This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2016.2537332, IEEE Transactions on Parallel and Distributed Systems

Scalability of Broadcast Performance in Wireless Network-on-Chip

Sergi Abadal, Albert Mestres, Mario Nemirovsky, Heekwan Lee, Antonio González, Eduard Alarcón, and Albert Cabellos-Aparicio

Abstract—Networks-on-Chip (NoCs) are currently the paradigm of choice to interconnect the cores of a chip multiprocessor. For hundreds or thousands of cores, though, conventional NoCs may not suffice to fulfill the increasing on-chip communication requirements given that the performance of such networks actually drops as the number of cores grows, especially in the presence of multicast and broadcast traffic. This not only limits the scalability of current multiprocessor architectures, but also sets a performance wall that prevents the development of architectures that generate moderate-to-high levels of multicast. In this paper, a Wireless Network-on-Chip (WNoC) where all cores share a single broadband channel is presented. Such design is conceived to provide low latency and ordered delivery for multicast/broadcast traffic, in an attempt to complement a wireline NoC that will transport the rest of communication flows. To assess the feasibility of this approach, the network performance of WNoC is analyzed as a function of the system size and the channel capacity, and then compared to that of wireline NoCs with embedded multicast support. Based on this evaluation, preliminary results on the potential performance of the proposed hybrid scheme are provided, together with guidelines for the design of MAC protocols for WNoC.

Index Terms—Network-on-Chip, Wireless On-Chip Communication, Design Space Exploration, Multicast, Broadcast, Latency, Throughput, MAC Protocols, Manycore Processors, Hybrid NoC

1 Introduction

In the ever-changing world of microprocessor design, multicore architectures are currently the dominant trend for both conventional and high-performance computing. These architectures consist of the interconnection of several independent processors or cores, as well as of a multilevel cache to improve overall performance. Communication between cores and the memory hierarchy is not only a requirement to ensure the correct operation of a multiprocessor, but also a main determinant of its performance [1].

Network-on-Chip (NoC), which consists of a fabric of routed interconnections, is currently the paradigm of choice for moderately-sized multiprocessors [2], [3]. As the number of cores grows, though, current NoCs face significant challenges that limit its theoretical scalability and that drive the need for new solutions at all levels of design [4]. For instance, the diminishing multicast performance of NoCs is foreseen to be a critical issue as we reach the manycore era: due to their point-to-point nature, multicast messages need to be replicated at the source or within the network. Without proper support, such approach increases the average communication latency and creates large levels of contention [5], leading to a remarkable slowdown of the execution speed of the multiprocessor [6]. As the average number of destinations per message grows, the performance drop only becomes more severe.

A major concern is that, despite its limited support, multicast and broadcast may play a decisive role in future architectures [7]. Cache coherency, arguably the main source of on-chip communication in shared memory multiprocessors, is currently implemented through directory-based schemes that use multicast to invalidate cache blocks on a shared write. Absorbing the increase of multicast requirements inherent to application scaling [8] or produced by imprecise tracking techniques [9] comes at the cost of increased latency, extra storage overhead or higher protocol complexity. To avoid these trade-offs, alternative schemes eliminate the restrictions imposed by full-bit directories and make intensive use of broadcast instead. These range from directory-like schemes like the AMD HyperTransport [10], to classical snooping protocols adapted to unordered networks [11], [12], and token coherence [13]. In message passing, collective primitives such as `MPI_Allgather` or `MPI_Allreduce` use multicast and are widely employed in parallel algorithms [7]. Finally, novel computing paradigms could be also multicast-driven: multiprocessors emulating neural network architectures communicate their cores through multicast "spike" messages [14].

Considerable research efforts have been recently devoted to reducing the penalty imposed by multicast communications and, therefore, relaxing the constraints on the design of massive multiprocessor architectures. Most works have progressively improved multicast support by augmenting conventional network interfaces and routers [5], [6], [15], [16], [17], [18], [19], [20], [21]. Alternatively, multicast support in high-radix switches has been proposed but not evaluated [22]. The advent of emerging interconnect tech-

- *Sergi Abadal, Albert Mestres, Eduard Alarcón and Albert Cabellos-Aparicio are with the NaNoNetworking Center in Catalonia (N3Cat), Universitat Politècnica de Catalunya, Barcelona, Spain. Corresponding E-mail: abadal@ac.upc.edu*
- *Mario Nemirovsky is an ICREA Senior Research Professor at the Barcelona Supercomputing Center (BSC), Barcelona, Spain.*
- *Heekwan Lee is with the Samsung Advanced Institute of Technology (SAIT), South Korea.*
- *Antonio González is with the Department of Computer Architecture at Universitat Politècnica de Catalunya, Barcelona, Spain.*

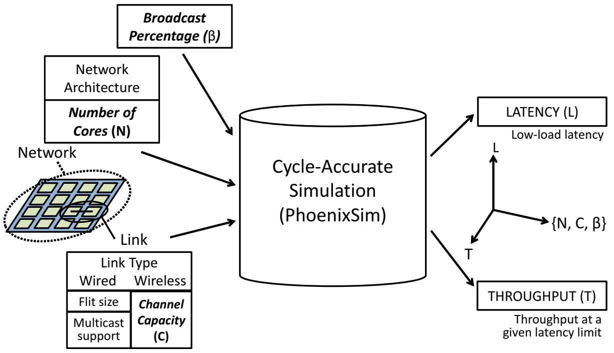


Fig. 1. Summary of the methodology of this work.

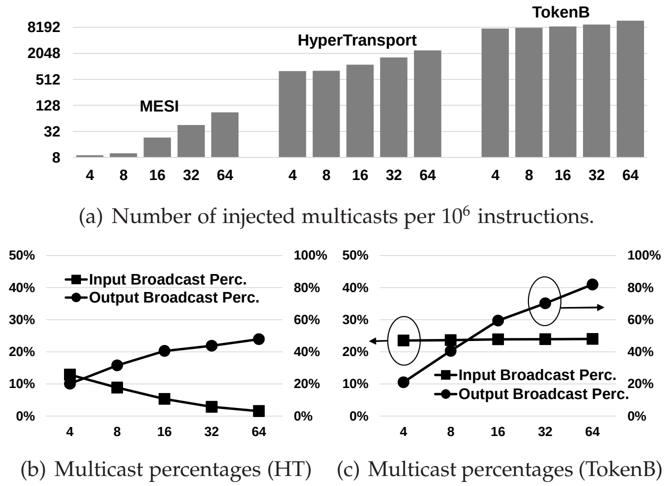
nologies also opens a set of opportunities within this context [23]. For instance, multicast methods have been extended to 3D topologies enabled by vertical stacking [24]. Further, the service of multicast traffic through dedicated broadcast channels by means of global RF transmission lines and nanophotonic waveguide bundles has been inspected [25], [26]. Finally, on-chip wireless communication stands as one of the most promising options given its inherent broadcast capabilities [7], [23], [27].

In all the aforementioned cases, substantial performance improvements have been reported with respect to baseline NoC designs. However, the scalability of these solutions remains unknown and must be investigated in order to evaluate their suitability for manycore processors. In this paper, we aim to address this issue by performing a design space exploration in the pathway to obtaining the scalability of different network schemes. To this end, we study how the latency and throughput vary with *a)* the number of nodes; *b)* the link capacity; and *3)* the percentage of broadcast traffic (understanding broadcast as an extreme case of multicast communication). Due to their unique suitability, we set the main focus to network architectures enabled by wireless on-chip communication. We analyze their performance and implementation cost, to then benchmark them against to that of wireline NoCs with multicast support.

The employed methodology is summarized in Figure 1 and detailed in Section 4. Basically, performance is modeled at the link level depending on the interconnect technology, and then used to obtain network-level metrics by means of cycle-accurate simulation. With it, the study aims to make a threefold contribution:

- To identify the most appropriate network architecture in different scenarios by capturing performance break-even points,
- To contextualize the performance of well-known Medium Access Control (MAC) techniques, providing guidelines for future design and optimization,
- To provide a preliminary evaluation of hybrid wired-wireless network architectures.

The remainder of this paper is as follows. Section 2 gives more details on the motivation of this paper and surveys related work. In Section 3, we provide some background on wireless on-chip networking and review the state of the art. The general framework of evaluation is described in detail in 4. Then, the results of the design space exploration are shown in Section 5 and discussed in Section 6. Section 7 concludes the paper.



(a) Number of injected multicasts per 10^6 instructions.

(b) Multicast percentages (HT) (c) Multicast percentages (TokenB)

Fig. 2. Multicast traffic as a function of the number of cores for three coherence schemes. We refer the reader to [8] for simulation details.

2 Motivation and Related Work

The existing gap between traffic requirements and communication performance forces architects to face different design trade-offs in multiprocessors. In cache coherence, snooping protocols in bus-based processors with a few cores have given way to directory-based schemes in denser processors with modern NoCs. Synchronization has been particularly affected by this fact, as it has become expensive by default and can degrade performance by a 40% in average and by seven times in extreme cases like *streamcluster* albeit representing a small fraction of the code [28]. The main reason is that these functions would greatly benefit from global communication schemes [7], [29], implying that in spite of the recent coherence-aware proposals [12], [30], multicast support will become a key design point for NoCs. Also, and in spite of the recent NoC-aware works in architecture [31], [32], this implies that avoiding multicast may become unaffordable at some point.

