Adaptive-Tree Multicast: Efficient Multi-destination Support for CMP Communication Substrate

Pablo Abad, Valentin Puente, Lucia. G. Menezo, Jose Angel Gregorio, *Member IEEE*

***Abstract*—Multi-destination communications are a highly necessary capability for many coherence protocols in order to minimize on-chip hit latency. Although CMPs share this necessity, up to now few suitable proposals have been developed. The combination of resource scarcity and the common idea that multicast support requires a substantial amount of extra resources is responsible for this situation. In this work, we propose a new approach suitable for on-chip networks capable of managing multi-destination traffic via hardware in an efficient way with negligible complexity. We introduce a novel multicast routing mechanism, able to circumvent many of the limitations of conventional multicast schemes. Adaptive-tree multicasting is able to maintain correctness for multi-flit multicast messages without routing restrictions, while also coupling correctness and performance in a natural way. Replication restrictions not only guarantee the presence of enough resources to avoid deadlock, but also dynamically adapt tree shape to network conditions, routing multicast messages through non-congested paths. The performance results, using a state-of-the-art full system simulation framework, show that it improves the average full system performance of a CMP by 20% and network ED2P by 15%, when compared to a state-of-the-art router with conventional multicast support and similar implementation cost.**

***Index Terms*—Chip Multiprocessor (CMP), Multicast and Broadcast communications, Network-on-Chip, Router Microarchitecture.**

I. Introduction

THE interconnection network has a critical role to play in general-purpose CMPs, and it should not be considered a passive component of memory hierarchy. The coupling of coherence protocols and network components provides a great opportunity to co-design the two components as an integrated structure. Tight integration of network logic and coherence protocol has already demonstrated that the joint effort of the two components provides clear performance benefits. Global ordering maintenance [6] and redundant multi-destination request filtering [7] are two examples of the potential benefits extracted from embedding additional logic inside network routers.

One of the basic issues where interconnection network can be helpful is in the optimization of multi-destination communications. A wide variety of implemented and proposed coherence protocols make use of this kind of communications, making the inclusion of hardware mechanisms desirable for their efficient support. Broadcast-based protocols such as TokenB [36] or Intel QPI [24] rely on broadcast requests in order to eliminate communication indirections through serialization points. The implementation of fast cache-to-cache accesses via broadcast messages comes at the cost of increased bandwidth requirements. The exclusion of a communication substrate able to efficiently manage this traffic overhead could make this kind of protocols much less attractive [25]. The point-to-point nature of communications in directory-based protocols could eliminate scalability problems, alleviating the need for multicast support. However, even directory coherence sometimes makes use of one-to-many communications. Protocols such as [32] perform block invalidations through messages sent to multiple sharers, and could also be able to extract performance benefits from multicast support.

In a communication subsystem without hardware multicast support the only way to perform multi-destination communications is by decomposing each multicast message into multiple unicast messages, one for each individual destination, while maintaining the interconnection network unaware of this kind of communications. The main limitations of this solution are the inefficient utilization of network resources (because of the reiterative resource utilization of decomposed messages belonging to the same multicast communication) and the waiting time overhead at injection queues (due to the unavoidable need to serialize the use of output ports).

In an attempt to minimize the overhead of message decomposition, some solutions have already been proposed for CMP environments [25][47]. These proposals rely on classical input-buffered router architectures, inheriting some of the limitations intrinsic to the underlying structure. Sub-optimal routing due to deadlock avoidance [33], an increased number of virtual channels to implement multi-destination communications [33] or serialization at crossbar traversal for replicated messages [35] are almost unavoidable limitations when working with this kind of structures.

The utilization of the Rotary Router [1] organization as a starting point enables us to explore a common problem from a

The authors are with the Electronics and Computers Department, University of Cantabria, 39005 Santander, Spain.
E-mail: {abadp, vpuente, monaster}@unican.es

new viewpoint. In this paper we present an on-network multicast support mechanism able to handle multi-destination communications in an extremely efficient way. Dynamically adapting multicast management to network conditions, we will be able to maximize network performance. The particular characteristics of the router will allow us to achieve these results with a minimal hardware overhead. This is an extended and improved version of the work presented in [3].

The rest of the paper is organized as follows: Section II provides detailed background on this research area, also analyzing the most relevant proposals in the on-chip context. Section III describes our proposal, including a detailed router structure description and provides the necessary correctness substrate. Section IV thoroughly analyzes performance through both synthetic traffic patterns and full-system evaluations. Finally, Section V states the main conclusions of the paper.

II. Background (state of the art)

Multicast hardware support has always been a hot topic in off-chip interconnection networks, generating a large number of proposals [12][14][31][33][35][42]. Among the multiple existing solutions, hardware-based schemes for multicast support can be divided into two main groups, path-based and tree-based multicast. The main difference between these two types of multicast support lies in the way messages are routed through the network in order to reach every destination. In path-based multicast, every message destination is covered in a sequential order, performing message replication only when an intermediate node is also a message destination. In contrast, tree-based multicast tries to minimize multicast communication latency, allowing replications at intermediate nodes even without belonging to the destination vector. In this paper we will propose a third multicast category named the adaptive-tree multicast, which can dynamically switch between path and tree-based solutions. The set of mechanisms able to implement this new category is the main contribution of this work. Any of these three solutions significantly improves the unicast approach. In this section we will analyze the advantages and disadvantages of each scheme, describing some of the most significant proposals and analyzing their suitability for CMP environments.

A. *Tree-based Multicast*

In tree-based multicast a multi-destination message is routed along a common path as far as possible. At a router where different destinations can be reached through different output ports the message is replicated and moved into different output channels. Each message copy is generated for a disjoint set of destinations. This branching continues as necessary until all destination nodes have been reached.

Tree-based routing has the advantage that no ordering of the destinations is required before injection in order to minimize distance. The shortest path between the source node and all destinations is always taken. The main problem of this scheme is the increased blocking probabilities at intermediate nodes. Message branches can create new dependencies leading to deadlock situations [31].

The straightforward solution to avoid deadlock could be the implementation of multicast trees by extending a deadlock-free unicast routing algorithm to handle multicast traffic. For example, a multicast tree algorithm can be designed for a 2D mesh based on Dimension Order Routing [33]. However, this is only correct if Virtual Cut Through flow control is employed or if multicast messages are restricted to 1-flit size. With wormhole flow control, channel reservation could lead to deadlock situations if longer multicast messages are employed, as shown in [33]. This work, named Double-Channel XY algorithm, implements deadlock avoidance with the inclusion of additional virtual channels. This mechanism divides the network into four sub-networks, each of them making use of a disjoint set of virtual channels. Moreover, the multicast destination set is broken up into four subsets, and each message copy is routed through a different sub-network following a Dimension-Order policy. Unfortunately, this mechanism was later demonstrated to be incorrect, leading to deadlock situations [12].

Due to the difficulties faced in guaranteeing deadlock avoidance, many tree-based multicast mechanisms rely on detection-recovery solutions to provide correctness [14][35][31]. Two examples of this are Branch Pruning [35] and HTA Multicast [31]. Branch Pruning relies on the storage of a local copy of the data carried by the multicast message and the utilization of one message flit to encode each message destination. Only one destination flit travels ahead of data flits, the rest of the header flits being placed at the last positions of the message. In this way, when a new branch is created or a deadlock situation is detected, the packet can be broken and the header flit causing the deadlock or moving to the new branch can make use of the local data copy to create a new message. The existence of the local data copy eliminates the need for flit-by-flit replication, breaking the possible dependencies between virtual channels. The HTA mechanism makes use of a special output queue to store messages considered to be in a potential deadlock situation. When this network is full and another message is potentially deadlocked, an interrupt is generated and the message is absorbed into the local host. These messages will be re-injected into the network after a predefined amount of time.

As well as the difficulties faced in providing correctness, in the presence of high network loads, the generation of new packets at intermediate nodes can increase network congestion supra-linearly [31]. Tree-based approaches perform message replication at intermediate nodes, without injection restriction and this uncontrolled replication favors the appearance of network areas where replicated messages rapidly exhaust buffering resources, increasing contention and causing the earlier appearance of network congestion [33].

B. *Path-Based Multicast*

In path-based multicast, multi-destination messages are routed through the network using information about one single destination. Once this destination is reached, a copy of the message is generated and consumed at the current router and the next position on the destination list is used to continue routing the message to its next destination. After reaching all the positions in the destination list, the message will be consumed without being replicated at this last position.

This scheme has two important advantages over tree-based solutions. First, it has a lower probability of message blocking (routing deadlock) since at most two channels are requested per message (one transit port and the consumption port). Maintaining the same routing rules for unicast and multicast communications is enough to avoid routing deadlock. Second, replication does not increase network occupancy, because every copy of a multicast message is immediately ejected from the network. This avoids situations where replications could exhaust network resources in some areas without injection queue intervention (therefore without their flow control mechanism).

The main problem of this multicast scheme is the length of the path covered to visit every destination node. In order to limit path length, a network partitioning strategy based on Hamiltonian paths is usually employed [33][42]. A Hamiltonian path traverses every node in a graph exactly once, which means that the last copy of a broadcast message will cover a distance equal to the number of network nodes. As network delay is usually critical for performance, there are multiple optimizations for multicast routing based in Hamiltonian paths. Dual-path, Multipath [33] and Column-path [42] algorithms are three examples of improved latency mechanisms. The dual-path algorithm simply constructs two Hamiltonian paths with the same shape but moving messages in opposite directions. This way, multicast destinations can be divided into two groups, generating two message copies traversing a shorter distance to reach every destination. The multipath mechanism relies on the same principle, dividing the network in this case into four disjoint subsets. Finally, the column-path algorithm divides the set of destinations into *2k* subsets (*k* being the number of mesh columns), such that there are at most two messages directed to each column [12][33]. In these cases each destination list is divided into disjoint sub-lists. One "sub-multicast" message is then created for each sub-list and sent along separate multicast paths. These approaches reduce path length, routing messages more efficiently, but inherit unicast-decoupling problems, such as packet serialization at injection queues and strong topology dependency.

