Deadlock-Free Multicast Routing Algorithm for Wormhole-Switched Mesh Networks-on-Chip

Everton Alceu Carara, Fernando Gehm Moraes
PUCRS – FACIN – Av. Ipiranga 6681 – Porto Alegre – 90619-900 - Brazil
carara@inf.pucrs.br, moraes@pucrs.br

Abstract

An important service in distributed systems, as multi-processors, is the ability to transmit multicast messages. Cache coherence protocols and parallel algorithms are examples of applications requiring multicast messages. To transmit a multicast message to n *targets, in networks-on-chip without multicast service, the source router must transmit* n *identical messages. Few works in the literature describe multicasting in NoCs. The goal of this work is to implement a deadlock free routing algorithm for wormhole-switched mesh NoCs, enabling to transmit simultaneous multicast messages. The dual-path multicast algorithm, used in multicomputers, is adapted to NoCs in this work. The dual-path multicast algorithm is implemented for circuit and packet switching. The evaluation comprises: (*i*) comparison between the algorithms; (*ii*) NoC performance as a function of the percentage of injected multicast messages; (*iii*) performance gains obtained with the dual-path algorithm when compared to the transmission of single unicast messages.*

1. Introduction

Performance gains provided by multiprocessor architectures (MPSoCs) are not only related to the computational power of the several processing elements. The interconnection architecture, responsible by the communication among the several processing elements, has an important contribution in the overall performance. NoCs can be seen as the main interconnection architecture responsible by the future of the multiprocessed technologies, which are rapidly prevailing in SoCs [1].

As MPSoC communication infrastructure is shifting from buses to NoCs, some buses services must be available in NoCs, in particular multicast communication. Multicast messages are responsible by the efficient execution of several parallel applications, as search algorithms, graph algorithms and matrix operations (e.g. inversion, multiplication). In addition, multicast messages are employed in management and network configuration, synchronization and cache coherence protocols.

Although bus-based interconnection architectures (single bus, segmented bus and hierarchical bus) are not scalable and provides low support to parallel transactions, these communication schemes natively supports multicast and broadcast services. In NoCs, either packet or circuit switching, multicasting depends on special algorithms that are deadlock sensitive. Furthermore, as multicast messages allocate several network resources, the overall system performance may be seriously degraded.

This work presents a deadlock-free adaptation of the dual-path multicast algorithm [2] for NoCs, targeting wormhole-switched 2-D mesh topology. The algorithm is implemented for packet and circuit switching modes. The impact of the multicast messages in the performance is evaluated, demonstrating the efficiency of the algorithm.

This paper is organized as follows. Section 2 discusses deadlock situations when multicast messages are transmitted in wormhole packet switching NoCs. Section 3 details the multicast algorithm implementation. Section 4 presents experimental results obtained from RTL simulations and Section 5 concludes this paper.

2. Deadlock in multicast wormhole

Several unicast deadlock-free routing algorithms have been proposed for wormhole meshes, which can be extended for multicast traffic. These algorithms guarantee the absence of cyclic dependence among communication channels (neighbor nodes) through routing constraints. Since unicast traffic allocates only one consumption channel (at the target node), there is

978-0-7695-3170-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ISVLSI.2008.18

no cyclic dependence among simultaneous unicast messages. However, deadlock-free unicast algorithms, when applied to multicast traffic, guarantee absence of deadlock only for one multicast message at a time. Due to the multi-target nature of the multicast traffic, several consumption and communication channels are allocated for only one message. That is the reason why simultaneous multicast messages are sensitive to deadlock on consumption channels.

Figure 1 illustrates an example of deadlock on consumption channels, due to two simultaneous multicast messages in a wormhole mesh. The consumption channel corresponds to the router → IP interface (local port of routers), represented in the Figure 1 by the arrow between input buffer and IP buffer (sink). Both messages are addressed to router 1 and 2.

