A MAC protocol for Reliable Broadcast Communications in Wireless Network-on-Chip*

Albert Mestres
N3Cat
Univ. Politècnica de Catalunya
Barcelona, Spain
amestres@ac.upc.edu

Sergi Abadal
N3Cat
Univ. Politècnica de Catalunya
Barcelona, Spain
abadal@ac.upc.edu

Josep Torrellas
I-acoma Group
University of Illinois
Urbana–Champaign, IL, USA
torrella@illinois.edu

Eduard Alarcón
N3Cat
Univ. Politècnica de Catalunya
Barcelona, Spain
eduard.alarcon@upc.edu

Albert Cabellos-Aparicio
N3Cat
Univ. Politècnica de Catalunya
Barcelona, Spain
acabello@ac.upc.edu

ABSTRACT

The Wireless Network-on-Chip (WNoC) paradigm holds considerable promise for the implementation of fast and efficient on-chip networks in manycore chips. Among other advantages, wireless communications provide natural broadcast support, a highly desirable feature in manycore architectures yet difficult to achieve with current interconnects. As technology advancements allow the integration of more wireless interfaces within the same chip, a critical aspect is how to efficiently share the wireless medium while reliably carrying broadcast traffic. This paper introduces the {Broadcast, Reliability, Sensing} protocol (BRS-MAC), which exploits the particularities of the WNoC context to meet its stringent requirements. BRS-MAC is flexible and employs a collision detection and notification scheme that scales with the number of receivers, making it compatible with broadcast communications. The proposed protocol is modeled and evaluated, showing a clear latency advantage with respect to wired on-chip networks and WNoCs with token passing.

CCS Concepts

•Networks → Link-layer protocols; •Computer systems organization → Multicore architectures;

Keywords

Chip Multiprocessor, Wireless Network-on-Chip, Protocols, Media Access Control, Broadcast, Latency, Throughput

*This work was supported in part by NSF under grants CCF-1012759, CNS-1116237, and CCF-1536795; and by the *Spanish Ministry of Economy and Competitiveness* under grant TEC2014-59583-C2-2-R.

1. INTRODUCTION

Multicore architectures achieve high performance at low operation frequencies by exploiting the parallelism inherent to most applications. To this end, Chip Multiprocessors (CMPs) integrate a number of processor *cores* that cooperate towards a common goal. Since more cores are expected to imply a higher potential for performance, recent years have seen a drastic increase in the number of cores per chip, which eventually led to the *manycore era*. These higher levels of integration have raised the importance of inter-core communication, to the point of becoming one of the main performance bottlenecks in manycore CMPs. Within this context, the Network-on-Chip (NoC) paradigm have gradually replaced bus-based interconnects due to its higher scalability, efficiency, and fault tolerance [8].

Despite being widely considered nowadays, the NoC paradigm also has several drawbacks when approaching the manycore horizon. Technology scaling reduces the intrinsic performance and efficiency of wires, discouraging their use as global links. NoC topologies with short links are thus preferred, decision that negatively impacts the performance of global and broadcast traffic. CMP architects are therefore compelled to avoid such types of communication, severely constraining their practical design space and progressively leading to slow and complex processors.

In light of these issues, novel interconnect paradigms have emerged to complement conventional NoCs [10]. Here we focus on the WNoC paradigm, which aims to interconnect cores via wireless communication. This approach is possible by virtue of the availability of Radio-Frequency (RF) antennas and circuits small enough for their on-chip integration [13]. The main advantages of WNoC with respect to other emerging alternatives are *(i)* its low latency for global messages, given by the speed-of-light propagation of RF waves; *(ii)* its non-intrusiveness and flexibility, given by the lack of a physical topology; and *(iii)* its inherent broadcast capabilities, given by the radiated nature of signals.