Regardless of whether multicast is avoided or intensively used, scaling parallel architectures and applications increases their multicast requirements. Data is generally distributed (and potentially shared) among a larger number of cores, causing coherence transactions to be more frequent and to involve a larger destination set [33], [34]. This implies that the multicast traffic per instruction increases with the system size for virtually any coherence protocol or interconnect, as shown in Figure 2a, which assumes a tiled architecture with private 32-kB L1-D/L1-I caches, 512-kB of shared L2 per core and three coherence protocols [8]. Results are the geometric mean of all the SPLASH-2 and PARSEC benchmarks. Additionally, Figures 2b and 2c show the percentage of multicast flits with respect to all the traffic for two multicast-intensive schemes. Even though multicasts may represent a little fraction of all the injected flits, in-network replication causes them to become accountable for a potentially huge percentage of the traffic. In TokenB, 25% of the injected traffic is broadcast and generates 80% of the flits served by the NoC.

Perhaps sparked by the expected increase in multicast traffic, considerable research has been directed towards improving its support in conventional NoCs using path-based and tree-based strategies. In the former case, a number of

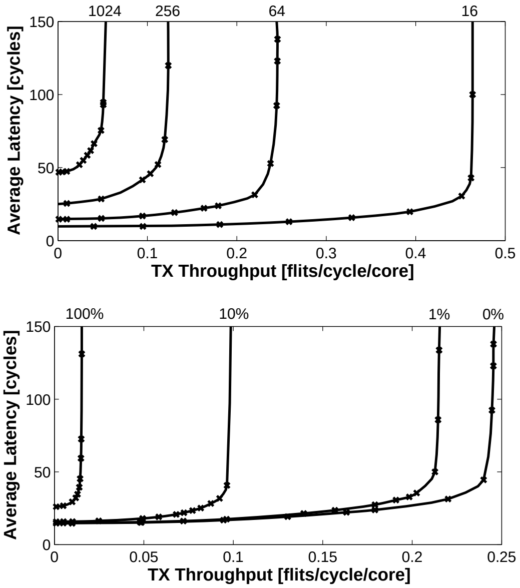


Fig. 3. Scaling of the performance of **MESH-FT** (see Section 4.4) as a function of the system size for unicast traffic (top, labels indicate the number of cores), and of the broadcast intensity in a 64-core system (bottom, labels indicate the percentage of broadcast transmissions).

copies are sent to separate chip partitions. Each copy is in turn replicated when reaching each destination: the copy is delivered and the original flit is forwarded to the next destination through a deterministic [15] or adaptive path [19]. In the latter case, the source injects a single message, which is replicated at intermediate routers and delivered to the intended destinations following a fixed [5] or balanced spanning tree [6], [20]. When compared, tree-based methods generally provide a lower latency than path-based multicast, but generate higher levels of contention that may lead to a reduced throughput. Since flit replication at the routers is a basic aspect in both cases, recent works have migrated from single-port allocation [5], [17] to multiport allocation [6], [16], [18], thereby reducing routing latency.

Despite the recent efforts, the increasing importance of multicast directly contrasts with how NoC performance is expected to scale with the system size. The sheer addition of more cores causes the average logical distance between processors to increase, affecting latency for all types of traffic. Figure 3a plots the latency-throughput characteristic of a mesh for different network sizes and assuming uniform random unicast traffic. Even considering an aggressive design with two cycles per hop, the low-load latency increases substantially and the throughput drops. Locality is normally enforced at upper layers to reduce the impact of the system size [1]; however, this principle does not apply to dense multicasts since flits still need to reach far-apart destinations [18], [20]. Moreover, flit forking increases contention and aggravates the performance degradation. This is clearly observed in Figure 3b, which plots the latency-throughput characteristic of a 64-core mesh with multicast support as a function of the percentage of broadcast traffic. Recently, Krishna and Peh presented a NoC that may alleviate scalability issues by enabling single-cycle multihop traversals [21]. However, a detailed study on how performance varies with the system size is missing.

Combined, the scalability trends mentioned above lead to a performance contradiction. On the one hand, the number of cores is scaled to speedup execution, which in turn increases the multicast communication requirements. On the other hand, NoCs are likely to be less effective when serving such increasingly important type of traffic as the system size increases. At some point, these opposite trends may cause multiprocessors to hit a performance wall unless scalable multicast support is provided.

One possible solution would be to employ shared-medium schemes, which are ideally suited to serve broadcasts yet clearly inefficient for unicast transmissions. Overlaying such a network over a conventional NoC would not only provide specialized support for multicasts, but also offload the main NoC and thus increase performance and efficiency for unicasts. If a single medium is shared among all cores, ordered multicast delivery can be ensured, helping to reduce the complexity of the processor architecture. However, globally shared-medium schemes will only be considered as serious contenders provided that their scalability is demonstrated. Among the different ways to implement such strategy [23], [25], [26], we chose to focus on the wireless RF paradigm due to its inherent broadcast capabilities and potential scalability [7].

3 Wireless Network-on-Chip

Recent advancements in on-chip antennas and transceivers have opened the door to the conception of Wireless Network-on-Chip (WNoC) architectures. As information is radiated and may be received by any other antenna regardless of its chip location, WNoC offers native broadcast capabilities and shows great promise towards the implementation of low-latency and adaptive schemes [7], [27]. Here, we provide some background on this area.

3.1 PHY: Towards Core-Level Wireless Communication

The physical layer (PHY) is the foundation over which wireless networks are developed. It defines how bits are transmitted over the wireless links and, thus, the design of the antenna and the transceiver. In a WNoC, the PHY module will basically serialize processor messages, modulate the resulting bits at a given frequency much higher than the processor clock, and deliver the modulated signal to the antenna. The inverse operation is performed at reception.

The decisions taken at the PHY level affect the raw transmission rate, area and power footprint of the solution. The transmission rate R, expressed in bits per second (bps), is given by:

$$R = B \cdot S_E, \tag{1}$$

where B is the frequency bandwidth of the link in Hz, and S_E is the spectral efficiency of the modulation in bps/Hz. Hence, transmission rates can be scaled by either *(a)* increasing B at the expense of an area and power cost that is roughly linear up to a certain limit imposed by the technology, or *(b)* using a modulation with higher S_E, which may come with non-linear area and power costs due to the need of a complex modulator. As we will see, technology evolution has orthogonal implications on these costs.

Current transceiver proposals for chip communication use frequencies around 60 GHz with simple modulations to minimize area and power. As a reference, the 65-nm CMOS implementation presented in [35] uses On-Off Keying (OOK) and achieves 16 Gbps with a bit error rate of 10^{-15} while taking 31.2 mW of power and 0.25 mm^2 of silicon area including an antenna of 0.02 mm^2 [36]. The same authors increase the data rate by roughly tripling B for a total of 48 Gbps, while consuming 97.5 mW and 0.73 mm^2. They also explore the possibility of using the Quadrature Phase-Shift Keying (QPSK), which doubles S_E with respect to OOK for a total of 32 Gbps assuming the original bandwidth. Due to the need of complex components, the power escalates up to 96 mW and the area becomes approximately 0.4 mm^2.

As shown in Section 5.6, these power and area figures would represent an overhead of 1% to 10% if integrated within current 18-core processors. To scale the design, it may become necessary to use clustering and to employ one wireless unit per cluster as extensively evaluated in related works [27], [37], [38], [39]. In spite of showing remarkable performance, efficiency, and thermal profile improvements, the potential of such hybrid approach for broadcast is still limited by its multi-hop nature.

In order to fully benefit from the unique flexibility and broadcast capabilities of WNoC, wireless communication must be provided for each individual core [7], [29]. However, integrating one transceiver per core in manycore settings will only be possible if their area and power are drastically reduced. This can be achieved by pushing the transceiver frequency possibly through technology downscaling. With this, the area of passive components like the antennas or the inductors is quadratically reduced, and it becomes easier to increase the bandwidth assuming a fixed power budget. In Section 5.6 we provide a rough estimation of the power and area of [35] when scaled down to 22-nm CMOS and operating at frequencies around 120 GHz.

Research on RF circuits and systems able to operate at 100 GHz and above is a reality. Recent survey works [40], [41] confirm that small-footprint antennas and transceivers operating at those frequencies will be available in the near future. Further, early prototypes at frequencies as high as 220 GHz have been presented [42]. Further down the road towards terahertz operation, graphene technologies are under intense research due to its exceptional properties [43], [44], [45]. Finally, the use of the recently proposed surface wave technology [23], [46] could significantly reduce the power consumption due to its highly improved propagation method. In all cases, we believe that the benefits of the WNoC application could represent a strong technology pull and drive new research on the field.

3.2 MAC: Medium Access Control in WNoC

The Medium Access Control (MAC) layer implements mechanisms to ensure that all nodes can access to the medium in a reliable manner. This plays a decisive role in determining the performance of any network as two simultaneous accesses to the same channel will fail and result into a waste of resources.

MAC protocol design has been a key research issue since the creation of the first computer networks. In the ALOHA system [47], nodes simply attempt to transmit whenever data is ready and wait for acknowledgment. A collision is assumed at the receiver if the CRC fails, in which case the transmitter is not acknowledged and will have to retry after waiting a random amount of time. With a similar structure, slotted ALOHA reduces the collision probability by only allowing the user to transmit in pre-defined time instants, whereas Carrier Sense Multiple Access (CSMA, [48]) does so by checking whether the channel is free before transmitting. Ethernet [49] uses CSMA with collision detection (CSMA/CD), which allows senders to prematurely abort transmissions to minimize the performance penalty. An alternative approach is token passing [50], where the node that possesses the token transmits and then hands it off to the next node of an ordered list, totally avoiding collisions yet at the cost of extra latency.

