LIGERO: A Light but Efficient Router Conceived for Cache Coherent Chip Multi Processors

PABLO ABAD, University of Cantabria
VALENTIN PUENTE, University of Cantabria
JOSE-ANGEL GREGORIO, University of Cantabria

Although abstraction is the best approach to deal with computing system complexity, sometimes implementation details should be considered. Considering on-chip interconnection networks in particular, underestimating the underlying system specificity could have non negligible impact on performance, cost or correctness. This paper presents a very efficient router that has been devised to deal with cache coherent chip multiprocessor particularities in a balanced way. Employing the same principles of packet rotation structures as in the Rotary Router, we present a router configuration with the following novel features: (1) reduced buffering requirements, (2) optimized pipeline under contention-less conditions, (3) more efficient deadlock avoidance mechanism and (4) optimized in-order delivery guarantee. Putting it all together, our proposal provides a set of features that no other router, to the best of our knowledge, has achieved previously. These are: (1') low implementation cost, (2') low pass-through latency under low load, (3') improved resource utilization through adaptive routing and a buffering scheme free of head-of-line blocking, (4') guarantee of coherence protocol correctness via end-to-end deadlock avoidance and in-order delivery, and (5') improvement of coherence protocol responsiveness through adaptive in-network multicast support. We conduct a thorough evaluation that includes hardware cost estimation and performance evaluation under a wide spectrum of realistic workloads and coherence protocols. Comparing our proposal with VCTM, an optimized state-of-the-art wormhole router, it requires 50% less area, reduces on-chip cache hierarchy energy delay product on average by 20% and improves the cache coherency chip multiprocessor performance under realistic working conditions by up to 20%.

Categories and Subject Descriptors: **C.1.2 [Processor Architectures]**: Multiprocessors, **C.2.1 [Computer-Communication Networks]**: Network Architecture and Design

General Terms: Design, Performance

Additional Key Words and Phrases: Router Microarchitecture, Cache-coherent CMP, Network on Chip

1. INTRODUCTION

On-chip point-to-point interconnection networks are an optimal solution for a wide spectrum of systems. As integration density grew over time, it became evident that routing packets is less complex and more energy-efficient than routing wires [Dally and Towles 2001]. As a consequence of this logical step, nowadays the utilization of on-chip networks is pervasive, being used from system-on-chip [Coppola et al. 2004] to high-performance processors [Park et al. 2010]. Although the basic elements of the network are similar in any environment, the systems where this paradigm is used often impose dissimilar requirements. This diversity can be manifested in terms of different constraints for programmability, energy consumption or implementation cost. For example, while for a high-performance general purpose processor, programmability is paramount, for SoCs, energy and implementation costs are the

This work has been supported by the MICCIN (Spain) under contract TIN2010- 18159 and the HiPEAC European Network of Excellence.

Authors' address: P. Abad, V. Puente, J.A. Gregorio, Electronics and Computers Department, University of Cantabria, Spain; email: {abadp, vpuente, monaster}@unican.es.

DOI*****

main issues. Similar reasoning could be applied to other specific systems such as GPUs, MPSoCs, etc. Therefore, it seems reasonable to consider the utilization scenario in the interconnection network design process. Extending the previous example to network features, while for a SoC, predictable network delay could be very relevant, high throughput support might be critical in the case of a high-performance general processor.

This work focuses on the proposal of a cost-efficient router micro-architecture for general-purpose chip multiprocessors (cc-CMP). In this environment the commonly accepted consensus is that shared memory is the most productive programming paradigm [Asanovic et al. 2006]. Additionally, if we take into account that complex on-chip cache hierarchies are inevitable, a cache coherence infrastructure is necessary in order to overcome the off-chip bandwidth wall. To maintain system correctness, coherence invariants for multiple copies of data blocks must be guaranteed. Although hardware support for this model is not essential, most commercial products are based on it. There is much more information available from the hardware perspective and it is faster and easier to preserve system correctness without impairing programmer productivity. Therefore, cc-CMP seems to present the most viable way to translate the transistor availability provided by Moore's law into performance, while maintaining system programmability.

These main characteristics of cc-CMPs impose a set of requirements that should (must, in some cases) be satisfied by the on-chip interconnection network. These requirements can be classified as correctness-oriented and performance-oriented ones. It should be noted that correctness requirements must be achieved at system level, not only at network level. Moreover, a network could obstruct system correctness if some required feature is not provided. For example, if we prevent deadlock, starvation and livelock, we can consider the network correct or anomaly-free [Duato 1995]. However, we cannot affirm the same for the whole system. If message-dependent deadlock is ignored in network design, system correctness will be put at risk [Song and Pinkston 2003]. Similar observations can be made for performance figures, where a key example is support for on-network multicast, which could have a large impact on system performance [Jerger et al. 2008]. In both cases, incorporating those features after network design could substantially increase the interconnection network complexity or render it unusable in practice.

Although most of the ideas presented in this paper could be applicable in other environments, such as interconnection networks for message passing systems or system on a chip, we focus our interest in cache coherent CMPs because they require all of them to be integrated in a single design. Additionally, a specific feature for the interconnection network in this environment is the wide range of usage scenarios due to the unforeseeable character of network traffic from the programmer's perspective. In contrast, in a message-passing scenario, the programmer of a cc-CMP is barely aware of the kind of traffic the application is generating. For example, subtle changes in OS memory mapping could induce large cache interference among threads at last level cache, generating great pressure on the network. This usage unpredictability makes it desirable to extend network range, i.e. optimal performance under low-latency and high-bandwidth conditions.

This paper advocates synergistically combining cc-CMP requirements with network design parameters from the very beginning to achieve a router microarchitecture, called LIGERO (LIGhtweight but Efficient ROuter), able to deal with all the aforementioned issues. LIGERO provides both correctness and performance requirements imposed in cc-CMP systems for a fraction of the cost of conventional routers. Even though it is a simple router, it includes a significant number of features in order to achieve high throughput at a limited cost. In

particular, the router is HoL blocking free, uses adaptive routing, has low pass-through latency under low-load conditions, supports on-network adaptive multicast and includes mechanisms for deadlock avoidance at both network and coherence protocol levels. Moreover, all these features are achieved while requiring less area and energy consumption than commonly used, state-of-the-art, deterministic wormhole routers. LIGERO reduces the latency-throughput-cost trade-off in the network, providing proficiency in terms of both simplicity and contention reduction. All these assertions have been demonstrated through an exhaustive evaluation process, comparing LIGERO with different state-of-the-art network configurations.

The rest of the paper is structured as follows: Section 2 stipulates the network design constraints in cc-CMP systems and describes how multiple state-of-the-art routers are affected by these constraints; Section 3 introduces the new router micro-architecture and operation details. Section 4 describes the evaluation framework, Section 5 analyzes the performance of the proposal and finally, Section 6 states the main conclusions of the paper.

2. MOTIVATION

2.1 Coherency Protocol and Interconnection Network Relationship

In cache coherent systems, split-transactions are necessary in order to achieve minimal performance requirements. The more optimized the coherence protocol is, the more intermediate states are required to reach the stable state [Vantrease et al. 2011]. Many intermediate state transitions require actions which involve the transmission of messages. Therefore, in a memory transaction a chain of messages, each one with a different purpose or nature, could be involved. For example, in a simple request-reply protocol, such as the one employed in Dash [Lenoski et al. 1992], two classes of messages can be identified: request commands and data responses. In more advanced protocols, such as Quick Path Interconnect [Intel 2009], there are six types. Even in the simplest case, this characteristic has to be considered during the interconnection network design process in order to guarantee system correctness. In a cc-CMP system the routers are connected to coherence or memory controllers. Those coherence controllers have a limited capacity to store pending memory transaction. If this buffering is exhausted, there is no way to drain additional messages from the network and deadlock could occur. To exemplify this, let's assume the simplest reactive traffic composed of request and reply, focusing our attention on a part of the system shown in Figure 1. At some point, we could reach a situation where all transit buffers and *A* and *C* consumption queues (located at coherency controllers) are swamped with request messages. To process the head message at each consumption queue we need to send a reply message. Nevertheless, neither messages *A* nor *C* can be injected because request packets are blocking the buffering resources at *B*. None of the responses can progress toward the transaction initiator, preventing the coherency controller from freeing buffering, precluding the consumption of more requests. In this situation, there is a cyclic dependency between the coherency controller and the network that will indefinitely delay the initial request consumption, which deadlocks the system.