On the one hand, most of the existing WNoC proposals highlight the latency benefits of the technology to implement point-to-point links between distant cores [7]. The main issue here is that RF transmission lines and nanophotonic networks are also good candidates for this application [10]. On

Table 1: Wireless Manycore Scenario Requirements

Metric	Value
Transmission Range	0.1–10 cm
Node Density	10–1000 nodes/cm^2
Throughput	10–100 Gbps
Latency	1–100 ns
Energy	1–10 pJ/bit
Bit Error Rate (BER)	10^{-15}

the other hand, less attention has been placed on the natural broadcast capabilities of the WNoC paradigm, which could have a transformative impact on manycore architectures and applications [1, 4]. It is important to note that, although RF transmission lines and nanophotonic networks have been also considered for a broadcast plane [9], design complexity and laser power problems may hinder their use as scalable and globally shared media.

To realize the broadcast-oriented WNoC approach, it is *desirable* to integrate one antenna every few cores and to share a small set of broadband channels. This, nonetheless, requires pushing the transmission frequency to the millimeter-Wave (mmWave) band and beyond [13, 15], and devising a Medium Access Control (MAC) strategy capable of coping with stringent requirements of the scenario.

In this paper, we focus on the MAC issue and propose BRS-MAC, a protocol specifically suited to the broadcast-oriented WNoCs. We observed that existing works on WNoC either rely on channelization or token passing schemes that do not scale well [7, 9], or map protocols to the on-chip scenario without considering broadcast patterns [6, 14]. Instead, our work takes advantage of the unique application context to propose a scalable and reliable solution, as well as to develop highly accurate performance models. We use such models to perform a rigorous analysis of its throughput and delay of BRS-MAC, and then compare them with those of representative NoCs designs. We demonstrate that, with BRS-MAC, the broadcast latency can be potentially reduced between one and two orders of magnitude.

The remainder of this paper is organized as follows. Sec. 2 provides background on the WNoC scenario. Sec. 3 analyzes the application context from the MAC standpoint. The design decisions of BRS-MAC are described in Sec. 4. Then, the performance models of BRS-MAC are derived in Sec. 5. Sec. 6 validates them and evaluates BRS-MAC. Sec. 7 concludes the paper.

2. BROADCAST-ORIENTED WIRELESS NETWORK-ON-CHIP

The WNoC paradigm emerges as a response to the communication needs of manycore processors, roughly quantified in Tab. 1. Increasing the core density implies a significant increment in the intensity, variability, and heterogeneity of a load that, in turn, needs to be served reliably while placing stronger emphasis on energy efficiency [10] and latency [18]. An evidence of the increasing heterogeneity of communication is the larger importance of multicast transactions, which become more frequent and reach more destinations [2]. This is detrimental to most NoC designs, where multicast packets need to be replicated multiple times, causing bursts of contention and eventually leading to severe performance losses. Advanced router designs [12] and high-radix topologies [5] may alleviate these issues, but high levels of broadcast will still *flood* the network and affect unicast traffic as well.

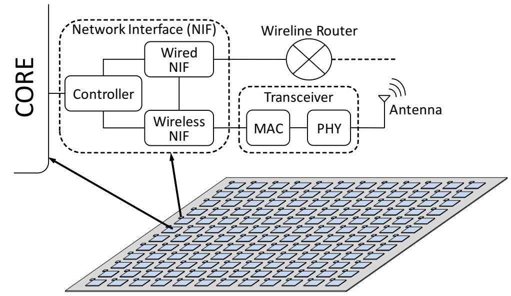

Figure 1: Schematic diagram of a manycore processor integrating a wireline NoC and a WNoC.

Using a WNoC to serve multicast traffic, for which it is uniquely suited, can be a way to tackle this heterogeneity problem. As depicted in Fig. 1, a broadcast-oriented wireless plane could complement a wired NoC that would mostly deal with unicast transmissions. Both planes would be coordinated through a controller, forming a hybrid scheme that has been demonstrated to achieve significant performance improvements in a wide variety of cases [1, 3].

