can be observed when all flows are routed using adaptive routing. In this case when router 7 North port is not available, disturbing flows may try alternative paths using router 7 West port, which in turn may disturb flow 2→22, since now this latter has to share the NoC resources with the disturbing flows.

The same behavior is observed when jitter is evaluated as shows Figure 5. Adaptive routing attenuates jitter, compared to deterministic routing, but do not eliminate it. The flow oriented routing in practice practically removed all jitter for flow 2→22, with almost all packets having the same latency. This result demonstrates the effectiveness of the flow differentiation at the IP level.

The good result obtained with the hot-spot scenario comes from *uncongested areas available*, allowing the adaptive routing algorithm to find alternative paths in these areas for the flow 2→22. A quite different situation arises when using the complement traffic distribution, with the *load evenly distributed* inside the network, without hot-spot regions. In the experiment conducted in this situation, all routing methods presented similar results with a small performance gain for the flow oriented routing for most disturbing injection rates. Such experiments highlight the benefits of differentiating flows when hot-spots may appear during some application execution. Meanwhile, they also present the *limitations* of the flow oriented routing when the search space for non-congested regions is restricted, as is the case for complement traffic distributions.

IV. CONCLUSIONS AND FUTURE WORK

The main contribution of this work is to show the benefits of exposing the NoC routing algorithm to the IP level. In this way, the design space exploration is expanded since the routing algorithm can be partially controlled. The proposed method has a minimal area overhead being general, simple, efficient and not targeted to a specific NoC, neither to a specific routing algorithm.

The main idea behind flow oriented routing is to restrict path exploration to flows with performance requirements, reducing the resources concurrence. There is no reason to enable BE flows to be routed adaptively, since they can unnecessarily disturb flows with some QoS constraints. Therefore, the evaluation of new routing proposals should change the way to evaluate performance, since, e.g., reduce the overall latency could not fulfill real-time requirements in MPSoCs with multiple applications running simultaneously.

As future work, the proposed scheme will be combined with other QoS mechanism to increase the support to applications with real time constraints.

ACKNOWLEDGMENTS. This research was supported partially by CNPq (Brazilian Research Agency), project 301599/2009-2.

REFERENCES

[1] Kumar, A.; et al *"Analyzing composability of applications on MPSoC platforms"*. Journal of Systems Architecture, 54(3-4), 2008, pp. 369-383.

[2] Nilsson, E.; Millberg, M.; Oberg, J.; Jantsch, A. *"Load distribution with the proximity congestion awareness in a networks on chip"*. In: DATE, 2003, pp. 1126-1127.

[3] Kumar, S.; Jantsch, A.; Soininen, J.-P.; Forsell, M.; Millberg, M.; Oberg, J.; Tiensyrja, K.; Hemani, A. "*A Network on Chip Architecture and Design Methodology"*. In: ISVLSI, 2002, pp.105-112.

[4] Ye, T.; Benini, L.; Micheli, G. *"Packetization and routing analysis of on-chip multiprocessor networks"*. Journal of Systems Architecture, 50(2-3), 2004, pp. 81-104.

[5] Hu, J.; Marculescu, R. "*DyAD-Smart Routing for Networks-on-Chip*". In: DAC, 2004, pp. 260-263.

[6] Chiu, G. *"The Odd-Even Turn Model for Adaptive Routing"*. IEEE Transactions on Parallel and Distributed Systems, v.7(11), 2000, pp. 729-738.

[7] Sobhani,A; Daneshtalab, M.; Neishaburi, M.; Mottaghi, M.; Afzali-Kusha, A.; Fatemi, O.; Navabi, Z. "*Dynamic Routing Algorithm for Avoiding Hot Spots in On-chip Networks*". In: DTIS, 2006, pp. 179-183.

[8] Li, M.; Zeng, Q.; Jone, W. *"DyXY – A Proximity Congestion-Aware Deadlock-Free Dynamic Routing Method for Networks on Chip"*. In: DAC, 2006, pp. 849-852.

[9] Ascia, G.; Catania, V.; Palesi, M.; Patti, D. *"Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip"*. IEEE Transactions on Computers, 57(6), 2008, pp. 809-820.

[10] Moraes, F.; Calazans, N.; Mello, A.; Moller, L.; Ost, L. "*HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip*". Integration the VLSI Journal, 38(1), 2004, pp. 69-93.

[11] Lin, X.; McKinley, P. K.; Ni, L. M. "*Deadlock-free Multicast Wormhole Routing in 2-D Mesh Multicomputers*". IEEE Transactions on Parallel and Distributed Systems, 5(8), 1994, pp. 793-804.

[12] Schroeder, M.; Birrell, A.; Burrows, M.; Murray, H.; Needham, R.; Rodeheffer, T.; Satterthwaite, E.; Thacker, C. "*Autonet: A High-Speed Self-Configuring Local Area Network Using Point-to-Point Links"*. IEEE Journal on Selected Areas in Communications, 9(8), 1991, pp.1318-1335.

[13] Glass, C. J.; Ni, L. M. "The Turn Model for Adaptive Routing". Journal of the Association for Computing Machinery, 41(5), 1994, pp. 874-902.