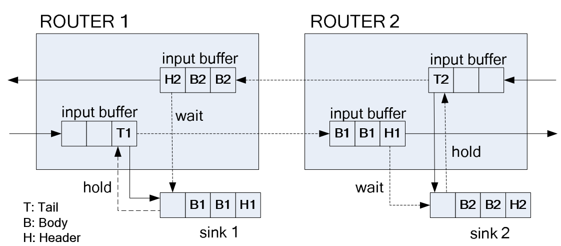

Figure 1. Deadlock in the consumption channels in wormhole-switched NoC [3].

Observe in router 2 that the packet 1 (suffix 1) cannot advance because the consumption channel (sink 2) is allocated by packet 2. The same situation can be observed in router 1, in which packet 2 cannot advance because the consumption channel (sink 1) is allocated by packet 1. Both packets are in a cyclic dependence creating a deadlock situation. Clearly, this situation does not arise in store-and-forward, since the packet only advances to the next router if there is enough space to store it. However, the maximum packet size, in this case, is limited by the size of the router input buffers, and the total latency increases.

To minimize the message latency, when it arrives at a destination router, the flits are simultaneously forwarded to the consumption channel and to the communication channel with the neighbor router. An alternative approach is the absorb-and-retransmit, where a message is first absorbed by the IP and then retransmitted to the neighbor router. This approach is similar to store-and-forward and besides the increase in latency, it is also sensitive to deadlock on consumption channels [4]. In [4], Boppana et al. presents a solution to eliminate this deadlock, by multiplexing (virtual channels) or by replicating the consumption channels.

3. Dual-Path multicast for NoCs

The dual-path multicast algorithm was adapted for NoCs in two versions: (*i*) using circuit switching, proposed in this paper; (*ii*) using packet switching, as proposed originally in [2] for multicomputers. The reasoning to employ circuit switching instead of packet switching is to send only one copy of the message, independently from the number of destinations [3]. A routing algorithm based on Hamiltonian paths [5] is used for both unicast and multicast messages. Thus, the use of the same routing algorithm for these two types of traffics eliminates deadlock, since there is compatibility on the messages routing [4].

3.1. Hamiltonian routing algorithm

Several routing algorithms are proposed in the literature to be employed in NoCs, from the deterministic XY to more adaptive routing ones, such as odd-even, up*/down* and turn-prohibition based. This work suggests the use of the Hamiltonian routing algorithm, due to the possibility to use it for multicasting, without incurring in deadlock.

A Hamiltonian path is a path in a graph with all vertex visited once. Each NoC router receives a label. In a NoC with *N* routers, the assignment of a label to a router is based on the position of the router on the Hamiltonian path, where the first router in the path is labeled 0 and the last is labeled *N-1*. Figure 2 illustrates one possible Hamiltonian path in a 4x4 mesh NoC.

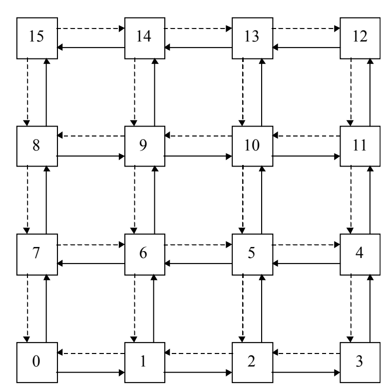

Figure 2. Example of Hamiltonian path in a 4x4 mesh.

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

The Hamiltonian routing algorithm works as follows. A message in a router with a label smaller than the destination router is forward to the greater neighbor router, which has a label smaller or equals to the destination router. Consider, for example, the source router 6 and 12 as the target router. The path taken by the message is: 6 → 9 → 10 → 11 → 12. In a similar way, when a message is in a router greater than the destination router, it is forwarded to the smaller neighbor router, which has a label greater or equal to the destination router. Consider, for example, the source router 13 and 7 as the destination router. The path taken by the message is: 13 → 10 → 9 → 8 → 7.

