

# Multi-channel Medium Access Control Protocols for Wireless Networks within Computing Packages

Bernat Ollé\*, Pau Talarn\*, Albert Cabellos-Aparicio\*, Filip Lemic†, Eduard Alarcón\*, and Sergi Abadal\*

\*NaNoNetworking Center in Catalunya (N3Cat), Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

†AI-driven Systems Lab, i2Cat Foundation, 08034 Barcelona, Spain

**Abstract**—Wireless communications at the chip scale emerge as an interesting complement to traditional wire-based approaches thanks to their low latency, inherent broadcast nature, and capacity to bypass pin constraints. However, as current trends push towards massive and bandwidth-hungry processor architectures, there is a need for wireless chip-scale networks that exploit and share as many channels as possible. In this context, this work addresses the issue of channel sharing by exploring the design space of multi-channel Medium Access Control (MAC) protocols for chip-scale networks. Distinct channel assignment strategies for both random access and token passing are presented and evaluated under realistic traffic patterns. It is shown that, even with the improvements enabled by the multiple channels, both protocols maintain their intrinsic advantages and disadvantages.

## I. INTRODUCTION

Efficient integrated networks at the chip scale within Systems-in-Package (SiPs) are a prerequisite for high performance in such computing systems. Currently, most systems incorporate a Network-in-Package (NiP) consisting of a set of on-chip routers and intra-/inter-chip wired links [1], [2]. However, recent scaling [3], [4], specialization [5], [6], and disintegration trends [7], [8] are increasing the pressure placed on the interconnect, to the point that new communication paradigms may be required [9], [10].

Among the emerging alternatives, wireless chip-scale communications stand as a promising contender [11]–[14]. This communication paradigm relies on the use of modulated electromagnetic waves for data transmission using the chip package as communications medium (Fig. 1). The resulting *wireless in-package links* provide low latency, inherent broadcast capabilities, and global reconfigurability.

Since the communications medium is shared, wireless in-package communications require Medium Access Control (MAC) protocols to avoid or manage wasteful collisions. In this scenario, MAC protocols generally reduce to variants of multiplexing, random access, or token passing [15]–[21]. Even though recent works have demonstrated that computing packages could support a few frequency [22], [23] and space channels [24], [25], it is still unclear how MAC protocols can benefit from them. This is because more than a few channels are needed to implement truly scalable frequency/space multiplexing techniques [15], and most importantly, because multi-channel variants of random access and token passing have not been explored yet.

This work is supported by the European Commission under H2020 grant WiPLASH (GA 863337) and HE grant WINC (GA 101042080).

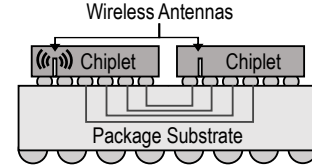


Fig. 1. Pictorial view of a wireless chip-to-chip communication link.

This paper aims to bridge this gap by focusing on the study of multi-channel versions of the two most representative protocol types in chip-scale scenarios, i.e. random access and token passing. In particular, the main contributions are as follows. We first describe the different ways we can extend random access and token passing with a small set of channels in Sec. II. Then, in Sec. III, we evaluate these protocol variants with traffic models typically used to mimic multiprocessor workloads [26]. This analysis sheds light on the impact of channel assignment on the protocol performance, as summarized in Sec. IV and concluded in Sec. V.

## II. MULTI-CHANNEL MAC PROTOCOLS

In this work, we describe three distinct channel assignment strategies for random access and token passing. As baselines, we take BRS [18] for random access and the baseline from [20] for token passing. The strategies presented here are not provably optimal, but they are simple (as required by the resource constraints of the chip-scale scenario) and representative of the potential techniques that can be used.

### A. Assignment Methods for BRS

In random access protocols such as BRS [18], nodes contend for channel access and back off if the channel is busy or there is a collision. Assuming  $N$  nodes, we study three ways to reduce the collision probability using  $N_c$  channels, namely:

**AS1:** Channels are assigned to nodes individually and randomly. When a node has a packet to transmit, the node is assigned a random channel. If the channel is busy or there is a collision, nodes undergo a random back off and also choose a random channel to use in the next attempt.

**AS2:** Each channel is assigned to  $\frac{N}{N_c}$  nodes statically following a uniform distribution, this is, assuming that all nodes have the same load (see Fig. 2, left). While this is not optimal for spatially unbalanced traffic, it serves as a baseline.