C. *On-Chip Multicast Proposals*

Despite the huge number of off-chip multicast proposals, the multi-destination issue has rarely been considered in on-chip interconnection networks in general or in CMPs in particular. In most cases, [17][18][19][23][30][38][39], it has been assumed that multi-destination communications can be implemented efficiently by simply dividing them into unicast packets. This assumption has mainly been motivated by resource scarcity and large bandwidth availability in this context. The strict conditions enforced by CMP environments impose serious limitations for the direct adoption of some off-chip solutions for multicast support. Mechanisms based on the utilization of large centralized buffers [49] or high radix switches [9] are not suitable under CMP area and power conditions. However, ignoring multicast issues does not make the multi-destination message problem disappear. As shown in [25], multicast traffic has a serious impact on CMP system performance, making it desirable to include hardware mechanisms to deal with this issue. This has encouraged some authors to search for efficient multicast support solutions, providing some recent proposals for on-chip environments that will be introduced in this section.

In [25] a multicast support scheme named Virtual Circuit Tree Multicasting (*VCTM*) was presented. *VCTM* is based on the generation of an exclusive virtual circuit for each new multi-destination set. A set-up phase and small tree-identification tables at each network router are required for circuit generation. In this way, multicast messages with the same destination set only need to carry information about virtual circuit identity to be routed properly. The main goal of this work is the minimization of header size for multicast messages. Multi-destination encoding is a difficult issue, and normally requires a large number of bits per message to be coded [15]. The main problem of this solution is that its performance is highly dependent on destination set variability. If the frequency of equivalent destination sets is reduced, the overhead introduced by setup phases can seriously harm performance. As tree tables are made up of Content-Addressable Memories [41], solving this problem through tables with a larger number of inputs would have a negative impact on area and power constraints.

Another mechanism to deal with routing encoding was presented in [47]. In this case, the authors make use of a logic-based routing implementation able to eliminate the need of routing tables for both unicast and multicast messages. This mechanism relies on the definition of network regions through a set of connectivity bits. Multicast destination sets are implemented as network regions where a tree-based broadcast operation is performed. The main benefit of this method is the drastic reduction of both area requirements and power consumption. However, network regions are statically defined at start time and cannot be dynamically modified, reducing the method's utility for the general case.

III. Adaptive-Tree Multicast Routing

Both path-based and tree-based multicast schemes provide a better solution than the unicast approach (dividing a multicast message into several unicast ones in the Coherence Controller), but their different characteristics have singular effects on network performance that must be analyzed. An example illustrating the effect of each solution is shown in Figure 1. Here, we depict the average latency evolution of the two multicast schemes and the unicast approach as the throughput applied to the network increases from 0.1 to 1 flit/cycle/router. The network is made up of 16 nodes, matched to a 4-ary 2-cube topology. Traffic distribution follows a uniform pattern (random) and 10% of communications are broadcast messages, addressed to every network router.

As can be seen, the tree-based approach obtains the best latency results under low load conditions. The distance overhead caused by multicast messages in path-based schemes has a negative effect on base latency, but is still a better solution than unicast decoupling. As network traffic increases, latency differences between the two schemes reduce, reaching a point where the tree-based network reaches its saturation point while the path-based network is still able to deal with higher throughput values. As path-based mechanisms only perform replication when a message is consumed, resource utilization remains under control for higher throughput values, obtaining better results than tree-based approaches.

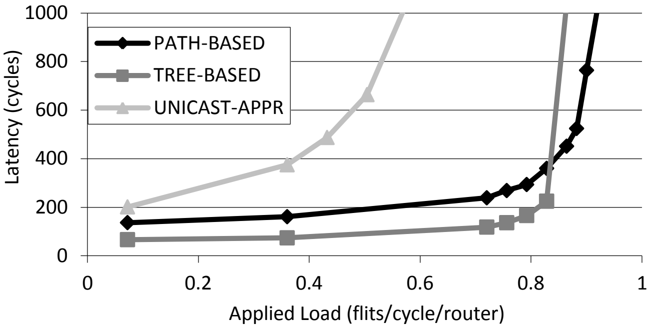


Figure 1. Average packet latency for different multicast solutions.

Depending on traffic demands, each multicast scheme has an optimal range of operation. If a multicast mechanism could somehow be linked with network pressure, we could extract the maximum benefit from network performance, switching from tree to path schemes and vice versa according to throughput values. This is the main contribution of the multicast scheme proposed, which is able to dynamically adapt multicast tree shape to network conditions. Messages will follow minimal distance tree routes under low load conditions, but the multi-destination route shape will gradually evolve to path-based routes, delaying the appearance of network congestion. In other words, we gradually increase tree length in order to reduce network replications and therefore congestion caused by additional messages in the network. To the best of our knowledge, this is the first time a hybrid multicast scheme has been proposed.

A. Underlying Router Structure: The Rotary Router

The Rotary Router [1] is an unconventional router structure in which the input and output ports are interconnected through two buffered rings (see Figure 2). Each buffered ring reaches every output port in the two opposite directions. Packets arriving at any input port (Input Stage) have to request access to one of the router rings. Once this access is granted, packets will enter the selected ring and start moving through the ring's buffers (Buffering Segment Stages) looking for a suitable output port (Output Stage). If the requested output port is granted, the message leaves the router ring and advances to the next router. On the contrary, if access to the output port is denied or it is not profitable, the message keeps on advancing through the router ring, looking for another suitable output port. If none of the profitable outputs for the message is granted, it will complete a whole lap inside the router ring and subsequently start a second lap. After a large enough predefined number of complete laps of a router ring, a packet is marked as miss-routable and can leave the router through the first available output port. This special organization was conceived to provide the network with desirable features such as the absence of centralized structures (arbiters or crossbar), Head-of-Line blocking [28] avoidance, absence of Virtual-channel requirements for correctness guarantee, fully adaptive routing and Virtual Cut-through flow control [29].

In the Rotary Router, network correctness relies on different resource reservation policies, which guarantee the permanent existence of enough free resources inside the network to make messages advance towards their destination. The deadlock avoidance mechanism, either inside the router or between routers, makes use of the Bubble Flow Control method [45], applying different restriction limits for injection and in-transit ports. A new packet will only be allowed to enter in a router ring (Buffering Segment Stage) through an in-transit Input Stage if there is room for at least two packets. If the new packet is trying to enter from the injection Input Stage, the bubble limit changes from two to at least three packets. The two-packet injection limit guarantees the continuous movement of messages inside router rings. The three-packet limit at injection ports forces the appearance of an additional resource (lifesaver hole) that can only be employed by in-transit messages, guaranteeing the advance of messages through the network. A detailed description of these mechanisms can be found in [1].

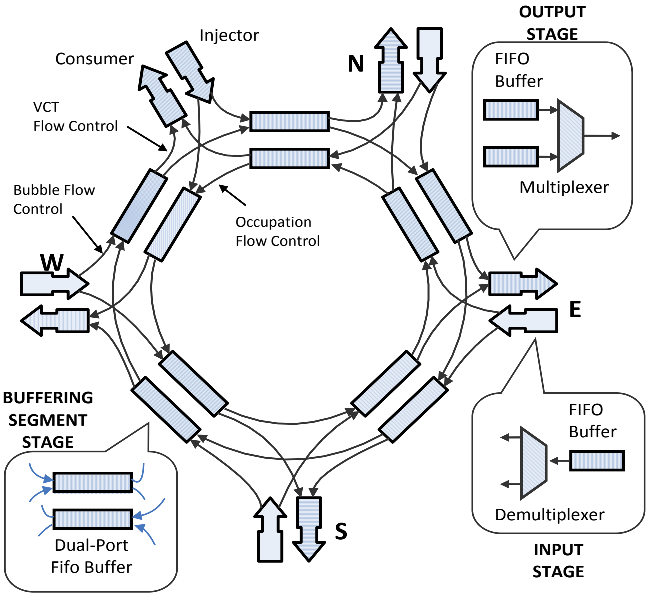


Figure 2. Rotary Router sketch.

Another significant aspect of the Rotary Router is the method of dealing with the end-to-end deadlock problem [2]. The reactive nature of the messages generated at each coherence controller establishes a dependency relation between them, known as the message-dependency chain [48]. The combination of this dependency and the finite nature of injection/consumption queues at network interfaces can cause the appearance of this anomaly. Any router proposed to work as part of a general purpose CMP supporting via-hardware coherence maintenance must provide a suitable solution. The usual solution for this problem in CMPs is to route each network-traffic class through different virtual or physical networks [22]. In both cases, the hardware overhead required to implement end-to-end deadlock avoidance is clear, requiring the replication of part or all the datapath hardware components. Moreover, virtual-channel based solutions also require increased complexity for control logic, artificially implementing priority-based arbitration policies.

The Rotary Router's internal rings provide the perfect substrate to overcome the hardware overhead required to perform message overtaking. The amount of buffering used by each message class is limited in accordance with the priority of the traffic. As their priority increases, messages will be allowed to occupy a larger portion of buffering resources. For example, for a Request-Reply protocol, low-priority messages (Requests) will only be allowed to access router rings when less than 50% of ring buffering resources are in use, but reply messages can make full use of buffering resources (excluding those reserved to guarantee that the network is routing-deadlock free) and therefore they can overtake the low-priority packet class.

