

OrthoNoC: A Broadcast-Oriented Dual-Plane Wireless Network-on-Chip Architecture

Sergi Abadal, Josep Torrellas, Eduard Alarcón, and Albert Cabellos-Aparicio

Abstract—On-chip communication remains as a key research issue at the gates of the manycore era. In response to this, novel interconnect technologies have opened the door to new Network-on-Chip (NoC) solutions towards greater scalability and architectural flexibility. Particularly, wireless on-chip communication has garnered considerable attention due to its inherent broadcast capabilities, low latency, and system-level simplicity. This work presents ORTHONOC, a wired-wireless architecture that differs from existing proposals in that both network planes are decoupled and driven by traffic steering policies enforced at the network interfaces. With these and other design decisions, ORTHONOC seeks to emphasize the ordered broadcast advantage offered by the wireless technology. The performance and cost of ORTHONOC are first explored using synthetic traffic, showing substantial improvements with respect to other wired-wireless designs with a similar number of antennas. Then, the applicability of ORTHONOC in the multiprocessor scenario is demonstrated through the evaluation of a simple architecture that implements fast synchronization via ordered broadcast transmissions. Simulations reveal significant execution time speedups and communication energy savings for 64-threaded benchmarks, proving that the value of ORTHONOC goes beyond simply improving the performance of the on-chip interconnect.

Index Terms—Network-on-Chip, Wireless On-Chip Communication, Broadcast, Hybrid NoC, Manycore Processors.

1 INTRODUCTION

Network-on-Chip (NoC) is becoming the dominant paradigm for communication between the components of a chip multiprocessor [1]. NoCs were actually conceived to solve the scalability issues of buses, but the advent of the manycore era has brought up several new challenges that, again, limit the scalability of multiprocessor architectures. From a communications perspective, increasing the core density implies a significant increment in the intensity, variability, and heterogeneity of a load that, in turn, must be served with higher energy efficiency and placing a larger emphasis on latency [2]. From a system perspective, reconfigurability and simplicity become desirable attributes [3].

The emergence of novel interconnect technologies represents a potential solution to most of the problems of current NoCs. Nanophotonics [4] or Radio Frequency (RF) transmission lines [5] promise lower latency for global links, as well as higher bandwidth density and intrinsic energy efficiency than conventional wires. Wireless on-chip communication has been also in the spotlight since it provides inherent broadcast capabilities, while being non-intrusive and more flexible than wired options [6], [7]. These features may compensate for the theoretically higher power consumption and lower bandwidth of this wireless technology with respect to other emerging alternatives.

This work focuses on the applicability of wireless on-chip communication in the context of manycore processors. A plethora of works have proposed to use this technology for the implementation of a set of long-range links over a

conventional NoC [8], [9], [10], [11]. This Wireless Network-on-Chip (WNoC) approach leverages the latency properties of wireless on-chip communication, attaining impressive performance and energy efficiency improvements with respect to traditional NoCs. However, it remains unclear whether this approach will be able to compete with the nanophotonics technology, which is expected to deliver even faster and more efficient chip-scale point-to-point (unicast) links in light of recent experimental advances [12].

Although the WNoC paradigm is uniquely suited to the broadcast of data, few works have explored such possibility [13]. An effective broadcast platform is desirable in many-core environments, but costly to implement due to either issues related to the routed nature of NoCs [14], the design complexity of RF interconnects [15], or the laser power constraints and the network-level complexity of nanophotonics [16]. In WNoCs, instead, implementing a broadcast plane can be performed by simply tuning all on-chip antennas to the same transmission frequency.

This paper presents ORTHONOC, a hybrid wired-wireless architecture that aims to make the most of the inherent broadcast capabilities of the wireless side. To this end, and unlike the existing hybrid NoCs, our proposal considers two independent network planes. The wireless plane is designed to minimize latency and provide ordered broadcast, whereas the wired plane is oriented to unicast traffic. Interaction between planes is not performed at the routers, but at the network interface. There, a hybrid controller coordinates the action of the wired and wireless interfaces by means of a policy easy to reconfigure at runtime. These decisions aim to keep ORTHONOC simple, flexible, and applicable over any wired topology.

ORTHONOC is evaluated in a wide variety of configurations to demonstrate the benefits of its dual-plane and broadcast-oriented approach. First, we compare OR-

- *Sergi Abadal, Eduard Alarcón and Albert Cabellos-Aparicio are with the NaNoNetworking Center in Catalonia (N3Cat), Universitat Politècnica de Catalunya, Barcelona, Spain. E-mail: abadal@ac.upc.edu*
- *José Torrellas is with the Department of Computer Science at the University of Illinois at Urbana-Champaign.*

THONOC with representative hybrid alternatives considering synthetic traffic with variable levels of broadcast. We observe that, in other NoCs, broadcasts generate a throughput bottleneck at the ejection links, i.e., those connecting the routers with the network interfaces [17]. ORTHONOC alleviates this bottleneck by increasing the bandwidth at the network edges, leading to not only a significant reduction of the energy and latency, but also a boost of the network throughput. Second, we integrate our hybrid network in a multiprocessor architecture suited to the ordered broadcast capabilities of the wireless plane [18]. By attaining significant execution speedups and energy savings in a wide set of benchmarks, we prove that ORTHONOC could make manycores faster and easier to scale.

The remainder of the paper is as follows. Section 2 refers to the importance of broadcast in manycore processors and motivates our design approach. Section 3 provides an overview of ORTHONOC, whereas Sections 4 and 5 detail the main design decisions and cost models. Then, Sections 6 and 7 evaluate two architecture-agnostic and architecture-oriented instances of ORTHONOC. Section 8 summarizes related works in the area and Section 9 concludes the paper.

2 MOTIVATION

On the Importance of Multicast and Broadcast. Ever since buses gave way to the NoC paradigm, architects have tried to avoid multicast and broadcast as these traffic patterns are highly suboptimal in NoCs. For instance, cache coherence is currently implemented via directory-based schemes that limit the use of multicast to the invalidation of cache blocks on a shared write. However, coherence transactions become more frequent and involve larger destination sets when scaling parallel programs [19], [20]. Scaling directory-based protocols is not an easy task either, as they gradually become slower, bigger, and harder to verify. To avoid this, some schemes eliminate the restrictions imposed by the directory and make intensive use of broadcast instead [14], [21], [22].

Figure 1(a) exemplifies this evolution by plotting the scaling trend of the multicast intensity in SPLASH-2 and PARSEC benchmarks [23]. The plots assume one directory-based (MESI) and two broadcast-based coherence schemes (HT, TokenB) over a typical L1-L2 hierarchy. MESI maintains a low multicast intensity, but at 64 cores becomes slow and cumbersome. On the other hand, HT and TokenB inject a significant amount of broadcast messages which, as shown in Fig. 1(b), become up to 80% of the overall traffic due to flit replication. Some of this traffic is in the critical path of the processor and, therefore, substantial execution speed-ups can be obtained if served well. Krishna *et al.* demonstrate an average gain of 12% (max 40%) for SPLASH-2 and PARSEC in a 64-core system with HT or TokenB [24]. Even better results can be expected at higher core counts.

Besides cache coherence, other functionalities are affected by the lack of an efficient broadcast platform as widely discussed in [6]. Thread synchronization has become expensive by default and can degrade performance of applications by a 40% in average albeit representing a small fraction of the code [18]. In message passing, widely employed collective primitives such as `MPI_Allgather` or `MPI_Allreduce` use multicast. Some novel programming

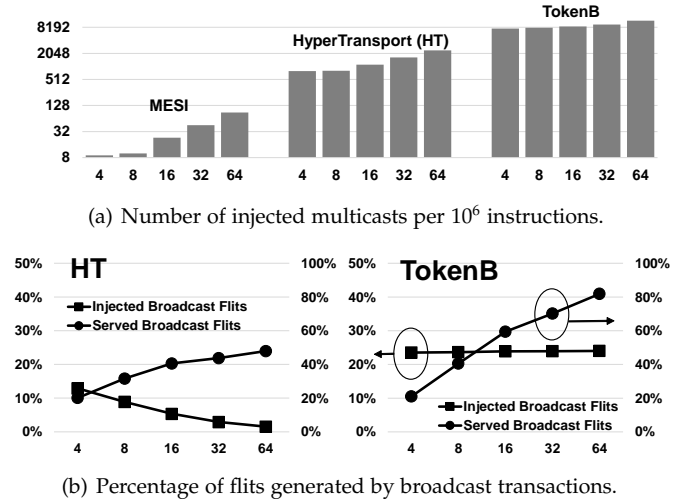


Fig. 1. Multicast and broadcast traffic as a function of the number of cores for three coherence schemes. We assume 32KB private I&D L1 caches and a 512KB slice of shared L2 per core, with 64B lines. We take the gmean over all SPLASH-2 and PARSEC benchmarks.

models and computing paradigms are also multicast-driven, e.g., neuromorphic architectures communicate among their cores through multicast *spike* messages [25].

Broadcast in Conventional NoCs. Current NoC designs use path-based and tree-based routing for efficient packet replication and multicasting [26]. Numerous optimization proposals [17], [27], [28] have provided important performance improvements, but have still left several issues to overcome. First, there is a fundamental tradeoff between the diameter of a NoC, which determines the broadcast latency, and the implementation cost of the required links and routers. Second, bursts of broadcast messages *flood* the entire network, reducing its performance also for unicast traffic. Third, the unordered nature of NoCs forces architects to devote additional resources [14] to guaranteeing a consistent view of the order of delivery of broadcasts, if required.

Broadcast via a Globally Shared Medium. One possible solution to the aforementioned issues would be to employ shared-medium schemes, which are ideally suited to serve broadcasts and global traffic, yet inefficient for local unicast transmissions. Overlaying such a network over any wired topology would not only provide better support for multicasts, but also offload the main NoC. This would increase the performance and efficiency of unicasts as well. Moreover, cores would see the same order of delivery if they all share the same medium, helping to reduce the complexity of the underlying architecture. This way, multiprocessors would be faster and easier to scale.

The main hurdle preventing the use of globally shared-medium schemes within manycore processors is scalability. Conventional buses were already discarded due to this same reason, and now the use of emerging interconnect technologies has been suggested instead. As mentioned above, several reasons discourage the use of RF interconnects or nanophotonics for broadcast in manycores, exactly where wireless on-chip communication shows unique promise.

