MRR: Enabling Fully Adaptive Multicast Routing for CMP Interconnection Networks

Pablo Abad, Valentin Puente and Jose-Angel Gregorio
University of Cantabria
{abadp, vpuente, monaster}@unican.es

Abstract

On-network hardware support for multi-destination traffic is a desirable feature in most multiprocessor machines. Multicast hardware capabilities enable much more effective bandwidth utilization as multi-destination packets do not need to repeatedly use the same resources, as occurs when multicast traffic must be decomposed in unicast packets. Although Chip Multiprocessors are not an exception in this interest, up to date, few fitting proposals exist. The combination of the scarcity of available resources 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 the Multicast Rotary Router (MRR), a router able to: (1) perform on-network multicast support with almost zero cost over the Rotary Router, (2) use a fully adaptive tree to distribute multicast traffic, (3) perform on-network congestion control extending network utilization range. The performance results, using a state-of-the-art full system simulation framework, show that it improves average full system performance of a CMP using a unicast Rotary Router in its interconnection network by 25%, and an input buffered router with multicast support by 20%.

1. Introduction

In off-chip interconnection networks, multicast hardware support has been a hot topic for many years, generating an enormous quantity of proposals. Just to cite a few, these include [19][21][23][30][31][33]. In contrast, the multi-destination issue has rarely been considered in on-chip interconnection networks in general or in CMPs in particular. In most cases, [18][27][2], it has been assumed that one-to-many (multicast) or one-to-all (broadcast) communications can be implemented efficiently in the optimized one-to-one (unicast) mechanism by the network interfaces or coherence controllers. This assumption is mainly motivated by resource scarcity in the on-chip interconnection network context and large bandwidth availability. Under these conditions, the off-chip solutions could not be adopted and this discouraged research into new approaches to solve the problem.

Nevertheless, as shown recently in [10], multi-destination traffic has a serious impact on CMP system performance. Without using any special mechanism, in a 4×4 mesh network under random traffic, if 1% of all injected packets[1] are multicast the saturation point drops significantly. The main reason for this drop in performance derives from the increased latency of messages. Replicating the messages in the source node causes a waste of bandwidth due to the reiterative resource use of unicast packets that belong to the same multicast message. Moreover, unicast decompositions for multi-destination packets increase the waiting time at their injection queues in each node because of the unavoidable need to sequence the use of the output links.

In this work, we introduce an innovative way to deal with multicast traffic under on-chip imposed constraints. The router, denoted by Multicast Rotary Router or MRR, is based on the new concept of router presented in [2]. The structural solutions and algorithmic procedures used by Rotary Router make it feasible to solve the problem of multi-destination traffic from a different perspective. The new approach provides on-net multicast support at negligible cost in terms of implementation. No extra hardware resources are needed compared with a conventional unicast Rotary Router, and by extension, MRR will be feasible

[1] We are using the terms “packet” and “message” indistinctly.

too for CMP networks. Additionally, the new proposal is able to perform the multicasting dynamically, taking into account the occupation of network resources. Utilizing only local information, MRR is able to use adaptively defined trees to perform multi-destination transactions. MRR uses on-net replication control which avoids network saturation, extending network operation range. To our knowledge, no other feasible fully-adaptive multicast technique for on-chip interconnection networks has been proposed to date.

The rest of the paper is structured as follows: Section 2 explains the motivation for adaptive multicasting. Section 3 describes our proposal based on the Rotary Router. In Section 4 the evaluation methodology is depicted. Section 5 thoroughly analyzes the performance of the proposal and finally, Section 6 states the main conclusions of the paper.

2. Adaptive-Tree Multicasting

There is an evident need for multicast communication in general-purpose chip multi-processors in order to efficiently handle their coherence protocols. As shown in [10], there is a wide variety of proposed coherence protocols that would benefit from multicast support. Thus, in directory-based protocols, such as the SGI-Origin protocol [20], it is found that the invalidation messages with multiple destinations can be up to 5% of total messages and without multicast support, their average latency increases to twice the network average latency. In the case of broadcast-based protocols, such as the token coherence protocol [24], the impact is clearly even greater. Simulation results show that the lack of hardware multicast support can double the execution time of some real applications. Similar requirements appear in common coherency protocols such as those employed in Intel QPI [15], the AMD Hypertransport [8] or those that will emerge as a natural consequence of new architectures [6][17][32], the growing number of processors [10] and the potentials and limitations of on-chip communications themselves.

However, the hardware mechanisms necessary to add multicast support to the network increase its cost. Additional deadlock conditions could arise and increased congestion could degrade performance. As a consequence, the complexity of the router design and the buffering requirements could get higher. On the one hand, the generation of replicas of a multicast packet in intermediate nodes reduces the occupation, but significantly increases the likelihood of deadlock as a consequence of the need to access several output ports simultaneously [19][21]. On the other hand, when the network load is high, the generation of new packets in the intermediate nodes increases network congestion supra-linearly [19].

In off-chip networks many solutions attempting to alleviate both effects have been proposed. They can be divided into two large groups [19]. The first one, called *path-based* multicast, is based on restricting the number and locations where replicas are performed, increasing the length of the path that multicast packets must follow to reach all of their destinations. This approach eases the deadlock avoidance, but forces the packet to follow longer paths thus increasing the latency under low to medium loads. A second group, known as *tree-based* multicast tries to increase the number of necessary resources (virtual channels, storage, etc.) in order to prevent / recover deadlock situations. In this case, in addition to increasing the hardware complexity, replication in the intermediate nodes, without intervention of the injection queues (therefore without their flow control mechanism), can easily flood the network. This reduces performance of the network at high loads. For example, Figure 1 shows the packet latency in a 4-ary 2-cube network under random traffic, with 10% of broadcast packets. Under low applied load, the best approach is the tree-based solution. The highest throughput is achieved by the path-based approximation. Both cases significantly improve the unicast approach (converting each multicast packet into several unicast ones at the source node).