This set of mechanisms providing network correctness (both routing and end-to-end deadlock) makes the Rotary Router an advantageous structure for multi-destination routing purposes. On the one hand, deadlock avoidance is not based on path restrictions, providing a high flexibility to route multicast messages through the network without requiring exclusive resources. On the other hand, messages circulate inside router rings in order to be arbitrated at each output port, which greatly facilitates the replication process. On-router replication can be performed simply allowing the packet to circulate until all replicas have been created at each specified output port. These special features, only present in the Rotary Router, have helped us to find a low-overhead solution that can deal with multi-destination traffic in a very efficient way.

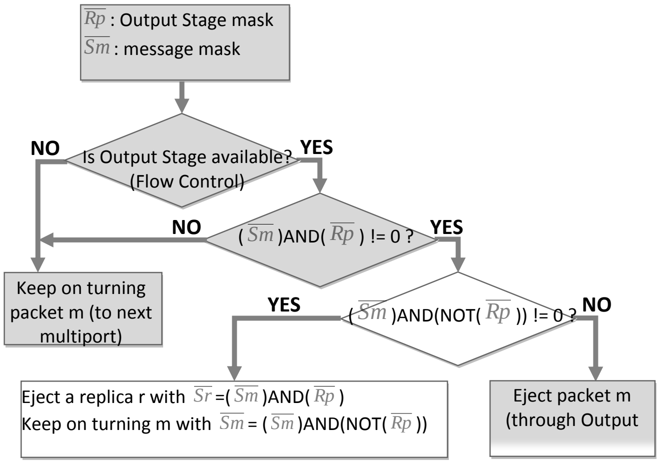

$\overline{Rp}$: Output Stage mask
$\overline{Sm}$: message mask
NO
Is Output Stage available? (Flow Control)
YES
NO
($\overline{Sm}$)AND($\overline{Rp}$) != 0 ?
YES
Keep on turning packet m (to next multiport)
YES
($\overline{Sm}$)AND(NOT($\overline{Rp}$)) != 0 ?
NO
Eject a replica r with $\overline{Sr}$ =($\overline{Sm}$)AND($\overline{Rp}$)
Keep on turning m with $\overline{Sm}$ = ($\overline{Sm}$)AND(NOT($\overline{Rp}$))
Eject packet m (through Output

Figure 3. Routing algorithm for unicast and multicast messages.

B. Multicast Mechanism Proposal

In order to manage unicast and multicast messages in a unified way, table-based routing seems to be the most suitable approach to physically implement the routing algorithm. At each router, a number of independent registers equal to the number of router output ports is needed. Each register is associated with one output port and consists of a bit-vector of length *N*, where *N* represents the number of network nodes. The value in each vector position will indicate if a specific node destination is reachable through that port at minimal distance. As arbitration is an independent process in each Output Stage, register distribution avoids possible contention in the routing process, eliminating simultaneous accesses to a unified routing table.

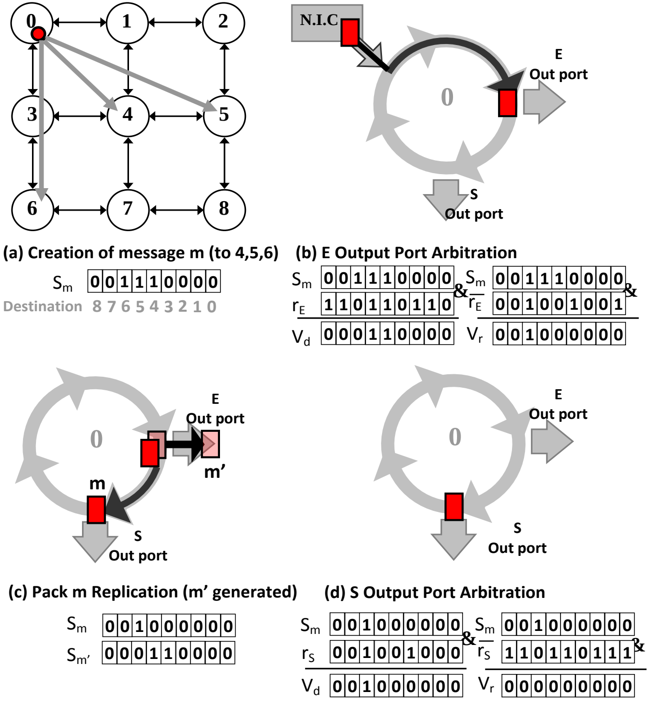
The remaining information needed to perform routing will be carried by the message header flit. Here we must reserve room for a bit vector of the same length as each register (equal to the number of network nodes *N*). In this case, each vector position indicates whether the message must be routed to a certain destination or not. Table distribution forces us to change the routing process to multiport arbitration logic. Every time a new message reaches the head of a Buffering Segment Stage, the arbitration process will check Output Stage availability in parallel with route computation. Making use of the header destination mask and the Output Stage routing mask, route calculation is reduced to a simple 1-gate operation. The result of an AND operation of both masks will tell us if any message destination is at minimal distance through that output port. Each 1-bit value at position *X* of the resulting vector indicates both that the position *X* is reachable through that output port and that this position is a message destination. The decision to make a request to that Output Stage will be based on the resulting vector value. If this vector has a non-zero value, a request is sent to the Output Stage. Otherwise, the message is forced to continue turning inside the router ring, and the request is forwarded directly to the next Buffering Segment Stage.

In multi-destination messages, a situation could occur where some of the destinations are reached through the arbitrated Output Stage, but in order to reach the rest of destinations the message should also keep on circulating inside the router ring. For this reason, messages with more than one destination must perform a second mask operation in order to decide whether to replicate or not. An AND mask operation indicates if any destination is reachable, but multicast messages also need to know if all destinations are reachable or only a sub-set can be reached. The replication decision will be made after performing a new AND operation on the negated port mask and the message mask. A non-zero value in the resulting vector indicates that only some message destinations are reachable through the port, starting the replication process. After the replication process, one of the messages will make use of the output port while the remaining copy will move to the next Buffering Segment Stage. Both messages must update their header mask values. The new mask value of the message leaving the router will be the result of the original AND operation, while the mask of the turning message will be the result of the AND operation with the negated port mask.

The whole routing process for both unicast and multicast messages is based on simple bit operations. Therefore, the routing process can be implemented with a simple and efficient algorithm, as shown in Figure 3. Port masks have enough flexibility to employ any routing strategy, ranging from a conventional deterministic policy such as Dimension Order Routing (DOR) to a fully adaptive scheme. Any routing policy could be built using the appropriate port masks. In this particular case, as the deadlock avoidance mechanism supports fully adaptive routing for the Rotary Router, this will be the policy implemented in port masks.

As an example, Figure 4 walks through the process of destination encoding and multicast routing, assuming port availability (near-zero applied load). In Figure 4.a router *#0* generates a multicast message with destination routers *#4*, *#5* and *#6*. After selecting a router ring the packet advances to the first output port (*E* Output port), where the arbitration process begins.

Figure 4.b shows the operations performed with message and port masks in order to obtain the vectors needed for arbitration. *Vd* indicates that at least one destination of the message is at minimal distance through that output port. *Vr* also has a non-zero vector value, which means that not every message destination can be reached through this port (node #6). Arbitration vectors indicate that the multicast message must start a replication process. The message requests access to both the Output Stage and the next Buffering Segment Stage. If both accesses are granted, the message is copied simultaneously to both positions, updating each header in a different way. If access to the Output Stage is denied due to the Flow Control policy, the message will keep on turning searching for another available output port.



(a) Creation of message m (to 4,5,6)

(b) E Output Port Arbitration

(c) Pack m Replication (m' generated)

(d) S Output Port Arbitration

Figure 4. Example of replication process and header updating.

Figure 4.c shows the resulting message headers after the replication process. As can be seen, the destination vector of the message at the Output Stage inherits only those destination values reachable through that port (nodes #4 and 5#), while the message that keeps on turning inside the router ring resets these values in its header vector, leaving only the pending destinations active (node #6). The message remaining at the router ring will advance until reaching the next output port (*S* Output port), where a new arbitration process takes place. Figure 4.d shows that the same operations are performed for arbitration, with a different result. In this case *Vd* again has a non-zero value, indicating that some destinations can be reached through port *S*. However, vector *Vr* has a zero value at every position, which means that every pending destination for this message corresponds to the current output port. In this case the message detects that no replication is needed, performing only a request to the Output Stage. Once this request is granted, the message will leave the router through this port without performing any replication. As can be seen, with this simple mechanism, under unloaded situations, the network is able to generate a minimal distance tree for each multicast message, reducing message latency by minimizing path distance and improving link utilization by sharing common links of the routes to all of their destinations.

C. Correctness: Deadlock Avoidance

The Rotary Router deadlock avoidance mechanism provides a sufficient degree of freedom to avoid path restrictions for both unicast and multicast communications. Nevertheless, the multicast replication algorithm cannot be directly applied because uncontrolled replication could lead to permanent blocking situations. Both routing and end-to-end deadlock avoidance mechanisms in the Rotary Router are based on occupation control, forbidding a packet to enter the network if it exhausts the buffering resources (Bubble Flow Control method [45]).

Both deadlock avoidance mechanisms implement their functionality at injection ports, deciding whether new messages are allowed to leave the Local Host or must wait before advancing. Under these conditions, multicast message replication could jeopardize correctness, because every time a replica message is generated within the network, the occupation level is increased without control from injection ports. For this reason, replicated messages could consume buffering resources reserved for deadlock avoidance.

To maintain the network deadlock free we re-size the Output Stage buffering to be able to hold at least two packets and the replication will only take place if after this process in the Output Stage there will still be room for a packet. In this way we guarantee that after consuming one of the two holes with a copy message, the remaining hole can still act as a lifesaver hole for advancing packets. Implementing this replication control in every multiport buffer (excluding those connected to a consumption port where it is not necessary), we guarantee that the replication process carried out by multicast messages does not interfere with the routing-deadlock avoidance mechanism.