This is known as message-dependent deadlock and it has been widely studied in cache coherent systems since the initial cc-NUMA prototypes [Lenoski et al. 1992]. DASH designers circumvent this issue reworking the network design, with separate physical networks for the requests and replies. In other words, they split the network resources, devoting a part to each class of messages. In other systems, such as the Alpha 21364 [Mukherjee et al. 2002] or Quick Path Interconnect, the classes of messages involved are too large for this approach and the solution is to use virtually separated networks for each class of messages. This approach is cost effective

because it only requires adding more virtual channels to the router. Although other methods such as recovery [Song and Pinkston 2003] could be feasible solutions, false positives, especially with long main memory latency, are hard to avoid. Consequently, such solutions negatively impact system stability. In general, an avoidance mechanism implemented within the interconnection network is preferable. This solution is used extensively not only in cc-CMP but also in application-specific systems where application traffic is reactive, including peer-to-peer streaming or slave locking [Murali et al. 2006; Hansson et al. 2007; Stok 2005].

Data race resolution in state-of-the-art coherence protocols usually requires some sort of packet ordering inside the interconnection network [Strauss et al. 2007; Agarwal et al. 2009; Raghavan et al. 2008; Martin et al. 2003]. For example, a deterministic path is sometimes desired to support in-order delivery, i.e. two packets sent with the same pair source-destination have to be consumed in the same order as injection. When adaptive routing is used in the network this requirement is not fulfilled, which can prevent some optimization or necessary transactions in the coherence protocol. For example, the router of Alpha 21364 uses adaptive routing in all traffic classes except one, where packets follow a deterministic path.

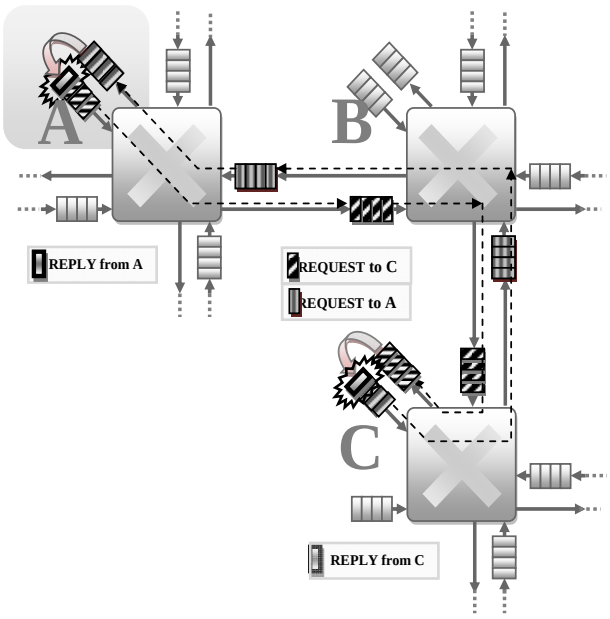

Figure 1. Message-Dependent Deadlock.

Any router proposal for cc-CMP must devote the necessary resources to deal with both message-dependent deadlock and in-order delivery. While ordered transactions are guaranteed with minimal overhead for conventional input-buffered routers, end-to-end deadlock can impose a prohibitive overhead in many state-of-the-art router proposals for networks on chip.

Finally, many cc-CMP protocols use multicast messaging to improve performance and/or simplify coherence protocol implementation. The large on-chip bandwidth availability makes the use of coherence protocols based on multicast communications attractive [Martin et al. 2003; Keltcher et al. 2003]. For most of these protocols, when a core misses in its private cache, it sends a multicast message that snoops the private caches of other cores in the chip, and respecting coherence invariants, it is possible to accelerate the access latency to shared data. On-chip support for multicast traffic implies not only that the memory transaction will be resolved faster but also

the network resource utilization can be optimized, which has a very relevant impact: the energy overhead of this type of protocols will be substantially reduced and the scalability will be greatly improved [Jerger et al. 2008].

2.2 Interconnection Network Performance and Cost under General Purpose Computing

The main performance metrics of interconnection networks are base latency and maximum sustainable throughput [Duato et al. 1997]. As Figure 2(a) suggests, the underlying fixed contributors to these metrics are the wire length and raw bandwidth limit. Minimal latency is affected by unavoidable factors, such as wire delay, and design-dependent parameters such as network topology, etc. The peak throughput or bandwidth limit depends on implementation choices, such as wire availability per link, link data rate, network bisection size, etc. However, observed latency in real systems is always far from minimal latency, the actual shape of the latency-load curve being strongly dependent on router micro-architecture. When the applied load is increased, at some point the traffic flow saturates and, even when far from peak throughput, the network cannot manage it. When the load increases, the likelihood of collisions among the packets escalates, increasing the average time required for delivering them. The applied load at which this occurs is at the maximum sustainable throughput. The difference between this point and the peak bandwidth limit is mainly affected by contention management policies. This is mostly determined by features included in the routers' micro-architecture, such as route adaptivity, queuing scheme, etc. For these reasons, improving maximum throughput usually requires increasing router complexity which negatively affects base latency.

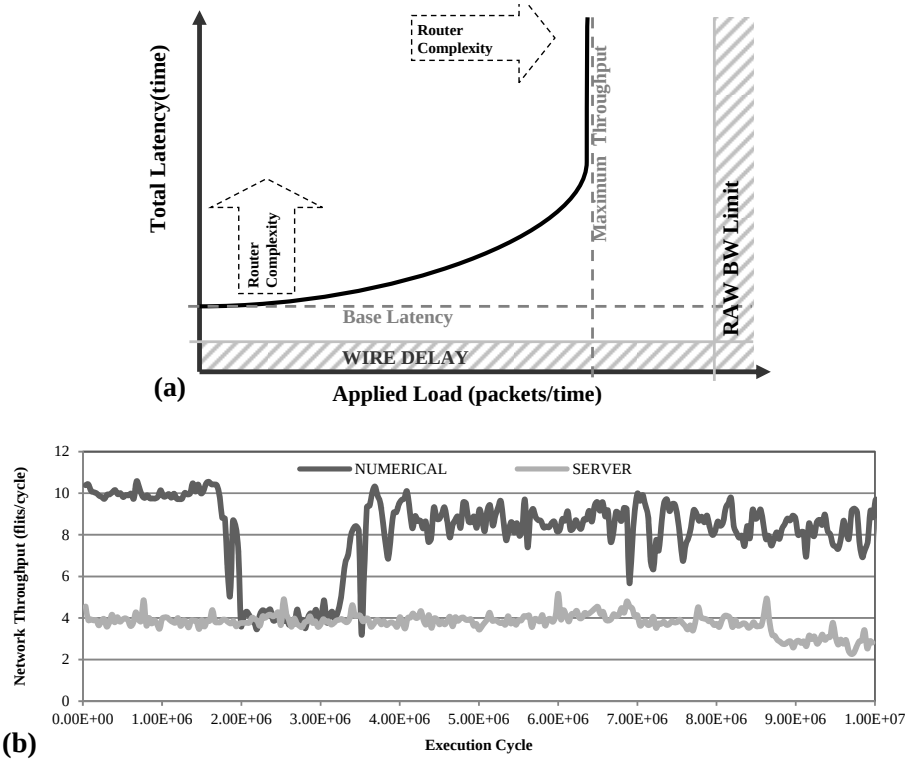

Figure 2. (a) Interconnection Network performance metrics. (b) cc-CMP network load during application execution.

As cc-CMPs are general purpose computing devices and the simultaneous improvement of both base latency and maximum throughput is complex, it is hard to decide which interconnection network metric to optimize in the design process. To

illustrate this, Figure 2(b) shows the applied load of two different applications running in the same cc-CMP, which uses a network with 16 routers. The configuration of the system is shown in Table I. For the SERVER application the offered load is quite low which makes a low base latency preferable. Nevertheless, for NUMERICAL applications, the network load is 120% higher which means that better contention management is recommendable; otherwise the average latency will be large and therefore the execution time of the application will be long.

The solutions employed to increase the maximum throughput achievable by the network usually have an associated overhead in terms of the amount of resources devoted to the router (cost). However, in the same way that correctness must be guaranteed at system level; cost impact of network design decisions must be analyzed in the context of a cc-CMP memory hierarchy. Nowadays it is usual to devote substantial chip resources to implement hierarchies with tens of megabytes. As the number of cores per chip grows, limited off-chip pin-count could jeopardize system scalability and one of the best ways to compensate that gap is to use large on-chip caches [Rogers et al. 2009]. In this environment, the interconnection network represents a minimal amount of the total resources compared to the remaining on-chip memory hierarchy. Therefore, although cost factors such as implementation cost or energy consumption of the network are relevant, their impact should be analyzed in the wider context of the global system.

2.3 State-of-the-art Router Proposals for a cc-CMP environment

In view of the correctness and performance requirements, the design of the interconnection network of this type of system can become challenging due to the difficult balance that must be achieved. In summary, network complexity, network performance and cc-CMP requirements must be balanced. A suitable design should deal with all of them in the best way possible, but a perfect coverage is an extremely hard task. In the literature, there are a vast number of approaches. For this reason, in this section we will focus our analysis on a reduced set with the most significant proposals made over the last few years.