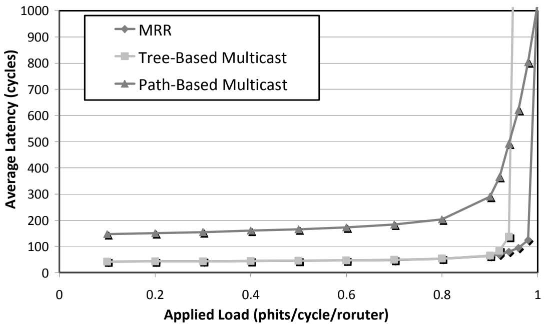
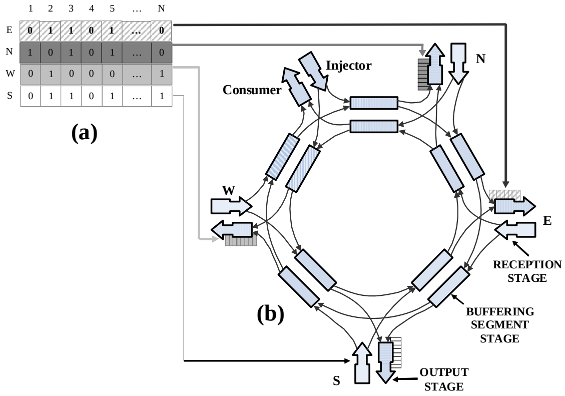


Figure 1. Average packet latency for different multicast solutions.

Our Fully-Adaptive Multicast mechanism behaves like a tree-based multicast policy at low or medium loads and like a path-based one when the network reaches saturation point. A router with such functionality will be able to adapt its behavior to the applied load, ranging progressively from a broader tree multicast distribution to a narrower one. When the network resources are lightly used, all destinations of each multicast packet could share the links that are common along the minimal route from source node. Therefore, destinations can be represented as the leaves of a tree whose branches are minimal routes from the root or source node (tree-based). Each new branch means a replica of the message. However, replicating a message not only depends on minimizing the distance to each destination, but also on the network congestion level. If a router that should perform a replica detects a high occupancy, it sends the message to any of its possible destinations without creating a new replica. Therefore, some of the new branches of the tree no longer route packets to the destination through a minimal route. In fact, in the unlikely extreme case of complete saturation of all network resources, each message should follow a path through all of its destinations, performing the replicas only for consumption (path-based). This implies that packet routing is carried out according to a tree that is being created dynamically, adapting to the congestion of the interconnection network. This adaptive behavior potentially enables the exploitation of the best features of both types of solutions mentioned above.



	1	2	3	4	5	…	N
E	0	1	1	0	1	…	0
N	1	0	1	0	1	…	0
W	0	1	0	0	0	…	1
S	0	1	1	0	1	…	1

Figure 2. Distributed Routing Table.

3. The Multicast Rotary Router (MRR)

3.1. Zero Cost: In-network Packet Multicasting

MRR foundations rely on the Rotary Router [2]. This architecture proposes an innovative organization for the router, and eliminates some common structures present in more classic architectures, such as global arbiters or crossbars. Figure 2(b) depicts the structure of this router for a bi-dimensional topology. The operation of the Rotary Router is based on two internal rings where packets circulate in opposite directions, looking for a suitable output port. This movement simplifies output arbitration, reduces contention and presents a noticeably better Energy-Delay trade-off than conventional router architectures. The deadlock avoidance mechanism is topology agnostic, making the router suitable for any kind of network topology. Additionally, with minimal modifications over this structure, it is possible to suppress the necessity of virtual channels in order to avoid protocol message-dependent deadlock, optimizing network buffering utilization and simplifying router control for any length of the message dependency chain generated by the coherence protocol [1].

```
if ( (is p not available) | | ( $\vec{s}_m \,\&\, \vec{r}_p = \vec{0}$ ) ) {
    Keep on Turning m;
}
else {
    if ( ( $\vec{s}_m \,\&\, \vec{r}_p \,!= \vec{0}$ ) && ( $\vec{s}_m \,\&!\, \vec{r}_p = \vec{0}$ ) ) {
        Eject packet m;
    }
    if ( ( $\vec{s}_m \,\&\, \vec{r}_p \,!= \vec{0}$ ) && ( $\vec{s}_m \,\&!\, \vec{r}_p \,!= \vec{0}$ ) ) {
        Eject a replica m’ with $\vec{s}_{m'} = \vec{s}_m \,\&\, \vec{r}_p$ ;
        Keep on Turning m with $\vec{s}_m = \vec{s}_m \,\&\, (!\, \vec{r}_p)$
    }
}
```

Figure 3. Routing algorithm with replication: initial version(C-like notation).

The characteristics of the Rotary Router make adding support for multi-destination packets almost straightforward. If each packet carries information about all its destinations, on-router replication can be performed by simply letting the packet circulate until all replicas have been created at each specified output port. The necessary information can be carried by the packet header using one bit per node. For an *N*-node network, *N* bits would be required [7].

The Rotary Router is topology agnostic. This makes table-based routing the best approach and facilitates handling of the routing of both unicast and multicast packets in a unified way. Taking into account that every packet will sequentially reach each output port, a per output port register could allow to identify the reachable destination nodes through minimal path. As a result, instead of a centralized routing table, as in Figure 2(a), we can distribute it around the output ports, as in Figure 2(b). Consequently, for an *N*-node network, each output port *p* requires an *N*-bit register mask $\vec{r}_p$ where $r_p(i)=1$ if node *i* can be reached through the port *p* at minimal distance. If the path is not minimal, $r_p(i)$ will be 0. For regularity reasons, the port corresponding to the consumer will also be provided with a register where only the position corresponding to the current node is set. In the Figure 4, the node represented is number zero.

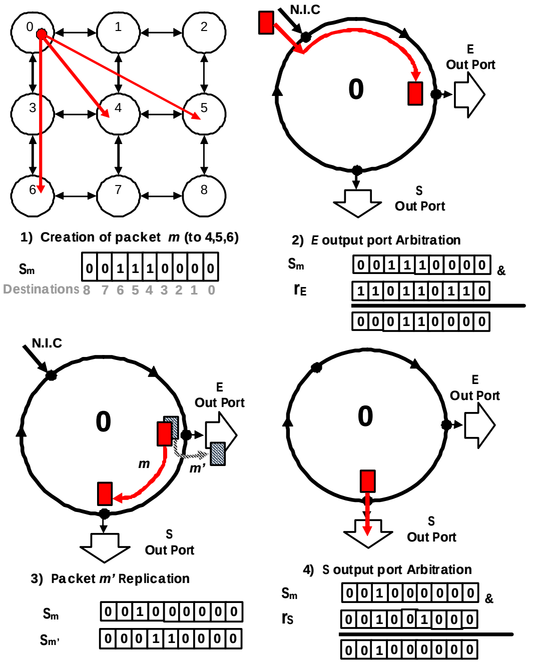


Figure 4. Example of message replication and header updating.