**AS3:** Channels are assigned to a variable number of nodes following a distribution that balances the load in each channel (see Fig. 2, right). To that end, nodes are ordered based on

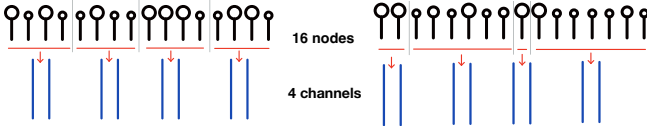


Fig. 2. Graphical representations of assignment techniques AS2 (left) and AS3 (right) for BRS assuming 16 nodes and 4 channels.

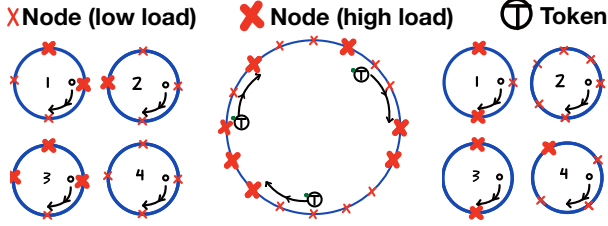


Fig. 3. Graphical representations of the different assignment techniques for token passing assuming 16 nodes and 4 channels.

the expected normalized load and assigned to each channel in order following a greedy algorithm.

### B. Assignment Methods for Token Passing

In token passing [27], typically, all  $N$  nodes are sorted forming a virtual ring and the token is passed in order through that ring. In a version with  $N_c$  channels, each channel can be a token. The design decisions then lie on the number of rings and the nodes that form each ring. For instance:

**AS1:** We assume as many rings as there are channels and map nodes uniformly to each ring. In other words, we distribute them in rings of  $\frac{N}{N_c}$  nodes, regardless of their expected load.

**AS2:** We assume a single virtual ring with multiple tokens circulating in it. In this case, tokens can jump over other tokens: when node  $i$  holds a token for multiple cycles during a transmission, idle tokens that arrive at  $i-1$  can jump to  $i+1$ .

**AS3:** This strategy is similar to AS1, but nodes are mapped to rings based on their expected load. This may lead to rings of different sizes, but similar in the expected overall load.

## III. PERFORMANCE EVALUATION

The architecture and application parameters are summarized in Table I. We implement both single-channel baselines and multi-channel versions of BRS and token passing as finite state machines in a modified version of Multi2sim that models wireless links and supports collision detection [28]. The protocols are stressed with synthetic traffic modeling uneven injection distributions (through the  $\sigma$  parameter) and bursty temporal behavior (through the Hurst exponent  $H$ ) [26]. The default values for the different parameters are  $N = 64$  nodes,  $N_c = 4$  channels,  $H = 0.5$  and  $\sigma = 1$ . Simulations are cycle-accurate.

In all cases, we compare the packet latency (in cycles) and throughput (in packets/cycle) of the different options. Given the high number of protocol strategies and traffic types, instead of plotting the classical latency-throughput curve, we make use of box plots that summarize the latency and throughput statistics. In our plots, the X axis shows the parameters under study. The plots have two Y axis: the left axis represents the latency and corresponds to the box plot values, whereas

TABLE I CHARACTERISTICS OF SIMULATED PROTOCOLS AND APPLICATIONS.	
Application	Synthetic traffic, $H=0.5-0.85$ , $\sigma=0.05-100$
System	$N=64-512$ cores, one antenna/core, 1-GHz clock
Network	80-bit packets (preamble: 20 bits), $N_c=1-4$ channels
Link	BRS [18], Token passing
Physical	On-Off Keying, 20 Gb/s

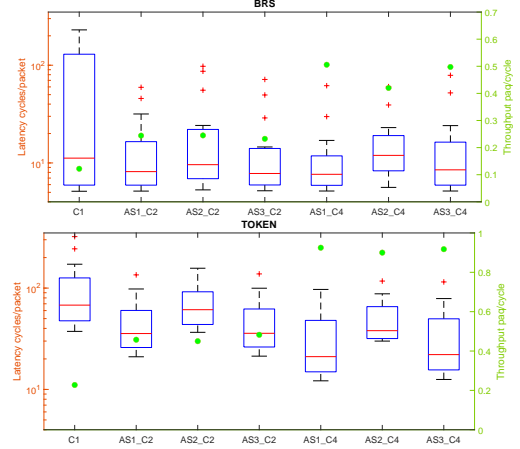


Fig. 4. Performance of multi-channel BRS (top) and token passing (bottom) for an increasing number of channels,  $C1$  to  $C4$ , and different assignments.

the right axis represents the throughput and corresponds to single-value markers of saturation throughput. Since a single packet takes 4 cycles in a single channel to be transmitted, the maximum throughput is 0.25 packets/cycle/channel.