The first three routers, named AERGIA [Das et al. 2010], EVC [Kumar et al. 2007] and WPF [Ma et al. 2012], make use of a set of virtual channels to implement optimized flow control or arbitration mechanisms. In AERGIA arbitration units prioritize packets according to their slack (cycles a packet can be delayed without affecting performance), while WPF makes use of an additional virtual channel per message type to implement adaptive routing, improving previous mechanisms such as [Duato 1995] by allowing packets to move back to adaptive channels after using an escape path. EVC makes use of the virtual channels to bypass intermediate routers. More aggressive solutions such as Elastic Buffers [Michelogiannakis et al. 2009] or MinBD [Fallin et al. 2012] propose buffering minimization or even the utilization of buffer-less routers like CHIPPER [Fallin et al. 2011]. The Elastic Buffer router employs pipelined links to store flits, eliminating input buffers. CHIPPER makes use of injection control policies and deflection routing to guarantee packet advance in a buffer-less network. The MinBD router is a CHIPPER optimization that incorporates a side-buffer at each router in order to reduce the number of deflections. Finally, some proposals such as VCTM [Jerger et al. 2008], FANOUT [Krishna et al. 2011] and the Multicast Rotary Router (MRR) [Abad et al. 2007, Abad et al. 2009] focus on providing support for specific cc-CMP requirements, such as multi-destination messages.

Of the routers described here, AERGIA, WPF and EVC deal with cost and performance elegantly, increasing network performance with minimal complexity impact. However, some important features concerning cc-CMP requirements are not

dealt with. According to section 2.1, each message type requires an exclusive set of virtual channels, making the final number of virtual channels prohibitive (for example, AERGIA's original implementation uses 6 VC, which combined with a coherence protocol with 6 message types requires a total number of 36 virtual channels at each input port). A similar problem is faced by CHIPPER, MinBD and Elastic Buffers. These routers minimize network cost with a minimal impact on performance. However, the absence of buffering or their special behavior prevents the utilization of virtual channels, and a physically independent network is required for each message type to avoid end-to-end deadlock [Michelogiannakis et al. 2009]. Finally, MRR and VCTM address cc-CMP correctness issues efficiently. In VCTM, multicast support does not increase the number of virtual channels required, while MRR implements an end-to-end deadlock avoidance mechanism which does not require virtual channels. However, neither of these two routers provides an optimal balance between performance and cost. MRR addresses performance efficiently at the expense of increased buffering requirements, while VCTM's limited complexity reduces performance. Although it partially solves VCTM limitations through a more elaborated multicast-tree generation and a specialized crossbar avoiding flit serialization, the FANOUT router requires additional virtual channels to avoid deadlock between different trees and still relies on deterministic routing policies for unicast messages. The target of LIGERO is to obtain an efficient balance between complexity and performance, while providing support for all cc-CMP requirements.

3. LIGERO

Most state-of-the-art routers use many concepts introduced by the Torus Routing Chip [Dally and Seitz 1986] 25 years ago. For instance, VCTM adds multicast support to a router microarchitecture that maintains the same routing, flow control, buffering policy and structure as its baseline. The AERGIA or EVC routers also perform a small set of modifications compared to the Torus Routing Chip, building more elaborated control logic able to provide quality-of-service or low-latency router traversals. Even those routers with more significant differences, such as bufferless structures, are still pretty similar when compared to the common baseline microarchitecture. Similarly, LIGERO does not start from scratch but from the original ideas introduced by the Rotary Router (and its multicast version, MRR), a radically new microarchitecture. With a similar philosophy to the one followed by its counterparts, LIGERO performs several modifications to the baseline structure to achieve a much more robust product with reduced complexity, but equivalent in terms of CMP supported features and performance.

LIGERO implements a more efficient deadlock avoidance mechanism based on the utilization of a deterministic escape path to reach destination. While livelock avoidance in the ROTARY router was based on probabilistic assumptions (after being misrouted a large enough number of times, a message will eventually reach its destination), LIGERO guarantees livelock avoidance forcing packets to follow a path traversing every network router under certain conditions. LIGERO implements additional connectivity for bypass router traversals. Every message must make use of ROTARY's internal rings, while this is not necessary for LIGERO, which provides lower latencies under low load conditions. Thanks to a novel deadlock avoidance mechanism and bypass paths, one of the internal rings can be eliminated while maintaining router correctness and minimizing performance impact, thus significantly reducing buffering requirements (with the subsequent area and energy benefits). Finally, a much more optimized in-order delivery mechanism minimizes the hardware required for implementation and allows LIGERO to support larger fractions of ordered traffic with minimal impact.

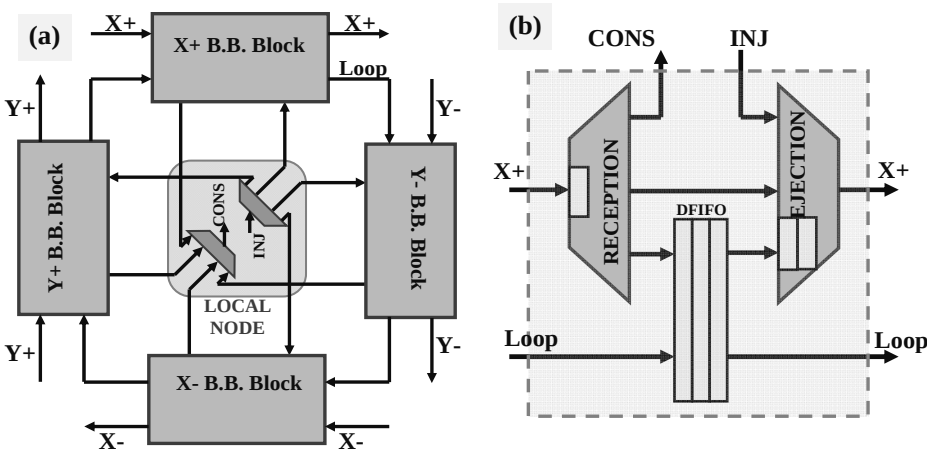

Figure 3. (a) Representation of a LIGERO router for a 2D topology. (b) Basic Building Block (denoted as BBB).

3.1 Description

LIGERO, like the ROTARY, is a fully modular architecture without centralized arbitration or switching fabric, but unlike its baseline router, it has a different structure of each one of its building blocks. Each input-output pair in the same direction and dimension has a similar structure. As can be seen in Figure 3(a), in the case of a bi-dimensional topology, the router is composed of four identical Basic Building Blocks. In this work, we particularize the proposal for well-suited CMP topologies such as the bi-dimensional torus. In Figure 3(b) a sketch of a Basic Building Block is depicted, which is composed of a dual-port FIFO buffer (DFIFO) and reception and ejection stages.

- The ***DFIFO*** is a multiport buffer with two input and output ports. By linking together an input-output pair of the DFIFO in each building block, a loop of buffers is created. For each packet stored, it must be decided whether the packet has to be ejected or must be moved forward through the internal loop to the DFIFO of the next building block. Packets must keep on moving in the loop of the buffers until reaching a profitable and available output port.
- The ***reception stage***, in packet-based multiplexing, has to choose between three different alternative outputs. The first one is consumption (CONS) which is chosen when the *local node* is the packet destination. The second one is *bypass* which is chosen if the following three conditions hold:
    - The packet destination is reachable following the same dimension and direction as the current packet movement.
    - There are no packets at the ejection stage waiting to use the output port (although packets can be at the DFIFO moving inside the loop).
    - The neighbor router has room for at least one packet. Given that the flow control applied in the router is Virtual Cut-through (VCT), the reception stage must be able to store at least one packet.

If any of the previous conditions are not met, the reception stage chooses the third exit, sending the packet to the DFIFO and forcing it to move continuously in the internal loop until reaching a profitable and available output port.

- The ***ejection stage*** regulates the access from the three different paths to the output port. On a round-robin basis this stage chooses which packet can

progress to the output link. As can be seen in Figure 3, there is a direct connection among the injection queue (INJ), the bypass line and the DFIFO queue and therefore this stage has to provide some sort of internal buffering to manage possible collisions. However, to maximize output link utilization and apply further improvements only the DFIFO's incoming path is required to have some buffering capacity. Bypass and injection incoming paths can be managed with only a latch.

To access the host or *local node*, it is necessary to interconnect the injection and consumption paths to each building block. This has been done with a conventional de-multiplexer and multiplexer respectively. For the multiplexer we assume that if the consumption queue is in use, packets have to wait in the reception stage of the incoming port. It is not worth providing any buffering in that multiplexer. Similarly, de-multiplexing the injection queue to all the ejection stages in the building blocks will not require additional buffers in the ejection stage. In Figure 3 both the multiplexer and de-multiplexer for the local node have four inputs and outputs respectively.