This routing algorithm, differently from the XY routing algorithm, may be partially adaptive, if the condition to forward the message is relaxed to "forward to one of the greater neighbor routers, which has a label smaller or equals to the destination router" (for target index greater than source index). In the example 6→12 the path could be 6 → 7 → 8 → 9 → 10 → 11 → 12, if the north port of router 6 is being used by another flow.

3.2. Connection-oriented dual-path multicast

The first step in the connection-oriented dual-path multicast algorithm is to establish a connection between the source router with all targets. The source router must separate the set of targets in two subsets, one with routers having labels greater than the label of the source router, and the second one with smaller labels. Each subset is inserted inside a *connection establishment packet*. The packet with labels greater than the source router sorts the labels in an increase order, whereas the other packet contains labels sorted in a decreasing order. Figure 3 illustrates the connection establishment packets, considering a multicast message from router 6 to routers 1, 3, 5, 8, 10, 12 and 14. The network may differentiate control packets (e.g. connection establishment) from data packets using dedicated bits in the packet header, or using a sideband signal.

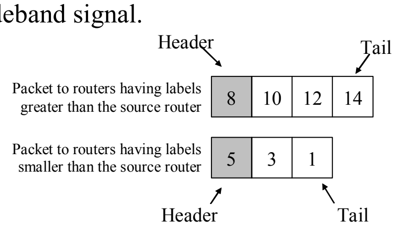

Figure 3. Connection establishment packets format.

The source router transmits the first packet (with higher labels), and waits the assertion of an acknowledgment signal from the last router. The same procedure is repeated for the second packet. After the reception of both acknowledgments, the second step of the algorithm may starts.

During the connection establishment, the Hamiltonian routing algorithm is executed based in the first flit of the packet (header). When the packet arrives in the router defined in the header, the consumption channel is reserved. The header in then removed, and this new packet is transmitted to the next router. This procedure is repeated while the packet size is different from zero. Figure 4 and Figure 5 illustrate the connection establishment in both directions.

The second step of the algorithm comprises the injection into the network of one copy of the message in both directions, using the connections previously established. The connections are released by the last message *flit*.

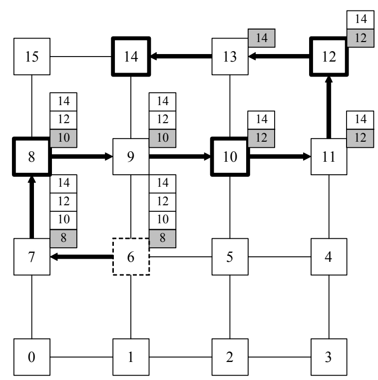

Figure 4. Connection establishment with routers having labels greater than the source router (label 6) using the Hamiltonian routing algorithm.

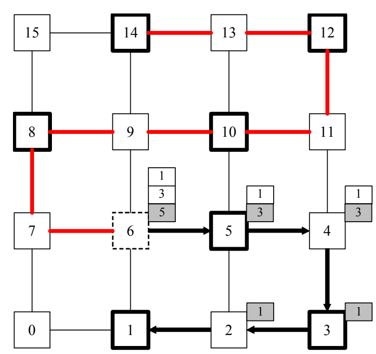

Figure 5. Connection establishment with routers having labels smaller than the source router (label 6) using the Hamiltonian routing algorithm.

When all targets have labels greater or smaller than the source router, only one connection packet is created. In this case, the source router waits only one acknowledgment before inject the message into the network.

3.3. Packet-switching dual-path multicast

As in the connection-oriented algorithm, two packets with sorted labels are created. The difference is that the multicast message is attached to the packet. Figure 6 presents such packets, using the same example of the previous algorithm. The network differentiates the labels from the multicast message using dedicated bits in the packet header.

The source router injects both packets into the network, without connection establishment. The multicast packet reaches the target routers according to the order defined in the header.

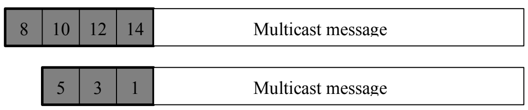

Figure 6. Packet-switching dual-path multicast packets.