The replication process will also be aware of occupation limits for the different classes of packets in order to avoid end-to-end deadlock. If the generation of a copy message implies reaching the limit for its packet-class, no new messages will be generated. A more formal description of the mechanisms employed for both routing and end-to-end deadlock avoidance can be found in the supplementary material of [4].

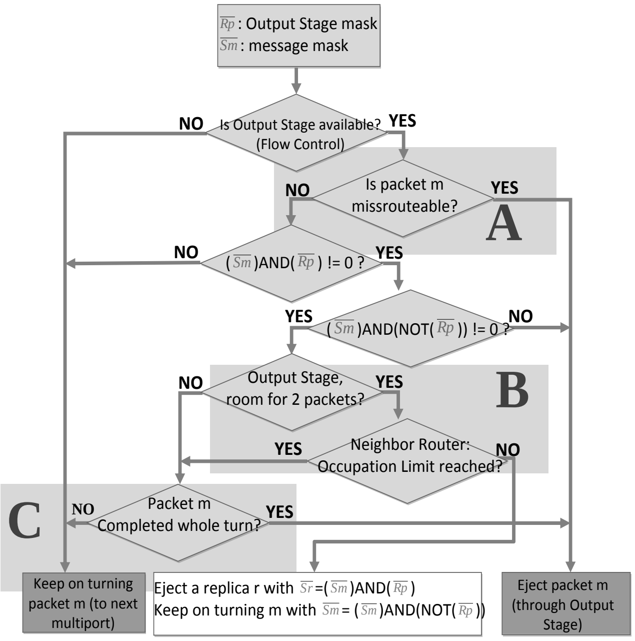

$\overline{Rp}$: Output Stage mask
$\overline{Sm}$: message mask
NO
Is Output Stage available? (Flow Control)
YES
NO
Is packet m missrouteable?
YES
A
NO
($\overline{Sm}$)AND($\overline{Rp}$) != 0 ?
YES
YES
($\overline{Sm}$)AND(NOT($\overline{Rp}$)) != 0 ?
NO
NO
Output Stage, room for 2 packets?
YES
B
YES
Neighbor Router: Occupation Limit reached?
NO
C
NO
Packet m Completed whole turn?
YES
Keep on turning packet m (to next multiport)
Eject a replica r with $\overline{Sr}$ =($\overline{Sm}$)AND($\overline{Rp}$)
Keep on turning m with $\overline{Sm}$ = ($\overline{Sm}$)AND(NOT($\overline{Rp}$))
Eject packet m (through Output Stage)

Figure 5. Routing algorithm with replication, deadlock-free version.

Every time the replication process must be interrupted by either of the two deadlock avoidance mechanisms, the routing decision will be taken according to the number of internal ring multiports traversed by the message. If the message is still completing the first turn, it will be forced to keep on turning inside the ring. In this way we ensure that every message will traverse the consumption port before leaving the router, performing message replication if it is being held at one of the vector destinations. After performing a complete lap, the message will be allowed to make use of the output port, but behaving like a unicast message. This means that the message can move to another router without replicating, maintaining the same destination mask in the header flit. Figure 5 shows the final algorithm governing both unicast and multicast messages, this time being aware of deadlock restrictions. Condition box A implements misrouting when requested by the deadlock avoidance mechanism. Box B contains the new conditions imposed in replication in order to avoid consuming the lifesaver hole. Finally, the box C statement describes the action taken when no replication is possible.

D. *Extending Operation Range: Adaptive Tree Multicast*

Output Stage buffering occupancy provides a good insight into router contention. A stalled message at an Output Stage indicates that the neighboring router is having problems advancing new messages coming from that direction. Raising the pressure on that network area by generating a new message would increase contention, adding one additional contender to highly requested resources. Due to deadlock avoidance, if the Output Stage occupation indicates a high level of congestion, multicast messages will try to make a first attempt to circumvent that port, moving through that link without being replicated when no more options are available. Network correctness also generates the behavior that links network pressure and multicast mechanism selection, performing implicit congestion control. In other words, the shape of the multicast tree will be self-adapted to the network utilization status. Under low load conditions, reduced Output Stage utilization allows multi-destination messages to follow a wide-tree path for packet destinations. An increase in network pressure will cause the appearance of congested areas, where messages remain stalled at Output Stages. In this case, multicast messages will be forced to follow path shapes with longer branches, trying to avoid increasing the pressure on congested areas. Finally, corner-case situations could lead to messages following a route with one single branch of length *M* (*M* being the number of multicast destinations). This implicit congestion control will allow us to benefit from the advantages of both multicast schemes simultaneously.

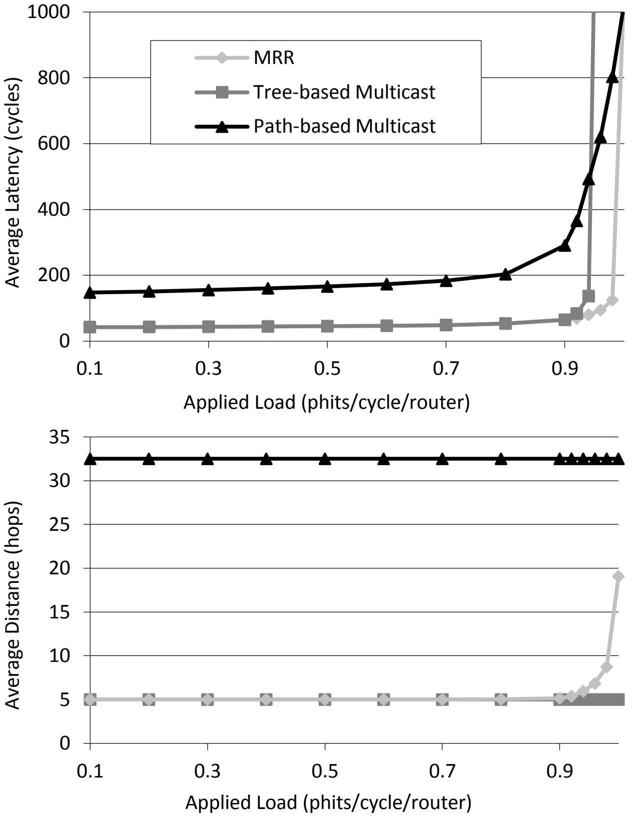


Figure 6. Traffic with Multicast Rotary Router(MRR) and different multicast mechanisms. (up) Average latency. (down) Average distance.

To provide a glimpse of the mechanism, we have performed a simple experiment analyzing tree depth evolution when replication control is applied. We also explored depth evolution's effect on performance. As reference values, path-based and tree-based mechanisms will be employed. In contrast to Adaptive-tree multicasting, these techniques have fixed values for tree depth. This means that the average distance (number of routers traversed) in multi-destination messages is constant for messages with equivalent destination masks, independently of the load applied to the network. Despite being distance-fixed, network latency still depends on contention values, dynamically evolving with the pressure exerted on the network. Figure 6 shows the results obtained for both distance and latency parameters. These numbers have been obtained for an 8×8 torus with only broadcast traffic. This is a corner-case configuration of multicast communications, different from real scenarios where multicast and unicast messages make use of network resources simultaneously. This configuration is employed to evaluate the effects of the adaptive-tree mechanism in a better way, isolating the effect of this mechanism on network performance.

The bottom of Figure 6 shows how deadlock avoidance increases tree depth. At low or medium loads the average distance of the Multicast Rotary Router (MRR) is close to a tree-based solution, because the absence of congestion allows messages to follow minimal routes. Consequently, similar latency values are observed for both cases, avoiding the initial delay penalty of longer paths. Notwithstanding, when the network gets closer to saturation, multicast messages start traversing congested areas. Circumventing these routers eliminating replication increases the average distance required to deliver multi-destination messages. However, as can be observed in the top of Figure 6, this increase reduces the effect of replications on network congestion, and consequently better maximum sustainable throughput is obtained. In this way, the adaptive-tree distribution of the MRR is able to obtain the best results for the latency-throughput network curve.

IV. Performance Evaluation

The simulation framework employed will allow us to perform experiments ranging from synthetic traffic patterns to exhaustive full-system simulation with complex workloads. This infrastructure is composed of four connected simulation tools. The full-system simulator Simics [34] has been extended with the GEMS timing infrastructure [37]. GEMS provides detailed models of both the memory system and a state-of-the-art processor. In order to achieve more detailed contention modeling for the interconnection network, the original network simulator of GEMS has been replaced with SICOSYS [46]. This simulator allows us to take into account most of the hardware implementation details with much higher precision. Finally, in order to perform energy estimations, SICOSYS has been connected to the Orion2.0 power simulator [27].

In the presence of unicast traffic patterns the Rotary Router has been proven to perform better than input-buffered routers [1]. Therefore, this evaluation must also clarify which part of performance improvement is achieved by the router structure itself and which part is caused by adaptive multicasting. To do so, the full-system evaluation will include results corresponding to structures without hardware multicast support, providing a clear decomposition of performance enhancements achieved by the different multicast schemes analyzed.