Packet rotation inside the router could be configured clockwise, as is shown in Figure 3(a), or counterclockwise by simply rotating the building blocks accordingly. We combine both types of router organization like a chessboard, as is shown in Figure 4(a) for a bi-dimensional torus. In this way, we balance link utilization through the implicit utilization of a zig-zag selection function. In other topologies a similar construction rule could also be applied in order to implement different selection functions. For example, in a mesh, an X/Y selection function is preferable [Dally and Towles 2001], and this can be done reordering the rotation direction of each router.

3.2 Network Perspective

3.2.1 Benefits

LIGERO supports fully adaptive routing, without requiring any data-path alteration. When a packet enters in the internal loop, it can leave the router by the first profitable and available port, which in contrast to deterministic routing, optimizes link utilization. For conventional routers, adaptive routing requires costly data-path replication in order to maintain the network deadlock free [Dally and Towles 2001] or deflection routing [Moscibroda and Mutlu 2009] which could introduce livelock issues.

In contrast to input-buffered routers [Karol et al. 1987], packet rotation structures such as ROTARY and LIGERO are Head-of-Line blocking free because when the profitable output port for a packet is unavailable, it is forced to move on to the next output port, making possible the progression of the following packets. To achieve this in conventional routers, centralized buffers or multi-port output buffering is required, which can be prohibitive in terms of cost.

The router pipeline length adaptively changes according to the network status. Under low-load conditions the packets continuing in the same direction will pass through the LIGERO router in only one cycle, spent in the reception stage. During this cycle the control logic evaluates conditions of bypass and uses the reception-ejection dedicated path if they are all met. Those packets that need to turn must go through additional stages, spent in the router's internal loop. The overhead caused by these extra cycles is minimal, because a maximum of 1 turn is required to reach any destination for the topologies proposed. Using a single latch at the ejection stage, it becomes feasible to travel the wire length to the next router in one additional cycle. Thanks to the absence of a central switching structure, no speculation is required to

perform router bypass. For the same reason, the designer has more flexibility to accommodate the router clock cycle to system requirements without affecting the bypass mechanism.

3.2.2 Correctness: Starvation, Livelock and Deadlock Freedom

To avoid network anomalies, LIGERO basically uses two mechanisms for packet movement guarantee. The intra-router approach borrowed from ROTARY and a newly developed inter-router mechanism.

a) Intra-router Packet Movement Guarantee: The packets located in a router's internal loop must be able to reach any of the output ports allowing them to make forward progress. This condition is fulfilled simply by restricting the addition of new packets into the loop; a packet stored in a reception stage can progress to the DFIFO stage only if there is room for at least two packets in it. Note that the policy is enforced locally in each building block by the reception stage. Therefore in the most adverse situation (all DFIFOs except one are exhausted), this rule guarantees that each router maintains space for at least one packet in the internal loop. To apply this condition, the DFIFO should have room for at least two packets. From a global perspective, the rule guarantees the existence of at least N packet holes in an N router network. From a local point of view, this construct guarantees non-blocking routers, through simple hardware structures and without virtual channels, which in contrast with conventional routers, is a remarkable achievement and constitutes the foundation of the whole idea.

b) Inter-router Packet Movement Guarantee*:* Any packet stored in the network will be able to reach the destination node in a finite number of cycles. To achieve this construct we need:

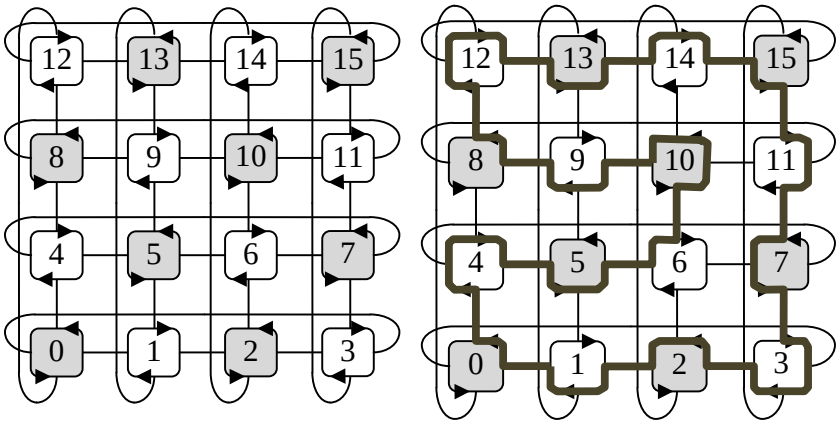

Figure 4. (a) Chessboard-like network configuration. (b) Embedded cyclic escape path.

b.1) Escape routing sub-function: After a predefined and high enough number of unsuccessful rotations in any router loop, a packet is marked as *on-escape* and it can leave the router using an escape routing sub-function [Duato 1995] output port. This condition holds for the packet until it reaches its destination. The routing sub-function is restricted to using only the network links that belong to a cyclic path, similarly to the scheme shown in Figure 4(b) for a 4-ary 2-cube, passing through all the nodes in the network. In any connected network, it is always possible to find a similar cyclic path. Following that path, *on-escape* packets will eventually reach their destination. Although this requires misrouting, under corner-case situations, the packet will reach its destination without any livelock issue. The effect on

performance of the solution is negligible because the injection control policy makes the proportion of on-escape marked packets negligible under any working conditions. Finally, forcing packets to follow a fixed route toward destination also eliminates the possibility of livelock situations. In contrast, other misrouting approaches require non-trivial mechanisms to avoid livelock issues [Vantrease et al. 2011; Moscibroda and Mutlu 2009; Hayenga et al. 2009].

b.2) Lifesaver hole existence: It is necessary to guarantee the existence of at least *N+1* packet holes in the whole network to allow the on-escape packets to reach their destinations. Intra-router Movement Guarantee assures the existence of *N* packet holes in the network. Increasing the restriction to inject new packets into the network, it is possible to assure the extra hole or *life-saver hole* by applying the following rule:

"**For injecting a new packet** into the network there must be **room** for **at least three packets** (three packet holes): two at the Basic Building Block (denoted local BBB) that includes the output port where the injection is performed and one at the reception stage of the neighboring router where it will be stored".

The restriction on the injection is applied using only local BBB information because the signaling of the room for a packet in the neighboring router is provided by the VCT flow control signals. Respecting this rule, after a new packet is injected, the existence of *N+1* packet holes in the network is guaranteed. In the worst case, when we inject a new packet in the network and simultaneously two packets fill up the local DFIFO (coming from the *reception stage* buffer of the local BBB and the DFIFO of the previous BBB in the loop) and the rest of the routers have space for one packet, *N+1* packet holes will exist in the network: room for two packets in the preceding router and one packet in the remaining N-1 routers.

Even in the extremely unlikely situation of the existence of only *N+1* holes in the whole network, eventually all the packets will be marked as on-escape and will follow the cyclic path to destination. Note that this movement is possible because of the existence of the *lifesaver hole*. In contrast with conventional routers, deadlock is avoided without requiring the utilization of virtual channels for escape routing sub-function as the "on-escape" packets cannot be blocked by "regular" packets due to the Intra-router Packet Movement Guarantee.

3.3 Cache Coherent Chip Multiprocessor Perspective

According to the previous discussion, the network is free of anomalies through the use of the simple proposal above, without reducing any significant performance feature. However, if we want to use LIGERO inside a cc-CMP system, we need to confront the issues discussed in Section 2.

3.3.1 Correctness I: End-to-end Deadlock Avoidance

The organization of conventional input-buffered routers imposes severe limitations on the available solutions to deal with end-to-end deadlock. Exclusive buffering per input port obliges adopting solutions at a port level, while FIFO buffering policies require the reservation of exclusive resources for each message type. For these reasons, the most commonly adopted solution consists in implementing separate buffering resources at each input port, known as virtual channels. In the case of packet rotation routers, the internal ring enables the sharing of buffering resources among every input port, allowing us to search for router-level solutions (instead of input-port level ones). Additionally, the continuous circulation inside the ring breaks up the strict FIFO ordering at buffers, making the reservation of exclusive resources for each message type unnecessary.

The mechanism proposed in LIGERO consists in a per-message-type flow control, where the priority of the traffic class is established according to its position in the message dependency chain [Song and Pinkston 2003]. According to its priority each traffic class is allowed to make use of a growing portion of router storage capacity. The basics of the mechanism are easily understood making use of a simple communication pattern, a request-reply protocol. In this case with two classes of traffic, reply traffic has higher priority. At each LIGERO router the requests can only occupy up to half of the buffering resources of the router's internal ring, while the replies can make use of all the buffering resources. In this way, as the internal ring has a non-FIFO behavior (forwarding is allowed due to continuous advance of packets), request messages can never exhaust the resources and stop reply messages. In the event of consumption queue overflow at any router, progress for reply (higher priority) messages is always guaranteed through the resources exclusively devoted to messages with higher priority (at least half of router capacity will be guaranteed). In other words, cyclic dependency between consumption and injection queues will never generate deadlock.