Wireless networks typically deal with a set of issues that complicate the protocol and reduce performance. For instance, the IEEE 802.11 standard for wireless local area networks [51] uses CSMA with collision avoidance (CSMA/CA) instead of CSMA/CD since collisions cannot be detected during transmission. It also provides a contention-free mode of operation and addresses issues related to mobility, asynchronous operation, or the *hidden terminal* problem [52], wherein nodes located in different transmission ranges cannot correctly assess if the medium is free.

In WNoC, the MAC mechanism is of fundamental importance since the medium will be densely populated and the load may be high. Additionally, the solution must be simple as large area and power overheads cannot be afforded. Fortunately, the on-chip scenario shows some unique characteristics that lead to simple but performant solutions, as elaborated in Section 6, *a)* the processor is driven by a unique system clock, reducing overall complexity and enabling the design of streamlined solutions; *b)* nodes are static and potentially within the same transmission range, eliminating hidden terminals; *c)* dedicated control wires may assist the protocol; and *d)* optimization is possible since the architect knows, and may even control, the whole system.

Related work in WNoC considers a variety of MAC designs. Most efforts implement *contention-free* schemes that avoid collisions by using orthogonal channels resulting from the use of different frequencies, codes, time slots or any combination thereof [27], [38], [39], [53]. The main downturn of this approach is its rigid nature: the bandwidth is typically allocated in a static manner, affecting performance if traffic bursts or hotspots appear. Also, the complexity of the required transceivers does not scale well with the number of channels. Seeking to avoid this, other works have explored more flexible ways to managing channel access. For instance, the protocol proposed in [36] uses the token passing strategy mentioned above. In [54], contention is avoided by exchanging control messages through the wired plane. In [55], multiple narrow channels are used by different nodes to simultaneously send requests to a common receiver, which grants access to the data channel to only one of them. While these approaches reduce the complexity issues of using orthogonal channels, management is either latency-constrained [36] or not global [54], [55].

TABLE 1
Simulation Parameters

System	400 mm^2 die, 1 V, 1 GHz, 16 to 1024 cores.
Traffic	Exponential and Pareto arrivals, uniform and hotspot distributions, 1- and 4-flit packets, 0 to 100% of broadcast
Wireline NoC	128 bits, XY routing, 4 virtual channels, credit-based flow control, fixed tree multicast, single- and multiport allocation. MESH: 2 cycles per hop. FBFLY: 3-7 cycles per hop.
Wireless NoC	Single channel, 8 to 160 Gbps[1], 1-cycle token passing delay, 1-cycle from and to central buffer. CSMA: Non-persistent, NACK burst, <1 cycle timeout, truncated exponential backoff, 8 maximum retries.

At the other end of the spectrum, *contention-based* schemes like ALOHA or CSMA have received less attention probably due to its random nature. The work in [56] proposes a slotted CSMA scheme with theoretically optimal persistence calculated *a priori*, and compares its performance with that of a token passing scheme. Given that these protocols perform better for different loads, the authors of [57] propose a scheme that dynamically switches between both depending on communication intensity.

4 Framework

The main objective of this work is to contextualize the performance of WNoC by comparing it against state-of-the-art wireline NoCs. To this end, we evaluate how the latency and throughput of different architectures scale as a function of the number of nodes N, the capacity of the wireless channel C and the percentage of broadcast traffic β. Table 1 shows a summary of the different simulation parameters.

The evaluation is performed with PhoenixSim [58], a cycle-accurate NoC simulator based on Omnet++. Although its goal is to provide a simulation framework for nanophotonic NoCs, it also includes a complete set of modules to evaluate conventional NoCs. On top of this, we have implemented the necessary modules for the simulation of wireless on-chip communication towards the study of WNoC.

4.1 Simulated Architecture

This work considers a conventional tiled architecture, where each tile contains one core, as well as a fraction of the memory hierarchy and a Network Interface (NIF) connected to the router and wireless transceiver associated to the core.

Figure 4 depicts the network side of the simulated architecture. In transmission, cores generate traffic and convey it to a controller, which redirects it to the appropriate NIF depending on a pre-defined policy. This policy can simply obey network performance reasons and be simple, like checking whether the message is broadcast; or complex as the proposed in [59], which takes decisions based on load and latency estimations for each network plane. It can also be defined using architectural reasons, e.g. a message is marked as latency-critical by the programmer or the coherence protocol. The NIFs prepare outgoing transmissions by performing end-to-end tasks such as address translation or flow/congestion control. Flow control units are sent to the corresponding network router in the electric case; whereas, in the wireless case, they go through the transceiver, which performs MAC procedures first. When access to the wireless network is granted, data is serialized into bits, which are coded and modulated in the PHY module. The resulting signal is then radiated by the antenna. In reception, the process is inverted to turn bits and flits into processor messages. It is worth to note that, in the wireless plane, the PHY module detects collisions and notifies upper layers accordingly.

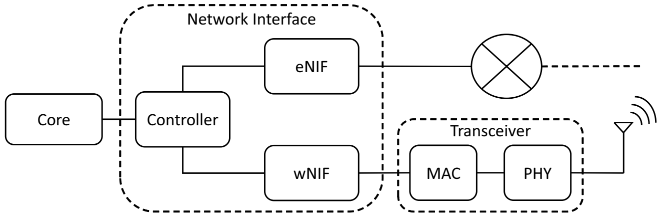


Fig. 4. Schematic diagram of the core-to-network architecture.

The scheme can be easily modified to perform core concentration. It can be symmetrically applied to both network planes by incorporating a local switch between a set of cores and the controller. In an asymmetric scheme, NIFs would be connected directly to a higher-radix router in the wired plane, while a multiplexer and an amplifier would be required between the NIF and the MAC module of the wireless plane. Note that, in any case, cores are connected to two separate network planes and that there is no need to modify the routing protocol in the main NoC.

Due to the flexibility of WNoC, this scheme can be applied in heterogeneous systems. In CPU+GPU organizations, or in processors with *small* and *large* cores like ARM's big.LITTLE, it may be desirable to provide different broadcast domains. This can be achieved by using different frequency channels and tuning the transceiver of each node to the frequency of its domain. To ensure correct operation, we also need to ensure that the distance between any two neighboring nodes is enough to neglect near-field effects at the antennas.

4.2 Traffic Generation

Initially, cores are modeled as generators of memoryless traffic with a constant arrival rate λ over time. Unless noted, all cores transmit with the same probability. Broadcasts represent a fraction β of all the packets, while the rest (1-β) are unicasts and their destination follows a uniform distribution. Even though we do not want to bind to specific architectures, we consider two packet sizes as commonly found in cache-coherent systems [1]: short for requests and long for responses (the size of one address and one address plus one data block, respectively). Here, we assume these to be equivalent to 1 flit and 4 flits.

This simple synthetic traffic is used to investigate the scalability of different networks under a broad range of conditions. To provide hints of performance in more realistic scenarios, we later perform a sensitivity analysis considering traffic bursty and hotspot traffic, which is found in most cache-coherent applications for communications in general [60] and multicast in particular [8]. To generate

[1] Wireless capacity values are chosen so that, including the propagation time, a flit can be transmitted in an integer number of cycles.

bursty traffic, we alternate ON/OFF periods, the length of which follows Pareto distributions defined by the Hurst exponent H [61]. This exponent takes values between 0.5 (exponential) and 1 (extreme burstiness). To model hotspot traffic, we use a gaussian parameter σ which takes values between 0 (concentrated) and ∞ (spread out) and describes the percentage of load that is assigned to each node [60]. More details can be found in [8].

4.3 Modeling Wireless Communication

In the simulator, wireless communication occurs through the exchange of messages between the PHY of different nodes. When a node sends data through the wireless plane, the simulator needs to internally determine:

Who receives the data? The PHY module of the sender delivers data to the nodes that are within its range and that use the same transmission channel. Since we assume that the transmission range of the antennas covers the whole chip and that all cores access to the same channel, data is broadcast to all nodes.

When is data received? The receiver will start receiving the data after the propagation time $t_p = d/c_0$, where d is the physical distance between the transmitter and the receiver and c_0 is the speed of light (we assume free-space propagation). The transmission process takes $t_t = l/C$, where l is the packet length and C is the capacity of the wireless channel.

How are collisions detected? During the propagation time, receiving nodes are still not aware that the medium is being used. This can lead to collisions, which are automatically detected by the simulated receivers whenever two incoming transmissions overlap in time. Colliding messages are internally discarded and the situation will be notified to the MAC module. The user can decide when to send this notification, which must be in agreement with how collisions are detected in a real scenario. If desired, our model also allows transmitters to detect collisions while sending data.

4.4 Investigated Network Architectures

In this paper, we assume a WNoC that basically consists of one large channel of capacity C that is shared among N cores. We seek to cover a representative portion of the design space by testing the scalability of such configuration considering three MAC protocols, and then comparing it with that of a conventional NoC with two topologies and two router designs. NoCs based on optical waveguides or RF transmission lines could be also evaluated, but have been left out of the study due to scalability constraints in terms of laser power and design complexity [40].

We simulate three wireless network architectures. In all of them, we consider that time is slotted at the system clock granularity. This simplifies the protocols and cannot be generally considered in conventional wireless scenarios due to synchronization issues. We also assume that, since the link budget is deterministic, negligible bit error rates (around 10^{-15}) can be assumed with proper power allocation and/or coding. This means that transmission errors will only be due to collisions and, thus, that acknowledging is not necessary in contention-free schemes. In [35], authors consider a link power budget of 26.5 dB, enough at 60 GHz for a transmission range in the centimeter scale.