A. *Network Configuration (counterparts)*

In order to contrast the effectiveness of *MRR* versus other multicast proposals, we have compared ours to two conventional deterministic input-buffered routers with *idealized* multicast support. In an attempt to mimic the usual configuration for on-chip proposals [6][9][17][21][25][30], both counterpart routers use wormhole flow control and implement deterministic routing. Moreover, separate virtual networks are employed to implement end-to-end deadlock avoidance. The first router, denoted BASE-MC, represents a minimal cost implementation. It includes scarce buffering capacity and uses a classical 5-stage pipeline [43]. Buffering per virtual channel is fixed to the minimal amount bounded by round-trip delay (2 flits for 1-cycle links). This router represents the minimal cost design. The second counterpart will be denoted ADV-MC. It will have generous buffering and pipeline optimizations. The pipeline of the router is optimized at 3 cycles, performing virtual channel allocation and switch allocation in the same cycle. A shared buffer similar to [30] is employed per input port. Although the buffer is able to store up to 5 flits per virtual channel, the capacity is dynamically partitioned according to traffic demands. ADV-MC represents a different design point with a similar implementation cost to the Rotary Router and it will allow us to compare raw performance directly. In order to support different numbers of message types while avoiding end-to-end deadlock, we use separate virtual channels for each message type in the two counterparts. As we will make use of a six-message-class protocol for the full system evaluation, to avoid both routing and end-to-end deadlock we need at least twelve virtual channels per physical port in the case of a torus network, because both wormhole routers make use of Dally's deadlock avoidance mechanism [16]. Multicast support in these two routers will be implemented through a solution similar to the one presented in [25]. For the sake of simplicity we will assume that both counterparts provide sufficiently large Destination Set CAMs to hold an unlimited number of multicast trees, eliminating the setup phase required to construct each multicast tree.

Router structures employing the unicast approach (replicating multicast messages at injection queues) will not be included in the first part of the evaluation. This comparison will be provided for full-system evaluation. In this way we will be able to clarify the overall effect on system performance of providing multicast support.

Table 1. Network configuration Parameters

Topology	4×4 Torus	8×8 Torus
Mcast Length	4 & 16 destinations	8 & 32 destinations
Mcast percentage	5%, 10%, 25%, 50%	5%, 25%, 50%
Message Types	6 (mcast messages belonging to types 1, 3 and 5)	
Message Size	1 flit (odd types) and 5 flits (even types)	
Traffic Pattern	Random, Bit Reversal, P. Shuffle, T. Matrix, Tornado	
Cyc. Simulated	200,000 (20,000 warm-up cycles discarded)	

In order to isolate the effect of the multicasting mechanisms, two additional design decisions have been taken. First, pipeline optimizations such as look-ahead signals [21][30] will not be supported. The implementation of base-latency optimizations is in most cases orthogonal to the evaluation of hardware multicast schemes and could hide the real effects of each solution evaluated. Second, a similar buffering capacity has been assumed for both *ADV-MC* (300flits) and *MRR* (325flits), while *BASE-MC* (120flits) limits its capacity to the minimal storage required to ensure correctness. Thus, we will be able to evaluate proposals with similar requirements as well as performance effects due to resource scarcity.

B. *Synthetic Traffic Evaluation*

Making use of different network sizes and traffic patterns we will evaluate each router's performance under different multicast scenarios. Different percentages of multicast traffic with a variable number of destinations will be evaluated. The multicast fraction is calculated as the portion of consumed messages originally belonging to a multicast packet. This means that for a 25% multicast fraction 1 out of every 4 messages consumed is a multicast replication.

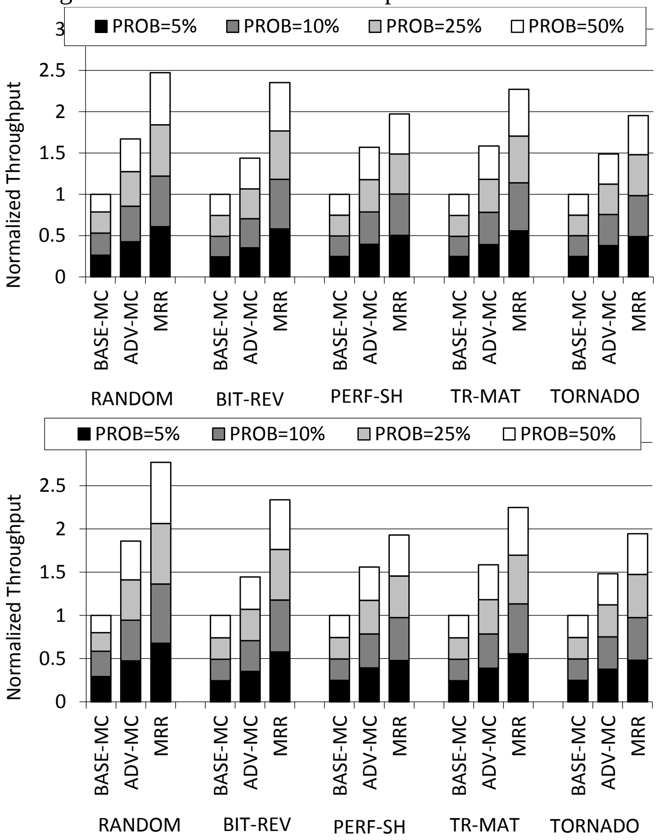


Figure 7. (above) 4-destination (below) broadcast, 4x4 torus, BASE-MC normalized throughput at maximum applied load.

As we will make use of the TokenB [36]coherence protocol for the full-system evaluation, traffic will be composed of six different message types, every router being enhanced to eliminate end-to-end deadlock. Each message type is numbered according to its position in the dependency chain. In order to emulate the reactive nature of coherence protocol traffic, the destination of an odd-type message is chosen according to the traffic pattern selected, whereas even-type messages are sent back to the sources of the originating message. Finally, message size follows a bimodal distribution, where messages belonging to odd types will have a size of 1 single flit, while the other types will make use of 5-flit messages. Table 1 summarizes all the configuration parameters previously described.

The first results of this evaluation are shown in Figure 7. This plot corresponds to maximum sustained throughput values for 4×4 torus network. These results have been obtained through the application of a constant load of 1 flit per cycle, and have been normalized for the *BASE-MC* case. Two different multicast lengths have been employed, covering multicast and broadcast scenarios. The top of Figure 7 represents the multicast scenario, with a set of 4 randomly generated destinations. In contrast, the results in the bottom of Figure 7 have been obtained for a one-to-all multicast configuration. In both cases the fraction of multicast messages has been modified from 5 to 50%. As can be seen, *MRR* is able to outperform both counterparts for each configuration analyzed. Comparing *MRR* results with those from the *BASE-MC* configuration we can observe that our implementation is able to double baseline results for worst-case patterns such as Perfect Shuffle or Tornado, obtaining nearly three times more throughput for a uniform traffic distribution with broadcast destination-sets. Changing this comparison to one where both implementations have a similar complexity (ADV-MC), we observe that these differences decrease, but we can still find significant improvements for traffic patterns such as random or bit reversal.

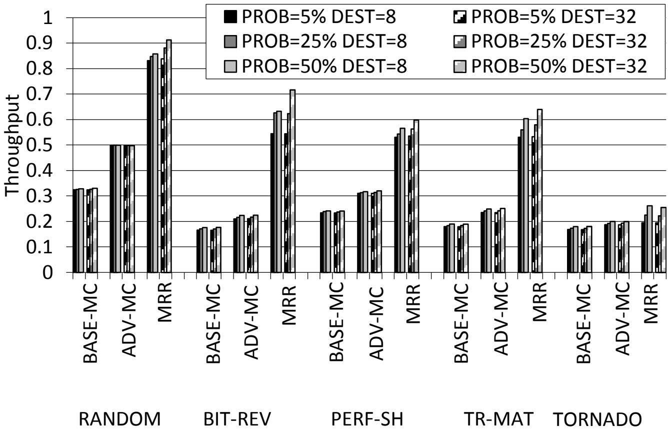


Figure 8. An 8×8 torus throughput at maximum applied load.

A similar experiment was performed for a different network size, showing 8×8 torus results in Figure 8. In this case throughput values have not been normalized. Destination-set lengths have been fixed to 8 and 32 nodes, represented by solid and striped columns respectively. The multicast percentage of total network traffic varies from 5% to 50% values, as in the previous case. The tendency observed on 4×4 torus networks is maintained for larger sizes. Again, *MRR* is able to outperform both counterparts for any of the configurations analyzed, being nearly double the *ADV-MC* performance for traffic patterns such as random or bit-reversal and obtaining even better results for bit-reversal distributions.

Another important conclusion can be extracted from these results. As can be clearly seen in the case of Bit-reversal traffic, *MRR* is able to extract substantial performance benefits from an increased fraction of multicast messages. This benefit is much less significant for input-buffered structures, where this increment shows no improvement at all for traffic patterns such as random. The unrestricted utilization of network links performed by *MRR* allows multicast messages to circumvent those network areas where unicast communications monopolize interconnect resources.

Finally, an 8×8 torus configuration has also been analyzed through the generation of a "pulse" of messages injected at maximum rate, determining the time required to consume all of them. Figure 9 shows the BASE-MC normalized time required to drain a pulse of 300,000 messages for each multicast configuration and traffic pattern. As can be seen, the Multicast Rotary Router is able to obtain more than a 50% time saving for most non-uniform traffic patterns.