Potential of a Dual-Plane Approach. Figure 2 illustrates the potential gains of combining a mesh NoC for unicasts with a globally shared medium for broadcasts. We use the models

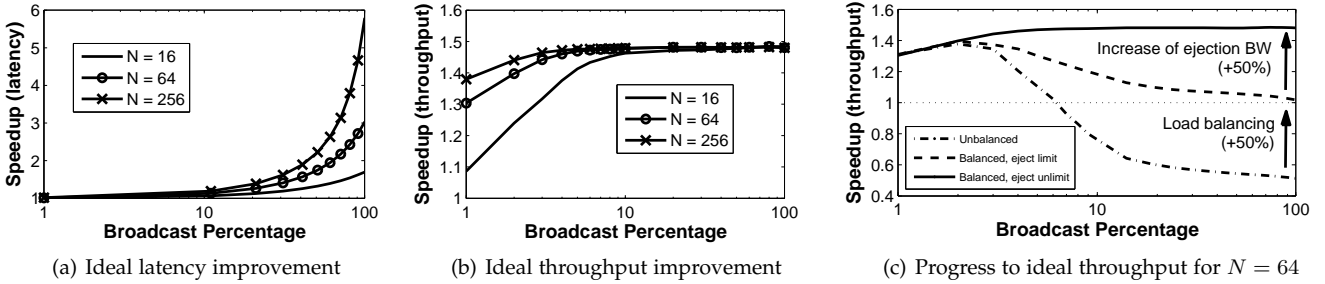


Fig. 2. Potential latency and throughput improvements of a mesh augmented with a globally shared medium for different system sizes N .

from [17] to calculate the latency and throughput limits of each network and then evaluate the speedups. In the hybrid case, we consider that the shared medium has a capacity of half flit per cycle. Even with this moderate bandwidth, Fig. 2(a) shows a reduction of the latency proportional to the system size N and the broadcast percentage β , from $\sim 15\%$ for $\beta = 10\%$ to a maximum of $5\times$. Fig. 2(b) shows that the throughput increases with β , yielding improvements of up to 40% already at $\beta = 1\%$. Against more costly high-radix topologies, a hybrid NoC would maintain the throughput advantage, but with lower latency speedups.

Two issues need to be overcome in order to fully exploit the throughput advantage of the hybrid approach. To exemplify them, Figure 2(c) plots the throughput improvement for 64 cores in three different scenarios. First, we assume that both the wired and wireless networks operate in isolation. Then, we consider perfect load balancing with either limited and unlimited bandwidth at the ejection links.

In the unbalanced case, the wireless plane can only offload the mesh up to a given broadcast percentage β_1 ($\beta_1 \approx 2\%$ for $N = 64$). Beyond β_1 , the wireless network saturates and the mesh becomes underutilized, up to a point where the overall throughput may even drop. Balancing the load alleviates this problem, but the throughput improvement still decreases after β_1 because the network cannot eject the broadcast flits fast enough. This is the case for most existing wired-wireless architectures, which integrate wireless interfaces at selected routers and use adaptive routing [10], [11], [29]: adaptive routing balances the load, but integration at the router level leaves the ejection links unchanged. We thus observe that increasing the bandwidth at the ejection links while balancing the load is necessary to maintain the throughput advantage beyond β_1 . This can be achieved by bringing the network planes as close to the computing tiles as possible.

3 OVERVIEW OF ORTHONOC

Orthos is a two-headed dog belonging to the greek mythology. ORTHONOC is named after this legendary creature since it is basically composed of two independent network planes or *heads*, both driven by a unique traffic steering policy that embodies the *core* of the architecture. *Ortho* is also a greek prefix often used to express uncorrelation between two variables or, in our case, two network planes.

Figure 3 pictorially represents the main idea of ORTHONOC. Each computing tile contains a number of processing cores with their respective instruction and data L1 caches, a slice of the shared L2 cache, other memory, and a Hybrid Network Interface (HNIF). The HNIF connects

the tile to a *wireless plane* by means of a transceiver and an antenna, and to a *wired plane* by means of a local router. Each network plane deals with a subset of the on-chip communication demands through its own Network Interface (NIF).

ORTHONOC leverages the unique properties of wireless on-chip communication by using the wireless plane mainly to transmit broadcast traffic. In the interest of simplicity, flexibility, and to provide full broadcast support, all wireless interfaces are tuned to a single set of broadband channels. Currently, CMOS millimeter-wave (mmWave) technologies (60–300 GHz) have shown to provide reasonable bandwidth density (~ 64 Gb/s/mm²) and power efficiency (~ 2 pJ/bit) [30], [31]. We will demonstrate that this is enough to provide compelling performance gains even if wireless interfaces are shared by a few computing cores. As CMOS technologies evolve and alternative technologies such as BiCMOS or graphene come into play, one can ultimately envisage integration on a per-core basis and aggregated speeds in the order of 100 Gb/s [16], [32], [33], [34].

The wireless plane is a natural complement to the wired plane, which achieves high throughput and moderate latency in the presence of unicast and local traffic. We initially consider a conventional mesh for its implementation given its scalability, regularity and simplicity. Note, however, that the principles of ORTHONOC can be applied over any topology and interconnect technology. For instance, longer term designs can employ a nanophotonic network capable of serving unicast and local traffic with outstanding power efficiency and bandwidth density [4], [35].

The main difference of ORTHONOC with respect to other hybrid proposals is the relative independence between network planes. This implies that a message will rarely switch between planes during its time-to-live. A traffic steering policy, enforced by the HNIF, determines the plane through which a message will be sent. This policy can be simple or complex, fixed or determined at runtime, and

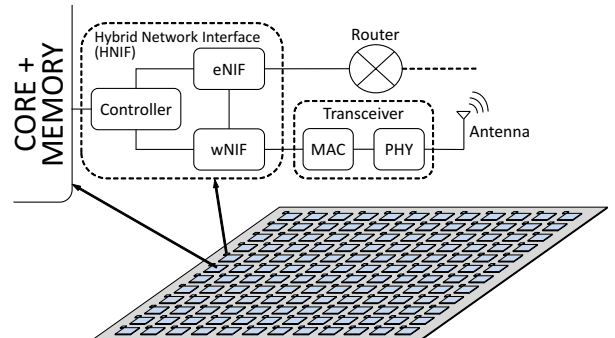


Fig. 3. Schematic representation of ORTHONOC with 144 cores.

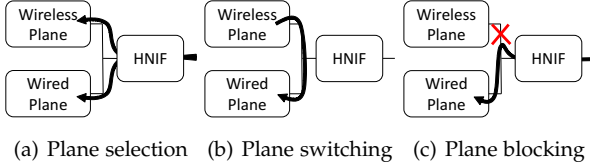


Fig. 4. Methods for load balancing in ORTHONOC.

agnostic or aware of the underlying multiprocessor architecture. In any case, the controller needs to be aware of the strengths and limitations of the wireless plane, which can become a performance and efficiency bottleneck if not used judiciously. Here, we consider a policy that distinguishes between unicast and broadcast traffic, to then provide an extensive justification of such choice in the evaluations.

Most of the design decisions of ORTHONOC seek to emphasize the system-level simplicity and natural broadcast capabilities of wireless communication. By separating both network planes, ORTHONOC simplifies the reasoning of aspects such as the network concentration, the routing protocol, or the dimensioning of buffers at the wireless interfaces. ORTHONOC also provides an opportunity to increase the broadcast throughput *over any hybrid architecture* by boosting the bandwidth at the network edges. Finally, by allowing all tiles to share the same set of channels, ORTHONOC provides fast broadcast with consistent order of delivery. This implies that three concurrent transmissions x, y, z by different processors can result in different interleavings, but all processors will observe the same interleaving (e.g. y, z, x). Consistent ordering is desirable in manycore processors as it allows to maintain certain memory consistency semantics, thereby simplifying the underlying architecture [6], [18].

Flexibility is another important facet of ORTHONOC. Adaptivity is of critical importance in the manycore era, where dark silicon constraints may force certain parts of the chip to be powered off and where multiprogramming workloads introduce high traffic variability. At the wireless plane level, ORTHONOC attains such flexibility by employing a Medium Access Control (MAC) protocol that naturally adapts to changes in the injection load and does not need to be reconfigured if a group of wireless interfaces is powered off. At the network level, flexibility is attained via three load balancing mechanisms (see Fig. 4):

- **Plane Selection:** directs packets to the appropriate network plane.
- **Plane Switching:** allows packets to change planes when they are heavily delayed.
- **Plane Blocking:** avoids packets to enter a heavily congested network plane.

These mechanisms are similar to those of congestion-aware routing [29], but with two particularities: in ORTHONOC, they are implemented at the network interface and can be easily coordinated globally as all nodes have the exact same view of the events happening in the wireless plane.

4 DESIGN DECISIONS

The design process of ORTHONOC requires addressing several issues present at different abstraction layers. Here, we detail a selection of them using a top-down approach.

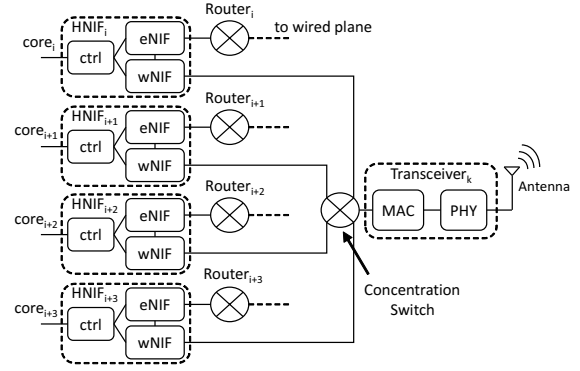


Fig. 5. Example of asymmetric concentration in ORTHONOC: Ort_1^4 or 4-way concentration in the wired plane and 1-way concentration in the wireless plane only.

4.1 Tile Architecture and Network Concentration

As shown in Fig. 3, ORTHONOC employs a tiled organization. In this work we consider that each tile is composed by one processor core with private 32-kB instruction and data caches, and a 512-kB bank of distributed L2 cache. Additionally, for the evaluation of an architecture-oriented application of ORTHONOC, we will include a small piece of memory called *Broadcast Memory* [18].