Let us consider that each core is equipped with a wireless communication unit composed by a transceiver and an antenna, as shown in Fig. 1. Currently, a 65-nm CMOS transceiver at 60 GHz reportedly reaches 16 Gbps with a BER of 10^{-15} at the chip scale, consuming 2 pJ/bit and 0.25 mm^2 [20]. This allows to collocate one antenna and transceiver every few cores, whereas downscaling trends and new RF technologies are expected to enable a per-core integration in the future [13, 15].

The proper characterization of the time-invariant on-chip propagation channel [16] allows to place all antennas within the same transmission range. Therefore, the scheme in Fig. 1 can provide natural support for broadcast as long as all antennas are tuned to the same channel. This, however, places a large pressure on the MAC layer due to the potentially large density of wireless nodes; a pressure that current MAC strategies for on-chip communication may not be able to withstand for the reasons detailed next.

Existing WNoC works do not share the broadcast-oriented philosophy due to the size of current antennas, and instead propose to lay a limited number of wireless point-to-point links over a wired topology to communicate distant cores. In these cases, the MAC protocol can consist in the creation and assignment of orthogonal channels via frequency, time, code, space multiplexing, or any combination thereof [7, 9]. Another approach is the use of coordinated access protocols such as token passing [7]. Both the channelization and coordinated access schemes proposed thus far offer a high throughput capability, but become impractical in high-density networks since each additional channel increases the complexity of the transceiver or reduces the overall performance. Also, these solutions are inherently rigid and may not perform well with highly variable traffic.

Few papers have explored random access or on-demand solutions, which can provide low latency when the load is moderate and adapt to changes in traffic. Dai *et al* propose the use of slotted Carrier Sense Multiple Access (CSMA) with optimal persistence calculated *a priori* [6]. Also, Mansoor *et al* present an adaptive scheme that switches between

CSMA and token passing depending on the level of contention [14]. These works show novelty, but do not detail how would multicasts be acknowledged or how the protocol scales. In next sections, we analyze the application context and present a protocol that aims to tackle these problems by exploiting the specificities of the on-chip scenario.

3. MAC CONTEXT ANALYSIS

The design of a MAC protocol is highly influenced by the application context. Here, we analyze three important facets that will guide the design of BRS-MAC.

Application Requirements: WNoC exhibits low latency, a feature highly desirable in the manycore scenario [1, 18]. Also, processors cannot allow errors as they would compromise the correctness of the program being run. Therefore, the protocol should be simple, fast, and reliable.

Chip Scenario: cores will have a very limited area and power budgets. Thus, the protocol should be lightweight and scalable. Besides this, it is worth noting that the physical landscape is static and known beforehand. This virtually eliminates the randomness of the propagation [16], allowing to *(i)* assume that all nodes can be placed within a single transmission range, *(ii)* disregard hidden terminal situations, *(iii)* reach consensus easily, *(iv)* consider collision detection schemes, and *(v)* create high-accuracy performance models. Finally, the existence of an underlying wired network can help in the design in different ways.

Traffic Characteristics: traffic in manycore settings is highly heterogeneous, but we envisage that WNoC will carry broadcast traffic. Therefore, we need reliability to be ensured even for broadcast messages. We also need the protocol to be flexible due to the high variability of traffic within and between applications [1, 2].

4. THE BRS-MAC PROTOCOL

The BRS-MAC protocol takes its name from the triplet {Broadcast, Reliability, Sensing} as it is designed to reliably transmit broadcast messages by taking advantage of the sensing opportunities offered by the chip environment. In essence, BRS-MAC is a solution between CSMA/CA and CSMA/CD thanks to the use of preamble-based collision detection [19]. Both the simplicity of the approach and the use of collision detection in a wireless network are unique to the on-chip context.

4.1 Basic Algorithm

When a node is ready to send data, it will only transmit if the channel is sensed idle, thereby preventing the interruption of on-going transmissions; otherwise, the node backs off. In the former case, the transmission starts and the protocol is as summarized in Fig. 2:

Step 1 – Preamble Transmission: the sender will transmit a preamble and then wait. Nodes that correctly receive these initial bits will remain silent during the remainder of the transmission, whereas nodes detecting a collision will start transmitting a Negative ACKnowledgment (NACK) in the next step.