When a given router receives the multicast packet, it reserves the consumption channel, removes from the header the matched label and establishes a connection with the next router. Once both ports reserved (consumption channel and some output port), the multicast packet is simultaneously transmitted to the local IP and the neighbor router.

3.4. Deadlock avoidance

In [4], Boppana et al. show that for a given routing algorithm, the minimum number of consumption channels per router to avoid deadlock is equals to the number of acyclic and disjoint subnetworks. As showed in Session 3.1, the Hamiltonian routing algorithm divides the network in two acyclic and disjoint subnetworks. Hence, two consumption channels per router are sufficient to avoid the deadlock on consumption channels (Section 2).

Both versions of the dual-path multicast algorithm replicate [4] the consumption channels to avoid deadlock. The consumption channel replication has a small impact in area, and improved performance when compared to virtual channels [6]. In the current implementation, the replication occurs in the Router → IP direction (Figure 7), resulting in two consumption channels per router. If one of the consumption channels is receiving data and the second consumption channel starts receiving data simultaneously, the NI connected to the router may be able to store at least one message (favors the network performance); or it waits the complete reception of the first message, only reserving the second consumption channel (small area overhead).

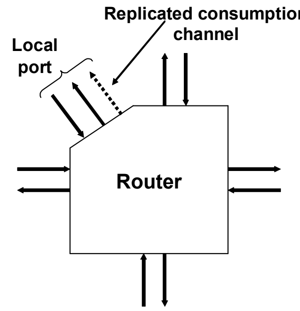

Figure 7. Consumption channel replication.

For multicast messages (or for connection establishment packets), the consumption channel 1 is reserved for routers with labels greater the source router, otherwise channel 2 is reserved. For unicast messages the free first consumption channel is used.

4. Results

Both dual-path multicast algorithms were described in synthesizable VHDL, deriving its structures from the *HERMES* NoC [7]. Each NoC router has an IP traffic generator connected to the local port. Performance figures are obtained from VHDL simulation. Table 1 presents the common NoC features.

Table 1. Common NoC features

Flit/Phit	8 bits
Flow Control	Credit Based
NoC Topology	Mesh 4x4
Routing Algorith	Hamiltonian Routing
Buffer Deph	16 positions

The two dual-path multicast algorithms (*PS* – packet switching e *CS* – circuit switching) are compared to the *individual* algorithm. In the *individual* algorithm, one unicast message is sent to each multicast target [4], resulting in *n* individual messages to *n* targets. The unicast traffic used in all experiments is connection oriented. The time spent to create multicast packets, sorting the list of targets, is not considered, since it is assumed this is executed by the IP connected to the router (e.g. a cache controller selects the processors to send a write invalidate based on its cache directory).

4.1. Multicast traffic

In this first experiment, only multicast traffic is injected into the network. Even if this is not a realistic traffic scenario in SoCs, the goal is to verify the behavior of the NoC under extreme traffic conditions. In this experiment, all 16 IPs transmit 200 100-flit multicast messages at the NoC rate. The number of targets varies from 2 to 14. Figure 8 presents the simulation time, in clock cycles, to all IPs transmit the 200 messages.

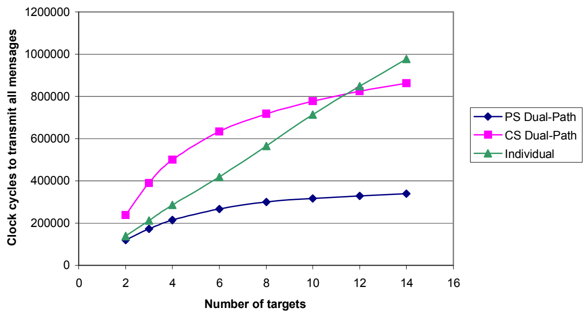

Figure 8. Performance of the multicast algorithms as a function of the number of targets.