The next element required to apply the routing is the packet header. As has been previously stated, the first phit of any packet, *m*, should carry a string of *N* bits, denoted by $\vec{s}_m$. Each position of $\vec{s}_m$ represents a single network router. The combined use of $\vec{r}_p$ and $\vec{s}_m$ will determine if the packet should be ejected at port *p* and/or replicated to the next multiport buffer. Based on both the message bit string $\vec{s}_m$ and the register mask $\vec{r}_p$ of the output port *p*, an efficient and simple algorithm (see Figure 3) can be implemented. In this way, it is possible to handle indistinctly multicast and unicast messages. If the actual port is not profitable or for some reason, it is not available (another packet is in transit or there is no room for the current packet in the *Output Stage*) the packet will keep going round inside the ring. Otherwise, if the packet can reach all of its destinations through the current output port it will be ejected through this port. If only a subset of its destinations are reachable by minimal path through *p*, i.e. $\vec{s}_m\ AND(NOT\,\vec{r}_p)$ is different from zero, a replica *m'* of the packet is ejected with its header updated to $\vec{s}_{m'} = \vec{s}_m\ AND\,\vec{r}_p$. Additionally, the original packet will keep on going round, but with its bit string updated including only the destinations pending, i.e. $\vec{s}_m = \vec{s}_m$ *AND (NOT* $\vec{r}_p$). If we look carefully, the algorithm is also valid for unicast traffic. If the destination of the packet is not reachable using a minimal path through *p*, $\vec{s}_m\ AND\ \vec{r}_p$ will be zero and, as expected, the packet will keep on going round the ring. Otherwise, if *p* is a profitable output port, $\vec{s}_m\ AND\ \vec{r}_p$ is not null, and obviously no other destinations will be pending ($\vec{s}_m\ AND\ (NOT\,\vec{r}_p)$=0) and the packet will be ejected.

The algorithm is applied to the packets at the head of each *Buffering Segment Stage*. The turning to the next *Buffering Segment Stage* or ejection to the *Output Stage* could be performed simultaneously if both are available. As an example and assuming a near-zero applied load, Figure 4 walks through the process of destination encoding and multicast routing, assuming port availability. Every time a NIC injects a new multicast packet its header bit string indicates all the selected destinations. In Figure 4(a), router #0 generates a packet with destination routers #4, #5 and #6. After selecting a router ring the packet advances to the first output port, where arbitration begins. Step 2 shows the result of bit computation. As some but not all multicast destinations are at minimal distance through this output port, the packet starts the replication process. At step 3 it can be observed how original and replicated packet bit strings are generated. The bit string of the ejected packet is updated, asserting only the reachable nodes through the ejection port, in this case routers #4 and #5. On the other hand, the original packet will have a new bit string where the destinations assigned to the ejected replica are cleared. Finally, the original message keeps on circulating inside the router, reaching the next output port. At step 4 a new arbitration process begins. This time every multicast destination of the packet (router #6 in the example) is reachable through this output port. For this reason, the packet will leave the router without being replicated again. With this simple mechanism the network is able to generate a tree for each multicast message, improving link utilization by sharing common links of the routes to all of its destinations.

3.2. Correctness: Deadlock avoidance

The resource utilization determines output port availability, and it will dynamically reshape the multicast distribution. Although it is necessary to address the network congestion exacerbated by the in-network traffic creation, correctness must be ensured first. In general, the most serious issue is that on-network replication can potentially violate the deadlock avoidance mechanism employed. In the case of the Rotary Router, packet replication could consume the extra holes required to guarantee message movement between nodes [2], violating network correctness. This Subsection introduces the changes in the previous algorithm required to keep the network deadlock free.

The network deadlock avoidance mechanism in the unicast Rotary Router [2] is maintained by the following rules:

1. New packets can only be *injected* ***in a router ring*** if there is at least space for two or more packets in the destination *Buffering Segment Stage* inside the ring.
2. New packets can only be *injected* ***in the network*** if there is at least space for three or more packets in the destination *Buffering Segment Stage* inside the selected ring.
3. If after a predefined number (and large enough) of complete turns in the same router the packet cannot advance through a minimal path to its destination, it will be eligible for misrouting, being able to leave the router through the first available output port.

Assuming Virtual Cut-through flow control (VCT) [16], rule *1* guarantees that packets at every ring in the network will never stop, rule *2* guarantees the existence of a *lifesaver* hole moving between all the network routers and rule *3* guarantees that any packet can use the lifesaver hole to advance to the next router under any load condition. Additionally, to avoid end-to-end deadlock without the presence of virtual channels, VCT was slightly adapted [1]. The modified flow control guarantees that low order traffic in the message-dependence chain, generated by the coherence protocol, never blocks higher order traffic, by limiting the total buffering utilization per router depending on the traffic priority.

At this point MRR does not obey rule 3, thus the lifesaver hole could be exhausted by the ejected packet replica while the original one is also still circulating inside the ring. If this happens, the network will be deadlock prone. In order to solve this problem, before replicating a message we must ensure that we are not consuming the lifesaver hole. In Figure 3, the first condition to process the packet is to check for *p* availability. In the unicast version, *p* is available if both the *Output Stage* buffer has space for one packet and the remote router ring buffering utilization for the level of traffic is below the limit (modified VCT flow control).

```
if ( (is p not available) || (( $\vec{s}_m \& \vec{r}_p = \vec{0}$ )
        &&(m is not misroutable)))                    A
{
   Keep on Turning m;
}
else {
   if ((( $\vec{s}_m \& \vec{r}_p != \vec{0}$ ) && ( $\vec{s}_m \&! \vec{r}_p = \vec{0}$ ))
        || (m is misroutable))                        A'
   {
        Eject packet m;
   }
   if ( ( $\vec{s}_m \& \vec{r}_p != \vec{0}$ ) && ( $\vec{s}_m \&! \vec{r}_p != \vec{0}$ ) ) {
        if (out_stage_buffer.space() >=2){            B
             Eject a replica m' with $\vec{s}_{m'} = \vec{s}_m \& \vec{r}_p$;
             Keep on Turning m with $\vec{s}_m = \vec{s}_m \& (! \vec{r}_p)$;
        }
        else Eject packet m;                          C
   }
}
```

Figure 5. Routing algorithm with replication: Deadlock free version.