Step 2 – Collision Handling: at this phase, nodes that detected the collision send a NACK signal through the same channel than data. The rest of nodes listen for NACK signals and cannot initiate a new data transmission. In the presence of a NACK signal, nodes assume there was a collision and therefore skip step 3. Before that, original senders cancel their transmission and back off, whereas the rest of nodes discard the preamble. In the absence of a NACK signal, the preamble transmission is considered successful and nodes proceed to step 3. By detecting collisions here, BRS-MAC reduces their penalty as it avoids the unnecessary transmission of the whole message.

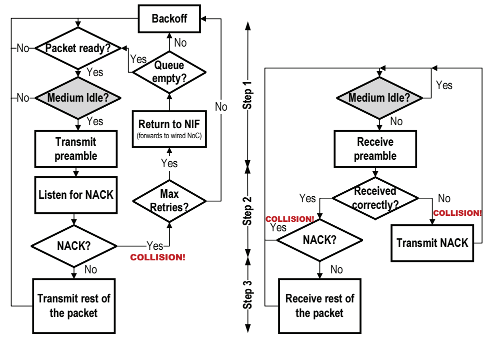

Figure 2: Flowchart of the BRS-MAC protocol for the transmitter (left) and the receiver (right).

Step 3 – Data Transmission: this step is only executed in the absence of collisions. Here, the source transmits the rest of the packet. The length of the data transmission depends on the packet size and transmission speed, both known since the transmission speed is fixed and the packet size can be indicated in the preamble with few bits as packet lengths take few different values [2].

4.2 Protocol Decisions

Channel Sensing and Collision Detection: This work considers that nodes can sense the presence of signals in the medium and detect collisions under certain conditions. This is an assumption unique to this scenario due to the following. In macro-networks, collision detection can only be performed in wired networks, e.g. Ethernet, where nodes can sense the channel while emitting. Outgoing and incoming signals have similar power, such that collisions can be detected by performing a simple comparison of those signals. In wireless systems, this scheme cannot be reproduced since received signals are generally masked by much stronger transmitted signals [19].

In light of the above, BRS-MAC leaves the collision detection responsibility to the receivers. In macro-networks, this approach would not be feasible as the channel generally changes with time. Instead, the uniqueness of the on-chip scenario with respect to the time-invariant and known propagation medium reinforces the practicality of collision detection. Different ways to detect collisions can be explored, namely: *(i)* a redundancy check performed to the preamble of the transmission; *(ii)* a comparison between the received amplitude or phase and the expected value depending on the source address, also indicated in the preamble; or *(iii)* the use correlation techniques to evaluate the integrity of unique signatures placed at the preamble [19].

Transmission Preamble: The preamble carries headers that allow to detect collisions in step 2 and determine the length of step 3. Its size should be large enough to ensure the reliability of the collision detection, but still short to reduce the penalty of collisions. Sec. 6 analyzes the impact of the preamble size on the performance of BRS-MAC.

Acknowledgment policy: To relax the requirements on the collision detection, BRS-MAC allows collisions to go unnoticed at certain nodes but later warns them using a NACK. Sending explicit ACK or NACK packets would be impractical as it would require the serialization of many messages. To avoid this *ACK implosion*, we resort to the collective communication scheme outlined in [17]. We model the NACKs as a *tone* and let a selection of nodes transmit it through the wireless medium at once. With a proper design, tones do not collide but are rather aggregated, which effectively scales with the number of nodes. The presence of signal is interpreted as a NACK and prompts the original transmitter to back off and retry.