Although concentration or heterogeneity can be applied at the processor side, we consider concentration to be performed at the network side. As exemplified in Fig. 5, the dual-plane structure of ORTHONOC allows to adopt asymmetric schemes, i.e. each network plane has a different degree of concentration, to better adapt to the general communication requirements of a given multiprocessor architecture. In the wired plane, concentration is achieved by increasing the radix of the local router; whereas in the wireless plane, we use a *concentration switch* to connect multiple cores to a single wireless transceiver. The concentration switch operates independently of the wired plane routers, arbitrating access to the transceiver in transmission and driving messages to HNIFs in reception.

In this work, we will explore different concentration configurations to provide a fair comparison with other wired-wireless architectures. Ort_i^j will denote ORTHONOC with i -way concentration in the wired plane and j -way concentration in the wireless plane.

4.2 Hybrid Controller

As shown in Fig. 3, cores are connected to the network through a HNIF composed of a hybrid controller and two network interfaces. The hybrid controller determines through which plane a message will be sent, a decision with a major impact on the performance of ORTHONOC.

Figure 6 sketches the generic design of a hybrid controller for ORTHONOC with support for *plane selection* and *plane blocking*. In transmission, the controller receives data from the processor tile and parses its contents to extract the header, which contains data that will drive a first decision on the network plane to use. In this example, the *message type* is the field of interest. The comparison of its value with a given condition generates a selection signal sel which, together with the *block* signal, feeds the demux guiding the

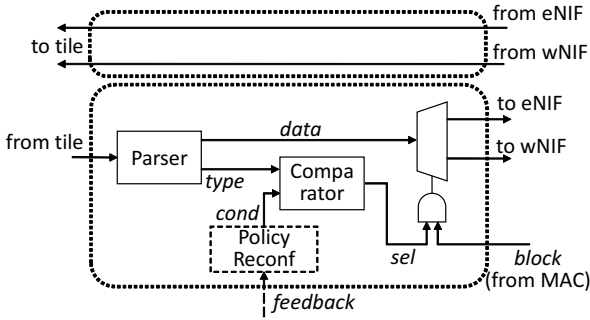


Fig. 6. Schematic representation of a hybrid controller implementing plane selection and blocking.

data to the appropriate network plane. Next, we elaborate on the plane selection and blocking functions.

Plane Selection: the process of choosing the network plane can follow a static or dynamic policy. In the latter case, the controller can receive *feedback* signals from other components, e.g. flow control messages or MAC queue information, to reconfigure the condition that drives the selection. In this work, we will distinguish between two broad types of plane selection policies.

On the one hand, *network-oriented* or *architecture-agnostic* controllers base their decisions on the characteristics of the message, the network plane, or the load [15]. In Section 6, we evaluate a simple network-agnostic broadcast policy: a message is sent through the wireless plane if it is broadcast, or through the wired plane otherwise. This way, the natural broadcast capabilities of WNoC are exploited.

On the other hand, *architecture-aware* controllers are co-designed with the architecture to optimize traffic steering. In Section 7, we evaluate a policy to speed up thread synchronization, which generally involves significant amounts of global communication [18]. In essence, messages related to synchronization are sent through the wireless plane, whereas the rest is sent through the wired plane. In this particular case, the consistent ordering delivered by the wireless plane allows synchronization variables to bypass the L1-L2 hierarchy, speeding up execution [18].

Plane Blocking: can be used when one of the planes suffers from congestion, in which case the controller temporarily deflects all packets towards the uncongested plane. To implement this variant of congestion-aware routing, our controller employs a *block* signal that comes from the MAC module and forces all packets to go through the wired plane. A similar mechanism could be employed in the reverse direction; however, the low bandwidth of the wireless plane discourages its use. Finally, note that plane blocking should not be used whenever consistent ordering is required among broadcast messages.

4.3 Network Interfaces

After the hybrid controller, the HNIF includes an interface for each network plane. While conventional NIF designs can be adopted for the wired plane, the interface of the wireless plane has a few peculiarities. Figure 7 shows the schematic representation of the wireless NIF employed in ORTHONOC. In transmission, the source and destination

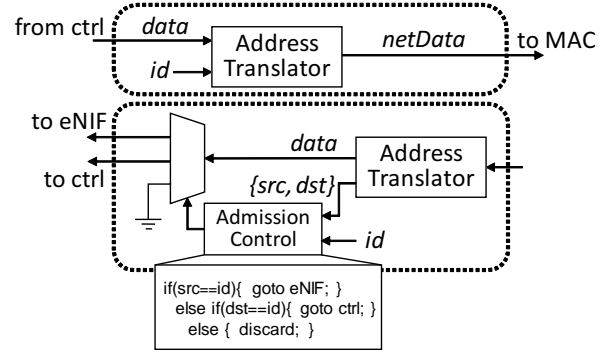


Fig. 7. Schematic representation of the NIF for the wireless plane, including pseudocode for the admission control module.

addresses are translated and attached to the outgoing data. In reception, the NIF implements two functions: admission control and plane switching. The former is necessary since all wireless messages reach all NIFs and is executed by comparing the source and destination addresses with the *id* of the destination NIF. If the destination addresses match, the packet is sent to the controller; otherwise, the packet is discarded. In our design, there is an exception to this rule, which is used to implement plane switching.

Plane switching: can be used when, due to sudden bursts of traffic, packets suffer large delays in the wireless plane. In that case, the MAC module returns queued packets to the NIF so that they can be sent through the wired plane. To this end, the NIF compares the source address with the local address and directs the message to the wired plane if the addresses match. Since plane switching may cause unordered delivery, it should not be used for messages requiring consistent ordering.

4.4 Channelization and RF Planning

Recent works have discussed the availability of multiple frequency channels for WNoCs [30]. In most wired-wireless architectures, these channels are used to implement orthogonal links between distant cores [7], [8], [29], [36]. Even if implemented with multiple channels, ORTHONOC considers a single broadband link shared by all cores instead. This way, a node uses all the wireless resources to broadcast a message when it gains access to the medium. Under this condition, the wireless network becomes an ordering point that guarantees that all processors will see the same order of delivery for concurrent transmissions.

Having multiple channels could also be interesting from an architectural perspective to implement multiple broadcast domains, which may be required to accommodate either multiple applications mapped within the same processor, or different components within the same system (e.g. CPU-GPU). However, this is out of the scope of this work.

4.5 Medium Access Control

The MAC mechanism plays a crucial role in any WNoC. Related works generally resort to contention-free mechanisms via multiplexing or variants of the popular token-passing protocol [7], [8], [36]. These methods do not scale well with the number of participating nodes due to the high cost of introducing new channels or the increase of the token round

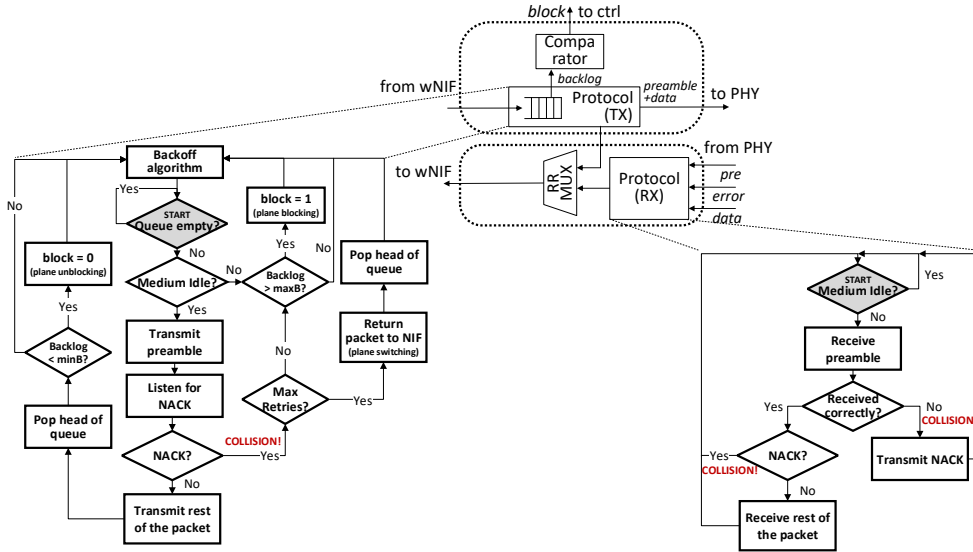


Fig. 8. Schematic representation of the MAC module, including the flowchart of the BRS-MAC protocol with support for load balancing.

trip time [37]. Better scalability is achieved with the protocol used in [29], where nodes request access by broadcasting short orthogonal request packets. However, this protocol is based on impulse radio techniques for which mmWave integrated solutions have not been explored.

ORTHONOC aims to maintain the broadcast advantage of WNoC in manycore environments, this is, even for a large number of wireless interfaces and highly variable communication patterns. For this, we adopt a family of protocols that let nodes to contend for the channel and resolve collisions in a distributed manner. The reason is that such contention-based protocols are generally more scalable than the contention-free alternatives and naturally adapt to hotspot traffic and other variations [37], as well as to changes in the number of available wireless interfaces. BRS-MAC [38], the protocol employed in ORTHONOC, maintains these advantages and minimizes the penalty of collisions via three techniques: preamble transmission, collision detection, and scalable acknowledging. With small modifications, ORTHONOC augments BRS-MAC with support for plane blocking and switching.

Basic algorithm: Figure 8 summarizes the BRS-MAC algorithm. Let us assume that the channel is slotted at the processor clock granularity. From the perspective of a transmitter T , the protocol works as follows: when T is ready to send data, it senses the channel. If busy, the node backs off and keeps checking until the medium is expected to be free. At that point, T transmits a fraction of the packet (i.e. the preamble) and then listens to the medium to check if there was a collision during the preamble transmission. If so, T and the rest of colliding nodes abort the transmission and try again later. Otherwise, the next cycles are used to send the rest of the message with guaranteed no collision. In this way, the penalty of a collision is reduced from the full transmission time to a preamble transmission time.

From the perspective of a receiver R , the algorithm is the following. R receives a preamble pre together with an error bit. If the error bit is set, R will notify the collision by sending a Negative ACKnowledgment (NACK) and will discard the preamble. Otherwise, R waits for either a collision notifi-

cation from another node, in which case the preamble is discarded; or the remainder of the ongoing transmission, in which case the full message is forwarded to the NIF.