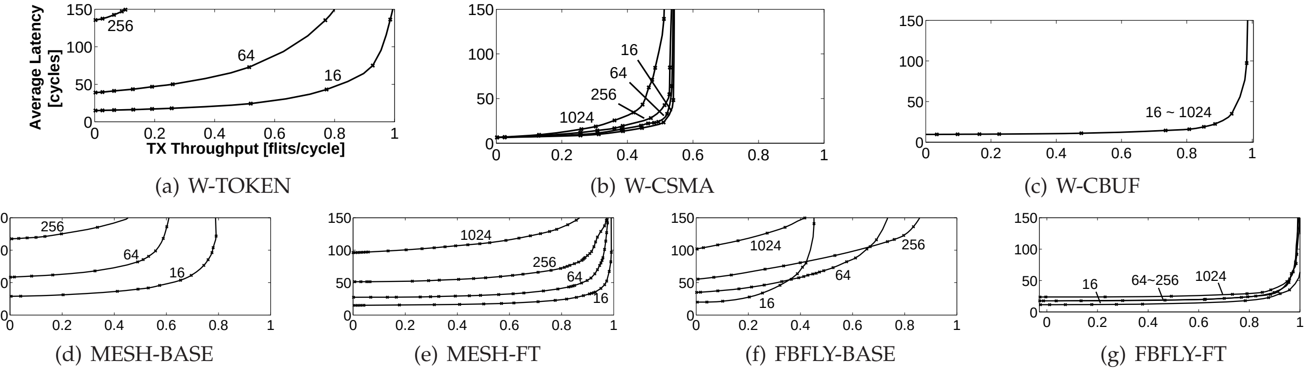
Wireless with Token Passing [W-TOKEN] - this category aims to represent a design family that relies in rigid strategies to avoid contention. In token passing, only the core that possesses the token is able to transmit [50]. One full packet can be transmitted in each round. We do not split long messages into flits here as the packet latency would be unacceptable, whereas bulk transmissions are not allowed for fairness reasons. Upon completion, or in case there is nothing to transmit, the token is handed off to the next core. We assume that the token passing is performed through a lightweight wired ring and is pipelined with the wireless transmission. Since the token somehow divides time in slots, we consider this as a feasible way to implement multiple access based on dividing the spectrum in orthogonal channels.

Wireless with Custom Carrier Sensing [W-CSMA] - this protocol represents a family of designs that let nodes contend for the channel (e.g. ALOHA and slotted ALOHA [47], CSMA [48]) as to provide certain flexibility at the cost of performance. In our case, we employed a protocol that augments a slotted ALOHA scheme with non-persistent CSMA aspects, and that adapts to the peculiarities of the chip scenario.

We divide time into slots at the processor clock granularity (1 GHz). When a node is ready to send data, it will begin a transmission at the next time slot provided that no other core is the middle of a transmission. This can be known by reading a *short/long* bit at the header of an on-going transmission (sizes are fixed) and ensures that collisions can only occur at the first slot of a transmission. If the medium is occupied or in the case of a collision, the sender backs off.

Receivers will detect collisions by checking redundancy or, since the power budget is deterministic, by observing that the power received is over a given limit. To minimize control overhead, we adopt a negative acknowledgment (NACK) strategy where receivers notify collisions by sending a tone. This acts as a jamming signal of sorts, and is based on the idea in [62]: the channel performs a collective AND operation, this is, the presence of tone implies that at least one receiver detected a collision. The transmitter will listen for the NACK during a time period by the end of the first transmission slot, or constantly if a separate frequency channel is used to this end. The sender will immediately cancel the transmission and retry later if a NACK is found, or continue with the transmission otherwise. In the unlikely case that a packet exceeds a given number of retries, it will be forwarded to an alternative network plane.

Wireless with Centralized Buffer [W-CBUF] - for the sake of comparison, we also study the performance of a centralized MAC scheme. With unlimited resources, it would be possible to have a single arbiter connected to every core with a one-cycle bidirectional link. When a node is ready to transmit data, it sends a request to the arbiter with its identity and size of the packet. The arbiter stores this information in a FIFO buffer and grants access to the node whose request is in the buffer head, waiting exactly the wireless transmission time between consecutive grants. Contention only appears when more than one node requests access during the same clock cycle and is resolved by the



(a) W-TOKEN (b) W-CSMA (c) W-CBUF

(d) MESH-BASE (e) MESH-FT (f) FBFLY-BASE (g) FBFLY-FT

Fig. 5. Latency-throughput curves for representative system sizes for 100% of broadcast traffic and a wireless capacity of one flit per cycle.

arbiter. This scheme therefore provides fair, ordered and contention-free access in a flexible way, with resources that are not available in traditional wireless networks. The main reason for evaluating this scheme with unlimited resources is to motivate unconventional MAC designs and to quantify the improvement margin of the protocols mentioned above.

Routed Mesh [MESH] - as baseline, we consider a wireline 2D mesh with two cycles per hop in the absence of contention, achievable with bypass strategies [20]. The choice of a mesh topology is backed up by its scalability in terms of complexity and performance, as well as by the fact that is extensively used as baseline design in the literature. Flow control is credit-based and uses the data links to exchange buffer utilization information between neighboring nodes. Tree-based multicast is studied due to its better latency scalability, which becomes a critical performance factor in many-core scenarios [63]. Note, though, that path-based and tree-based methods become very similar for broadcast (e.g. column-path routing [15] is equivalent to fixed tree with XY routing). We consider two router microarchitecture flavors: single-port allocation in **MESH-BASE** routers, and multiport allocation with multicast crossbar in **MESH-FT**. The delay is proportional to and independent of the number of flit replications in the former and latter case, respectively.

Flattened Butterfly [FBFLY] - given its much lower diameter, this topology represents a more aggressive competitor. As described in [64], FBFLY generally employs 4-way concentration and then connects each router with every router in its row or in its column. Thus, less than 4 hops suffice to reach any core from any other core. This comes at the cost of increasing the radix of the routers and complicating the design of its arbiter and crossbar as the network is scaled [63]. These designs do not scale well in terms of area and power (see Section 5.6), but are evaluated as to include low-diameter NoCs in the design space exploration. Due to the increase in design complexity, we assume that the pipeline depth logarithmically increases with the router radix, for a total of 3 to 7 cycles per hop. As in MESH, we consider fixed tree routing and two types of router: single-flit allocation in **FBFLY-BASE** and multiport allocation in **FBFLY-FT**.

4.5 Performance Metrics

Network performance is evaluated as follows. On the one hand, we measure latency as the time passed between the generation of one message and its the complete reception at *all* the intended destinations. Therefore, results include the delay introduced by the NIF at both ends, typically one clock cycle each. We consider that the controller also takes one clock cycle both at transmission and reception.

On the other hand, throughput is measured from the transmitter perspective, which implies that a multicast message will only be counted once despite being received by more than one core. Throughput results are expressed in flits per cycle to facilitate the comparison between networks with different topology or bandwidth, and always account for the aggregate of both the unicast and broadcast flows.

To obtain the latency-throughput characteristic of each network architecture, we gradually increase the offered load. For simplicity, though, we will generally show two critical metrics as exemplified in Figure 5f: the low-load latency and the saturation throughput. The former corresponds to the average communication latency in the presence of mild contention, and models the performance of the network for at least 50% of the maximum admitted load with reasonable accuracy. The latter evaluates the throughput of the network for very high loads. For the sake of fairness, we measure the throughput obtained when the latency reaches a given common limit instead that of simply measuring the load at which the network saturates (both metrics may differ). The limit is set to 150 cycles, a value commensurate with the latency of accessing data in the main memory. Relating the throughput to a bounded latency is important since, in the manycore scenario, the execution speed of the processor *is mainly constrained by latency, not throughput* [63].

5 Results

In the following, we first inspect how the performance of each network architecture scales with the number of cores N. We then fix the system size and investigate the impact of the wireless channel capacity C and of the broadcast percentage β on performance.

5.1 Scaling the Number of Nodes N

Figure 5 shows the latency-throughput characteristics of the considered network architectures for a set of representative system sizes $N = \{16, 64, 256, 1024\}$, assuming a wireless capacity of one flit per cycle and only broadcast traffic. As intuition suggests, the increase in system size has a significant impact upon performance. In conventional NoCs, the latency increases due to the growth of the number of hops

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2016.2537332, IEEE Transactions on Parallel and Distributed Systems

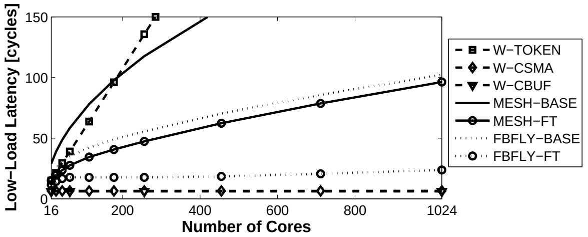


Fig. 6. Low-load latency of broadcast transmissions as a function of the number of nodes N for $C = 1$.

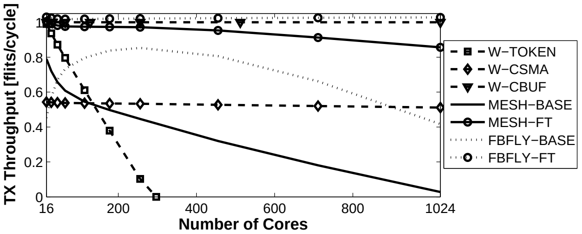


Fig. 7. Throughput of broadcast transmissions at the maximum admissible latency (150 cycles) as a function of the number of nodes N for $C = 1$.

required to reach all destinations, while the throughput suffers a contained drop. In wireless NoCs, the performance both in terms of latency and throughput depends on whether the protocol is fixed or works on demand. A notable case is that of **W-CSMA**, which saturates significantly earlier than the rest of alternatives. This is basically due of collisions: retries compete with newly generated packets, causing the throughput to gradually become lower than the offered load. Next, we analyze the results in more detail.