To maintain the network deadlock free, port availability has to be redefined. In fact, if the *Output Stage* buffer has space for two packets, then it is safe to make a replica because in the worst case the lifesaver hole will be the remaining hole. As the lifesaver hole can never be at the consumption *Output Stage* buffer, simply providing room for one packet is enough for consuming an instance of a multicast packet. Then, to maintain the network deadlock-free a fourth rule will be needed:

4. A multicast message *can only be replicated* and ejected if the *Output Stage* buffer for the output port has room for at least two packets. The rule does not apply to consumption ports.

Note that this rule does not limit the utilization of *p*, just the replication. In other words, a multi-destination message can still be eligible to use an output port with room for just one packet. Although under this condition, rule 4 forbids the production of a new replica this only means that a branch of the tree has been pruned, (which would have been generated if the reply had been permitted).

In summary, Figure 3's algorithm should be redefined to guarantee deadlock freedom, the final version being shown in Figure 5. First of all, in addition to the original algorithm the conditions boxed in *A* and *A´* indicating if rule 3 applies. Second, for replicating a packet there must be at least room for two packets in the Output Stage (Box B). Otherwise, if *p* is a profitable output port only for some destinations in a multicast packet but there is not enough space for a new replica (condition *B* does not hold), the packet will leave the router with all its destinations unaltered (Box C). Again, no special treatment is required for unicast traffic.

3.3. Performance: On-network congestion control

The proposed deadlock avoidance mechanism is not only suitable to keep network correctness but it is also able to improve performance. The replication probability is correlated to the network occupancy level. The replication arises when inter-router channels are lightly loaded (out port availability implies low channel utilization). In heavily loaded resources the replication is limited. Moreover, the shape of the multicast tree will be self-adapted to the network utilization status. Under low load conditions the multicast will follow a wide-tree for packet destinations whereas under heavy load it will be closer to a deep-tree.

The solution adopted to avoid deadlock by restricting the replication process not only enables reshaping the multicast tree but also performs an implicit congestion control. Under high applied loads, network routers are not able to deliver messages at the same rate as they are produced. They are accumulated in the injection queues and each message must compete with all previously generated ones. It is well known that injection restriction is an effective way to reduce congestion [3][28]. However, replicated messages are generated bypassing the injection queues. Additionally, it is also well known [21] that they can create little hot-spots that reduce the mobility of packets trying to cross through the router where replication is taking place. For this reason, the mechanism for avoiding packet deadlock by restricting replication is also a simple and efficient mechanism for avoiding congestion caused by multicast traffic.

To prove the previous asseverations, we will analyze the evolution of multicast trees depth when replication control is applied. Also, we will explore depth evolution effect on performance. As reference values, path-based and tree-based multicasting solutions will be employed. In contrast to Adaptive-tree multicasting, these techniques have fixed values for tree depth. In order to ensure path-based and tree-based mechanisms correctness for the RR, unlimited output-stage buffering will be used to ensure deadlock freedom without replication control. For an 8×8 torus with only broadcast traffic Figure 6 shows how deadlock avoidance increases tree depth. At low or medium loads the average distance of MRR is closer to a tree-based multicast, and consequently similar latency is observed. Notwithstanding, when the network is closer to saturation average distances of packets in MRR are noticeably higher than in tree-based multicast. However, this increase in the distance the packets travels causes a lower congestion and consequently better maximum sustainable throughput, as observed in Figure6.down. In summary, adaptive-tree distribution of MRR combines the efficiency of tree-based distributions when the network congestion is low and path-based when the network is saturated in a simple but effective way.

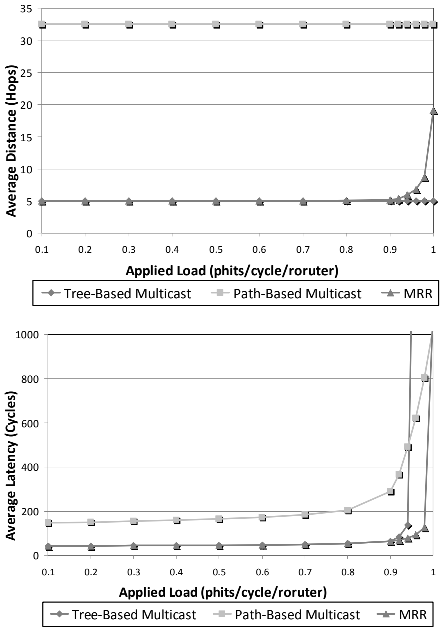


Figure 6. Broadcast traffic with MRR and different multicast distributions: (up) Avg distance, (down) Avg latency.

4. Evaluation Methodology

4.1. Simulation Framework

In order to know the real impact of MRR on full-system performance, we will employ a complex simulation infrastructure composed of four connected simulation tools. The full system simulator Simics [22] has been extended with the GEMS timing infrastructure

[25]. 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 [29]. This simulator, although slower, allows us to take into account most of the hardware implementation details with much higher precision. Moreover, in order to perform power consumption estimations SICOSYS has been connected to the Orion power simulator [11]. This framework will allow us to perform exhaustive full-system simulation with complex workloads and also detailed modeling of the most relevant system modules at architectural level.

Table 1. Simulated CMP parameters.

Number of Cores	16
Window Size / outstng req. per CPU	64/16
Issue Width	4
L1 I/D cache	Private, 32KB, 2-way, 64-Byte block, 1-cycle
Direct Branch Pred.	4KB YAGS
Indirect Branch Pred.	256 entries(cascaded)
L2 cache	16MB SNUCA, token coherence protocol, 16x16 banks, 4 banks/router
L2 cache bank	64KB, 16-way, 3-cyc, pseudo LRU, 64-Byte block
Main Memory	4GB, 260 cycles, 320 GB/s
Command size	16 bytes
Network Topology	8x8 torus
Network link	128 bits / 1 cycle

4.2. Full-System Configuration and Workloads

The simulated system is a 16-processor CMP with shared S-NUCA L2 based on [5]. The protocol, based on Token Coherence [24], requires a hierarchy of six classes of messages to be implemented. We have chosen this coherence protocol because it is extremely multicast sensitive. 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. If the L1 does not receive the necessary tokens after a fixed timeout, a new multicast message with the same destinations is generated. This coherence protocol also requires point-to-point packet ordering for some specific actions (persistent req. activations and deactivations). As they represent a small fraction of the network traffic we have chosen to decompose the multicast messages of those transactions in unicast packets. The main parameters of the simulated system are shown in Table 1.