BRS-MAC uses the well-known exponential backoff algorithm, which sets a waiting period proportionally to the number of collisions to maximize the network utilization. The range is set between 0 and $2^i - 1$ cycles, where i is updated after every collision or successful transmission. For increased fairness, the backoff counter is attached to messages so that it can be shared among all cores. We refer the reader to [38] for more details on the transmission, backoff, and acknowledgment policies.

Support for load balancing: Figure 8 shows a sketch of the MAC module of ORTHONOC which, besides integrating the protocol, provides support for the plane blocking and switching. Both mechanisms modulate the pressure applied to the wireless plane, minimizing the energy and time wasted on collisions. On the one hand, plane blocking is performed by means of the *block* signal, which originates at the queue of the MAC protocol: the blocking is set/lifted when the backlog of the queue is higher/lower than predefined thresholds with hysteresis, which indicates presence or absence of contention. On the other hand, plane switching is performed when a maximum number of retries is exceeded. To this end, the packet is simply popped out of the queue and sent back to the wireless NIF, which will act as defined in Sec. 4.3.

4.6 Physical Layer

At the physical layer of design, modulation and coding are two important design decisions. In consonance with other works in the area [9], [30], [36], we choose a simple modulation leading to the use of a transceiver with affordable power and area overheads. Yu *et al* presented an On-Off Keying (OOK) implementation in the 30–90 GHz range capable of providing up to 48 Gb/s [30], enough for the evaluations carried out in this paper. With technology advances, transceivers providing even larger bandwidths at 100–300 GHz bands are expected [16], [39]. An additional consideration is the bit error rate, for which most WNoC

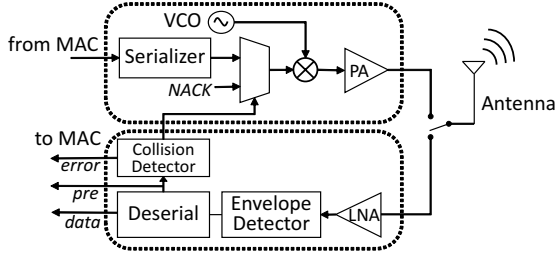


Fig. 9. Schematic representation of the PHY module.

works have assumed to be commensurate to that of a wire ($\sim 10^{-15}$) [8], [30]. With this, the need for additional error detection or correcting codes is prevented.

Figure 9 details the building blocks required to implement the OOK transceiver of ORTHONOC. The main novelty of this scheme is the added support for preamble-based collision detection and notification as per the MAC layer requirements. To this end, the receiver separately deserializes the preamble and the rest of data, whereas a collision detector checks the correctness of the preamble. Such verification can be performed via different simple methods as discussed in [38]. The resulting *error* signal drives the MAC protocol and triggers the transmission of a NACK signal to notify the collision.

4.7 Underlying wired NoC

We consider an aggressive mesh NoC with embedded multicast support as both the baseline network used for comparison and the wired plane of ORTHONOC. The choice of a mesh topology is backed up by its low radix, ease of layout, reasonable performance, and extensive use as baseline in the literature. Note, in any case, that the benefits of ORTHONOC are applicable to virtually any wired topology.

To provide a fair comparison with ORTHONOC, we consider a fast router microarchitecture with sophisticated multicast support. Routers are assumed to implement a two-stage pipeline with virtual bypass [17], which allows to minimize the routing latency in the absence of contention. To support multicast efficiently, the router is augmented with multiport switch allocation and a multicast crossbar [17], so that a flit can be simultaneously allocated and replicated in multiple outputs. The routing protocol is a wormhole, dimension-ordered XY with spanning tree for multicasts. Each router has 10 flit buffers shared among 6 Virtual Channels (VCs) and, unless noted, the datapath width is of 128 bits. The link delay is one cycle.

5 IMPLEMENTATION COST MODELS

Next sections evaluate both the performance and the implementation cost of ORTHONOC and of several alternatives. In the following, we detail the area and energy models employed to this end.

5.1 Area Models

In order to calculate the area overhead of ORTHONOC, we need to take into account all the components required to implement its wired and wireless planes.

Area of the wired plane: In a conventional NoC, the number of links and routers as well as their characteristics can be

easily inferred from the topology. Our area occupation estimates are directly based on the hardware implementation of a full-swing router with virtual bypass and multicast support presented in [17]. We use DSENT [40] to scale their area overhead figures to our design point, as well as to calculate the area of the links.

Area of the wireless plane: In ORTHONOC, we need to account for the area consumed by the antennas and the transceiver circuits. The number of such devices will depend on the core count and the concentration applied to the wireless plane. If there is concentration, we need to take into consideration the area of the switch devoted to both arbitrating access to the shared transceiver and driving received messages to the NIFs.

The area of the OOK transceiver is obtained by extrapolating numbers from the state-of-the-art hardware implementations for on-chip communication. The work in [30] reports two 65-nm designs with one and three channels, which operate at 16 Gb/s and 48 Gb/s while taking 0.25 mm² and 0.73 mm² of silicon area, respectively. Weissman *et al* describe a 65-nm design that provides 6 Gb/s with less than 0.1 mm² of area without the measurement pads [31]. In [36], the authors assume that a 40-nm transceiver operating at 32 Gb/s can have an area overhead in the range of 0.05–0.1 mm². With these figures and empirical scaling projections [16], [32], we obtain conservative estimates at different technology nodes. For the antenna, we use data from existing on-chip implementations at 60 GHz [41].

Summary: Table 1 shows the area of representative building blocks of ORTHONOC’s network planes for the design points assumed in this work. To contextualize these numbers, we also include the area taken by the L1 and L2 caches considered throughout this work. We use CACTI [42] to calculate their implementation cost.

5.2 Energy Models

To evaluate the bit energy of ORTHONOC, we need to consider the energy consumed in the wired plane for unicast flows and in the wireless plane for broadcast flows.

Energy of a wired transmission: the energy required to transmit a single bit through the wired plane depends on the average number of hops H as well as on the energy required to perform one hop E_{hop} as

$$E_{bit}^{wired} = H \cdot E_{hop} = H(E_{link} + E_{router}), \quad (1)$$

where E_{link} and E_{router} are the bit energies required to traverse a link and a router, respectively. On the one hand, we take the power consumption reported in the hardware implementation of [17] (~ 25 mW per router at $\sim 70\%$ of the maximum throughput) as an average for E_{hop} . Since current power modeling tools maintain relative accuracy, we use DSENT to scale their figures to our design points. On the other hand, evaluating the number of hops H requires knowledge on the logical distance between transmitter and receiver, which is determined by the topology, the message type, and the traffic pattern. For instance, $H_{ucast} = \frac{2k}{3}$ and $H_{bcast} = k^2 - 1$ for a $k \times k$ mesh and uniform random traffic.

Energy of a wireless transmission: our design spends energy in the wireless transceiver and, if any, in the concentration switch. To calculate the energy of the wireless

TABLE 1
Area of memory and network components (mm²).

Component	Area (45nm)	Area (22nm)
Wired Link (1 mm)	1.81·10 ⁻⁴	0.65·10 ⁻⁴
Router (5 ports)	0.394	0.095
Router (8 ports)	0.712	0.171
Concentration Switch (4-way)	0.038	0.009
Transc. + Antenna (64 Gb/s)	0.8	0.45
L1 (Inst+Data)	0.86	0.21
L2 slice	3.77	0.9

transceiver we need to take into consideration two peculiarities. First, the MAC protocol considered in this work achieves a very low latency at the cost of letting nodes to contend for the channel. Thus, collisions can occur and result in an energy waste. Remind that BRS-MAC protocol reduces the penalty of collisions by transmitting a preamble of size L_{pre} and then checking for collisions before continuing. To account for these effects, we have that

$$E_{bit}^{wless,B} = E_{OK}(1 + \frac{L_{pre}}{L} N_{re}), \quad (2)$$

where E_{OK} is the energy per bit of a successful transmission, N_{re} is the average number of retransmissions per successful transmission, and L is the average packet size. The preamble and packet lengths are known, whereas N_{re} is obtained via simulation or remains as a parameter.

The performance of ORTHONOC is compared against two hybrid alternatives where token passing is used in the wireless plane. Transmissions cannot collide in this case, but incur in a token passing overhead instead. Therefore, the energy in token passing networks is

$$E_{bit}^{wless,T} = E_{OK}(1 + \frac{L_{tok}}{L} H_{tok}), \quad (3)$$

where L_{tok} is the length of the token and H_{tok} is the average number of hops performed between transmissions. If N_{wi} is the number of wireless transceivers, $H_{tok} = N_{wi}/2$ at low loads and uniform random traffic.

To evaluate E_{OK} , we must take into consideration that all messages are received by all the transceivers tuned to the same frequency regardless of their intended destination. Therefore, the energy of a successful transmission is

$$E_{OK} = [E_{tx} + E_{conc,tx}] + (N_{wi} - 1)[E_{rx} + E_{conc,rx}], \quad (4)$$

where E_{tx} and E_{rx} are the energies consumed by the transmitting and receiving part of a transceiver, respectively, and N_{wi} is the number of active wireless transceivers. If the design applies concentration at the wireless network side, the switch between the HNIFs and the transceiver consumes an extra $E_{conc,tx}$ and $E_{conc,rx}$.

To calculate the bit energy consumed by the OOK design considered in this work, we use values from existing hardware implementations. In the literature, 65-nm transceivers with total circuit efficiencies of 7.3 pJ/bit [11], 1.95 pJ/bit [30], and 1.5 pJ/bit [31] can be found. Other works provide extensive discussions on the feasibility of more efficient transceivers implemented in 28 nm (1.3 pJ/bit, in [29]) or 22nm (≤ 1 pJ/bit, in [11], [32]). In light of these figures and applying empirical scaling projections [16], we obtain reasonable estimates at 45nm and 22nm. We assume the same energy consumption for any transceiver pair, even

TABLE 2
Energy consumption of network components (fJ/bit).

Component	Energy (45nm)	Energy (22nm)
5-Port Router Traversal	113	28
8-Port Router Traversal	121	31
Link Traversal (1 mm)	40	23
Concentration Switch (4-way)	70	18
Transceiver (TX+RX)	1650	1000

though power allocation could be performed on a per-transceiver basis to reduce the cost even further [43].