In order to guarantee a portion of each router buffering for each type of traffic, we only need to have separate stop signals per traffic class between adjacent routers. At each router the number of packets per traffic class is counted and the stop signals are raised accordingly. When the number of traffic classes involved in the message dependency chain is greater than two, no data-path modifications are required, only inter-router handshaking. To achieve the same functionality in a conventional router, separate virtual or physical networks are required for each class of traffic. This implies increasing the number of virtual channels or the number of physical networks required. For blocking routers without virtual channels, separate physical networks will be required.

3.3.2 Correctness II: In-Order Delivery Support

Two of the most remarkable features of LIGERO, adaptive routing and the possibility of forwarding inside the router loop, can also make it difficult to support in-order delivery. Consecutive packets that should reach their destinations in order may be shuffled because they can follow different paths (adaptivity) or perform different numbers of laps of the internal loop (non-blocking switching) in any of the intermediate routers.

Path diversity is eliminated forcing in-order traffic to make use of deadlock-free deterministic routing. In the case of a torus topology, DOR routing over the embedded mesh is enough to eliminate path diversity, thus also guaranteeing routing-deadlock freedom. The additional control logic required to implement DOR routing for ordered transactions is minimal. One single bit in the header flit identifies ordered messages, while almost the same control logic as that used for adaptive routing can be employed. At each multiport buffer, if an ordered message is detected, a request for the ejection stage is only generated if the previous dimension has been exhausted, which can be easily done by comparing current and destination positions in the required dimension.

Buffering at the reception and ejection stages is created through FIFO structures. The multiport buffers of each BBB also follow a FIFO policy, guaranteeing message ordering at all these stages. However, the loop formed by multiport buffers can cause packet reordering if two ordered transactions are allowed to make use of the loop simultaneously and they perform a different number of laps. Overtaking inside the router is avoided by restricting the utilization of the internal loop to only one ordered message per input port. This mechanism only requires one control signal per input port for implementation. Every time an ordered message advances from a reception

module to the internal loop, this signal is activated, stopping newly ordered messages from reaching the router loop from the same reception stage. This stop signal is deactivated once the ordered message reaches a profitable ejection module, where, once again, strict ordering is guaranteed. The same stop signal employed to avoid overtaking inside the loop eliminates the possibility of an ordered message performing router bypass. Otherwise, messages could be overtaken despite loop restrictions, by making use of the bypass path.

It should be noted that the control mechanisms used to guarantee in-order delivery are based on path (more restrictive routing) and temporal (blocked loop access) restrictions to ordered messages, but they do not impose additional conditions on loop occupation level. If occupation policies are also respected for ordered messages (according to their priority), in-order delivery mechanisms are compatible with the end-to-end deadlock avoidance in LIGERO. Additionally, the presence of ordered transactions is not restricted to only one message class. The mechanism is still valid for protocols with ordered transactions in different positions of the message dependency chain and it also works correctly if one message class mixes ordered and un-ordered transactions. The restrictions imposed by ordered messages strongly limit the potential performance benefits of LIGERO if these are the dominant kind of transactions. However, more aggressive ordering policies will not be cost-effective when working under the reasonable assumption that in-order messages will only make up a small percentage of traffic, which is true for most currently employed coherence protocols.

The small buffering provision of LIGERO makes it possible to support ordered traffic. To support this requirement in other deflection-based routing strategies, such as the one employed in the CHIPPER [Fallin et al. 2011] and MRR [Abad et al. 2009] routers, complex solutions would be required. Most of the proposals do not even contemplate the existence of such a requisite. If coherence protocol or system maintenance procedures require in-order delivery, a separate physical network will be necessary. In conventional routers, such as VCTM [Jerger et al. 2008], in-order traffic is naturally supported because the routing is deterministic and the switching is blocking.

3.3.3 Benefits: Adaptive On-Chip Multicast Support for Coherence Traffic

As mentioned earlier, many coherence protocols use multicast messaging to accelerate data race solution, reduce storage overheads or simplify protocol design. It is straightforward to add fully adaptive in-network multicast support to LIGERO using a similar mechanism to MRR [Abad et al. 2009]. As the internal loop organization is analogous, we can also distribute the routing table around the output ports employing register masks as in [Abad et al. 2009]. At each BBB, a register (named routing mask) consisting of a bit-vector of length *N* (where N is the number of network nodes) will indicate those nodes reachable through its ejection stage at minimal distance. The remaining information is carried by the message header flit, where a bit-vector (message mask) of the same length is employed to encode message destination nodes. Those packets advancing through the internal loop must perform two1-gate bit-to-bit operations indicating whether all, part or none of the destinations are reachable through that ejection stage. While the packet advances through the internal loop, replication can be performed at each BBB where the masks indicate those destinations that can be reached from that port. To support deadlock-free replication in an internal loop we need room for at least two packets in the *ejection stage* of the incoming path from the internal loop. Otherwise, we could exhaust the *lifesaver hole* necessary to guarantee network level deadlock freedom in the replication process. This establishes the minimum buffering capacity for the ejection stage at two packets. As in the case of MRR, the restriction on replicating

messages provides fully adaptive multicast tree support, which means that under low-load conditions the multicast follows a wide-multicast tree, while under heavy-load conditions it tends to follow a path. The multicast tree allows fast packet delivery when the network is empty if the multicast path minimizes the number of packet replications. Replicating packets inside the network increases the level of contention, which potentially could destabilize it if it is heavily loaded. Adaptive multicast support is so complex in conventional routers that it is not usually considered even in off-chip networks [Dally and Towles 2001].

The packets at the reception stage must perform a similar mask operation to check whether the message has reached one of its destinations, employing their message mask and an additional mask (destination mask) where only the destination position is activated in the bit-vector. In order to maintain arbitration simplicity in LIGERO, multicast messages will only be allowed to perform router bypass when the operation of message and routing masks indicates that all destinations of the incoming packet at the reception stage are reachable through the bypass path. If bypass conditions are fulfilled, the packet moves directly from the *reception stage* to the *output link*. Other routers, such as [Jerger et al. 2008; Hu et al. 2011], have also advocated support of multicast in on-chip networks but none of them have been able to combine the features that LIGERO provides: low cost, adaptive multicast and router bypass for multicast traffic.

4. EVALUATION FRAMEWORK

4.1 System Configuration

The system simulator is based on the GEMS [Martin et al. 2005] infrastructure, where the original network simulator has been replaced by TOPAZ [Abad et al. 2012] in order to enable precise modeling of the network behavior. DSENT [Chen 2012] and Cacti 6.5 [Muralimanohar et al. 2007] have been employed to model cost issues in the whole on-chip cache hierarchy. Processor energy is not considered in this study. The main parameters of the simulated system are shown in Table I. The simulated CMP has 16 aggressive OOO processors with static shared S-NUCA L2. System layout uses a folded torus to connect the 16 L2 banks. Each router connects a processor and an L2 bank. Cores operate at 4GHz and memory subsystem at 2GHz.

Table I. Main parameters of the simulated system.

Number of Cores	16@4GHz
Win Size / outs.req. per CPU	128/16
Issue Width	4
L1 I/D cache	Private, 32KB, 2-way, 64B block, 2-cycle
L2 cache	16MB SNUCA, 16-way, 16 banks, 1 bank per router, 5-cycle
Coherence Protocol	Broadcast-Based or Directory-Based
Main Memory	4GB, 250 cycles, 320GB/s
Command Size	16 bytes
Network Topology	4-ary 2-cube (16x1 banks + 16 cores)
Network Link (width/lat)	128 bits / 1 cycle

In order to better understand the potential effect of the different working conditions of the network, we chose two different coherence protocols based on Directory and Broadcast. Both protocols will be constructed over a Token Coherency framework [Martin et al. 2003]. The correctness substrate of this framework allows us to evaluate our proposal for different policies, high performance or bandwidth efficiency. The goal of the TokenB high-performance policy is better average on-chip

access time achieved by broadcast-based protocols. TokenB makes use of substantially more bandwidth than a directory protocol. For this reason, we will also evaluate a policy with lower bandwidth requirements. TokenD makes use of the token correctness substrate to emulate a directory protocol, resulting in a protocol with the similar bandwidth and latency sensitivity. Similarly to other coherence protocols [Park et al. 2010; Intel 2009], TokenB and TokenD design requires six classes of traffic. The starvation avoidance mechanism requires in-order delivery.

The workloads used in this study are three multi-programmed and eight multi-threaded workloads (numerical and transactional) running on top of the Solaris 9 OS. The numerical applications, (FT, IS, SP, LU) are part of the NAS Parallel Benchmarks (OpenMP implementation). The transactional benchmarks (Apache, JBB, Zeus, OLTP) correspond to the Wisconsin Commercial Workload suite [Martin et al. 2005], released by the authors of GEMS in version 2.1. The remaining class corresponds to multi-programmed workloads using part of the SPEC CPU2000 benchmark. Multiprogrammed workloads are evaluated in rate mode (one instance of the program per available processor) and with reference inputs. The mix of benchmarks employed covers broad utilization scenarios for the network, ranging from very low to very high load.