The workloads considered in this study are two multi-programmed and nine multithreaded workloads running on top of Solaris 9 OS. We have selected a broad spectrum of workloads composed of a mixture of three different classes of applications. Two of the classes are multithread applications, being numerical and transactional applications. A summary is provided in Table 2.

Table 2. Workloads considered in our study.

Benchmark	Description
Wisconsin Commercial Workload Suite	
Apache	Task-parallel web server
Jbb	Java middleware application
Zeus	Pipelined web server
Oltp	Pseudo TCP-C on-line trans. processing
NAS Parallel benchmark	
FT	3-D partial diff. eq. solution using FFTs
IS	Integer sort
SP	Scalar Pentadiagonal solver
BT	Block Tridiagonal solver
LU	LU solver
SPEC2000 Multiprogrammed	
Twolf	Place & Route simulator, 16 instances
Gcc	C optimizing compiler, 16 instances
MCF	Combinatorial optimization, 16 inst.

The numerical applications are part of NAS Parallel Benchmarks (OpenMP implementation version 3.2.1 [14]). The transactional benchmarks correspond to the Wisconsin Commercial Workload suite [4], released by the authors of GEMS in 2.1version. The other class are multi-programmed workloads using part of SPEC2000CPU [13] applications. The benchmarks are evaluated in rate mode (one instance of the program per available processor) and with reference inputs.

For each simulation point a variable number of runs are performed with pseudo-random perturbation in order to estimate workload variability [14]. All the results provided have a 95% confidence interval.

4.3. Synthetic Traffic Configuration

Prior to full-system simulation and in order to clarify the performance benefits of MRR an evaluation based on synthetic traffic will be carried out. The main parameters employed in this evaluation are summarized in Table 3. In all cases the traffic pattern of multicast traffic is uniform, whereas unicast traffic will be different. There will be 8 or 64 destination nodes of multicast traffic, emulating a directory protocol or a broadcast-based coherence protocol respectively.

5. Performance Evaluation

5.1. Competitive Counterparts

In the presence of unicast traffic patterns the Rotary Router has proven to perform better than input buffered routers [1] [2]. Therefore, we must clarify which part of performance improvement is achieved by the router structure itself, and which part is caused by the adaptive multicasting. The simplest algorithm evaluated consists of breaking down multicast messages into multiple unicast at the injection.

Table 3. Main Synthetic Traffic Characteristics.

Topology	8-ary 2-cube
Message Size	5 phits
Mcast % of total traffic	1%, 5%, 10% and 15%
Mcast nº of destinations	8 or 64
Unicast Traffic Pattern	Uniform, bit-reversal, perfect-shuffle, matrix-transpose, tornado
Cycles simulated	200,000 (20,000 warm-up)

In order to contrast the effectiveness of MRR versus other multicast proposals, we have compared our proposal to a conventional deterministic input-buffered router without multicast support, denoted by BASE using Bubble Flow Control to avoid deadlock [28]. Additionally, a router with idealized multicast support, structurally similar to the baseline router and denoted by BASE-MC, has been considered in the comparison. Although BASE-MC uses a solution similar to the one presented in [10], the router has some peculiarities. The technique employed by [10], known as Virtual Circuit Tree Multicasting, dynamically generates DOR trees to deliver multicast messages, avoiding setup latency but increasing the router complexity. BASE-MC will assume sufficiently large Destination Set CAMs to hold an unlimited number of active multicast trees and no setup packets are required to construct each multicast tree. As no adaptive routing is performed for the BASE-MC, no special actions (like the decomposition in unicast packets performed by MRR) are needed in order to guarantee point-to-point ordering. Finally, the Rotary Router as defined in [1] is included in the study, so enabling the quantification of the real impact on performance enhancement of MRR.

In order to isolate the effect of multicasting mechanisms all routers will present similar characteristics. All of them use Virtual Cut-through [16] as flow control. A similar buffer capacity has been assumed for all of them. The BASE routers have 10-phit FIFO statically allocated queues and 6 virtual channels per input port due to the message dependency chain length of the coherence protocol used [24]. The sizes chosen are those where optimal ED2P is achieved in BASE routers. In order to keep the storage area per router constant, in the Rotary Router and MRR, buffer capacity is 20 phits in the Buffering Segment Stage, 20 phits in output stages and 10 phits in input stages. Note that according to [1] no virtual channels are required in order to avoid end-to-end deadlock. In this way, the total storage capacity per router is 300 phits, which requires less than 5KB per router. Note that each router connects to 256KB of L2, and consequently router storage represents 1.8% of L2 banks served. In order to maintain hardware simplicity, none of the routers employ hardware techniques to optimize pipeline length under low load conditions [18][27].

5.2. Synthetic Traffic

Figure 7 shows the best-result, BASELINE normalized, average base latency for different unicast traffic patterns, different proportions and number of destinations of multicast traffic in an 8×8 torus. Adding multicast support to the router leads to a noticeable improvement due to the elimination of the serialization at injection queues. When the number of destinations or the percentage of multicast traffic increases the improvement is greater since the waiting time at injection queues is substantially higher.

Studying the routers without multicast support, the RR router exhibits a lower latency than BASE. This is due to the HOL blocking produced in the injection queue of BASE router, which RR minimizes. All unicast packets that belong to the same multicast destinations must wait to access the output port sequentially, whereas in the RR the simultaneous access to the output port and the next buffering segment reduces serialization. Moreover, RR is fully adaptive whereas BASE is deterministic and consequently the BASE serialization could appear in any network queue. As we can see, if the number of the destinations is increased, the differences are higher.