The energy values mentioned above include both the transmitter and the receiver. To derive E_{tx} and E_{rx} , we need specific ratios from the literature. For instance, the 65nm OOK transceiver design presented in [30] devotes 53% and 47% to power the transmitter and receiver, respectively. An alternative design given in [11] consistently yields a 59% to 41% ratio across different technology nodes. We adopt the latter as it is desirable to minimize the energy consumed at the $N_{wi} - 1$ receivers.

Summary: Table 2 shows the dynamic energy consumed by the components of ORTHONOC's network planes for the design points assumed throughout this work.

6 ARCHITECTURE-AGNOSTIC ORTHONOC: HYBRID DESIGN SPACE EXPLORATION

In this section, we explore the architecture-agnostic integration of a broadcast-oriented WNoC within a hybrid network. We target two design points, namely, a 64-core ORTHONOC implemented with 45nm technology, and a 256-core ORTHONOC implemented with 22nm technology. A representative fraction of the network-level design space is covered as we consider different degrees of concentration, system sizes, and percentages of broadcast traffic.

For the sake of comparison, we also consider two hybrid wired-wireless architectures from the literature: MWNOC [10] and HCWiNOC [11]. The choice aims to be representative of the works in this field, the majority of which integrate the wireless interfaces within the wired plane to implement a small-world or a hierarchical topology. To provide a fair comparison, ORTHONOC is evaluated in different flavors, including Ort_X^4 which has a similar number of antennas than MWNOC and HCWiNOC.

6.1 Evaluation Framework

We use PhoenixSim [44] to explore the effectiveness of architecture-agnostic configurations serving synthetic traffic. To this end, we augmented PhoenixSim with wireless communication modules and our HNIF design. Table 3 details the parameters and variables considered in the evaluation. Basically, we set the broadcast percentage as the main variable and assess the performance of the broadcast policy detailed in Section 4.2. We also measure its implementation cost with the methods explained in Section 5.

Evaluated Networks: The performance of ORTHONOC is compared with that of a conventional baseline NoC and two hybrid wired-wireless architectures. Hybrid networks have been modeled considering a fixed wireless bandwidth per channel and their routing has been optimized for broadcast traffic. Note that we do not link the origin of such traffic to

TABLE 3
Simulation Parameters for the Design Space Exploration

Common Parameters	
System Technology	20×20 mm ² die, 1 V, 1 GHz, 64/256 tiles CMOS 45nm for 64 tiles, 22nm for 256 tiles
Baseline NoC	
Topology	2D MESH, 128-bit links, 1-cycle delay
Routers	1-/2-cycle delay (bypass/no bypass), wormhole XY, 6 VCs, 10 flit buffers
Multicast	fixed tree routing, multiport allocation and crossbar
ORTHONOC Design	
Concentr. Controller	none/4-way wireless and/or wired Broadcast, 1-cycle delay
MAC	BRS-MAC (1-flit preamble, NACK burst, exp. back-off), max. 3 retries, 4-flit block, 2-flit unblock
PHY	Single broadcast domain, 2 cycles/flit (64 Gb/s)
Wired Net.	Baseline NoC with no changes
Workload Characteristics	
Arrivals	Poisson, Uniformly distributed source
Msg. Size	1 and 4 flits (same probability)
Hop Dist.	Uniform random for unicasts
Broadcast	0–100% (def: 0%)

(Explored variables are shown in bold)

any particular cache coherence mechanism. Next, we briefly describe the compared wired-wireless architectures.

MWNoC [10] communicates clusters of cores via a small-world topology. To form this topology, a selected set of the routers is augmented with a wireless interface so that the average hop distance is minimized. Since all the wireless interfaces are tuned to the same frequency band, MWNoC has increased broadcast capabilities. Access is arbitrated using a token-passing protocol. To model MWNoC, we consider six wireless interfaces for $N = 64$ and then scale the design by employing more wireless interfaces and keeping the degree of concentration at the clusters. Additionally, we optimistically assume a two-cycle delay and a quarter-flit wireless energy for each hop of the token.

HCWiNoC [11] is a regular wired-wireless architecture that augments a concentrated mesh. Specifically, each wireless interface is connected to a group of four adjacent routers. Each wireless interface is tuned to two frequency channels shared with all the interfaces within the same row and column, respectively. Therefore, HCWiNoC requires $\sqrt{N}/2$ frequency channels, which may be impractical for

high N . Arbitration among nodes within the same row or column is performed with token passing. The token passing overheads are the same than for MWNoC.

For the sake of fairness, we compare the baseline mesh against the versions of ORTHONOC without concentration in the wired plane. Then, we compare a concentrated mesh with the architectures that overlay a wireless plane over a concentrated mesh: MWNoC, HCWiNoC, Ort_4^1 and Ort_4^4 . Finally, note that nanophotonics have not been considered due to the difficulty of scaling optical broadcast topologies.

Evaluation metrics: On the one hand, we measure latency as the time passed between the generation of a message and its complete reception at *all* the intended destinations. For simplicity, we report the average communication latency for low loads, as this models the performance of the network for at least 40% of the maximum admitted load with high accuracy [17], [37]. On the other hand, the throughput accounts for the aggregate of both the unicast and broadcast flows and is measured from the transmitter perspective, i.e., messages with multiple receivers are only counted once. We report the maximum admitted throughput to assess the network performance at high loads.

6.2 Network Performance

Figure 10 shows how the network latency of the different architectures improves with respect that of a mesh for different broadcast intensities and system sizes. We observe that all the hybrid architectures introduce a certain latency improvement, but it is ORTHONOC without network concentration that achieves the best latency of all configurations, with speedups values close to the upper bound estimated in Fig. 2(a). Network concentration reduces the latency advantage of ORTHONOC. Finally, note the existence of a break-even point with respect to MWNoC and HCWiNoC that occurs around 5% and 20% for 256 cores. This difference is of a few cycles but, as we will see, comes at the cost of a huge increase in energy consumption.

Figure 11 shows the throughput improvement. All the hybrid architectures, even those based on concentrated meshes, outperform the mesh NoC for traffic with more than ~5% broadcast (a bit less for $N = 256$). Due to the use of

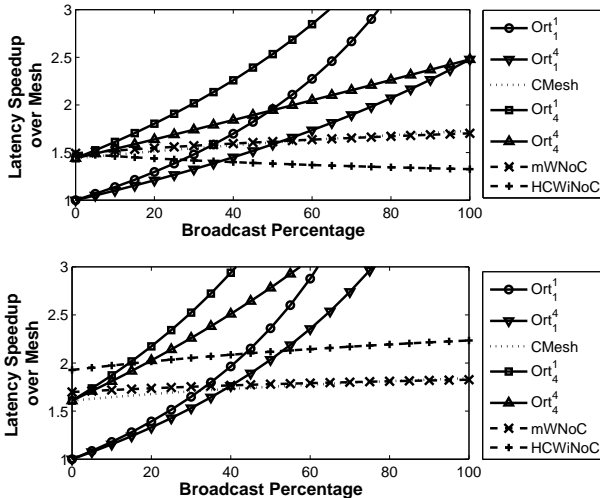


Fig. 10. Latency speedup with respect to a mesh as a function of the broadcast traffic percentage for $N = 64$ (top) and $N = 256$ (bottom).

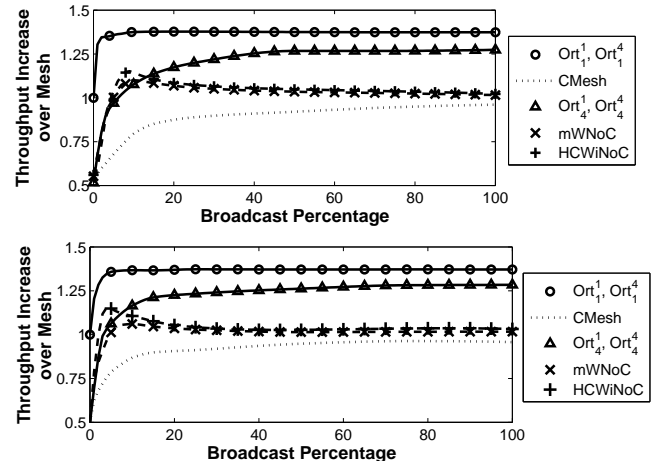


Fig. 11. Throughput improvement of the different architectures as a function of the broadcast traffic percentage for $N = 64$ (top) and $N = 256$ (bottom).

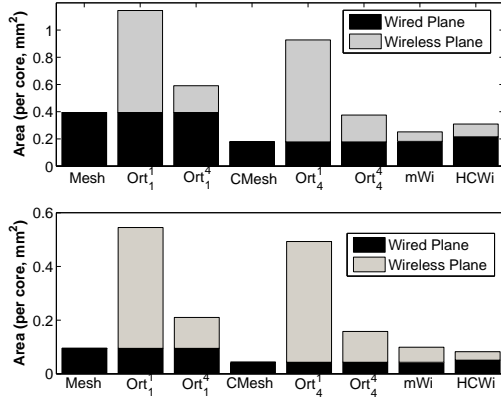


Fig. 12. Per-core area of the evaluated architectures for 64 cores (45nm, top) and 256 cores (22nm, bottom).

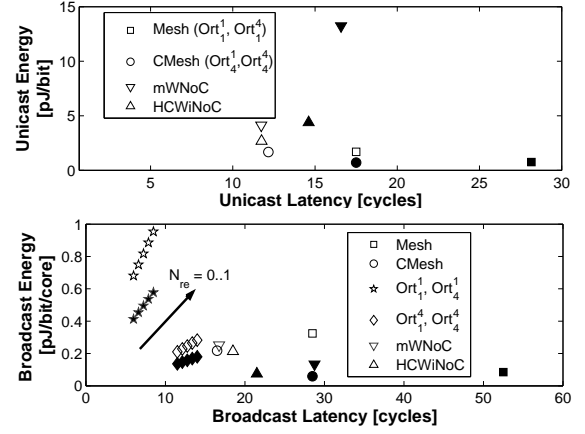
multiple wireless channels, HCWiNoC achieves a slightly better throughput also around 5% broadcast. However, from that point onwards, where the network is limited by the ejection links, MWNOC and HCWiNoC start losing the throughput advantage with respect to a regular mesh. In contrast, the dual-plane structure of ORTHONOC increases the bandwidth at the ejection points and, thus, sustains a 25% to 40% throughput improvement even with fully broadcast traffic. Note that these values are close to the upper bound estimated in Fig. 2(b).