4.2 Counterparts and Memory Hierarchy Cost Modeling

Next we will detail all router micro-architecture implementation costs considered in the evaluation. We use DSENT models in order to estimate area and energy requirements for every counterpart router. We will estimate the remaining parts of the on-chip cache hierarchy using CACTI 6.5. The three counterparts selected for evaluation are: a network based on the MRR [Abad et al. 2009], another one based on the VCTM Router [Jerger et al. 2008] and a third network based on CHIPPER [Fallin et al. 2011]. Some system-level correctness issues were not taken into account in the original evaluation of some routers (such as VCTM and CHIPPER). In this work we will evaluate both the original version of each router and a modified one guaranteeing system-level correctness. The election of these counterparts was motivated by how all of them cover cc-CMP requirements in the design space. As discussed in Section 2, there are three basic properties: network complexity, network performance and support for system level requirements. A suitable design should cover all of them in the best way possible. CHIPPER is a good design that deals with the first two points elegantly. MRR deals with the last two points nicely. VCTM partially deals with the last point and deals reasonably well with the first one. We will compare all of them across different coherence protocols and workloads. In this way, the comparison will provide the reader with a better perspective of the benefits of LIGERO which attempts to cover all key aspects simultaneously.

In LIGERO, to guarantee network and coherence protocol correctness, the minimal buffering per building block is six packets: one is required at the reception stage, two are required at the ejection stage and three are required at the DFIFO. The local node interface does not require buffering on the router side. Therefore, the whole router must include room for a minimum of 24 packets. According to Table I, the packet size is 80 Bytes and the flit size is 128 bits, which represents 2 KB for each router. The MRR router will require 10 two-port buffers with room for four packets in order to work properly, i.e. 325flits (5KB) are necessary per router.

The original proposal of CHIPPER is unable to avoid message-dependent deadlock or provide in-order delivery. In order to evaluate how these limitations affect system performance, we will use two different versions of the router. The initial proposal, denoted as CHIPPER-UNLIMIT, will assume unlimited consumer capacity. Under such circumstance, it is safe to use a single physical network to manage all traffic

classes. We use a separate physical zero-cost network with a conventional router to manage in-order traffic. As no virtual network can be implemented, under realistic conditions (i.e. limited consumer capacity) separate physical networks are required to guarantee message-dependent deadlock avoidance. To maintain fixed wire link width at 128 bits, optimistically instead of six, we will assume four separate physical networks with 32-wire links. We will denote this approach CHIPPER-REAL. We will continue to assume zero-cost for in-order networks. Neither CHIPPER-REAL nor CHIPPER-UNLIMIT has support for multicast traffic.

Table II. Energy per event and Area.

	ENERGY (pJ)					AREA (mm^2)(%)	
	B. Write	B. Read	S. Trav.	L. Trav.	Static	Router	Net/Cache
VCTM UNLIMIT	3.38	3.16	1.17	26.56	17.5	0.0931	0.433%
VCTM REAL	7.19	6.85	1.17	26.56	40.2	0.1923	0.897%
CHIPPER UNLIMIT	--	--	1.17	26.56	1.15	0.0216	0.101%
CHIPPER REAL	--	--	0.31	6.64	0.289	0.0017	0.007%
	I. Stage	O. Stage	MP. Buf	L. Trav.	Static	Router	Net/Cache
MRR	2.77	2.88	4.04	26.56	43.35	0.2447	1.138%
LIGERO	2.81	2.94	3.49	26.56	23.52	0.1326	0.617%

With respect to VCTM, a first approach called VCTM-UNLIMIT assumes unlimited consumer queues and uses four virtual channels per port in order to avoid network-level deadlock and to use virtual-channel flow control(Document_not_found, n d). As in the original proposal, buffering per virtual channel will be dynamically allocated using 24-flit buffers (1.5KB per router). The realistic version, denoted VCTM-REAL, requires six virtual networks to avoid message dependent deadlock. Therefore, as two virtual channels are required to avoid network-level deadlock, we increase the number of virtual channels to twelve. To accommodate this number of virtual channels without affecting performance we need to increase the dynamic buffering capacity to 60 flits, 300flits (4.5KB) being necessary for the whole router. The centralized crossbar makes this router even more costly than MRR. It should be noted that in order to increase the credibility of our evaluation, the characteristics of LIGERO´s counterparts have been selected trying to reach their optimum performance–cost ratio, sometimes being a little bit unfair on our proposal.

Table II summarizes the area foot-print, both as an absolute value and comparing it to L1 and L2 area, and energy required by a flit to cross a router in the absence of contention, assuming 2GHz clock, 32 nm technology and 5mm wire length. For reference, the table includes the cost ratio of the network with respect to the remaining parts of the cache hierarchy. According to CACTI, each L2 bank requires 19.2 mm2, has a leakage of 182.59 pJ/cycle and requires 2408 pJ per access. L1 caches require 2.24mm2, have a leakage of 16.3 pJ/cycle and require 530.54 pJ per access.

5. PERFORMANCE EVALUATION

Figure 5 shows the performance results for each coherence protocol, application and router. The scenario for the interconnection network is broad and diverse. For the directory coherence protocol, in most applications the extended network's maximum sustainable throughput is not relevant enough to compensate for added delay at low load. Except in some high load applications, such as IS, in routers without bypass, such as the MRR router, system performance drops slightly. In contrast, LIGERO's low-load latency provides a slight advantage in most applications, being 10% faster than MRR. CHIPPER-UNLIMITED performs similarly to LIGERO except in high load applications where performance drops by up

to 20%, when the network is not behaving as expected. CHIPPER-REAL performs poorly because network link splitting makes packets longer (20 flits for response and 4 flits for commands), which increases average latency and contention. Although further optimizations performed on CHIPPER could improve its performance results, the differences are so high in a realistic scenario that even the improved router MinBD [Fallin et al. 2012], where a throughput improvement between 2% and 8% is reported, would not be able to significantly change the conclusions extracted from these results (on average, execution time is nearly twice LIGERO's).

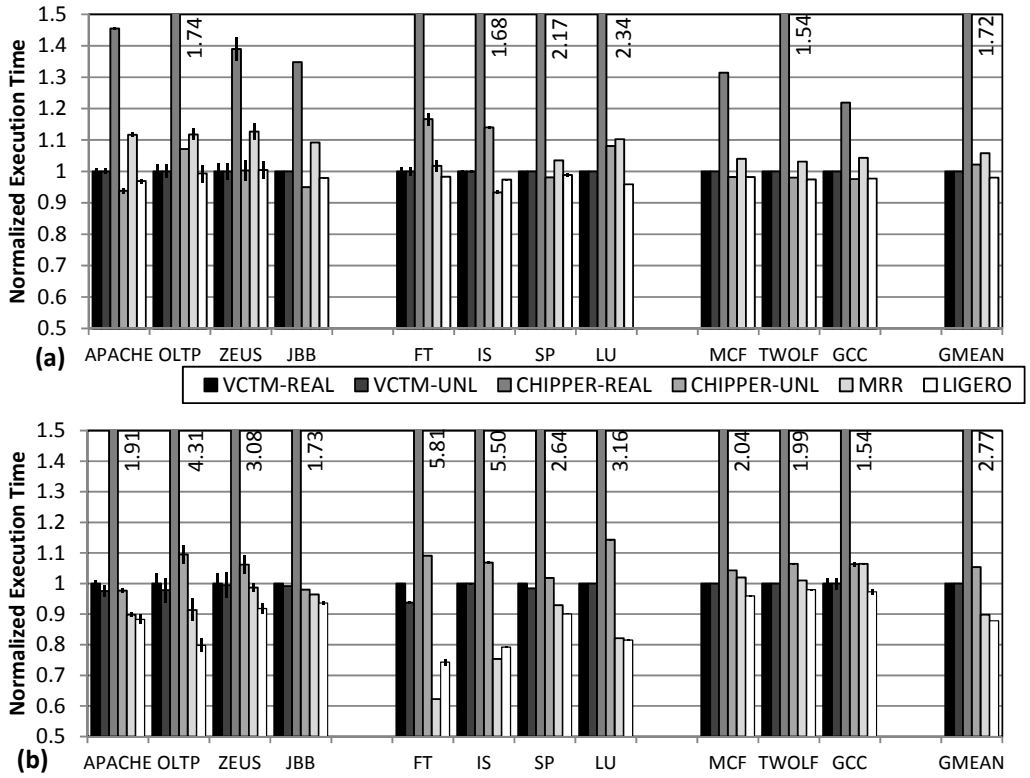

Figure 5. VCTM-REAL Normalized Performance Results (a) Directory-Based Coherence Protocol. (b) Broadcast-Based Coherence Protocol.