As expected, the number of clock cycles to transmit all messages in the *individual* algorithm grows linearly with the number of targets, since each new target adds a new message. On the other hand, the PS and CS dual-path algorithms are less sensitive to the number of targets.

The lower performance of the CS dual-path algorithm comes from the large connection establishment time, due to congestions inside the network. This congestion arrives since the CS dual-path algorithm reserves network resources and must wait for at least one acknowledgement.

The performance of the PS dual-path algorithm is superior, due the absence of path reservation. Once one router is able to reserve the consumption port and one output port, the multicast message may be forwarded.

4.2. Simultaneous multicast and unicast traffic

In this second experiment, multicast and unicast flows are simultaneously injected into the network. All IPs sent 200 100-flit messages, at a relative rate to the channel bandwidth equal to 14% (below the network saturation point). Among the 200 messages per IP, 10% are multicast and 90% unicast. The number of targets varies from 2 to 14. Figure 9 presents the results.

Comparing Figure 8 (higher injection rate) to Figure 9 (lower injection rate), the impact of the connection establishment is reduced, since there is more idle time between packets, making the CS dual-path algorithm faster than the *individual* algorithm. The PS dual-path algorithm surpasses the CS dual-path and *individual* algorithms for higher and lower injection rates.

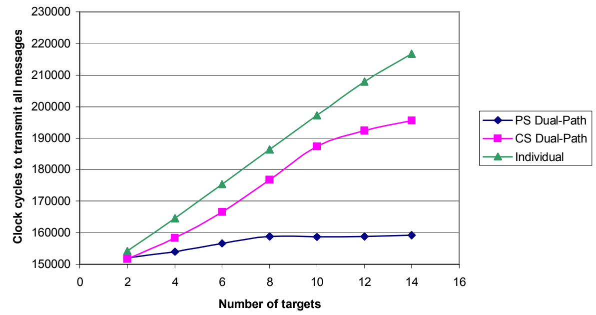

Figure 9. Traffic 10% multicast and 90% unicast, at a relative injection rate of 14%.

Figure 10 presents the performance of the multicast algorithms as function of the percentage of the multicast messages sent by IPs. All multicast messages have 4 targets.

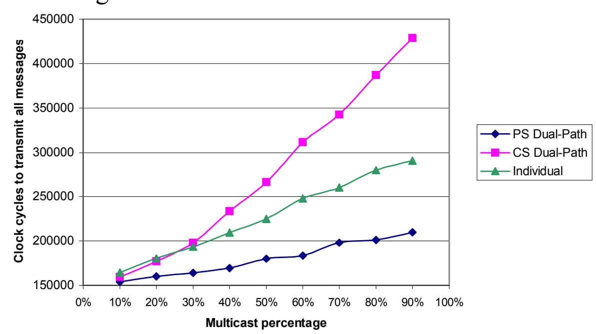

Figure 10. Performance of the multicast algorithms considering the percentage of multicast messages.

For small injection rates, small number of targets (4) and a small percentage of multicast messages the three algorithms have similar performance (difference below 10%). The performance of the PS dual-path algorithm is superior to both algorithms in all other cases. The performance of the *individual* algorithm is a function of the number of targets. The CS dual-path algorithm is recommended only for small injection rates due to the connection establishment time.

4.3. Impact of the multicast traffic over the unicast traffic

Figure 11 compares the impact of the multicast traffic over pure unicast traffic. Each IP transmit 200 100-flit messages, at a relative rate varying between 9% and 50%. The scenarios with multicast messages, 10% of the messages are multicast, and the number of targets is fixed at 8.

For small injection rates, as 14%, the PS dual-path algorithm penalizes only 7% of the network performance, while CS dual-path and individual algorithms penalize 28% and 38% respectively the network performance. For typical injection rates, between 20% and 25%, the PS dual-path has an average impact on the performance around 10%. If an higher injection rate is considered, as 50%, the impact on the performance of the PS dual-path is 32% and 65% for the individual algorithm.