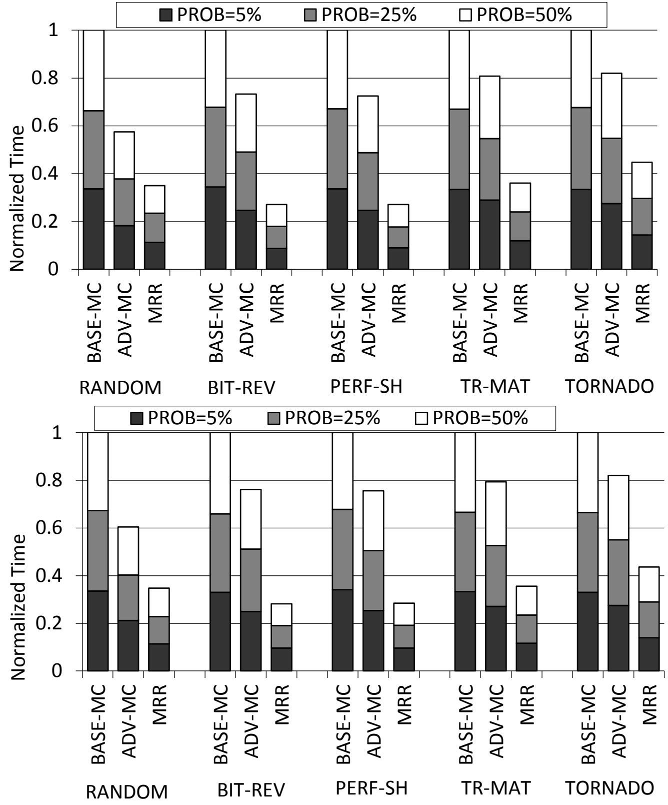


Figure 9. 8x8 Torus, BASE-MC normalized time to consume a 300,000 message pulse, (above) 8-destination multicast, (below) 32-destination multicast.

C. Full-System Evaluation

The simulated system in this section will be a 16-processor CMP with shared S-NUCA L2 based on [10]. The main parameters of the simulated configuration can be seen in Table 2. The workloads selected cover different utilization scenarios, such as desktop, numerical or server-based applications. The multithreaded numerical applications are part of the NAS Parallel Benchmarks (OpenMP implementation version 3.2 [26]) and the PARSEC software [11]. Transactional applications correspond to the Wisconsin Commercial Workload suite [8]. Finally, desktop workloads use part of the SPEC CPU2006 suite [50] running in rate mode (multi-programmed). For each application a variable number of runs is performed with pseudo-random perturbation in order to estimate workload variability [8]. All the results provided have a 95% confidence interval.

Token coherence protocol [36] is extremely multicast sensitive, being a perfect test bench for the network configurations proposed. Each time an L1 miss occurs, a multicast message is generated and sent to the rest of the L1 caches and to an L2 bank. These actions involve an important amount of network traffic, multicast hardware support being critical for this protocol. Persistent request activations and deactivations require the generation of multicast and point-to-point ordered communications. As they represent a small fraction of network traffic, we have chosen to decompose those transactions in unicast messages for the Rotary Router. Thus we are able to avoid the inclusion of special routing masks for ordered messages, which are not allowed to perform adaptive routing.

TABLE 2. MAIN SYSTEM PARAMETERS.

PROCESSOR		MEMORY HIERARCHY	
Number of Cores	16	L1 I/D Cache	32KB, 4-way, 1 cycle
Frequency	2GHz	L2 Cache	16MB SNUCA, 16 Banks
Window Size	64	L2 Cache Bank	1MB, 16-way, 5-cyc, pseudo LRU
Outstanding Requests	16	Main Memory	4GB, 250 cycles, 320GB/s
Issue Width	4	Block Size	64 Bytes
		Command/ Data Size	16/80 Bytes (command/data)

Finally, in this section we will also provide energy-related results. Energy consumption of network components has been obtained through the Orion2.0 tool [27]. Table 3 shows the energy consumption of the main router events for the three counterparts. Orion2.0 energy values obtained for control logic are at least two orders of magnitude lower than the values obtained for datapath components. Only those actions consuming a significant amount of energy, included in Table 3, will be considered in the evaluation. Network results were obtained for a 2GHz operating frequency, 1V operating voltage and 45nm technology. The switch model selected was Matrix-Crossbar, while buffers were modeled as SRAM blocks. Additionally, the energy consumed by the different on-chip cache levels will also be measured, making use this time of the CACTI6.5 software [40]. Assuming the same technology and operating frequency employed in Orion2.0 and the L1 and L2 bank sizes specified in Table 2, each L1 bank access requires 530.54pJ and each L2 bank access 2408pJ.

TABLE 3. ENERGY REQUIREMENTS ESTIMATED WITH ORION 2.0.

BASE-MC		ADV-MC		MRR	
	E(pJ/flit)		E(pJ/flit)		E(pJ/flit)
Buff Write	1.03	Buff Write	3.18	Input Stage	6.51
Buff Read	6.21	Buff Read	11.8	Output Stage	7.08
SW Trav.	14.93	SW Trav	14.93	Buff S. Stage	11.83
Link	18.16	Link	18.16	Link	18.16

The first results provided in the top of Figure 10 correspond to the energy consumed by each network configuration in order to complete each application execution. As expected from the energy values obtained with Orion2.0, the network configured with BASE-MC routers is the one with the lowest energy values. In this case, scarce buffering reduces the energy required to read and write flits from/to buffers, reducing the energy required by network components. The network configuration denoted as ADV-MC implements larger and more sophisticated buffers, requiring more energy for buffer operations. Finally, each time a message traverses an MRR router multiple Buffer Stage traversals are performed (approximately 2.3-2.5 for the applications evaluated), making this router the most energy consuming of the three counterparts analyzed.

On the other hand, performance results for the three router micro-architectures are shown in the bottom of Figure 10. As can be observed, in this case we obtain opposite results from those obtained for Energy, and *MRR* outperforms all the counterparts consistently. Comparing the three implementations, *MRR* is 20% faster than *BASE-MC* on average, and 10% faster than *ADV-MC*. These results, when compared with those using synthetic traffic, suggest that most of the applications maintain the system under medium-to-low load. However, with bandwidth-demanding applications, such as *FT* or *IS*, the throughput benefits of *MRR* are clear. For these two applications *MRR* is able to nearly halve the time required for application execution, demonstrating the important performance degradation caused by resource scarcity.

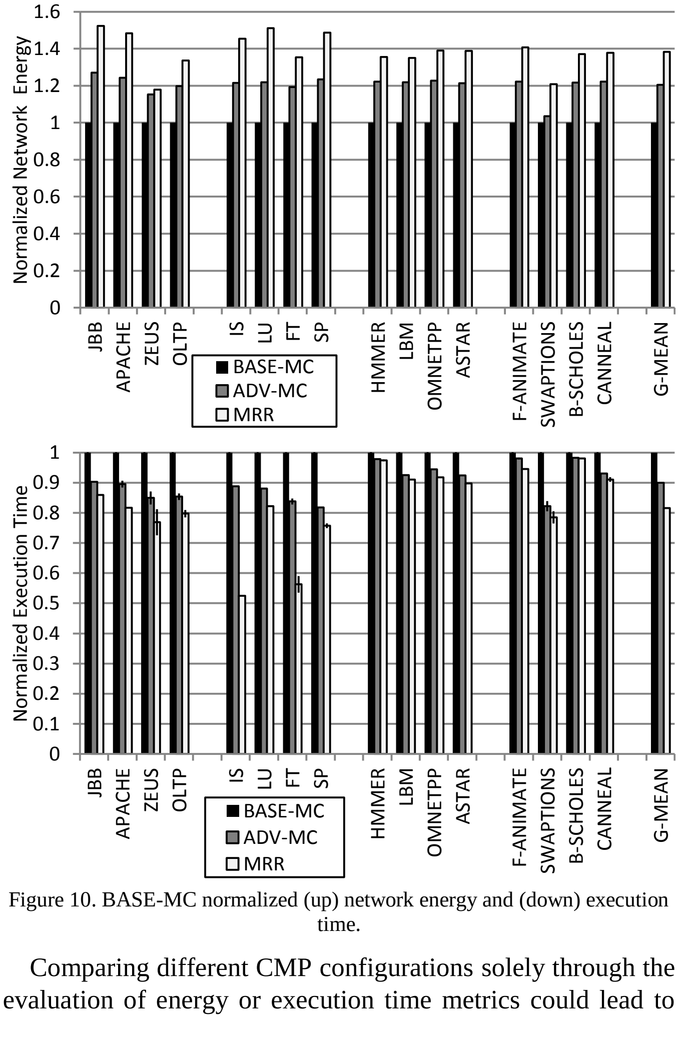


Figure 10. BASE-MC normalized (up) network energy and (down) execution time.

Comparing different CMP configurations solely through the evaluation of energy or execution time metrics could lead to misleading conclusions [20]. We need a metric able to consider both quantities simultaneously, exposing which configuration is able to provide minimum power at a given performance or more performance for the same power. The way to obtain these results is by taking the product of energy and delay or EDP [20]. Additionally, another mistake arises when comparing full-system metrics (such as execution time) with partial metrics (network energy). In order to perform a fair comparison, both metrics should provide results for the same group of system components. Including L1 and L2 cache events in our energy evaluation will enable us to provide more meaningful results than those obtained for network components only.

Figure 11 shows the Network+Cache Energy Delay Product. The contribution of network and cache components to total EDP has been separated, in order to better understand the results provided. The lower part of each bar, in solid color, represents L1 and L2 caches, while the upper part with degraded color corresponds to network components. On average, the EDP reduction achieved by the Multicast Rotary Router is clear. The significant performance benefit compensates the extra dynamic energy consumed by continuous packet movements between Buffering Segment Stages. MRR is able to reduce EDP by nearly 20% on average when compared to a baseline implementation and by 20% compared to a state-of-the-art router with similar implementation cost. It should be noticed that core energy consumption or static energy consumed by leakage currents has not been taken into account. As static energy is proportional to execution time and processor energy could be considered constant independently of network configuration, their inclusion in EDP results would increase the differences between our proposal and the rest of the counterparts, improving the results obtained even more.