6.3 Implementation Cost

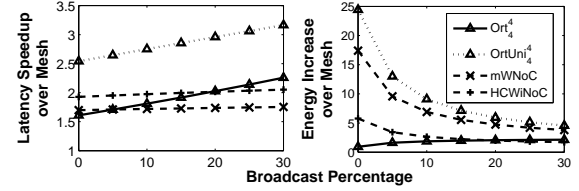
Figure 12 shows the per-core area required by the different networks. It is observed that, due to the conservative scaling rule employed for the transceiver, the contribution of the wireless plane to the total area becomes more significant at 22nm. A comparison between the wired-wireless architectures reveals that ORTHONOC is commensurate to MWNOC and HCWiNoC as long as it employs similar levels of concentration at the wireless plane. For instance, Ort_4^4 is only $1.2\times$ larger than MWNOC and still represents less than 10% than the tile area in a $20\times 20\text{mm}^2$ chip while providing substantial speedups as shown earlier.

Figure 13(a) explores the tradeoff between the energy and the latency of network transactions for the evaluated architectures. For unicasts, the top chart of Fig. 13(a) shows that MWNOC and HCWiNoC consume $1.5\text{--}20\times$ more energy than the other alternatives to achieve a latency gain of 20% at most. This is mainly because all tuned wireless receivers demodulate all messages regardless of their intended destinations, therefore wasting energy. In contrast, ORTHONOC keeps the energy consumption low by using the wired plane to transmit unicast messages.

For broadcasts, it is observed in the bottom chart of Fig. 13(a) that the increase of load (represented by means of the average number of retries N_{re}) causes an increment of ORTHONOC's latency and energy. The smallest latency is obtained with ORTHONOC without concentration, whereas the best energy efficiency is achieved with a concentrated mesh. Likewise to MWNOC and HCWiNoC, ORTHONOC with wireless plane concentration sacrifices performance to reduce the cost of broadcast transfers. However, with 4-way



(a) Latency and energy of unicast and broadcast transmissions for 64 nodes (45nm, white symbols) and 256 nodes (22nm, black symbols).



(b) Relative network latency and efficiency as functions of the broadcast percentage, including $OrtUni_4^4$.

Fig. 13. Performance–Efficiency tradeoff of the evaluated architectures.

concentration, ORTHONOC can still achieve a similar energy efficiency than MWNOC or HCWiNoC while being at least $\sim 30\%$ faster at zero load (15% for moderate load).

6.4 Discussion

The tradeoff observed in Figure 13(a) justifies the use of the wired plane of ORTHONOC to serve unicast messages. To explore this further, we modified the controller of Ort_4^4 so that long-range (≥ 5 hops) unicast messages are also transmitted through the wireless plane. We refer to this version of ORTHONOC as $OrtUni_4^4$. Figure 13(b) shows that $OrtUni_4^4$ exhibits a similar behavior than MWNOC and HCWiNoC: the use of the wireless plane cuts latency in half even for small broadcast percentages, but at the expense of consuming $5\times$ to $25\times$ more energy than the baseline mesh. In fact, the use of the wireless plane to serve long-range traffic becomes harder to justify as faster and more efficient router microarchitectures [17] and links [12] appear.

Finally, it is worth noting that most of the other existing wired-wireless architectures in the literature [8], [9], [29], [36] share most of the design principles with MWNOC and HCWiNoC: intertwining of both network planes, moderate number of antennas, and contention-free MAC. Therefore, their scalability trends will arguably be commensurate.

7 ARCHITECTURE-ORIENTED ORTHONOC: WISYNC

Synchronization in the form of locks and barriers is suited to the capabilities of ORTHONOC, as locks are oftentimes latency-sensitive and barriers generate global and broadcast traffic. To improve the support for them, we integrate

TABLE 4
Simulation Parameters for the Architecture-oriented Exploration

General Parameters	
Chip	20×20 mm ² die, 1 V, 1 GHz, 64 tiles, 22nm
Tiles	1 core/tile, private I&D L1s, bank of shared L2
L1 Cache	32KB, 2-way, 2-cycle, 64B lines
L2 Cache	512KB banks, 8-way, 6-cycle, 64B lines
Coherence	MOESI directory (+WiSync with ORTHONOC)
Main mem	4 controllers at the corners, 110-cycle latency
BMEM	16-kB, 2-cycle, 64-bit entries
Baseline NoC	
As described in Table 3	
ORTHONOC Design	
As described in Table 3. Changes: Controller: synch., MAC: 20-bit preamble, PHY: 1–20 Gb/s	
Workload Characteristics	
Benchmarks	SPLASH-2 (default set), PARSEC (<i>simsml</i> set)
Locks	Spinlock
Barriers	Centralized barrier

ORTHONOC within WiSync [18], a multiprocessor architecture that uses a small Broadcast Memory (BMEM) besides the main L1-L2 hierarchy. This piece of memory contains locks and barriers, which are kept coherent through a basic protocol that updates variables in all tiles on every write. This specific function can only be carried out through the wireless plane of ORTHONOC due to its unique broadcast and ordering consistency properties.

In this section, we compare the execution speed of WiSync with that of a conventional cache hierarchy without the BMEM. The comparison is performed for moderate-to-low values of the wireless capacity in order to show that the advantages of ORTHONOC go beyond simply improving the network performance metrics. Note that the original WiSync design considers a wireless speed of ~ 20 Gb/s. We refer the reader to [18] for more details.

7.1 Evaluation Framework

We employ the cycle-level execution-driven simulator Multi2sim [45] to test an architecture-aware implementation of ORTHONOC. In this case, the hybrid controller routes synchronization messages from the BMEM to the wireless plane, and from the cache hierarchy to the wired plane. Since it is crucial to guarantee that the wireless network acts as an ordering point, plane blocking and switching are deactivated. However, these functions are not necessary because the load injected to the wireless network is small in this particular architecture.

The entire architecture is modeled in Multi2sim with the parameters shown in Table 4. In this case, we run the entire SPLASH-2 [19] and PARSEC [20] benchmark suites. We implement spinlocks and centralized barriers for the baseline, and then adapt them to make use of the BMEM for ORTHONOC. Applications are run up to 10 times to reduce the time variability introduced by the MAC backoff mechanism. To evaluate the implementation cost, we use the methodology outlined in Section 5.

Evaluated Networks: The performance of ORTHONOC+WiSync is compared with that of a conventional architecture with MOESI coherence and a baseline NoC. mWNoC and HCWiNoC cannot be included in this

analysis since they do not guarantee a consistent ordering in the delivery of broadcast messages.

Evaluation Metrics: we measure the execution time of the parallel section of the different applications for the two architectures, to then calculate the relative speedup. We also measure the average delay in accessing synchronization variables as well as the effective use of the wireless channel.

7.2 Application Speedup

Figure 14(a) shows the speedup in terms of execution time of the architecture based on ORTHONOC, which is of 25% in average. This is a significant speedup considering that the baseline architecture employs a very aggressive NoC; the analysis in [18] reveals that by considering a router pipeline design without bypassing and a routing latency of three cycles, the average speedup is increased up to a 39%.

Barrier-intensive applications like *ocean* or *streamcluster* show the highest speedup, whereas improvements in lock-intensive applications depend on the case. To cite some examples, *radiosity* shows significant improvements, while *bodytrack* does not due to the large amount of calculations between locks. A particular case is that of *fluidanimate*, which uses a huge array of locks that does not fit within the small BMEM and, therefore, scarcely uses the wireless plane.

The speedups are quite consistent as long as the data rate is over 2 Gb/s. At this design point, the latency of the wireless plane would double (at best) that of the wired plane. Yet still, the speedups are maintained because of the delays introduced by the coherence protocol. In the baseline, misses to synchronization variables go through the cache hierarchy in a process that often takes several communication transactions; whereas ORTHONOC simplifies coherence procedures, achieving much lower access latencies. To illustrate this, Figure 14(b) shows the speedup of accessing synchronization variables. The latency of stores is reduced up to two orders of magnitude, whereas loads are much more frequent and are sped up by 4–6 \times .

A final comment is that the speedups are obtained by using the wireless channel less than 1% of the time in average (maximum $\sim 3\%$, in *streamcluster*). This is because WiSync targets latency-critical traffic that often stands in the critical path of the processor and, hence, has a large influence on the execution speed. Given such a low throughput requirements, plane switching and blocking can be safely deactivated to enforce the ordering constraint needed to implement WiSync. This also implies that there is a large room for improvement in applications or architectures that may make a more intensive use of the wireless plane.

7.3 Implementation Cost

In light of the results shown above, WiSync can achieve substantial performance gains with simple antennas and transceivers. This has an impact on the area occupied by the transceiver, which can be reduced from the 0.45 mm² previously assumed for 64 Gb/s to a conservative value of 0.12 mm² for 16 Gb/s. The energy is maintained at 1 pJ/bit.

Figure 15 shows the tile area breakdown assuming the use of energy-efficient Atom Silvermont cores and of simple transceivers. At 22nm, Atom Silvermont processors take an area of approximately 2 mm² without accounting for the

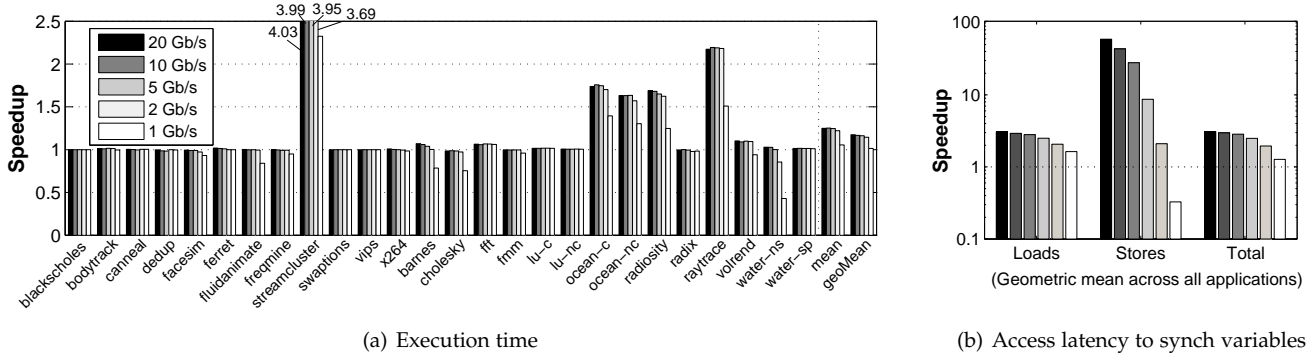


Fig. 14. Speedups of ORTHONOC+WiSync running PARSEC and SPLASH-2 for different wireless data rates.