The behavior of the low-load latency as a function of the system size is shown in Figure 6. Three different scaling trends can be clearly identified. First, the latency of **W-TOKEN** scales linearly with the number of cores due to delay introduced by the arbitration phase. Since the token is passed through a ring, the latency scales as the average hop distance of such topology, $\mathrm{O}\big(N/2\big)$. Due to their on-demand nature, the latency of the rest of wireless schemes remains flat given that, at low loads, a node will most likely be able to transmit immediately. In the wired NoCs, the latency scales proportionally to the average hop distance of the topology: $\mathrm{O}\big(\sqrt{N}\big)$ in **MESH** and almost constant in **FBFLY**. In both topologies, the latency observed in their base configurations is considerably higher due to the additional delay incurred by the flit forking process. (**FBFLY** actually loses its scalability advantage since every flit spends $\propto N$ cycles in each router).

For the sake of comparison, it is important to remark the results from related work. The NoC implemented in [20] is expected to show a similar scaling trend than **MESH-FT**, but with a lower absolute value since it attains one cycle per hop. The scheme in [21] is a mesh with unconventional multi-hop bypass links, and achieves a latency (not counting router-to-processor communication) as low as 5.6 clock cycles for 64 cores. Although its scalability is theoretically close to **FBFLY-FT** as it is potentially able to reach all cores in two network hops, several technological assumptions need to be made to confirm this potential [65].

Figure 7 illustrates how the throughput of the different schemes scales with the system size. In **MESH-BASE** and **FBFLY-BASE**, the throughput decreases with the number of cores mainly because latency scaling induces the network to reach the latency limit at lower loads. With unbounded latency, the increase in terms of bisection bandwidth would mostly compensate the increase in number of destinations per message and the throughput would remain constant. This is the case for **RMESH-BASE** and **FBFLY-BASE**, which are below the latency limit and almost achieve maximum throughput. Given by the inherent broadcast nature of wireless NoCs and in spite of having a much lower bisection bandwidth, **W-CBUF** and **W-CSMA** also achieve a rather flat scaling with a lower absolute value in the latter case. Finally, we have that **W-TOKEN** is clearly dominated by the token passing delay and that it is not able to provide any throughput with acceptable delay beyond a few hundreds of cores.

Again, we compare our results with those of related work. The 16-core NoC implemented in [20] reports a throughput similar to that our **MESH-FT**. Since it is able to perform single-cycle hops, it will probably provide better throughput scalability than the meshes evaluated here. The work in [21] reports a throughput of ∼0.9 flits per cycle for 64 cores. This value, and its expected scalability, directly competes with **MESH-FT**, **FBFLY-FT**, or **W-CBUF**. The rest of wireless alternatives will need to improve to be comparable to it in large systems.

5.2 Scaling the Channel Capacity C

For the conditions evaluated above, wireless strategies are capable of consistently achieving very low latencies with moderate-to-high throughput. However, we have assumed a bandwidth of one flit per cycle thus far, which is around 160 Gbps in a system running at 1 GHz and with 128-bit links and including the propagation latency. As discussed in Section 3.1, these figures may not be available in the near future. Therefore, it is important to understand the dependence between performance and channel capacity in order to guide the design of future WNoCs.

Scaling the channel capacity C impacts upon the latency of any wireless communication through the transmission time as $t_t = l/C$, where l is the packet length. The propagation time, which also contributes on the communication latency, is dependent on the chip size and therefore assumed constant. The arbitration overhead is dependent on the arbitration scheme and, in the absence of load, remains constant (zero, two and $N/2$ cycles for **W-CSMA**, **W-CBUF** and **W-TOKEN**, respectively). For all this, the low-load latency approaches a fixed lower bound as we increase the channel capacity. To cite an example, the latency increases from ∼6.5 to ∼44 cycles in **W-CSMA** when scaling down the capacity from 160 to 8 Gbps. In large systems, these figures can still compete with most wired options.

Varying the channel capacity C also has a direct impact upon the throughput. Basically, the throughput increases linearly with the channel capacity if the propagation time is neglected. However, the propagation time becomes significant at high speeds and imposes an increasing overhead. To transmit a 128-bit flit in 8 cycles at 1 GHz, the propagation time requires the capacity to be increase only by 2.5%; whereas to transmit it in 1 cycle, the increase is of 25%.

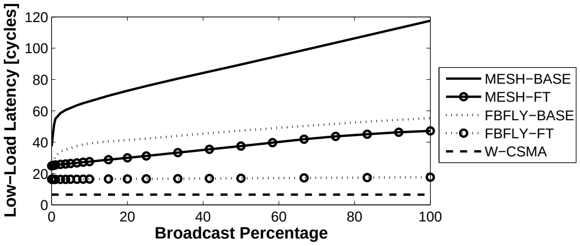


Fig. 8. Low-load latency as a function of the percentage of broadcast traffic β for $N = 256$ and $C = 1$.

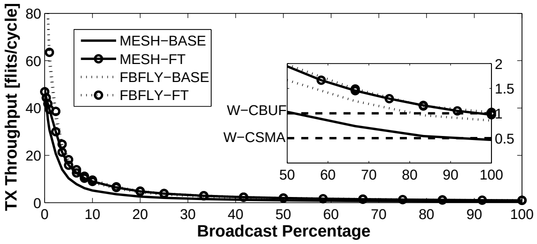


Fig. 9. Throughput at the maximum admissible latency (150 cycles) as a function of the percentage of broadcast traffic β for $N = 256$ and $C = 1$.

5.3 Scaling the Broadcast Percentage β

The performance of wired topologies is generally inversely proportional to the number of destinations per message. On the contrary, wireless schemes treat all messages as broadcasts and, thus, their performance is independent of the percentage of broadcast traffic. Therefore, $\beta = 100\%$ is a clearly unsuitable case for any NoC based upon point-to-point links and evaluating performance only in such scenario would be unfair. Here, we inspect the performance of **MESH** and **FBFLY** as a function of the percentage of broadcast traffic.

In the absence of contention, the latency of a broadcast transmission is equivalent to the latency of reaching the furthest destination. As shown in Figure 8, this causes the latency to drop as the broadcast probability decreases. The impact is more patent in the base configurations, since it takes several cycles to complete the flit forking process in each router. Remarkably, there is no break-even point of latency even with the best wired alternative: given enough channel capacity and due to its unique one-hop communication capabilities, the latency of any wireless transmission will be always the lowest.

From a throughput perspective, the percentage of broadcast traffic has a very large impact on performance. As illustrated in Figure 9, the throughput is of a several tens of flits per cycle in the wired topologies and decreases as each message has to reach more destinations. Approaching $\beta = 100\%$, the performance of wireline and wireless schemes become comparable despite the huge difference in terms of bisection bandwidth, to the point that break-even points with **MESH-BASE** are achievable given enough wireless capacity and MAC efficiency. This, together with the latency results above, suggests that the wireless plane could be used not only for broadcast transmissions, but also for selected latency-sensitive unicasts to further enhance performance.

On the one hand, we estimate that the work in [20] will provide curves very similar to that of **MESH-FT**. On the other hand, it remains unknown how the multihop capabilities of [21] will affect throughput in mixed traffic, given that broadcasts that occupy several router ports within the same clock cycle may greatly affect unicast transmissions.

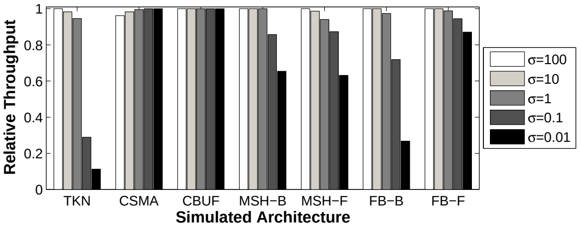


Fig. 10. Throughput for different spatial injection distributions, from spread out ($\sigma = 100$) to extremely hotspot ($\sigma = 0.01$), with $N = 64$, $\beta = 100\%$ and $C = 1$.

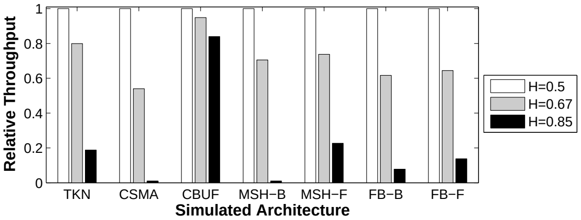


Fig. 11. Throughput for different temporal injection distributions, from exponential ($H = 0.5$) to extremely bursty ($H = 0.85$), with $N = 64$, $\beta = 100\%$ and $C = 1$.

5.4 Sensitivity Analysis

Several works have confirmed that on-chip traffic is often injected by a small subset of nodes and in a bursty manner [8], [60]. This fact has a notable impact upon the performance of the chosen NoC and, thus, should also influence its design. Here, we evaluate the sensitivity of the different network architectures to the traffic characteristics using, to this end, synthetic traffic generated with the method described in 4.2. Due to the random nature of the method, we performed 15 runs for each design point and calculated the geometric mean. Results are normalized to the exponential uniform random traffic case.

Hotspot traffic: spatial concentration typically reduces the network throughput due to the uneven use of resources. Figure 10 illustrates this effect by plotting the throughput for different levels of hotspot traffic. As we reduce σ, the injection process becomes more concentrated, significantly impacting the performance of most networks. **W-TOKEN** suffers an important reduction since the token needs to travel around the ring even if the processors willing to transmit are highly clustered, whereas **W-CBUF** and **W-CSMA** perform well independently of the injection profile. Concentration is even helpful in the latter case, as it reduces the average number of contending stations. In wired schemes, concentration creates congestion around the source, particularly in those configurations with high per-hop time, e.g. **FBFLY-BASE**.