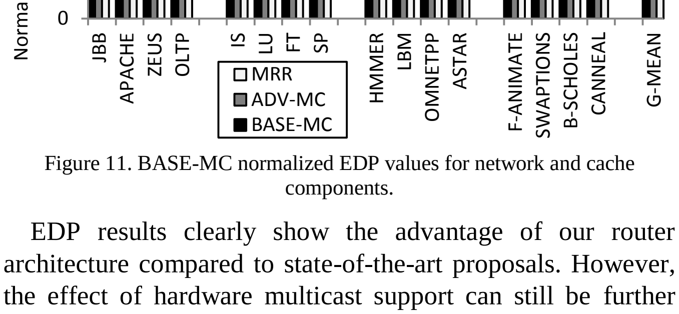


Figure 11. BASE-MC normalized EDP values for network and cache components.

EDP results clearly show the advantage of our router architecture compared to state-of-the-art proposals. However, the effect of hardware multicast support can still be further clarified. To do so, results obtained for both *MRR* and *ADV-MC* configurations have been compared with their implementations without multicast support in order to show the improvement margin obtained merely by adding this feature. Thus, four different router configurations will be employed for this last experiment; two implementations without multicast support (ADV-UC and ROTARY) and two implementations with hardware-based multicast support (ADV-MC and MRR). Routers without multicast capabilities must decompose multi-destination messages into multiple messages with a single destination. This fact multiplies the amount of times flits are written and read from buffers, and also the number of switch and link traversals performed. This overhead imposes a severe penalty which can be quantified.

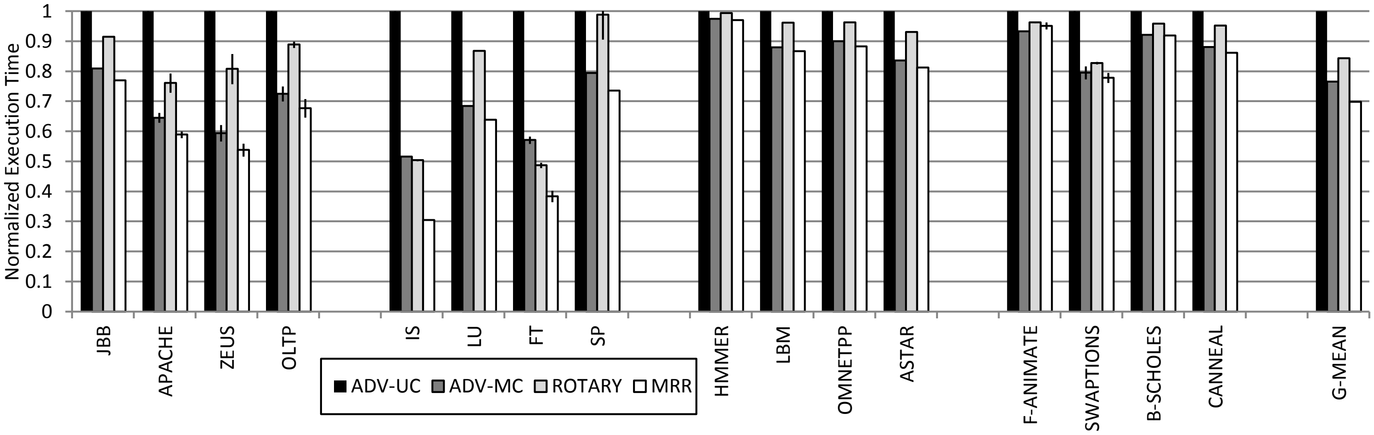


Figure 12. Performance benefits of multicast support, ADV-UC normalized values.

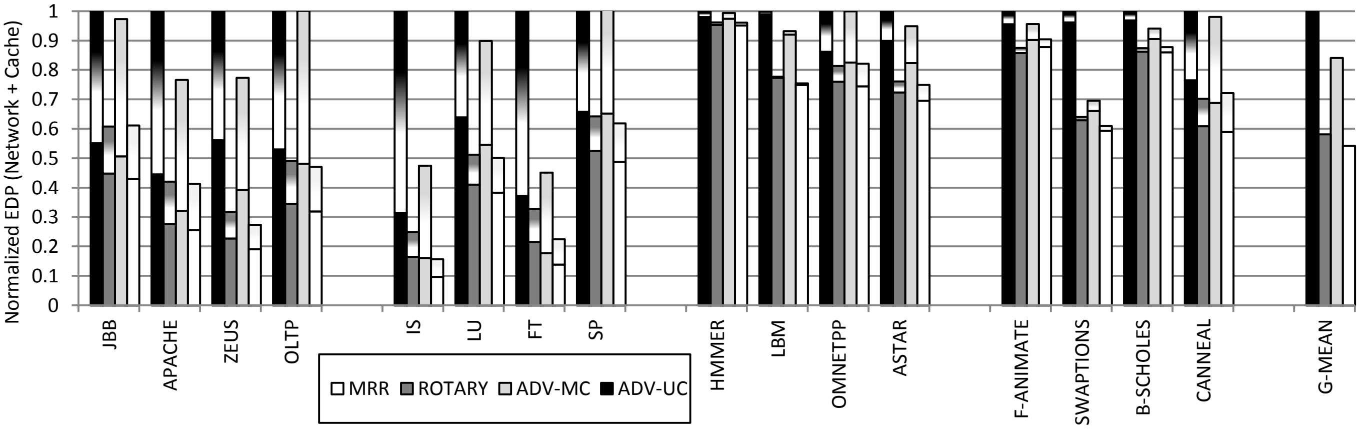


Figure 13. EDP Benefits of multicast support, ADV-UC normalized values.

As expected, Figure 12(above) shows the clear performance benefits of multicast support. The more efficient bandwidth utilization makes *ADV-MC* more than 20% faster on average than its base implementation. A similar result can be observed for the *MRR*, in this case improving execution times by nearly 20% on average. Finally, the joint benefits of multicast hardware support in terms of energy and performance provide the EDP results presented in Figure 13(below). On average, the count of network events is reduced by three times when moving from a configuration without multicast support to a router with it. As expected, this fact has an important effect on EDP results. As can be seen, multicast support translates into 50% EDP saving when compared to the baseline router implementation.

V. Conclusions

The unique properties of the Rotary Router have enabled us to propose a multicast mechanism especially targeted to CMP architectures. Mask-based routing allows us to provide an improved router version with almost negligible hardware overhead, making use of a unified mechanism for unicast and multicast routing issues, avoiding control logic replication and reducing the routing process to simple 1-gate operations. This solution has also achieved better performance results than the other current proposals analyzed.

Adaptive Multicasting is the main contribution of this paper. Linking network pressure to the multicast mechanism has demonstrated to be an extremely efficient solution from the performance point of view. The dynamic evolution from tree-based to path-based multicast as the throughput applied to the network grows can delay the appearance of network congestion, achieving better overall performance results than applying static solutions. Performance metrics have been obtained for a wide range of traffic types, loads, numbers of destinations and multicast traffic fractions and the proposal obtains the best performance results in most cases.

Finally, full-system evaluation has clearly shown the importance of multicast support for a CMP memory hierarchy. Both for the Rotary Router and Input-buffered structures, the mere inclusion of this feature can reduce application execution time by more than 25%. However, those applications with high traffic demands have shown that network congestion can be as important as multicast support. For this reason, the additional benefits of combining multicast support with a high-performance router micro-architecture have enabled us to obtain the best overall performance results.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful and valuable comments. Additionally, the authors also want to thank Jose Angel Herrero, the group´s system manager, for his continuous support and enormous effort in maintaining the computing infrastructure required to provide the results in this document.

This work has been supported by the Spanish Ministry of Science and Innovation, under contracts TIN2010-18159, and by the HiPEAC2 European Network of Excellence.

REFERENCES

[1] P. Abad, V. Puente, P. Prieto, J.A. Gregorio, “Rotary Router: An Efficient Architecture for CMP Interconnection Networks”, International Symposium on Computer Architecture (ISCA), pages 116-125, June 2007.

[2] P. Abad, V. Puente, J.A. Gregorio, “Reducing the Interconnection Network Cost of Chip Multiprocessors”, IEEE International Symposium on Networks-on-chip (NOCS), pages 183-192, February 2008.

[3] P. Abad, V. Puente, J.A. Gregorio, “MRR: Enabling Fully Adaptive Multicast Routing for CMP Interconnection Networks”, High-Performance Computer Architecture (HPCA), pages 355-366, February 2009.

[4] P.Abad, V. Puente, J.A. Gregorio, “Balancing Performance and Cost in CMP Interconnection Networks”, Accepted for publication at IEEE Transactions on Parallel and Distributed Systems, June 2010.

[5] N.R. Adiga, et al., “Blue Gene/L torus interconnection network”, IBM Journal of Research and Development, vol 49, no. 2, pages 265-276, March 2005.

[6] N. Agarwal, L.S. Peh, N.K. Jha, “In-Network Snoop Ordering (INSO): Snoopy Coherence on Unordered Networks”, International Conference on High-Performance Computer Architecture (HPCA), pages 67-78, February 2009.

[7] N. Agarwal, L.S. Peh, N.K. Jha, “In-Network Coherence Filtering: Snoopy Coherence without Broadcasts” International Symposium on Microarchitecture (MICRO), pages 232-243, December 2009.

[8] A.R. Alameldeen, M.K. Martin, C.J. Mauer, K.E. Moore, M. Xu, D. J. Sorin, M.D. Hill, D.A. Wood, “Simulating a $2M Commercial Server on a $2K PC”, IEEE Computer, pages 50-57, February 2003.

[9] J. Balfour, W. J. Dally, “Design Tradeoffs for Tiled CMP On-chip Networks”, International Conference on Supercomputing (ICS), pages 187-198, 2006.

[10] B. Beckmann, D. Wood, “Managing Wire Delay in Large Chip-Multiprocessor Caches”, International Symposium on Microarchitecture (MICRO), pages 319-330, December 2004.