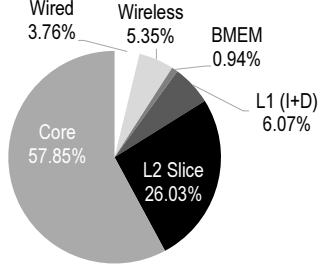


Fig. 15. Area breakdown of WiSync tiles at 22nm.

last-level caches [46]. We then use data from Table 1 to calculate the area of the different caches and of the wired and wireless communication planes, including the NIFs. It is worth noting that the BMEM and the wireless interface take $\sim 6\%$ of the tile area.

Figure 16 plots the total communication energy that WiSync saves with respect to the baseline per each application. In average, WiSync consumes 33% less communication energy than the baseline. Most applications save energy because, in WiSync, many data races occurring in the L1–L2 hierarchy to update synchronization variables are avoided, eliminating unnecessary transmissions. With its many barriers, *streamcluster* is a notable example of this, as it only requires the transmission of one tenth of the messages and consumes $4.7\times$ less than the baseline. Out of the 26 evaluated benchmarks, only 5 consume more energy mainly due to the extra cost of wireless transmissions.

8 RELATED WORK

The importance of broadcast in manycore architectures, highlighted in Sec. 2, has motivated a large body of research in NoCs with conventional and/or emerging interconnects.

Multicast in wired NoCs: improving multicast support in conventional NoCs has been addressed at different levels. At the router microarchitecture, optimizations in the switch allocation and traversal stages allow serving the same flit to multiple outputs within the same clock cycle, thereby reducing the hop delay [28]. At the routing level, different works have proposed adaptive routing techniques to increase the saturation throughput of both path-based [47] and tree-based multicast [24]. At the system level, high-radix topologies have been inspected in works seeking to reduce the network diameter [27], [48] for all types of traffic. Alternatively, Krishna *et al.* developed an asynchronous multihop approach that could provide broadcast capabilities

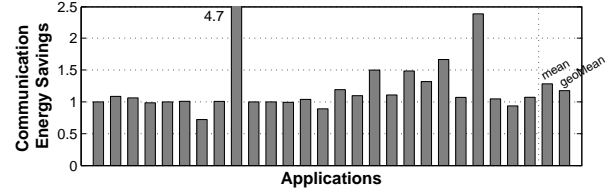


Fig. 16. Communication energy saved by WiSync at 22nm.

in as few as two hops [49]. These solutions have led to important improvements, but still suffer from scalability, ordering, and flooding issues that ORTHONOC can address.

Wireless RF: most hybrid architectures integrate a set of wireless links within the topology, either following a hierarchical approach [8], [11], [29], [36] or small-world principles [9], [10] aiming to reduce the diameter of the network. These proposals are not oriented to broadcast, which has been shown to be a strong advantage of WNoC in manycores [37]. In spite of this, a few works have discussed evaluated architectures with multicast capabilities. In [10], the authors tune all wireless units of their hybrid network to the same frequency and provide a brief evaluation of the resulting broadcast performance. Duraisamy *et al.* present a multicast-aware architecture with network coding [29]. In both cases, however, the network remains unordered and limited by the bandwidth of the ejection links.

Transmission Lines (TL): the transmission of EM waves through integrated TLs maintains the latency and broadcast advantages of the wireless approach, yet with higher efficiency and bandwidth density since waves are guided rather than radiated [5]. The use of TLs has been thus far limited to hybrid networks similar to those proposed for wireless [50]. Additionally, the use of TLs to distribute broadcast signals has been inspected in [15]. However, the scalability of the approach is compromised by the presence of signal reflections within the TL. To combat this, it is recommended to not include more than a few inlets and outlets per TL segment and use amplifiers to connect different segments. This is energy consuming and introduces additional design constraints.

Nanophotonics: modulation and transmission of light through on-chip waveguides augments the advantages of TLs with even higher intrinsic energy efficiency and outstanding bandwidth density [4]. Although integration within a dual-core computing system has been recently demonstrated [12], applying this technology in a manycore

scenario is still quite challenging. Also, the broadcast scalability of the approach is highly questionable due to laser power scaling issues [16]. Instead, existing hybrid proposals overlay a nanophotonic bus or crossbar over a conventional mesh [51], or over a wireless network [35] to leverage the unique properties of optics. Note, in any case, that the design principles of ORTHONOC can be applied to any combination of interconnect technologies.

9 CONCLUSION

This work has presented ORTHONOC, a hybrid wired-wireless architecture composed of two independent network planes driven by a hybrid controller that can be agnostic or aware of the architecture. With the architecture-agnostic approach, ORTHONOC achieves significant speedups over other hybrid NoCs by offloading the wired plane from traffic for which it is inefficient. Simulation results show up to 30% latency improvement, 25% throughput improvement, and higher energy efficiency with a similar number of wireless interfaces than other wired-wireless designs. With the architecture-aware approach, ORTHONOC's consistent order of delivery enables the design of faster and simpler multiprocessor architectures. The evaluation of ORTHONOC as an accelerator of thread synchronization for SPLASH-2 and PARSEC benchmarks yields, in average, an execution speedup of 25% and an energy saving of 33% with less than 5% of area overhead.

ACKNOWLEDGMENTS

This work was supported by the Catalan Government under grant 2014SGR-1427 and by the Spanish State Ministry of Economy and Competitiveness under grant PCIN-2015-012.

REFERENCES

- [1] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of Network-on-chip," *ACM Computing Surveys*, vol. 38, no. 1, pp. 1–51, 2006.
- [2] D. Sánchez, G. Michelogiannakis, and C. Kozyrakis, "An Analysis of On-Chip Interconnection Networks for Large-Scale Chip Multiprocessors," *ACM Transactions on Architecture and Code Optimization*, vol. 7, no. 1, p. Article 4, 2010.
- [3] D. Bertozzi, G. Dimitrakopoulos, J. Flich, and S. Sonntag, "The fast evolving landscape of on-chip communication," *Design Automation for Embedded Systems*, vol. 19, no. 1, pp. 59–76, 2015.
- [4] R. G. Beausoleil, P. J. Kuekes, G. S. Snider, S.-Y. Wang, and R. S. Williams, "Nanoelectronic and Nanophotonic Interconnect," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 230–247, 2008.
- [5] M.-C. F. Chang, V. Roychowdhury, L. Zhang, H. Shin, and Y. Qian, "RF/wireless interconnect for inter- and intra-chip communications," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 456–466, 2001.
- [6] S. Abadal, B. Sheinman, O. Katz, O. Markish, D. Elad, Y. Fournier, D. Roca, M. Hanzich, G. Houzeaux, M. Nemirovsky, E. Alarcón, and A. Cabellos-Aparicio, "Broadcast-Enabled Massive Multicore Architectures: A Wireless RF Approach," *IEEE MICRO*, vol. 35, no. 5, pp. 52–61, 2015.
- [7] S. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Wireless NoC as Interconnection Backbone for Multicore Chips: Promises and Challenges," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 228–239, 2012.
- [8] S.-B. Lee, S.-W. Tam, I. Pefkianakis, S. Lu, M.-C. F. Chang, C. Guo, G. Reinman, C. Peng, M. Naik, L. Zhang, and J. Cong, "A scalable micro wireless interconnect structure for CMPs," in *Proceedings of the MOBICOM '09*, 2009, p. 217.
- [9] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, and C. Teuscher, "Scalable Hybrid Wireless Network-on-Chip Architectures for Multi-Core Systems," *IEEE Transactions on Computers*, vol. 60, no. 10, pp. 1485–1502, 2010.
- [10] S. Deb, K. Chang, X. Yu, S. P. Sah, M. Cosic, P. P. Pande, B. Belzer, and D. Heo, "Design of an Energy Efficient CMOS Compatible NoC Architecture with Millimeter-Wave Wireless Interconnects," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2382–2396, 2013.
- [11] A. K. Kodi, A. I. Sikder, D. Ditomaso, S. Kaya, D. Matolak, and W. Rayess, "Kilo-core Wireless Network-on-Chips (NoCs) Architectures," in *Proceedings of the NANOCOM '15*, 2015, p. Art. 33.
- [12] C. Sun, M. T. Wade, Y. Lee, J. S. Orcutt, L. Alloatti, M. S. Georgas, A. S. Waterman, J. M. Shainline, R. R. Avizienis, S. Lin, B. R. Moss, R. Kumar, F. Pavanello, A. H. Atabaki, H. M. Cook, A. J. Ou, J. C. Leu, Y.-H. Chen, K. Asanović, R. J. Ram, M. A. Popović, and V. M. Stojanović, "Single-chip microprocessor that communicates directly using light," *Nature*, vol. 528, no. 7583, pp. 534–538, 2015.
- [13] A. Karkar, T. Mak, K.-F. Tong, and A. Yakovlev, "A Survey of Emerging Interconnects for On-Chip Efficient Multicast and Broadcast in Many-Cores," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 58–72, 2016.
- [14] B. Daya, C.-H. O. Chen, S. Subramanian, W.-C. Kwon, S. Park, T. Krishna, J. Holt, A. P. Chandrakasan, and L.-S. Peh, "SCORPIO: a 36-core research chip demonstrating snoopy coherence on a scalable mesh NoC with in-network ordering," in *Proceedings of the ISCA-41*, 2014, pp. 25–36.
- [15] J. Oh, A. Zajic, and M. Prvulovic, "Traffic steering between a low-latency unswitched TL ring and a high-throughput switched on-chip interconnect," in *Proceedings of the PACT*, 2013, pp. 309–318.
- [16] S. Abadal, M. Iannazzo, M. Nemirovsky, A. Cabellos-Aparicio, H. Lee, and E. Alarcón, "On the Area and Energy Scalability of Wireless Network-on-Chip: A Model-based Benchmarked Design Space Exploration," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1501–13, 2015.
- [17] S. Park, T. Krishna, C.-H. Chen, B. Daya, A. Chandrakasan, and L.-S. Peh, "Approaching the theoretical limits of a mesh NoC with a 16-node chip prototype in 45nm SOI," in *Proceedings of the DAC-49*, 2012, pp. 398–405.
- [18] S. Abadal, E. Alarcón, A. Cabellos-Aparicio, and J. Torrellas, "WiSync: An Architecture for Fast Synchronization through On-Chip Wireless Communication," in *Proceedings of the ASPLOS '16*, 2016, pp. 3–17.
- [19] S. Woo, M. Ohara, E. Torrie, and J. Singh, "The SPLASH-2 programs: Characterization and methodological considerations," *ACM SIGARCH Computer Architecture News*, vol. 23, no. 2, pp. 24–36, 1995.
- [20] C. Bienia, S. Kumar, J. Singh, and K. Li, "The PARSEC benchmark suite: characterization and architectural implications," in *Proceedings of the PACT '08*, 2008, pp. 72–81.
- [21] P. Conway and B. Hughes, "The AMD Opteron Northbridge Architecture," *IEEE Micro*, vol. 27, no. 2, pp. 10–21, 2007.
- [22] M. Martin, "Token Coherence: decoupling performance and correctness," in *Proceedings of the ISCA-30*, 2003, pp. 182–193.
- [23] S. Abadal, R. Martínez, J. Solé-Pareta, E. Alarcón, and A. Cabellos-Aparicio, "Characterization and Modeling of Multicast Communication in Cache-Coherent Manycore Processors," *Computers and Electrical Engineering (Elsevier)*, vol. 51, no. April, pp. 168–183, 2016.
- [24] T. Krishna, L.-S. Peh, B. Beckmann, and S. K. Reinhardt, "Towards the ideal on-chip fabric for 1-to-many and many-to-1 communication," in *Proceedings of the MICRO-44*, 2011, pp. 71–82.
- [25] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the spinnaker system architecture," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, 2013.
- [26] M. Palesi and M. Daneshmand, *Routing Algorithms in Networks-on-Chip*. Springer, 2014.
- [27] N. Abeyratne, R. Das, Q. Li, K. Sewell, B. Giridhar, R. G. Dreslinski, D. Blaauw, and T. Mudge, "Scaling towards kilo-core processors with asymmetric high-radix topologies," in *Proceedings of the HPCA-19*, 2013, pp. 496–507.
- [28] F. A. Samman, T. Hollstein, and M. Glesner, "Multicast parallel pipeline router architecture for network-on-chip," in *Proceedings of DATE '08*, 2008, pp. 1396–1401.
- [29] K. Duraisamy, Y. Xue, P. Bogdan, and P. P. Pande, "Multicast-Aware High-Performance Wireless Network-on-Chip Architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 1126–1139, 2017.