Retransmission policy: BRS-MAC uses the widely known Binary Exponential Backoff (BEB) strategy, where the backoff length r depends on the number of attempts att as

$$r = r_0(2^{att} - 1). \quad (1)$$

r_0 is the minimum backoff, assumed to be equal to the average transmission time T here. This approach dynamically modifies the assertiveness of the protocol depending on the load; a choice driven by the variability of traffic outlined in Sec. 3. Also, to bound the maximum latency, packets that exceed a given number of retries will be forwarded to the wired network by returning the packet to the wireless NIF which, upon observing that the source address of the packet matches with the local network *id*, will redirect the message to the wired NIF. This ensures that all packets are delivered, fact that is not guaranteed in conventional wireless networks where the packet is discarded after a given number of retries.

5. PERFORMANCE MODELS

This section presents the analytical models for the performance of BRS-MAC. We use the models for non-persistent CSMA from [11] as starting point. We maintain the definitions of throughput S, offered traffic G, and delay D. We keep the propagation time a and add the preamble time b as main parameters. All variables are normalized by the average transmission time T.

The models also maintain most of the system assumptions from [11], which let us obtain closed-form expressions. We basically consider that the time required to switch between TX and RX modes and to detect the channel busy are negligible. All arrivals (including retries) follow a Poisson process and are uniformly distributed among an infinite population, which is reasonable in manycore processors.

5.1 Propagation Time in Manycore Processors

As in traditional models [11], we first consider an equal (worst-case) propagation time $a \simeq \frac{WC\sqrt{2}}{v_p L}$ between nodes, where $W \sim$ 1cm is the side of a square chip, C is the channel capacity, v_p is the propagation speed, and $L \sim$ 100bits is the average packet length. This assumption is generally taken due to the unknown and dynamic position of nodes. However, the static and time-invariant nature of the channel [16] allows to determine the exact propagation time among nodes and, therefore, create more accurate performance models. To this end, we relate the average distance between cores with the a parameter as follows.

Let us define α as the average distance among any pair of nodes in a square of diagonal one. If nodes densely distributed within a grid, which would be the case of a manycore processor, α can be evaluated using *Square Line Picking*. Such method calculates the average distance $\Delta(2)$ among two arbitrary points in a unitary square as $\Delta(2) = 0.512$. Normalizing, we obtain the distance between cores as

$$\alpha = \frac{1}{N(N-1)} \sum_{\forall i} \sum_{\forall j \neq i} d(i,j) \simeq \frac{\Delta(2)}{\sqrt{2}} \simeq 0.3687, \quad (2)$$

where $d(i,j)$ represents the distance between nodes i and j.

5.2 Throughput Model

The throughput of the network is calculated as:

$$S = \frac{E\{U\}}{E\{B\} + E\{I\}} \quad (3)$$

where $E\{U\}$, is the expected occupancy of the channel of successfully transmitted messages. The term $E\{B\} + E\{I\}$ represents a cycle, i.e. the time between two transmissions taking into consideration the average duration of the busy and idle periods, respectively. The throughput takes a value between 0 and 1, and *can be seen as a metric of the effective use of the wireless medium* [11].

The expected duration of successfully used slots $E\{U\}$ can be obtained by multiplying the useful transmission time and the probability of success, which corresponds to the probability that no terminal starts a transmission during the propagation time a. By assuming Poisson traffic and an identical propagation time among all nodes, we have

$$E\{U\} = e^{-aG}. \quad (4)$$

The expected duration of the idle period $E\{I\}$ is given by:

$$E\{I\} = \frac{1}{G}. \quad (5)$$

The expected duration of the busy period $E\{B\}$ takes into account the propagation times on the duration of successful and colliding transmissions. In the former case, the channel is occupied during $1 + 2a$; whereas, in the latter case, the *jamming* nature of the NACK burst ensures that the cancellation takes effect after $b + 2a$. Thus, we have:

$$E\{B\} = e^{-aG}(1 + 2a) + (1 - e^{-aG})(b + 2a). \quad (6)$$

Finally, we obtain the throughput with Equation (3) as

$$S = \frac{e^{-aG}}{e^{-aG}(1-b) + b + 2a + 1/G}. \quad (7)$$

Exact propagation time: considering the exact propagation time, the expected duration of successfully used slots and of busy periods (now noted as $E\{U^e\}$ and $E\{B^e\}$) change. In the former case, $E\{U^e\}$ is proportional to the probability of not interrupting a current transmission, which now takes a different value for each pair of nodes. Assuming independent and equally distributed traffic, $E\{U^e\}$ becomes the average probability for each pair of links

$$E\{U^e\} = \frac{1}{N} \frac{1}{N-1} \sum_{\forall i} \sum_{\forall j \neq i} e^{a_{i,j}G}. \quad (8)$$

Here, it is worth noting that this expression cannot be simplified. However, when the probability of collision is low, it is possible to use the Taylor approximation for the exponential term ($e^x \simeq 1 - x$) and apply Eq. (2) to obtain

$$E\{U^e\} \simeq 1 - G\alpha a. \quad (9)$$

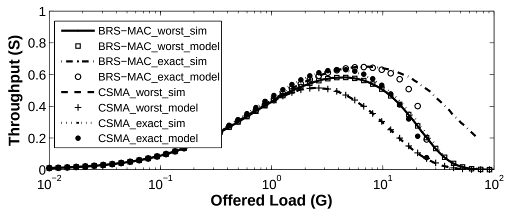

Figure 3: Simulated and analytical throughput for worst-case and exact propagation times.

In the case of $E\{B^e\}$, we need to evaluate the average duration of the busy period for each pair of nodes i, j. This requires re-evaluating Equation (6) which, using the Taylor expansion approximations outlined above, becomes

$$E\{B^e\} \simeq 1 + (2+\alpha)a - (1-b)G\alpha a. \qquad (10)$$

Thus, the throughput S^e can be re-calculated using Equation (3), which yields

$$S^e \simeq \frac{(1 - G\alpha a)}{1 + (2+\alpha)a - (1-b)G\alpha a + 1/G}. \qquad (11)$$

5.3 Delay Model

Here, we calculate the average time required to successfully transmit a packet. We define the transmission delay D as the time between the generation of a packet and its correct delivery to *all* receivers. This can be expressed as

$$D = N_{re}R + N_c T_{KO} + T_{OK}, \qquad (12)$$

where N_{re} stands for the average number of retransmissions, $R = \sum_1^{N_{re}} r/N_{re}$ is the average duration of the backoff period, N_c stands for the average number of collisions per transmission; whereas T_{OK} and T_{KO} are the delays incurred by successful and colliding transmissions, respectively.

To calculate the average number of retransmissions N_{re}, note that the offered rate G includes retransmissions, whereas the throughput S only considers successful attempts. Thus,

$$N_{re} = \frac{G}{S} - 1 \qquad (13)$$

To calculate the number of collisions per packet N_c, let us denote P_b as the probability of finding the channel busy within a cycle. Thus, we have that $1 - P_b = \frac{a+1/G}{E\{B\}+E\{I\}}$ as arrivals see the medium free in idle periods plus when a transmission is still being propagated [11]. From this, we can obtain the average number of attempts that find the channel free as $(1-P_b)\frac{G}{S}$ [11]. Since the average number of collisions is the number of attempts minus one, we have

$$N_c = \frac{a + 1/G}{E\{B\} + E\{I\}}\frac{G}{S} - 1. \qquad (14)$$

Exact propagation time: when considering exact propagation times, the delay model is slightly different. We basically need to consider S^e instead of S in Eq. (14), and reevaluate the average number of collisions as

$$N_c^e = \frac{\alpha a + 1/G}{E\{B^e\} + E\{I\}}\frac{G}{S^e} - 1. \qquad (15)$$

6. VALIDATION AND RESULTS

Next, we validate the BRS-MAC protocol models and evaluate BRS-MAC in a wide variety of configurations and using the non-persistent CSMA as baseline for comparisons. Unless noted, we assume $a = 0.1$, $b = 0.1$, and a transmission time of $T = 1$ ns, achievable with speeds around 100 Gbps considering the length of packets in the CMP scenario [2].