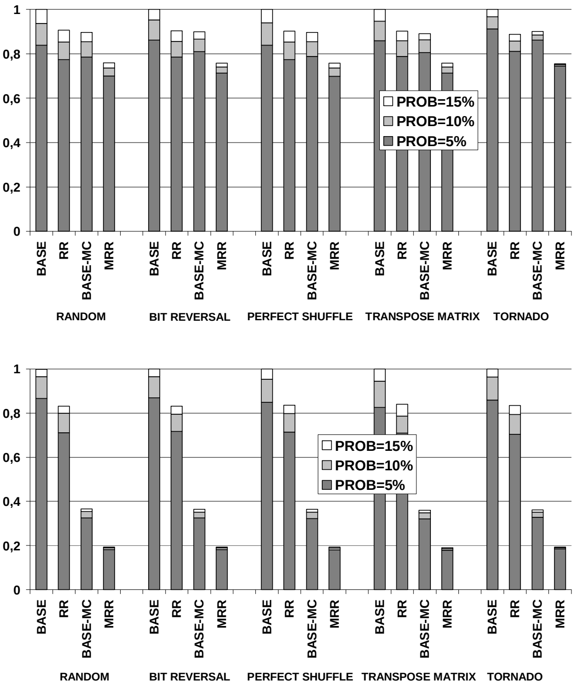


Figure 7. Baseline Normalized Low-load latency for (up) 8-destination multicast, (down) broadcast.

Focusing attention on multicast routers, MRR outperforms BASE-MC. MRR can carry out packet replicas at buffering segment stage simultaneously due the dual output port. Consequently, under low contention conditions multiple replicas of the same packet could leave the router almost simultaneously. This is a clear advantage over any input buffered router, but without the cost of output-queuing or centralized buffering structures [30]. Similarly to [10], in BASE-MC the crossbar scheduling for multicast packets is sequential. In this way, the crossbar arbiter is output-port based, which has the best cost/performance ratio [26]. In spite of simultaneously selecting all the required output ports, each replica is processed individually. In [10] VCTM, due to the flit-level multiplexing required for wormhole flow control, every flit of a multicast packet sequentially requests all output ports before proceeding through the crossbar. As packet multiplexing is more delay-efficient than flit multiplexing [9], BASE-MC multicast packet delay is even more favorable than the original VCTM. This discussion makes it clear that BASE-MC is not unfairly penalized compared to the VCTM router, consequently MRR will also outperform the [10] router in base latency. As previously suggested, mixing different flow controls in this comparison could blur the analysis. Although VCTM is not explicitly compared, the work inspired the design choices taken in BASE-MC.

To evaluate the effectiveness of each router in terms of maximum sustainable throughput, a constant load of 1 phit per cycle and per router is applied. The best-case normalized accepted load for different traffic patterns, multicast percentage and number of destinations is shown in Figure 8. Focusing our attention on multicast routers, MRR is still the best performer and by contrast BASE-MC behavior suffers noticeably. While resource utilization by multicast traffic is more efficient in BASE-MC than RR, its performance is worse. As we can see, increasing the proportion of multicast traffic and the number of destinations, slightly improves BASE-MC performance. Nevertheless, when non uniform pattern traffic is employed BASE routers perform poorly. This is due to the fact that Rotary Routers use adaptive routing, avoid HOL and have less contention due to the lack of a centralized arbiter. In Tornado traffic, which is an almost worst-case traffic, adaptive routing impact is maximal.

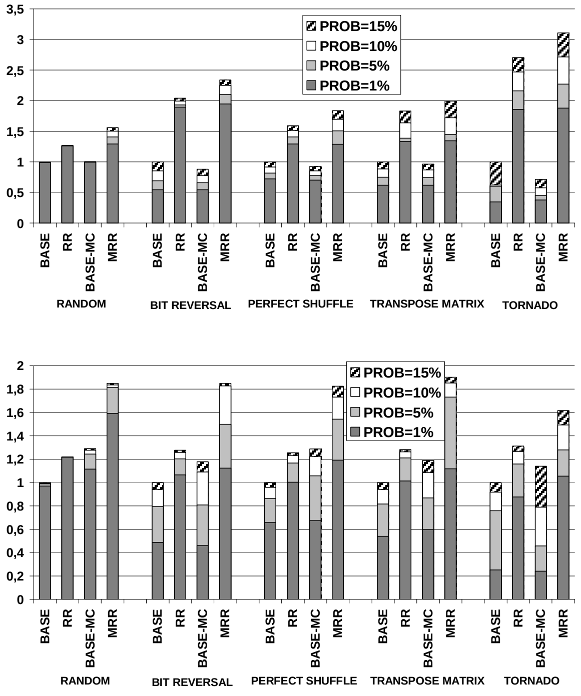


Figure 8. Baseline normalized throughput at maximum applied load (up) 8-destination multicast, (down) broadcast.

In MRR we have not only all the benefits of RR but also the better resource utilization of multicast traffic. Due to this, the network throughput could be improved up to 20% in some cases. Note that with broadcast, even 1% of multicast traffic implies that MRR outperforms RR by up to 33%.

Surprisingly, in some non-uniform unicast traffic, BASE-MC has worse throughput than the BASE router. Here, two confronted effects appear. On the one hand, multicast support improves resource utilization and so tends to improve performance. On the other hand, uncontrolled replication under high load conditions could destabilize the network under certain conditions [21]. To observe these two effects in more detail, Figure 9 shows the throughput evolution for two different destination patterns. As we can see, under uniform unicast traffic, multicast support helps to improve the throughput slightly due to the first effect. Notwithstanding, with bit-reversal unicast traffic the network becomes clearly unstable after reaching the saturation point. Although BASE-MC reaches a slightly better result at saturation, beyond this point the performance falls. Something similar happens with other non uniform traffic patterns. If we compare MRR versus RR that effect never arises due to the on-network replication restriction.

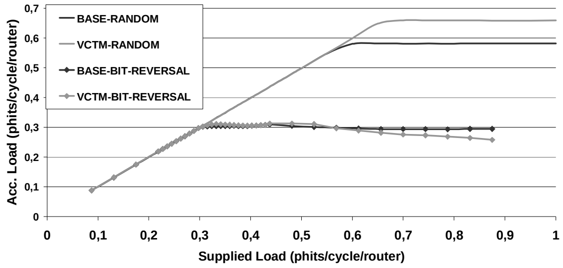


Figure 9. Throughput evolution with 10% broadcast traffic for Uniform and Bit-reversal traffic patterns.