Bursty traffic: Figure 11 shows the performance of the different schemes for increasing levels of burstiness. All networks see their performance reduced due to the backlogging of flits in routers and interfaces during packet bursts. This increases the mean latency and reduces the achievable throughput due to momentary congestion. **W-CSMA** is a

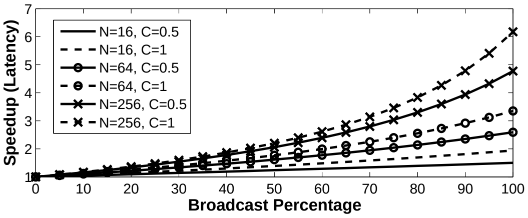


Fig. 12. Latency speedup of a hybrid NoC with respect to **MESH-FT** as a function of the broadcast percentage for different system sizes and channel capacities.

particularly concerning case, as the probability of collision increases with the burstiness of traffic. Congestion is also aggravated in wired schemes, which see the admissible throughput to drop substantially. On the other hand, **W-TOKEN** and **W-CBUF** perform reasonably well due to their collision-free and short pipelined nature.

5.5 Performance of a Hybrid Network Architecture

We conceive WNoC as a latency-driven and broadcast-oriented plane that will serve global traffic, in complement to a throughput-oriented wireline NoC that will transport the rest of the communication flows [7]. In light of the results above, it is reasonable to think that such hybrid network architecture will not only greatly reduce the latency of broadcast messages, but also achieve a significant throughput boost by relieving the wired plane of having to serve such type of traffic. Note, though, that this represents a small fraction of the hybrid architecture design space [27].

Here, we evaluate our proposed hybrid scheme by comparing the performance of **MESH-FT** with and without an overlaid WNoC with centralized buffer. We inspect how the potential improvement scales with the system size, the capacity of the wireless channel and the percentage of broadcast traffic. We use the scheme depicted in Figure 4, configuring the controller so that it forwards unicast and broadcast messages to the wired and wireless planes, respectively. We put particular emphasis on the results for $\beta \leq 25\%$, as these correspond to the range of broadcast percentages found in cache-coherent processors as detailed in Figure 2 [8]. Results for $\beta > 25\%$ could be also of interest given that new multicast-intensive architectural methods may arise if the cost of multicast is reduced [7], [29].

Figure 12 illustrates the improvement of the low-load latency as a function of all the variables considered throughout the paper. Here, a value of 4 implies that the hybrid architecture goes four times faster. The improvement is quite consistent and, as the intuition suggests, increases with the broadcast percentage, the channel capacity and the system size. Assuming $N = 256$, the latency is reduced by a 20% for $\beta \approx 10\%$, cut in half by $\beta \approx 50\%$ and to one third for $\beta \approx 70\%$. A similar tendency, yet with lower absolute gains, is obtained when comparing the hybrid architecture to **FBFLY-FT**.

Figure 13 shows the throughput improvement (or deterioration) as a function of all the design variables. Up to a given broadcast percentage β_1, the throughput of the wired plane is limited by bisection bandwidth and, therefore, the addition of the wireless plane increases performance (over 2X speedup in some cases). This percentage range is reduced as the network scales and the bisection bandwidth increases. Beyond that point β_1, the wired plane becomes limited by the ejection links and the inclusion of the wireless plane only helps to reduce the overall latency. At very high percentages, the speedup is the throughput difference between **W-CBUF** and a **RMESH-FT**. Although the throughput improvement is sensitive to the capacity of the wireless channel, improvements can be achieved over a significant range of broadcast percentages even with a modest capacity. Finally, note that throughput results are similar when compared to **FBFLY-FT**, but not shown for brevity.

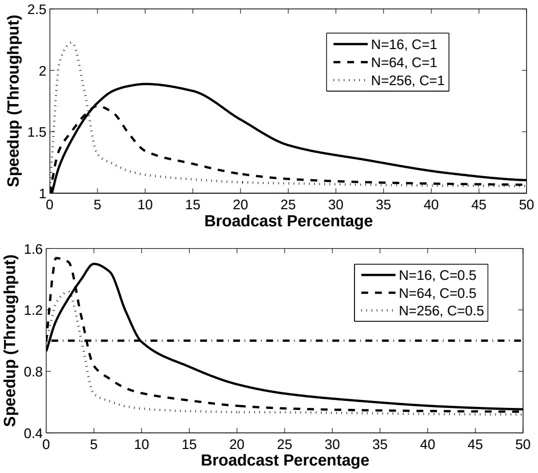


Fig. 13. Throughput speedup of a hybrid NoC with respect to **MESH-FT** as a function of the broadcast percentage for different system sizes and channel capacities.

5.6 Implementation Cost

To evaluate the costs of the proposed WNoC, we take base on rough but conservative scaling trends [25], [40] to extrapolate a reasonable design point from the transceiver designs proposed in [35] and outlined in Section 3.1. On the one hand, a sublinear downscaling in terms of area can be considered as a conservative rule of thumb given that passive components (e.g. antennas, inductors) scale quadratically with technology and assuming that active components approximately maintain their size. On the other hand, device scaling implies faster transistors and generally higher bandwidths assuming a constant power envelope. For RF transmission lines, a 40% increase of bandwidth per technology node was predicted in [25]. To account for the additional losses of the wireless channel, we conservatively assume a bandwidth-to-power ratio increase of around 25% per generation.

Table 2 compares the aforementioned area and power figures with that of different NoC implementations reported in the literature. In [3], the authors describe the mesh NoC of the Intel's 80-core Polaris processor. This design can be considered similar to **MESH-BASE**, as it does not implement any specific multicast support. To improve multicast performance, subsequent designs have explored the use of shorter router pipelines and multiport arbitration like the considered in **MESH-FT** [20], as well as broadcast ordering capabilities [12]. Lower diameter topologies like **FBFLY-BASE** have been also estimated at 32 nm [63]. Reasonable power and area are obtained for different system sizes and link widths, although recognizing serious scalability issues due to the increase in crossbar size and buffering.

We also include data from two recent designs that also consider broadcast capabilities enabled by novel RF-interconnect and nanophotonic technologies. Even though transmission line complexity and laser power issues may limit their scalability, they are serious contenders that need to be considered. On the one hand, Oh et al [59] propose to augment a conventional mesh with a global RF transmission line. Laying down the transmission line and the required transceivers takes a significant portion of area, and less power than the wireless option given that, in transmission lines, signals are guided instead of radiated and losses are reduced. On the other hand, a all-optical 64-core NoC based on global broadcast buses is proposed in [26]. The estimated power is high compared to the rest of alternatives, but provides a huge broadcast bandwidth of 320 bits per clock cycle. Area measurements are not provided.

To put these numbers in context, we complete Table 2 with the area and power consumption of two popular 22-nm cores, namely the high-performance Xeon Haswell and the energy-efficient Atom Silvermont. The thermal design power of an 18-core Haswell chip at 2.1GHz is 135W [66]. Correcting for frequency, we roughly estimate a per-core power of 5W. A similar reasoning is perform for the 8-core Silvermont chip, which works at 1.7GHz with a thermal design power of 12W [66]. Area numbers are supplied by the literature.

In overall, it is shown that a 22-nm transceiver would have an area and power consumption commensurate to that of current and future NoC designs, while representing between 1% and 10% of the area and power consumed by current core designs. The use of a secondary control channel in any of the wireless schemes (to transmit the NACKs in **W-CSMA**, for instance) would incur into additional overhead, yet much lower than for the data channel. These figures could be reduced by means of transceiver optimization, the system-level techniques described in Section 6.4, or the use of surface wave technologies bringing propagation losses down to transmission line levels [23], [46]. In any case, it is worth noting that the cost of the wireless channel for broadcasts could be in part compensated by the fact that the wired plane can be simplified.

6 Discussion

The results shown in Section 5 confirm the potential of wireless on-chip communication for low-latency global communication. Specifically, it is observed that wireless NoCs *1)* can outperform any wired topology in terms of latency; *2)* have potential to offer a throughput comparable to that of their wireline counterparts; and *3)* should be restricted to broadcast traffic and perhaps a small fraction of unicast messages. For all this, adding a wireless plane to a wired NoC will result in significant latency and throughput improvements in a wide range of settings.

These conclusions are valid provided that a flexible and efficient MAC mechanism is implemented. Next, we discuss possible optimizations and future research lines for the alternatives evaluated in this paper.

6.1 Optimizing W-CSMA

The good scalability of **W-CSMA** with respect to the system size makes it an interesting approach for broadcast-based wireless NoCs. However, its efficiency still needs to be improved in order to consider it as a serious contender. There are different design facets that affect performance and that must be carefully considered when optimizing a CSMA-like protocol.

Persistence is another important aspect to consider [48]. When seeing the channel busy, the p-persistent protocol grants access with probability p immediately after the ongoing transmission ends, or backs off with probability $1-p$. In this work, we assumed a non-persistent protocol ($p=0$). However, there exists an optimal persistence value depending on the traffic characteristics. In typical CSMA environments, this value is hard to find and to change in a consistent way at runtime in conventional wireless networks, resulting in suboptimal performance. In a chip environment, though, p could be evaluated more precisely given the relatively high knowledge on the traffic and even dynamically changed using a single broadcast message. Backoff times could be also precisely determined using the same principles, avoiding recurrent collisions in communication-intensive phases.