[11] C. Bienia, “Benchmarking Modern Multiprocessors”, Ph.D. Thesis, Princeton University, January 2011.

[12] R. Boppana, S. Chalasani, C. Raghavendra, “On Multicast Wormhole Routing in Multicomputer Networks”, Symposium on Parallel and Distributed Processing, pages 722-729, 1994.

[13] D.M. Brooks, et al, “Power-Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors”, IEEE Micro, Vol. 20, Issue 6, pages 26-44, November 2000.

[14] G. Byrd, N. Saraiya, B. Delagi, “Multicast Communication in Multiprocessor Systems”, International Conference on Parallel Processing (ICPP), pages 196-200, August 1989.

[15] C. Chiang, L.M. Ni, “Multi-address Encoding for Multicast”, International Workshop on Parallel Computer Routing and Communication, pages 146-160, 1994.

[16] W. Dally, C.L. Seitz, “Deadlock-Free Message Routing in Multiprocessor Interconnection Networks”, IEEE Transactions on Computers, pages 547-553, May 1987.

[17] R. Das, O. Mtulu, T. Moscibroda, C.R. Das, “Aérgia: Exploiting Packet Latency Slack in On-Chip Networks”, International Symposium on Computer Architecture (ISCA), pages 106-116, June 2010.

[18] N. Eisley, L.S. Peh, L. Shang, “In-Network Cache Coherence”, International Symposium on Microarchitecture (MICRO), pages 321-332, December 2006.

[19] C. Gomez, M. Gomez, P. Lopez, J. Duato, “BPS: A Bufferless Switching Technique for NoCs”, Workshop on Interconnection Network Architectures” pages 1-6, 2008.

[20] R. Gonzalez, M. Horowitz, “Energy Dissipation in General Purpose Microprocessors”, IEEE Journal of Solid-State Circuits, Vol. 31, No. 9, pages 1277-1284, September 1996.

[21] P. Gratz, C. Kim, R. McDonald, S.W. Keckler, D. Burger, “Implementation and Evaluation of On-Chip Network Architectures”, International Conference on Computer Design (ICCD), pages 477-484, October 2006.

[22] A. Hansson, K. Goossens, A. Radulescu, “Avoiding Message-Dependent Deadlock in Network-Based Systems on Chip”, VLSI design, 2007.

[23] M. Hayenga, N. E. Jerger, M. Lipasti, “SCARAB: A Single Cycle Adaptive Routing and Bufferless Network”, International Symposium on Microarchitecture (MICRO), pages 244-254, December 2009.

[24] Intel Corporation, “An Introduction to the Intel Quickpath Interconnect”, White paper, Document Number 320412-001US, 2009.

[25] N.E. Jerger, L.S. Peh, M.H. Lipasti, “Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support”, International Symposium on Computer Architecture (ISCA), pages 229-240, June 2008.

[26] H. Jin, M. Frumkin, J. Yan, “The OpenMP Implementation of NAS Parallel Benchmarks and its Performance”, NAS Technical Report, October 1999.

[27] A. Kahng, B. Li, L.S. Peh, K. Samadi, “ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration”, Design Automation and Test in Europe (DATE), April 2009.

[28] M.J. Karol, M.G. Hluchyj, S.P. Morgan, “Input versus Output queuing on a space-division packet switch”, IEEE Transactions on Communication, vol. 35, no. 12, pages 1347-1356, December 1987.

[29] P. Kermani, L. Kleinrock, “Virtual Cut-Through: A New Computer Communication Switching Technique”, Computer Networks, vol.3, pages 267-286, September 1979.

[30] A. Kumar, P. Kundu, A.P. Singh, L.S. Peh, N.K. Jha, “A 4.6 Tbits/s 3.6GHz Single-cycle NoC Router with a Novel Switch Allocator in 65nm CMOS” International Conference on Computer Design (ICCD), pages 63-70, October 2007.

[31] D.R. Kumar, W.A. Najjar, P.K. Srimani, “A New Adaptive Hardware Tree-Based Multicast Routing in K-Ary N-Cubes”, IEEE Transactions on Computers, Vol. 50, no. 7, pages 647-659, July 2001.

[32] J. Laudon, D. Lenoski, “The SGI Origin: A cc-NUMA Highly Scalable Server”, International Symposium on Computer Architecture (ISCA), pages 241-251, June 1997.

[33] X. Lin, L.M. Ni, “Deadlock-Free Multicast Wormhole Routing in Multicomputer Networks”, International Symposium on Computer Architecture (ISCA), pages 116-125, 1991.

[34] P.S. Magnusson, M. Christensson, J. Eskilson, D. Forsgen, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, B. Werner, “Simics: A Full System Simulation Platform”, IEEE Computer, Vol. 35, no. 2, pages 50-58, February 2002.

[35] M. Malumbres, J. Duato, J. Torrellas, “An Efficient Implementation of Tree-Based Multicast Routing for Distributed Shared-Memory Multiprocessors”, IEEE Symposium on Parallel and Distributed Processing, pages 186-189, October 1996.

[36] M.M.K. Martin, M.D. Hill, D.A. Wood, “Token Coherence: Decoupling Performance and Correctness”, International Symposium on Computer Architecture (ISCA), pages 182-193, June 2003.

[37] M.M.K. Martin, D.J. Sorin, B.M. Beckmann, M.R. Marty, M. Xu, A.R. Alameldeen, K.E. Moore, M.D. Hill, D.A. Wood, “Multifacet´s General Execution-driven Multiprocessor Simulator (GEMS) Toolset”, Computer Architecture News (CAN), pages 92-99, September 2005.

[38] T. Moscibroda, O. Mutlu, “A Case for Bufferless Routing in On-Chip Networks”, International Symposium on Computer Architecture (ISCA), pages 196-207, June 2009.

[39] R. Mullins, A. West, S. Moore, “Low-Latency Virtual-Channel Routers for On-Chip Networks”, International Symposium on Computer Architecture”, pages 188-197, June 2004.

[40] N. Muralimanohar, R. Balasubramonian, N. Jouppi, “Optimizing NUCA Organizations and Wiring Alternatives for Large Caches with CACTI 6.0”, International Symposium on Microarchitecture (MICRO), pages 3-14, 2007.

[41] K. Pagiamtzis, A. Sheikholeslami, “Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey”, IEEE Journal of Solid-State Circuits, vol. 41, no. 3, pages 712-727, March 2006.

[42] D.K. Panda, S. Singhal, P. Prabhakaran, “Multidestination Message Passing Mechanism Conforming to Base Wormhole Routing Scheme”, Parallel Routing and Communication Workshop, May 1994.

[43] L.S. Peh, W.J. Dally, “A Delay Model and Speculative Architecture for Pipelined Routers”, International Symposium on High Performance Computer Architecture (HPCA), pages 255-266, January 2001.

[44] F. Petrini, J.Duato, P. Lopez, J.M. Martinez, “LIFE: A Limited Injection, Fully AdaptivE, Recovery-Based Routing Algorithm”, International Conference on High Performance Computing (HIPC), pages 316-321, 1997.

[45] V. Puente, R. Beivide, J.A. Gregorio, J.M. Prellezo, J. Duato, C. Izu, “Adaptive Bubble Router: A Design to Improve Performance in Torus Networks”, International Conference on Parallel Processing (ICPP), pages 58-67, 1999.

[46] V. Puente, J.A. Gregorio, R. Beivide, “SICOSYS: An Integrated Framework for studying Interconnection Network in Multiprocessor Systems”, IEEE Euromicro Workshop on Parallel and Distributed Processing, pages 15-22, January 2002.

[47] S. Rodrigo, J. Flich, J. Duato, M. Hummel, “Efficient Unicast and Multicast Support for CMPs”, International Symposium on Microarchitecture (MICRO), pages 364-375, November 2008.

[48] Y.H. Song, T.M. Pinkston, “A Progressive Approach to Handling Message-Dependent Deadlock in Parallel Computer Systems”, IEEE Transactions on Parallel and Distributed Systems, Vol. 14, no. 3, 2003.

[49] C.B. Stunkel, J. Herring, B. Abali, R. Sivaram, “A New Switch Chip for IBM RS/6000 SP Systems”, ACM/IEEE Conference on Supercomputing, November 1999.

[50] The Standard Performance Evaluation Corporation. SpecCPU2006, http://www.spec.org/cpu2006

Pablo Abad was born in Reinosa, Cantabria. He received his BS, MS and PhD degree from the University of Cantabria, Spain, in 2003 and 2010 respectively. He currently works as assistant professor of Computer Architecture. His research interests are focused on on-chip interconnection network design, as well as their interaction with the rest of system components as part of CMP memory hierarchy.

Valentin Puente was born in Vendejo, Cantabria. He received the BS, MS and PhD degree from University of Cantabria, Spain, in 1995 and 2000 respectively. He is currently an Associate Professor of Computer Architecture at the same University. His research interests include interconnection networks, multithreaded architectures, and performance evaluation. He is a member of the IEEE Computer society.

Lucia G. Menezo was born in Santander, Cantabria. She received her BS and MS from the University of Basque Country, Spain, in 2007. She is currently working on her PhD degree with a 4-year-scholarship from the University of Cantabria, Spain. Her research is focused on designing cache coherence protocols for enhancing CMP performance.

José Angel Gregorio was born in Bareyo, Cantabria (Spain). He received his BS, MS and PhD in Physics (Electronics) from the University of Cantabria, in 1978 and 1983, respectively. He is currently a professor of computer architecture in the Department of Electronics and Computers in the same University. His research interests include parallel and distributed computers, interconnection networks, and performance evaluation of computers and communication systems. He is a member of the IEEE Computer society.