Finally, normalized Energy-delay^2 Product (ED2P) metrics are provided in Figure 10. This metric is the most adequate to evaluate energy-performance trade-offs in high performance systems like our scenario. To determine those figures we divided the energy consumed by the square of maximum achievable throughput in stable network conditions. As we can appreciate, MRR performance is advantageous compared to its counterparts. Due to the idealized Destination Set CAMs in BASE-MC, its power consumption is not taken into account. Even under these clearly unfavorable circumstances for MRR, it is capable of greatly outperforming BASE-MC results. Only when a large proportion of broadcast packets is present are the differences constrained, being minimal in some specific traffic.

5.3. Full-system results

From the full-system perspective, the performance results are shown in Figure 11.up. As we can see, MRR outperforms all the counterparts consistently. It is clear that hardware support for multicast becomes worthwhile. This observation is shared by [10], although in that work, instead of full-system simulation, trace-driven was used. Centering our attention on RR, the performance boost obtained by MRR is even more noticeable, which is more remarkable if we take into account the small cost required. On average MRR makes a chip multiprocessor 25% faster than RR for the analyzed loads. Results also show that the impact on performance of decomposing point-to-point ordered traffic into multicast messages is negligible.

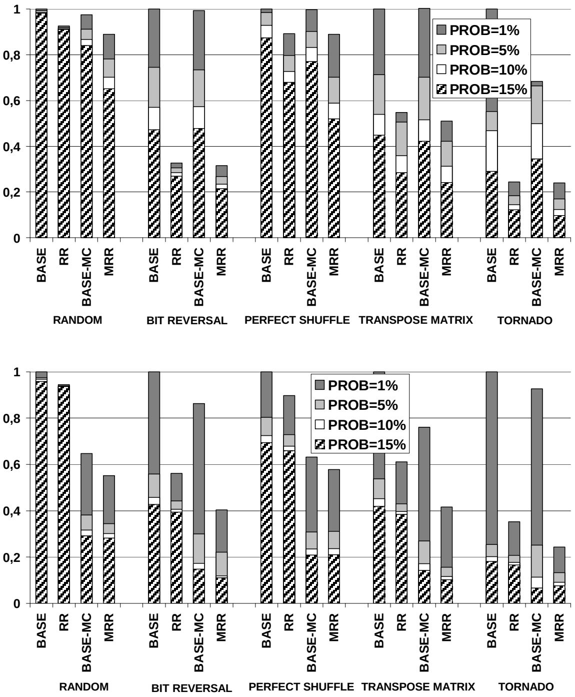


Figure 10. Baseline Normalized ED2P at maximum applied load (up) 8-destination multicast, (down) broadcast.

Comparing the two multicast routers, MRR is 20% faster than BASE-MC on average. These results, when compared with those obtained using synthetic traffic, suggest that most of the applications maintain the system under medium-to-low load, because the BASE-MC outperforms the RR router in most cases, as shown in Figure 7 (note that the coherence protocol uses broadcast traffic). However, with high bandwidth demanding applications, such as FT, non-multicast router RR could outperform BASE-MC, as expected from the synthetic traffic results in Figure 8. With more aggressive processor architectures this effect could be aggravated and extended to other benchmarks. In contrast, MRR's advantage is consistent under any application because the router performs better at high load level and at low load levels.

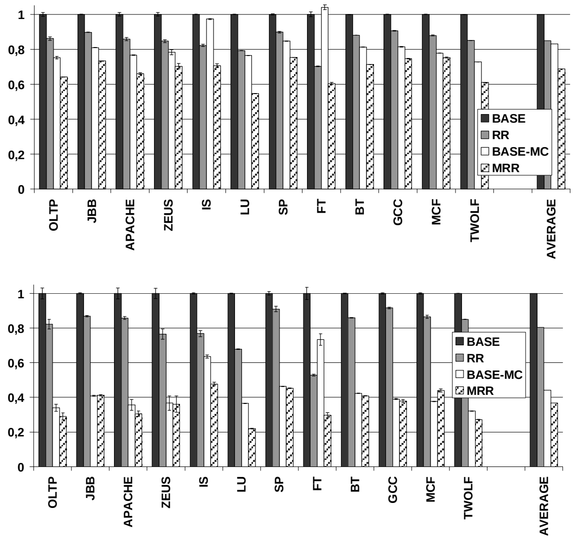


Figure 11. Base-line normalized (up) Execution time, (down) Interconnection Network ED2P.

Finally, we analyzed the impact of the proposal on the energy-delay trade-off. In Figure 11.down we provide the ED2P for the interconnection network employing the different router architectures and the applications considered. Note that the rest of the system power is not accounted for, and given that the architecture is kept constant from router to router, the relative differences could differ but not the tendency. In fact, it is predictable that better performance results lead to even more reduction in full-system ED2P. In any case, just looking at these results it is clear that multicast designs significantly reduce power consumption. Thus both MRR and BASE-MC clearly outperform non multicast routers. However, while MRR always maintains its advantage, for applications such as FT, the execution time degradation was so high that it was impossible for BASE-MC to outperform RR results. This leads to an important observation, namely, that for some applications, an apparently less power-hungry router such as the BASE-MC can waste energy since its inefficiency in handling the high number of messages imposed by the application increases the execution time.

This is not the case of the MRR because it is able to adapt to the network load. It is worth noting in Figure 11.down that the proposed multicast mechanism, on average, ED2P is reduced by half with respect to RR.

6. Conclusions

We have presented a new adaptive multicast mechanism based on a router, especially targeted to CMP architectures. Due to the special characteristics of the router implementation, the mechanism is so simple that the increased cost of hardware implementation compared to the unicast version is almost negligible and the performance achieved is greater than the best of current proposals.

Performance metrics have been obtained for different traffic characteristics, applied loads, number of destinations and proportions of multicast traffic. In all cases our proposal clearly outperforms current proposals. The same applies for the time reduction of real applications running on a complete system simulator.

Therefore, our proposal makes the Multicast Rotary Router a very competitive element for building interconnection networks of Chip Multiprocessors where multicasting is present, even in relatively low proportions.