Knowledge or even control on the traffic characteristics should be systematically exploited to optimize the protocol beyond the conventional CSMA design decisions. Broadcast traffic shows considerable predictability in some architectures due to recurrent memory access patterns [8]. Such predictability is even higher if we consider the well-studied phenomenon of application phase behavior [67]. This means that with *multicast source prediction* [8] some applications could see unprecedented levels of CSMA performance. This process could even be assisted by the programmer or the compiler, which could insert instructions that would tell the protocol how to operate (e.g. define the backoff length, set priorities) to maximize performance and, most importantly, combat the detrimental effects of traffic burstiness.

Multiprogramming also opens an unconventional design space. In these workloads, different sets of cores execute different applications, each with its own phased communication requirements. In this case, the assertiveness of the protocol could still be managed on a per-application basis, but should be coordinated among the different applications (similarly to the way prefetchers are adjusted in multicore settings [68]) to maintain fairness and performance.

6.2 On the feasibility of W-CBUF

Although the **W-CBUF** strategy performs remarkably well, it is an unrealistic option due to the implementation complexity. Achieving centralized arbitration requires small control packets to traverse global links and an N-to-1 multiplexer, or being buffered, within the same clock cycle. This is a daunting task as the system size and frequency increase.

One alternative way to perform centralized arbitration would be to use the low bandwidth channel in the wireless medium to exchange access requests and grants. However, these can still collide. This can be avoided by performing arbitration by means of a hierarchy of simple multiplexers and on-chip wires that progressively lead to the central

TABLE 2
Per-Tile Area and Power Comparison

Ref.	Cores	Topology	Technology	Voltage	Frequency	Width	Area	Power
[35]	N	Wireless	65 nm (22 nm)	1 V	1 GHz	16 b (32 b)	0.25 mm^2 (0.1 mm^2)	31.2 mW
						32 b (64 b)	0.4 mm^2 (0.16 mm^2)	96 mW
						48 b (96 b)	0.73 mm^2 (0.3 mm^2)	97.5 mW
[3]	80	Mesh	65 nm	0.7 V	1.7 GHz	39 b	0.34 mm^2	98 mW
[12]	36	Mesh	45 nm	1.1 V	1 GHz	137 b	0.36 mm^2	139 mW
[20]	16	Mesh	45 nm	1.1 V	1 GHz	64 b	0.32 mm^2	27 mW
[63]	128	FBFly	32 nm	0.9 V	2 GHz	144 b	0.18 mm^2	78 mW
[26]	64	Optical Bus	22 nm	1 V	2.5 GHz	320 b	-	187.5 mW
[59]	64	RF Bus	22 nm	1 V	1 GHz	16 b	0.48 mm^2	7.8 mW
Atom Silvermont (22 nm)							2.5 mm^2	$\sim$1 W
Xeon Haswell (22 nm)							21.1 mm^2	$\sim$5 W

buffer. From an implementation standpoint, recent works [12] have shown that similar lightweight networks may consume less than the tile area and power. Our network performance results, not shown for brevity, demonstrate that this approach would cause an increase in latency but would have a negligible impact on throughput. This way, the problem boils down to striking a balance between latency requirements and implementation complexity.

6.3 Tearing down the performance barrier of W-TOKEN

The rigidity of the token passing scheme is the main barrier that prevents **W-TOKEN** from being a valid alternative in the manycore scenario. We assumed that passing the token takes one clock cycle per node, but this is clearly not enough in light of the results shown above, especially for hotspot traffic. By making the token ring asynchronous and allowing the token to traverse multiple nodes within the same clock cycle (assuming that these nodes do not have anything to transmit), the performance of **W-TOKEN** would greatly improve. This could be implemented with multi-hop asynchronous schemes such as the recently proposed in [65].

6.4 System-level cost reduction

The implementation cost of the wireless alternatives is basically driven by the area and power consumed by the transceiver. As pointed out in Section 5.6, these costs can be reduced by means of technology scaling, circuit optimization, and the use of simple modulations. Here, we point out some system-level design decisions that can be taken to further push the area and power down.

Power gating: in manycore processors, cores will likely be dynamically turned on and off to save power. The flexibility of the wireless approach allows transceivers to be power gated with their associated core without affecting network performance, fact that is generally not possible with routers.

Fine-grained power allocation: a fixed power consumption is assumed in Section 5.6. The reality, however, is that power can be allocated on a per-core basis statically based on its position or dynamically based on traffic demands [69].

Concentration: as outlined in Section 4.1, k cores can be clustered and connected to a single transceiver. This approach would increase latency by a few cycles and threaten the ordering guarantees, but would also reduce the area and power by a factor of k.

7 Conclusions

Motivated by the expected lack of efficient multicast mechanisms in manycore environments, we have evaluated the broadcast scalability of different WNoC schemes and compared it to that of aggressive NoC designs. The analysis considers full broadcast support in WNoCs through the integration of antennas on a per-core basis and the sharing of a single broadband channel among all cores. Besides enabling the ordered delivery of broadcast traffic, this scheme provides a latency up to one order of magnitude lower than the best evaluated wireline counterpart. Beyond a few hundreds of cores and in spite of its much lower bisection bandwidth, WNoC attains a broadcast throughput commensurate to that of conventional NoCs. For all this, we envisage a hybrid network architecture where a WNoC will serve broadcast traffic and a conventional NoC will transport the rest of communication flows. With such scheme, the latency is reduced dramatically for high levels of broadcast, whereas the throughput is significantly increased for low levels of broadcast. The improvement becomes more patent as the system size increases, ensuring the suitability of such hybrid approach in the manycore scenario. To achieve such goal, though, we stress the need of a channel capacity commensurate to the rate at which cores can inject data, as well as of a flexible and reasonably efficient MAC protocol. The latter requirement can be either met with common MAC protocols or amply exceeded by virtue of protocols that take advantage of the unique optimization advantages of the multiprocessor scenario.

Acknowledgments

The authors gratefully acknowledge support from INTEL's Doctoral Student Honor Program, as well as from Samsung's Global Research Outreach (GRO) program. This work has been also partially supported by the Catalan Government through a FI-AGAUR grant and by the Spanish State *Ministry of Economy and Competitiveness* under grant aid PCIN-2015-012.

References

[1] J. Hennessy and D. Patterson, *Computer architecture: a quantitative approach*. Morgan Kaufmann, 2012.

[2] 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, jun 2006.

[3] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, 2008.

[4] D. Bertozzi, G. Dimitrakopoulos, J. Flich, and S. Sonntag, "The fast evolving landscape of on-chip communication," *Design Automation for Embedded Systems (Springer)*, vol. 19, no. 1, pp. 59–76, 2015.

[5] N. E. Jerger, L.-S. Peh, and M. Lipasti, "Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support," in *Proceedings of the ISCA-35*. Ieee, jun 2008, pp. 229–240.

[6] T. Krishna, L. 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.

[7] 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.

[8] 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)*, 2016.

[9] A. Agarwal, R. Simoni, J. Hennessy, and M. Horowitz, "An evaluation of directory schemes for cache coherence," in *Proceedings of the ISCA-15*, 1988, pp. 280–289.

[10] P. Conway and B. Hughes, "The AMD Opteron Northbridge Architecture," *IEEE Micro*, vol. 27, no. 2, pp. 10–21, 2007.

[11] K. Strauss, X. Shen, and J. Torrellas, "Uncorq: Unconstrained snoop request delivery in embedded-ring multiprocessors," in *Proceedings of MICRO-40*, 2007, pp. 327 – 342.

[12] 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.

[13] M. Martin, "Token Coherence: decoupling performance and correctness," in *Proceedings of the ISCA-30*, 2003, pp. 182–193.

[14] 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.

[15] R. Boppana, S. Chalasani, and C. Raghavendra, "Resource deadlocks and performance of wormhole multicast routing algorithms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 6, pp. 535–549, jun 1998.

[16] F. A. Samman, T. Hollstein, and M. Glesner, "Multicast parallel pipeline router architecture for network-on-chip," in *Proceedings of DATE '08*. ACM Press, 2008, pp. 1396–1401.

[17] P. Abad, V. Puente, and J.-A. Gregorio, "MRR: Enabling fully adaptive multicast routing for CMP interconnection networks," in *Proceedings of HPCA '09*. Ieee, feb 2009, pp. 355–366.

[18] L. Wang, Y. Jin, H. Kim, and E. Kim, "Recursive partitioning multicast: A bandwidth-efficient routing for Networks-on-Chip," in *Proceedings of the NoCS '09*, 2009, pp. 64–73.

[19] M. Daneshtalab, M. Ebrahimi, T. C. Xu, P. Liljeberg, and H. Tenhunen, "A generic adaptive path-based routing method for MPSoCs," *Journal of Systems Architecture*, vol. 57, no. 1, pp. 109–120, 2011.

[20] 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 '12*, 2012, p. 398.

[21] 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.

[22] K. Sewell, R. G. Dreslinski, T. Manville, S. Satpathy, N. Pinckney, G. Blake, M. Cieslak, R. Das, T. F. Wenisch, D. Sylvester, D. Blaauw, and T. Mudge, "Swizzle-Switch Networks for Many-Core Systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 278–294, jun 2012.

[23] 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.

[24] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, J. Flich, and H. Tenhunen, "Path-Based Partitioning Methods for 3D Networks-on-Chip with Minimal Adaptive Routing," *IEEE Transactions on Computers*, vol. 63, no. 3, pp. 718–733, 2014.

[25] 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 '08*, 2008, pp. 191–202.