- [30] X. Yu, J. Baylon, P. Wettin, D. Heo, P. Pratim Pande, and S. Mirabasi, "Architecture and Design of Multi-Channel Millimeter-Wave Wireless Network-on-Chip," *IEEE Design & Test*, vol. 31, no. 6, pp. 19–28, 2014.
- [31] N. Weissman and E. Socher, "9mW 6Gbps Bi-directional 85-90GHz Transceiver in 65nm CMOS," in *Proceedings of the EuMIC '14*, 2014, pp. 25–28.
- [32] S. Laha, S. Kaya, D. W. Matolak, W. Rayess, D. DiTomaso, and A. Kodi, "A New Frontier in Ultralow Power Wireless Links: Network-on-Chip and Chip-to-Chip Interconnects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 186–198, 2015.
- [33] O. Markish, B. Sheinman, O. Katz, D. Corcos, and D. Elad, "On-chip mmWave Antennas and Transceivers," in *Proceedings of the NoCS '15*, 2015, p. Art. 11.
- [34] D. Fritsche, P. Stärke, C. Carta, and F. Ellinger, "A Low-Power SiGe BiCMOS 190-GHz Transceiver Chipset With Demonstrated Data Rates up to 50 Gbit/s Using On-Chip Antennas," *IEEE Transactions on Microwave Theory and Techniques*, vol. PP, no. 99, pp. 1–12, 2017.
- [35] M. A. I. Sikder, A. K. Kodi, M. Kennedy, S. Kaya, and A. Louri, "OWN: Optical and Wireless Network-on-Chip for Kilo-core Architectures," in *Proceedings of the HOTI-23*, 2015, pp. 44–51.
- [36] D. DiTomaso, A. Kodi, D. Matolak, S. Kaya, S. Laha, and W. Rayess, "A-WiNoC: Adaptive Wireless Network-on-Chip Architecture for Chip Multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3289–3302, 2015.
- [37] S. Abadal, A. Mestres, E. Alarcón, M. Nemirovsky, A. González, H. Lee, and A. Cabellos-Aparicio, "Scalability of Broadcast Performance in Wireless Network-on-Chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3631–3645, 2016.
- [38] A. Mestres, S. Abadal, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "A MAC protocol for Reliable Broadcast Communications in Wireless Network-on-Chip," in *Proceedings of the NoCArc '16*, 2016.
- [39] Z. Wang, P. Y. Chiang, P. Nazari, C. C. Wang, Z. Chen, and P. Heydari, "A CMOS 210-GHz fundamental transceiver with OOK modulation," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 3, pp. 564–580, 2014.
- [40] C. Sun, C. Chen, and G. Kurian, "DSENT - A Tool Connecting Emerging Photonics with Electronics for Opto-electronic Networks-on-Chip Modeling," in *Proceedings of the NoCS '12*, 2012, pp. 201–210.
- [41] F. Gutierrez, S. Agarwal, K. Parrish, and T. S. Rappaport, "On-chip integrated antenna structures in CMOS for 60 GHz WPAN systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1367–1378, 2009.
- [42] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "CACTI 6.0: A Tool to Model Large Caches," Tech. Rep., 2009.
- [43] A. Mineo, M. Palesi, G. Ascia, and V. Catania, "Runtime Tunable Transmitting Power Technique in mm-Wave WiNoC Architectures," *IEEE Transactions on VLSI Systems*, vol. 24, no. 4, pp. 1535–1545, 2016.
- [44] J. Chan, G. Hendry, A. Biberian, K. Bergman, and L. P. Carloni, "PhoenixSim: A Simulator for Physical-Layer Analysis of Chip-Scale Photonic Interconnection Networks," in *Proceedings of DATE '10*, 2010, pp. 691–696.
- [45] R. Ubal, P. Mistry, D. Schaa, H. Ave, and D. Kaeli, "Multi2Sim: A Simulation Framework for CPU-GPU Computing," in *Proceedings of the PACT '12*, 2012, pp. 335–344.
- [46] "Intel Corporation. Intel Products. ark.intel.com, 2015."
- [47] M. Daneshlab, M. Ebrahimi, T. C. Xu, P. Liljeberg, and H. Tenhunen, "A generic adaptive path-based routing method for MP-SocS," *Journal of Systems Architecture*, vol. 57, no. 1, pp. 109–120, 2011.
- [48] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, "Kilo-NOC: A Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees," in *Proceedings of ISCA-38*, 2011, pp. 401–412.
- [49] T. Krishna and L.-S. Peh, "Single-Cycle Collective Communication Over A Shared Network Fabric," in *Proceedings of the NoCS '14*, 2014, pp. 1–8.
- [50] M. F. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S.-W. Tam, "CMP Network-on-Chip Overlaid With Multi-Band RF-Interconnect," in *Proceedings of the HPCA-14*, 2008, pp. 191–202.
- [51] G. Kurian, J. Miller, J. Psota, J. Eastep, J. Liu, J. Michel, L. Kimmerling, and A. Agarwal, "ATAC: A 1000-Core Cache-Coherent Pro-

cessor with On-Chip Optical Network," in *Proceedings of the PACT*, 2010, pp. 477–488.



Sergi Abadal is Project Director at the NaNoNetworking Center in Catalonia, Universitat Politècnica de Catalunya, where he also obtained his PhD in computer science engineering (2016). In 2013, he was awarded by INTEL within his Doctoral Student Honor Program. He has given 6 invited talks and co-authored more than 10 journal and 15 conference papers. His research interests include on-chip networking, many-core architectures, and graphene-based wireless communications.



Josep Torrellas is the Saburo Muroga Professor of Computer Science at the University of Illinois at Urbana-Champaign. He is the Director of the Center for Programmable Extreme Scale Computing and a Fellow of IEEE (2004), ACM (2010), and AAAS (2016). He is also a member of the Computing Research Association (CRA) Board of Directors. He has served as the Chair of the IEEE Technical Committee on Computer Architecture (2005-2010) and as a Council Member of CRAs Computing Community Consortium (2011-2014). His research interests include multicore architectures, low-power design, and extreme-scale computing. Torrellas has a PhD in Electrical Engineering from Stanford University.



Eduard Alarcón is an associate professor at the the Universitat Politècnica de Catalunya, where he obtained his PhD in electrical engineering in 2000. He has coauthored more than 400 scientific publications, 8 book chapters and 12 patents, and has been involved in different national, EU and US R&D projects. He was elected IEEE CAS society distinguished lecturer, member of the IEEE CAS Board of Governors (2010-2013), Associate Editor for IEEE TCAS-I, TCAS-II, JOLPE, and Editor-in-Chief of JETCAS. His research interests include nanocommunications, energy harvesting, and wireless energy transfer.



Albert Cabellos-Aparicio is an assistant professor at Universitat Politècnica de Catalunya, where he obtained his PhD in computer science engineering in 2008. Also, he is co-founder and scientific director of the NaNoNetworking Center in Catalonia. He has been a visiting researcher at Cisco Systems and Agilent Technologies and a visiting professor at the KTH, Sweden, and the MIT, USA. He has given more than 10 invited talks and co-authored more than 20 journal and 50 conference papers. His research interests include nanocommunications and software-defined networking.