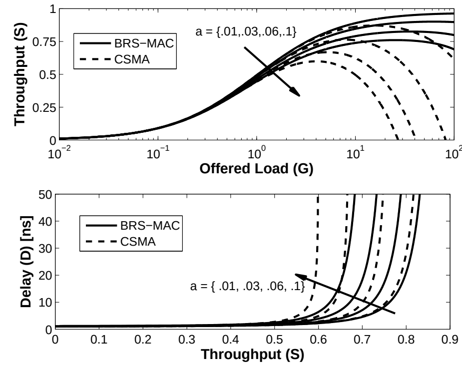

Figure 4: Throughput and delay characteristics as functions of the propagation time.

To validate the models, we implement the protocols in an in-house simulator based on Omnet++. Fig. 3 compares the throughput obtained with the simulator and the models for both BRS-MAC and CSMA, considering both worst-case and exact propagation time. The plots reveal a good agreement between the model and the simulation results for $G \leq 10$, which is the input range of interest. A significant performance improvement is also observed when considering the exact propagation time. In light of its validity, we will henceforth use the exact propagation time model.

Fig. 4 plots the throughput and delay characteristics of the evaluated protocols considering different propagation times. Increasing the propagation time negatively impacts on performance due to an increase of the collision rate. Thanks to its early reaction to collisions, BRS-MAC outperforms CSMA for all propagation times, with the peak being 10%–26% higher. For a target latency of 50 ns, BRS-MAC admits 2%–13% more throughput.

Let us now evaluate the impact of the preamble time on the performance of BRS-MAC. Fig. 5 shows the throughput and delay characteristic of BRS-MAC for different preamble lengths. The plot reveals that handling collisions using the same channel than data implies an overhead respect to the ideal CSMA model. This is compensated when collisions are handled at the beginning of the transmission ($b = 0.1$), which in fact improves throughput up to a 27%. A similar behavior is observed with respect to the delay: at 50 ns, the throughput increases up to 13% with respect to CSMA. Note that $b = 0.1$ corresponds to a preamble of around 10–30 bits in light of the average packet length in CMPs [18].

To contextualize the performance of BRS-MAC, we finally compare its throughput-delay characteristic with that of a WNoC with token passing (*TOKEN*) and a very aggressive wired NoC with multicast support (*MESH*). In *TOKEN*, the passing of the token takes 1 ns per node and cannot be overlapped with the data transmission; whereas in *MESH*, routers are provided with multiport allocation, multicast crossbars, and balanced tree routing algorithm [12], which allow each hop (link+router) to take 2 ns. All networks have the same per-node bandwidth and all packets are broadcast. Fig. 6 shows the results of the comparison, revealing that both *MESH* and *TOKEN* see their performance reduced when the number of cores increases due to the in-

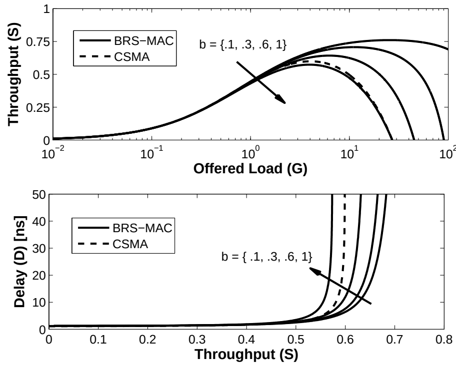

Figure 5: Throughput and delay as functions of the position of the preamble.