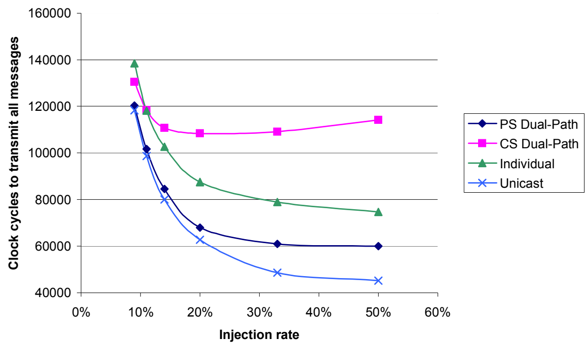

Figure 11. Impact of the multicast traffic over a unicast traffic (8 targets, 10% multicast).

Figure 11 enables also to quantify the performance gains when including in NoCs multicast services. This can be achieved comparing the curve *individual* to the curve PS dual-path. For injection rates between 20% and 50% the performance gains is 20% when including multicast services in the NoC (for 8 targets, 10% multicast).

4.4. Area overhead

Table 2 presents the dual-path area overhead for a five ports router, targeting a Virtex 2VP30 FPGA. The area overhead is the same for both algorithm versions. For a single router, a 20.2% area overhead is observed when implementing the dual-path multicast algorithm. Part of the additional logic corresponds to the multicast header analysis during the routing. The major part of the area overhead is induced by the channel consumption replication, which avoids the deadlock, and increases the NoC throughput.

Table 2. Dual-path area overhead, targeting a Virtex 2VP30 FPGA.

Resource	5 ports router	5 ports router with dual-path	Available
Slices	302	363	13696
LUTs	603	726	27392
Flip Flops	189	211	29060

5. Conclusions

This paper presented an adaptation of the dual-path multicast algorithm for NoCs in two versions, varying the switching mode. The deadlock avoidance is achieved replicating the consumption channels. This feature enables the injection of several simultaneous multicast messages into the network.

The packet switching version is superior to the connection-oriented version, due to the absence or resources reservation. The claimed advantage of the connection oriented version presented in the literature is that only a single copy of the message need to be injected in the network, but the connection establishment with all destinations increases the congestion inside the NoC, and penalizes its performance.

Two other important results are advanced. The first one concerns the comparison of the PS dual-path to the *individual* algorithm. This comparison quantified the performance penalty in NoCs without multicast services. The second one regards the impact of the multicast traffic over the unicast traffic. Employing the PS dual-path the performance impact of the multicast traffic over the unicast traffic, for typical application rates, is smaller than 10%.

Future work includes: (*i*) evaluate the impact on power consumption in NoCs when multicast messages are inject; (*ii*) evaluate the performance of the multicast algorithms replacing traffic generators by processors (real MPSoC environment).

6. Acknowledgements

This research was supported partially by CNPq (Brazilian Research Agency), project 300774/2006-0.

7. References

[1] Bjerregaard, T.; Mahadevan, S. "*A survey of research and practices of Network-on-chip*". ACM Computing Surveys, v.38(1), 2006, pp. 1-51.

[2] 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.

[3] Lu, Z.; Yin, B.; Jantsch, A. *"Connection-oriented Multicasting in Wormhole-switched Networks on Chip"*. In: Emerging VLSI Technologies and Architectures (ISVLSI), 2006.

[4] Boppana, R. V.; et al. S. *"Resource Deadlocks and Performance of Wormhole Multicast Routing Algorithms"*. IEEE Trans. on Parallel and Distributed Systems, 9(8), 1998, pp. 535-549.

[5] Harary, F. "Graph Theory". Reading, MA: Addison-Wesley, 1972.

[6] Carara, E; Moraes, F.; Calazans, N. "*Router Architecture for High-Performance NoCs*". In: SBCCI, 2007.

[7] Moraes, F.; et al. "*HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip*". Integration the VLSI Journal, v.38(1), Oct. 2004, pp. 69-93.