When the coherence protocol is broadcast based, the network is much more relevant. Although with this protocol, MRR behaves much better, LIGERO is still the best performer. For this protocol the remaining routers perform much worse; there is more than 20% performance loss. In these contended situations, CHIPPER is the worst performer due to longer packets and lack of support for multicast traffic. VCTM routers have static multicast support whereas LIGERO and MRR provide adaptive multicast.

More than the absolute performance differences, the most interesting result is that LIGERO is the best performer in most conditions. Nevertheless, according to Table II, CHIPPER variants are much cheaper in terms of area and power. In order to combine, performance-cost tradeoffs, Figure 6 provides the Energy-Delay product (EDP) of all combinations. The energy of the whole cache hierarchy has been measured in this evaluation, showing in the graph the fractions corresponding to the network components (upper bar) and to the two cache levels (lower bar). As can be seen, the network contributes in a limited way to the memory hierarchy, which makes performance improvements more important than energy saving in most cases. For example, in directory protocol, the MRR energy overhead degrades EDP by up to

20% compared to VCTM routers, but in broadcast-based protocol, the greater network pressure is compensated by lower cache energy consumption, making the system more efficient than the VCTM router. Usually energy estimations do not take into account that the network is just another part of a bigger system. We need to look at the whole picture to understand the benefits of an idea. When this is taken into account, CHIPER-REAL seems to be infeasible for a cc-CMP system, with up to 23 times the EDP of VCTM. CHIPPER-UNLIMITED has a reasonable EDP under low load conditions but the reader should be aware that under such conditions the system is not deadlock free. When we combine the simplicity and good performance of LIGERO in all the applications and coherence protocols, it is clearly the best performer, saving up to 20% in EDP with respect to VCTM.

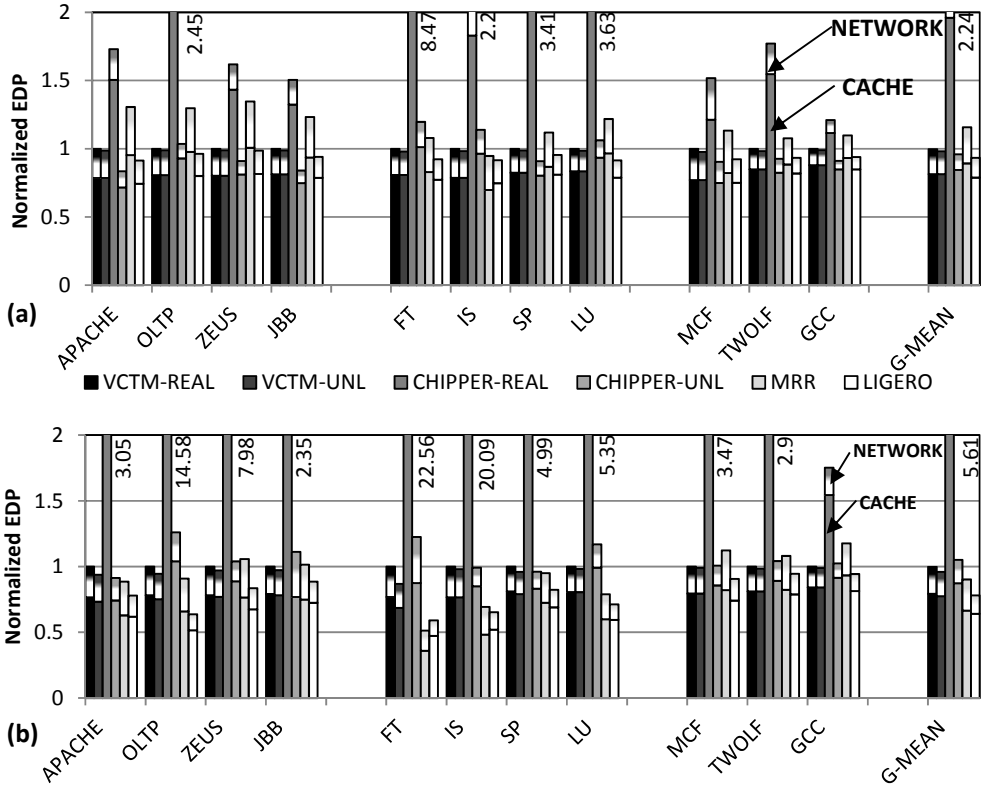

Figure 6. VCTM Normalized on-chip Memory Hierarchy Energy Delay Product (a) Directory-Based Coherence Protocol. (b) Broadcast-Based Coherence Protocol.

In order to better understand the previous results, Figure 7 shows raw network performance under synthetic traffic conditions in steady state. Mimicking each coherence protocol, separate performance is shown with and without multicast traffic. At injection time, each packet is categorized uniformly among six classes of traffic. Packet length distribution is bimodal, with 50% probability of being 80 bytes and 50% of being 16 bytes. As can be seen in the Figure 7(a), in the absence of multicast traffic, LIGERO, both VCTM implementations and the ideal configuration of the CHIPPER router present very similar latency curves. If we compare these results with those obtained for applications making use of directory-based coherence protocol (where the fraction of multicast messages is lower), we will observe a similar result. In this case performance differences are minimal, and LIGERO is able to obtain the best results thanks to its slightly better maximum throughput numbers. In the case of MRR, its better throughput values are not able to compensate for the lack of base-latency optimizations.

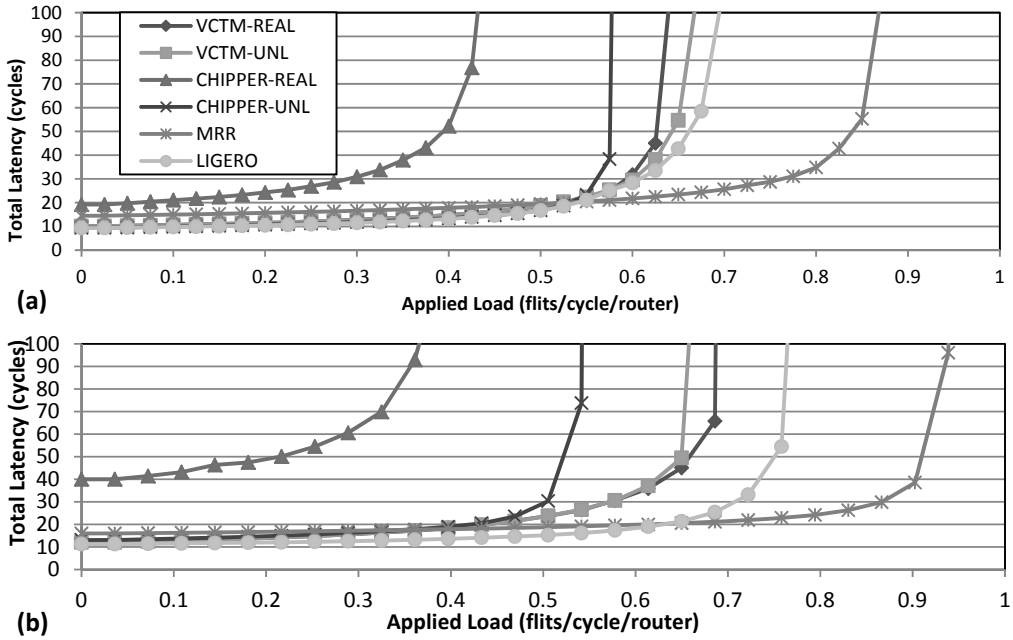

Figure 7. 4×4 Torus performance with uniform traffic: (a) Unicast Traffic, (b) 15% (at consumption) of multicast traffic.

In Figure 7(b), where a fraction of the traffic pattern becomes multicast, we clearly observe the benefits of LIGERO, which still maintains the same base-latency values while improving its maximum sustained throughput, with values closer to those obtained by MRR. This traffic configuration better mimics a system with a broadcast-based coherence protocol. If we observe the results in Figure 5 and compare them with this last synthetic traffic evaluation, we can again observe a similar behavior in both cases. LIGERO, with the best tradeoff between base-latency and maximum throughput, is the router with the best results in the applications.

6. CONCLUSIONS

Based on packet rotation router structures we have been able to create a new router micro-architecture that successfully reconciles implementation cost and provision of features for a cc-CMP system. Even though LIGERO is remarkably simple, it has all the features achievable with routers with much higher implementation costs. Integrating known mechanisms and completely developing new ones, the router proposed presents a set of characteristics that make it the most efficient structure for the target system. LIGERO is HoL blocking free, uses adaptive routing, has optimized pass-through latency in low-load situations, can work with any topology, is deadlock free at both network and coherence protocol levels, and supports on-network adaptive multicasting. All of these characteristics allow it to improve throughput at a fraction of the cost of other state-of-the-art routers.