[26] R. Morris, E. Jolley, and A. K. Kodi, "Extending the Performance and Energy-Efficiency of Shared Memory Multicores with Nanophotonic Technology," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 83–92, 2014.

[27] 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.

[28] C.-K. Liang and M. Prvulovic, "MiSAR: Minimalistic Synchronization Accelerator with Resource Overflow Management," in *Proceedings of the ISCA-42*, 2015, pp. 414–426.

[29] S. Abadal, E. Alarcón, A. Cabellos-Aparicio, and J. Torrellas, "WiSync: An Architecture for Fast On-Chip Synchronization through Wireless-Enabled Global Communication," in *Proceedings of the ASPLOS '16*, 2016.

[30] M. Lodde, J. Flich, and M. E. Acacio, "Heterogeneous NoC Design for Efficient Broadcast-based Coherence Protocol Support," in *Proceedings of the NoCS '12*. Ieee, may 2012, pp. 59–66.

[31] L. Cheng, N. Muralimanohar, K. Ramani, R. Balasubramonian, and J. B. Carter, "Interconnect-Aware Coherence Protocols for Chip Multiprocessors," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 2, pp. 339–351, 2006.

[32] W.-C. Kwon, T. Krishna, and L.-S. Peh, "Locality-oblivious cache organization leveraging single-cycle multi-hop NoCs," in *Proceedings of ASPLOS '14*, 2014, pp. 715–728.

[33] 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.

[34] 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.

[35] X. Yu, J. Baylon, P. Wettin, D. Heo, P. Pande, and S. Mirabbasi, "Architecture and Design of Multi-Channel Millimeter-Wave Wireless Network-on-Chip," *IEEE Design & Test*, vol. 31, no. 6, pp. 19–28, 2014.

[36] 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.

[37] 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.

[38] 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.

[39] D. Matolak, A. Kodi, S. Kaya, D. DiTomaso, S. Laha, and W. Rayess, "Wireless networks-on-chips: architecture, wireless channel, and devices," *IEEE Wireless Communications*, vol. 19, no. 5, 2012.

[40] S. Abadal, M. Iannazzo, M. Nemirovsky, A. Cabellos-aparicio, 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. 1501–1513, 2015.

[41] 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.

[42] B. Khamaisi, S. Jameson, E. Socher, and S. Member, "A 210 227 GHz Transmitter With Integrated On-Chip Antenna in 90 nm CMOS Technology," *IEEE Transactions on Terahertz Science and Technology*, vol. 3, no. 2, pp. 141–150, 2013.

[43] I. Llatser, C. Kremers, A. Cabellos-Aparicio, J. M. Jornet, E. Alarcón, and D. N. Chigrin, "Graphene-based nano-patch antenna for terahertz radiation," *Photonics and Nanostructures - Fundamentals and Applications*, vol. 10, no. 4, pp. 353–358, 2012.

[44] Y. Wu, K. A. Jenkins, A. Valdes-Garcia, D. B. Farmer, Y. Zhu, A. Bol, C. Dimitrakopoulos, W. Zhu, F. Xia, P. Avouris, and Y.-M. Lin, “State-of-the-art graphene high-frequency electronics,” *Nano letters*, vol. 12, no. 6, pp. 3062–7, jun 2012.

[45] S. Abadal, E. Alarcón, M. C. Lemme, M. Nemirovsky, and A. Cabellos-Aparicio, “Graphene-enabled Wireless Communication for Massive Multicore Architectures,” *IEEE Communications Magazine*, vol. 51, no. 11, pp. 137–143, 2013.

[46] A. J. Karkar, J. E. Turner, K. Tong, R. Al-Dujaily, T. Mak, A. Yakovlev, and F. Xia, “Hybrid wire-surface wave interconnects for next-generation networks-on-chip,” *IET Computers & Digital Techniques*, vol. 7, no. 6, pp. 294–303, nov 2013.

[47] L. Roberts, “ALOHA packet system with and without slots and capture,” *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[48] L. Kleinrock and F. Tobagi, “Packet Switching in Radio Channels: Part I–Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics,” *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1400–1416, 1975.

[49] R. M. Metcalfe and D. R. Boggs, “Ethernet: distributed packet switching for local computer networks,” *Communications of the ACM*, vol. 19, no. 7, pp. 395–404, 1976.

[50] D. Clark, K. Pogran, and D. Reed, “An introduction to local area networks,” *Proceedings of the IEEE*, vol. 66, no. 11, pp. 1497–1517, 1978.

[51] B. Crow, I. Widjaja, L. Kim, and P. Sakai, “IEEE 802.11 Wireless Local Area Networks,” *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997.

[52] C. L. Fullmer and G.-L.-A. J. J, “Solutions to hidden terminal problems in wireless networks,” in *Proceedings ACM SIGCOMM*, 1997, pp. 39–49.

[53] V. Vijayakumaran, M. P. Yuvaraj, N. Mansoor, N. Nerurkar, A. Ganguly, and A. Kwasinski, “CDMA Enabled Wireless Network-on-Chip,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 10, no. 4, pp. 1–20, may 2014.

[54] D. Zhao and Y. Wang, “SD-MAC: Design and Synthesis of a Hardware-Efficient Collision-Free QoS-Aware MAC Protocol for Wireless Network-on-Chip,” *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1230–1245, 2008.

[55] R. Wu and D. Zhao, “Load adaptive multi-channel distribution and arbitration in unequal RF interconnected WiNoC,” in *Proceedings of the ISCAS ’14*, 2014, pp. 1973–1976.

[56] P. Dai, J. Chen, Y. Zhao, and Y.-H. Lai, “A study of a wire-wireless hybrid NoC architecture with an energy-proportional multicast scheme for energy efficiency,” *Computers and Electrical Engineering (Elsevier)*, vol. 45, pp. 402–416, 2015.

[57] N. Mansoor and A. Ganguly, “Reconfigurable Wireless Network-on-Chip with a Dynamic Medium Access Mechanism,” in *Proceedings of the NoCS ’15*, 2015, p. Art. 13.

[58] J. Chan, G. Hendry, A. Biberman, 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.

[59] 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 ’13*, 2013, pp. 309–318.

[60] V. Soteriou, H. Wang, and L. Peh, “A Statistical Traffic Model for On-Chip Interconnection Networks,” in *Proceedings of MASCOTS ’06*, 2006, pp. 104–116.

[61] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of Ethernet traffic (extended version),” *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[62] J. Oh, M. Prvulovic, and A. Zajic, “TLSync: support for multiple fast barriers using on-chip transmission lines,” in *Proceedings of ISCA-38*, 2011, pp. 105–115.

[63] D. Sanchez, 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. Art. 4, 2010.

[64] J. Kim, J. Balfour, and W. J. Dally, “Flattened butterfly topology for on-chip networks,” in *Proceedings of the MICRO-40*, 2007, pp. 172–182.

[65] T. Krishna, C. Chen, W. Kwon, and L. Peh, “Smart: Single-Cycle Multihop Traversals over a Shared Network on Chip,” *IEEE Micro*, vol. 34, no. 3, pp. 43–56, 2014.

[66] “Intel Corportation, Intel Products. ark.intel.com,” 2015.

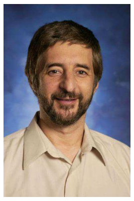
[67] T. Sherwood, S. Sair, and B. Calder, “Phase tracking and prediction,” *ACM SIGARCH Computer Architecture News*, vol. 31, no. 2, pp. 336–349, 2003.

[68] E. Ebrahimi, O. Mutlu, C. J. Lee, and Y. N. Patt, “Coordinated control of multiple prefetchers in multi-core systems,” in *Proceedings of the MICRO-42*, 2009, pp. 316–326.

[69] 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. PP, no. 99, 2015.

Sergi Abadal is a PhD student at the NaNoNetworking Center in Catalonia, Spain, at the Universitat Politécnica de Catalunya. His research interests include on-chip networking, many-core architectures, and graphene-based wireless communications. Abadal has an MSc in information and communication technologies from the Universitat Politécnica de Catalunya.

Albert Mestres is a PhD student at the NaNoNetworking Center in Catalonia, Spain, at the Universitat Politécnica de Catalunya. His research interests include network protocols, new architectures for the Internet, and machine learning. Mestres has an MSc in telecommunication engineering from the Universitat Politécnica de Catalunya.



Mario Nemirovsky is a Catalan Institution for Research and Advanced Studies (ICREA) Senior Research Professor at the Barcelona Supercomputing Center. His research interests include high-performance computing, the Internet of Things, and emerging on-chip interconnect technologies. Nemirovsky has a PhD from the University of California, Santa Barbara.

Heekwan Lee is a researcher in Samsung Electronics, Seoul, South Korea. His research interests include coding theory, cryptography, and Information theory and Security. Lee has a PhD in Electrical Engineering from University of Southern California, Los Angeles.

Antonio González is full professor at the Universitat Politécnica de Catalunya and leads the ARCO research group. His research interests include computer architecture, processor microarchitecture, and code generation and optimization techniques. González has a PhD in computer science and engineering from the Universitat Politécnica de Catalunya.

Eduard Alarcón is an associate professor at the NaNoNetworking Center in Catalonia at the Universitat Politcnica de Catalunya. His research interests include nanocommunications, energy harvesting, and wireless energy transfer. Alarcón has a PhD from the Universitat Politcnica de Catalunya.

Albert Cabellos-Aparicio is an associate professor at the NaNoNetworking Center in Catalonia at the Universitat Politécnica de Catalunya. His research interests include graphene technology, nanocommunications, and software-defined networking. Cabellos-Aparicio has a PhD in computer science engineering from the Universitat Politécnica de Catalunya.