7. Acknowledgments

This work has been supported by the Ministry of Education and Science of Spain, under contracts TIN2007-68023-C02-01, Consolider CSD2007-00050 and by the HiPEAC European Network of Excellence. The authors would like to thank the reviewers for their helpful comments.

8. References

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

[2] Abad, P., Puente, V., Gregorio, J.A., and Prieto, P. “Rotary router: an efficient architecture for CMP interconnection networks”. 34th International Symposium on Computer Architecture (ISCA), pp.116-125. June 2007.

[3] Adiga, N. et al. “An Overview of the BlueGene/L Supercomputer”. ACM/IEEE Supercomputing Conference, pp.60. 2002.

[4] Alameldeen, A.R., Mauer, C.J., Xu, M., Harper, P.J., Martin, M.M.K., Sorin, D.J., Hill, M.D., and Wood, D.A. “Evaluating non-deterministic multi-threaded commercial workloads”. 5th Workshop on Computer Architecture Evaluation Using Commercial Workloads, pp.30–38. February 2002.

[5] Beckmann, B. and Wood, D. “Managing Wire Delay in Large Chip-Multiprocessor Caches”. 37th International Symposium on Microarchitecture (MICRO), pp.319-330. December 2004.

[6] Burger, D. et al. “Scaling to the end of silicon with EDGE architectures”. IEEE Computer, Volume 37, No 7, pp.44-55. 2004.

[7] Chiang, C. and Ni, L.M. “Multi-address Encoding for Multicast”. Proceedings of the First International Workshop on Parallel Computer Routing and Communication, Springer-Verlag, pp.146-160. 1994.

[8] Conway, P. and Hughes, B. “The AMD Opteron Northbridge Architecture”. IEEE Micro, Vol. 27, Issue 2, pp.10-21. March 2007.

[9] Dally, W. and Towles, B. “Principles and Practices of Interconnection Networks”. Morgan Kaufmann Publishers Inc. 2003.

[10] Enright Jerger, N., Peh, L.S. and Lipasti, M. “Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support”. 35th International Symposium on Computer Architecture (ISCA). June 2008.

[11] Hang-Sheng Wang, Xinping Zhu, Li-Shiuan Peh, and Malik, S. “Orion: a power-performance simulator for interconnection networks”. 35th International Symposium on Microarchitecture (MICRO), pp.294-305. November 2002.

[12] Held, J., Bautista, J., and Koehl, S. “From a Few Cores to Many: A Tera-scale Computing Research Overview”. Intel Research (White Paper). 2006.

[13] Henning, J. “Spec2000: Measuring CPU performance in the new millennium”. IEEE Computer, pp.28–35. July 2000.

[14] Jin, H., Frumkin, M., and Yan, J. “The OpenMP Implementation of NAS Parallel Benchmarks and its Performance”. NASA Ames Research Center, Technical Report NAS-99-01. October 1999.

[15] Kanter, D. “The Common System Interface: Intel's Future Interconnect”. Real World Technologies. 2007.

[16] Kermani, P. and Kleinrock, L. “Virtual Cut-Through: A New Computer Communication Switching Technique”. Computer Networks, Vol. 3, pp.267-286. September 1979.

[17] Kim, C., Burger, D., and Keckler, S.W. “An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches”. ACM SIGPLAN Notices, Vol. 37, Issue 10, pp.211-222. October 2002.

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

[19] Kumar, D., Najjar, W., and Srimani, P. “A new adaptive hardware tree-based multicast routing in k-ary n-cubes”. IEEE Transactions on Computers, Vol. 50, No. 7, pp.647-659. 2001.

[20] Laudon, J. and Lenoski, D. “The SGI Origin: A ccnuma Highly Scalable Server”. 24th International Symposium on Computer Architecture (ISCA), pp.241-251. 1997.

[21] Lin, X. and Ni, L.M. “Deadlock-free multicast wormhole routing in multicomputer networks”. 18th International Symposium on Computer Architecture (ISCA), pp.116-125. 1991.

[22] Magnusson, P. et al. “Simics: A full system simulation platform”. Computer, Vol. 35, No. 2, pp.50-58. 2002.

[23] Malumbres, M.P., Duato, J., and Torrellas, J. “An Efficient Implementation of Tree-Based Multicast Routing for Distributed Shared-Memory Multiprocessors”. 8th IEEE Symposium on Parallel and Distributed Processing (SPDP), pp.186. 1996.

[24] Martin, M., Hill, M., and Wood, D. “Token Coherence: decoupling performance and correctness”. 30th International Symposium on Computer Architecture (ISCA), pp.182-193. June 2003.

[25] Martin, M.M.K. et al. “Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset”. SIGARCH Computer Architecture News, Vol. 33, No. 4, pp.92-99. November 2005

[26] Mukherjee, S.S., Silla, F., Bannon, P., Emer, J., Lang, S., and Webb, D. “A comparative study of arbitration algorithms for the Alpha 21364 pipelined router”. 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp.223-234. October 2002.

[27] Mullins, R., West, A., and Moore, S. “Low-latency virtual-channel routers for on-chip networks”. 31st International Symposium on Computer Architecture (ISCA), pp.188-197. June 2004.

[28] Puente, V., Beivide, R., Gregorio, J., Prellezo, J., Duato, J., and Izu, C. “Adaptive bubble router: a design to improve performance in torus networks”. International Conference on Parallel Processing, pp.58-67. 1999.

[29] Puente, V., Gregorio, J., and Beivide, R. “SICOSYS: an integrated framework for studying interconnection network performance in multiprocessor systems”. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, pp.15-22. September 2002.

[30] Sivaram, R., Panda, D.K., and Stunkel, C.B. “Efficient Broadcast and Multicast on Multistage Interconnection Networks Using Multiport Encoding”. IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 10, pp.1004-1028. 1998.

[31] Stunkel, C., Herring, J., Abali, B., and Sivaram, R. “A New Switch Chip for IBM RS/6000 SP Systems”. ACM/IEEE Supercomputing Conference, pp.16. 1999.

[32] Taylor, M.B. et al. “The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs”. IEEE Micro, Vol. 22, Issue 2, pp.25-35. 2002.

[33] Turner, J. “An optimal nonblocking multicast virtual circuit switch”. INFOCOM '94. Networking for Global Communications., 13th Proceedings IEEE, Vol. 1, pp.298-305. 1994.