REFERENCES

ABAD, P., PUENTE, V., GREGORIO, J.A., AND PRIETO, P. 2007. Rotary router: an efficient architecture for CMP interconnection networks. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, 116-124.

ABAD, P., PUENTE, V., AND GREGORIO, J.A. 2009. MRR: Enabling fully adaptive multicast routing for CMP interconnection networks. In *Proceedings of the International Symposium on High Performance Computer Architecture (HPCA)*, 355-366.

ABAD, P., PRIETO, P., MENEZO, L., COLASO, A., PUENTE, V., AND GREGORIO, J.A. 2012. TOPAZ: An Open-Source Interconnection Network Simulator for Chip Multiprocessors and Supercomputers. In *Proceedings of the International Symposium on Networks-on-Chip (NOCS)*, 99-106.

AGARWAL, N., PEH, L.-S., AND JHA, N.K. 2009. In-Network Snoop Ordering (INSO): Snoopy coherence on unordered interconnects," In *Proceedings of the International Symposium on High Performance Computer Architecture (HPCA)*, 67-78.

ASANOVIC, K., BODIK, R., AND CATANZARO, B. 2006. The landscape of parallel computing research: A view from Berkeley. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-183.

CHEN, C.-H.O., KURIAN, G., WEI, L., MILLER, J., AGARWAL, A., PEH, L.S., AND STOJANOVIC, V. 2012. DSENT – A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling. In *Proceedings of the International Symposium on Networks-on-Chip (NOCS)*, 201-210.

COPPOLA, M., LOCATELLI, R., MARUCCIA, G., PIERALISI, L. AND SCANDURRA, A. 2004. Spidergon: a novel on-chip communication network. In *Proceedings of the IEEE International Symposium on System-on-Chip*, 15-15.

DALLY, W.J., AND SEITZ, C.L. 1986. The torus routing chip," *Distributed Computing, 1*, 187-196.

DALLY, W.J. AND TOWLES, B. 2001. Route packets, not wires: on-chip interconnection networks. In *Proceedings of the Design Automation Conference* (DAC), 684-689.

DAS, R., MUTLU, O., MOSCIBRODA, T., AND DAS, C.R. 2010. Aergia: Exploiting Packet Latency Slack in On-Chip Networks. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, 106-116. DUATO, J. 1995. A theory of deadlock-free adaptive multicast routing in wormhole networks. *IEEE Transactions on Parallel and Distributed Systems (TPDS),* vol. 6, 976-987.

DUATO, J. YALAMANCHILI, S., AND NI, L. 1997. Interconnection Networks: An Engineering Approach, *IEEE Press*, 1997.

FALLIN, C., CRAIK, C., AND MUTLU, O. 2011. CHIPPER: A low-complexity bufferless deflection router. In *Proceedings of the International Symposium on High Performance Computer Architecture (HPCA)*, 144-155.

FALLIN, C., NAZARIO, G., YU, X., CHANG, K., AUSAVARUNGNIRUN, R., AND MUTLU, O. 2012. MinBD: Minimally-Buffered Deflection Routing for Energy-Efficient Interconnect. In *Proceedings of the International Symposium on Networks on Chip (NOCS)*, 2012.

HANSSON, A., GOOSSENS, K., AND RĂDULESCU, A. 2007. Avoiding Message-Dependent Deadlock in Network-Based Systems on Chip. *VLSI Design*, 1-10.

HAYENGA, M., JERGER, N.E., AND LIPASTI. M. 2009. SCARAB: a single cycle adaptive routing and bufferless network. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, 244-252.

HU, W., LU, Z., JANTSCH, A., AND LIU, H. 2011. Power-efficient tree-based multicast support for networks-on-chip. In *Proceedings of the Asia and South Pacific Design Automation Conference*, 363–368.

(INTEL), 2009. An Introduction to the Intel ® QuickPath Interconnect. 1-22.

JERGER, N.E., PEH, L.S., AND LIPASTI, M. 2008. Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support. In *Proceedings of the International Symposium on Computer Architecture (ISCA),* 229-240.

KAROL, M., HLUCHYJ, M., AND MORGAN, S. 1987. Input Versus Output Queueing on a Space-Division Packet Switch. *IEEE Transactions on Communications*, Vol. 35, 1347-1356.

KELTCHER, C.N., MCGRATH, K.J., AND AHMED, A. 2003. The AMD Opteron processor for multiprocessor servers," *IEEE Micro, 23*, 66-76.

KRISHNA, T., PEH, L.S., BECKMANN, B.M., AND REINHARDT, S.K. 2011. Towards the Ideal On-Chip Fabric for 1-to-Many and Many-to-1 Communication. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, 71–82.

KUMAR, A., PEH, L.S., KUNDU, P., JHA, N.K. 2007. Express Virtual Channels: Towards the Ideal Interconnection Fabric. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, 150-161.

LENOSKI, D., LAUDON, J., GHARACHORLOO, K., WEBER, W.-D., GUPTA, A., HENNESSY, J., HOROWITZ, M., AND LAM, M.S., 1992. The Stanford Dash multiprocessor. *Computer*, Vol. 25, 63-79.

MA, S., JERGER, N.E., AND WANG, Z. 2012. Whole Packet Forwarding: Efficient Design of Fully Adaptive Routing Algorithms for Networks-on-Chip. In *Proceedings of the International Symposium on High Performance Computer Architecture (HPCA)*, 467-478.

MARTIN, M., HILL, M., AND WOOD, D.A. 2003. Token Coherence: a New Framework for Shared-Memory Multiprocessors," *IEEE Micro*, 108-116.

MARTIN, M.M.K., SORIN, D.J., BECKMANN, B.M., MARTY, M.R., XU, M., ALAMELDEEN, A.R., MOORE, K.E., HILL, M.D., AND WOOD, D.A. 2005. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset. *ACM SIGARCH Computer Architecture News*, Vol. 33, 99-107.

MICHELOGIANNAKIS, G., BALFOUR, J., AND DALLY, W.J. 2009. Elastic-Buffer Flow Control for On-Chip Networks. In *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, 151-162.

MOSCIBRODA, T., AND MUTLU, O. 2009. A case for bufferless routing in on-chip networks. *ACM SIGARCH Computer Architecture News*, Vol. 37, 196-207.

MUKHERJEE, S., BANNON, P., LANG, S., SPINK, A., AND WEBB, D. 2002. The Alpha 21364 network architecture. *IEEE Micro*, 26-35.

MURALIMANOHAR, N., BALASUBRAMONIAN, R., AND JOUPPI, N. 2007. Optimizing NUCA Organizations and Wiring Alternatives for Large Caches with CACTI 6.0. In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, 3-14.

MURALI, S., MELONI, P., ANGIOLINI, F., ATIENZA, D., CARTA, S., BENINI, L., MICHELI, G., AND RAFFO, L. 2006. Designing Message-Dependent Deadlock Free Networks on Chips for Application-Specific Systems on Chips. In *Proceedings of the International Conference on Very Large Scale Integration*, 158-163.

PARK, C., BADEAU, R., BIRO, L., CHANG, J., SINGH, T., VASH, J., WANG, B. AND WANG, T. 2010. A 1.2 TB/s on-chip ring interconnect for 45nm 8-core enterprise Xeon® processor. In *Proceedings of the IEEE International Solid-State Circuits Conference* (ISSCC), 180-181.

RAGHAVAN, A., BLUNDELL, C., AND MARTIN, M.M.K. 2008. Token tenure: PATCHing token counting using directory-based cache coherence," In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, 47–58.

ROGERS, B.M., KRISHNA, A., BELL, G.B., VU, K., JIANG, X., AND SOLIHIN, Y. 2009. Scaling the Bandwidth Wall: Challenges in and Avenues for CMP Scaling. In *Proceedings of the International Symposium on Computer Architecture (ISCA),* 371-379.

SAMMAN, F.A., HOLLSTEIN, T., AND GLESNER, M. 2010. Adaptive and Deadlock-Free Tree-Based Multicast Routing for Networks-on-Chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 18, 1067-1080.

SONG, Y.H. AND PINKSTON, T.M. 2003. A progressive approach to handling message-dependent deadlock in parallel computer systems. *IEEE Transactions on Parallel and Distributed Systems*, Vol 14, 259-275.

STOK, P. 2005. Dynamic and Robust Streaming in and between Connected Consumer-Electronic Devices. *Kluwer*. Dordrecht.

STRAUSS, K., SHEN, X., AND TORRELLAS. J. 2007. Uncorq: Unconstrained Snoop Request Delivery in Embedded-Ring Multiprocessors," In *Proceedings of the International Symposium on Microarchitecture (MICRO)*, 327-342.

VANTREASE, D., LIPASTI, M.H., AND BINKERT N., 2011. Atomic Coherence: Leveraging nanophotonics to build race-free cache coherence protocols. *In Proceedings of the International Symposium on High Performance Computer Architecture (HPCA*), 132–143.