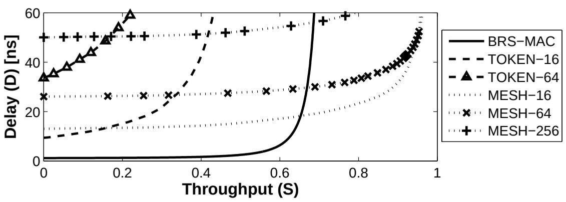

Figure 6: Delay as a function of the throughput for different NoC designs.

crease of the network diameter and of the average token passing delay, respectively. The performance of BRS-MAC does not change with the number of nodes [3] and cuts down the latency down to a few nanoseconds until ∼60% of the per-core capacity is reached. Note that BRS-MAC would still achieve the best latency even reducing the wireless network capacity to $C \approx 10$ Gbps.

7. CONCLUSIONS

We have presented BRS-MAC, a protocol for the reliable delivery of broadcast traffic in WNoCs. The key design aspects are the preamble-based collision detection and the scalable NACK scheme. Through performance modeling, we have shown that BRS-MAC achieves a peak throughput 27% higher than conventional CSMA protocols, and a latency between one and two orders of magnitude lower than aggressive NoCs for moderate loads.

8. REFERENCES

[1] 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*, pages 3–17, 2016.

[2] S. Abadal, R. Martínez, J. Solé-Pareta, *et al.* Characterization and Modeling of Multicast Communication in Cache-Coherent Manycore Processors. *Computers and Electrical Engineering (Elsevier)*, 2016.

[3] S. Abadal, A. Mestres, E. Alarcón, *et al.* Scalability of Broadcast Performance in Wireless Network-on-Chip. *IEEE Transactions on Parallel and Distributed Systems*, PP(99):1–14, 2016.

[4] S. Abadal, B. Sheinman, O. Katz, *et al.* Broadcast-Enabled Massive Multicore Architectures: A Wireless RF Approach. *IEEE MICRO*, 35(5):52–61, 2015.

[5] N. Abeyratne, R. Das, Q. Li, *et al.* Scaling towards kilo-core processors with asymmetric high-radix topologies. In *Proceedings of the HPCA-19*, pages 496–507, 2013.

[6] 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)*, 45:402–416, 2015.

[7] S. Deb, A. Ganguly, P. P. Pande, *et al.* Wireless NoC as Interconnection Backbone for Multicore Chips: Promises and Challenges. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(2):228–239, 2012.

[8] Y. Hoskote, S. Vangal, A. Singh, *et al.* A 5-GHz mesh interconnect for a teraflops processor. *IEEE Micro*, 27(5):51–61, 2007.

[9] A. Karkar, T. Mak, K. Tong, *et al.* A Survey of Emerging Interconnects for On-Chip Efficient Multicast and Broadcast in Many-Cores. *IEEE Circuits and Systems Magazine*, 16(1):58–72, 2016.

[10] J. Kim and K. Choi. Exploiting New Interconnect Technologies in On-Chip Communication. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(2):124–136, 2012.

[11] 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*, 23(12):1400–1416, 1975.

[12] 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*, pages 71–82, 2011.

[13] S. Laha, S. Kaya, D. W. Matolak,*et al.* 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*, 34(2):186–198, 2015.

[14] N. Mansoor and A. Ganguly. Reconfigurable Wireless Network-on-Chip with a Dynamic Medium Access Mechanism. In *Proceedings of the NoCS '15*, 2015.

[15] O. Markish, B. Sheinman, O. Katz, *et al.* On-chip mmWave Antennas and Transceivers. In *Proceedings of the NoCS '15*, Art. 11, 2015.

[16] D. Matolak, S. Kaya, and A. Kodi. Channel modeling for wireless networks-on-chips. *IEEE Communications Magazine*, 51(6):180–186, 2013.

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

[18] 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*, 7(1), 2010.

[19] S. Sen, R. Roy Choudhury, and S. Nelakuditi. CSMA/CN: Carrier Sense Multiple Access with Collision Notification. In *Proceedings of the MobiCom '10*, pages 25–36, 2010.

[20] X. Yu, J. Baylon, P. Wettin, *et al.* Architecture and Design of Multi-Channel Millimeter-Wave Wireless Network-on-Chip. *IEEE Design & Test*, 31(6), 2014.