### A. Number of Channels

Here, we discuss the results shown in Fig. 4 for BRS and token passing and an increasing number of channels.

**Latency.** In general, it can be observed that BRS is less stable than token in terms of latency as the range of values is larger, with a higher number of outlier points. However, BRS has a much better zero-load latency than token since, in BRS, the protocol allows nodes to start transmitting immediately when the channel is sensed idle. This fact also can explain why independently of the parameters evaluated here (assignment, number of channels) the minimum latency is quite similar. The worst-case latency, however, clearly improves when having multiple channels, as the high load is distributed over multiple channels. On the other hand, in token passing, nodes must wait until they possess the token to start transmitting. For this reason, when the number of nodes is large,  $N = 64$  in this case, the system remains idle much longer.

**Throughput.** The results for token passing depict a rather stable increase in saturation throughput as more channels are added, regardless of the assignment method. This could be due to the use, by default, of non-bursty and non-hotspot traffic to evaluate scalability. On the other hand, the results for BRS illustrate a different behavior than in token passing. Firstly, BRS cannot reach a saturation throughput as high as token passing. The main reasons are that channel contention and multiple collisions lead to channel waste and, hence, to a reduced throughput. Furthermore, BRS is more irregular than token passing in terms of saturation throughput as it depends

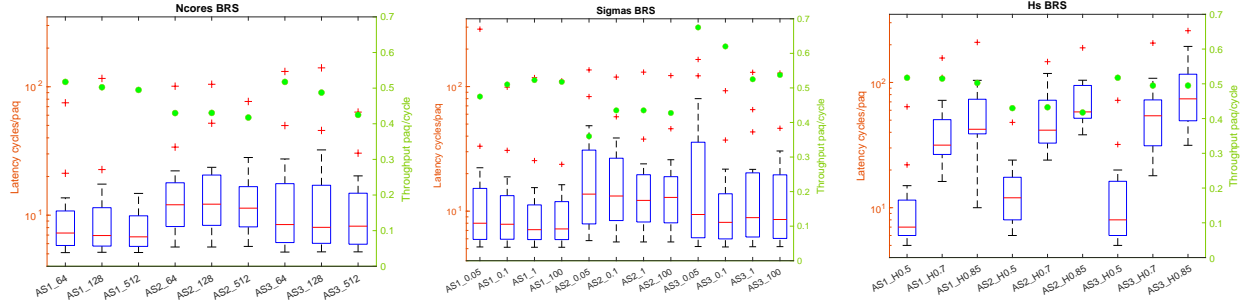


Fig. 5. Performance of multi-channel BRS protocol for an increasing number of nodes,  $N=64-512$  (left graph), different spatial concentration levels,  $\sigma=0.1-100$  (center graph), different temporal burstiness levels,  $H=0.5-0.85$  (right graph), and different assignment techniques.

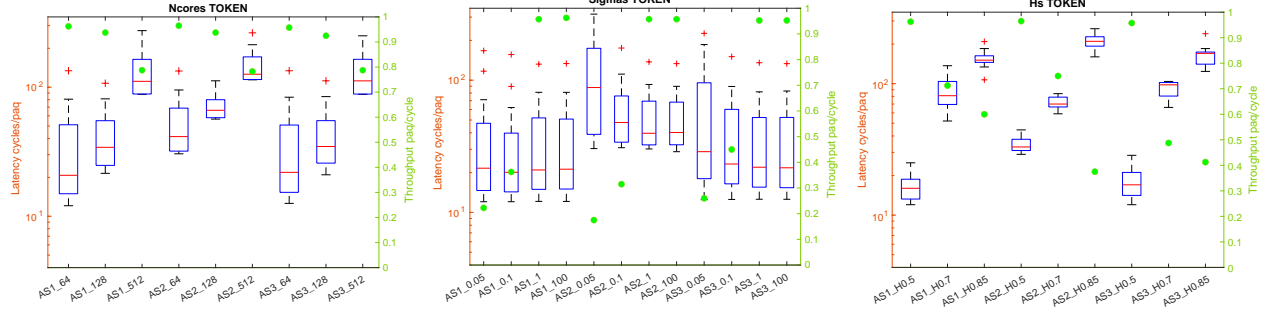


Fig. 6. Performance of multi-channel token passing protocol for an increasing number of nodes,  $N=64-512$  (left graph), different spatial concentration levels,  $\sigma=0.1-100$  (center graph), different temporal burstiness levels,  $H=0.5-0.85$  (right graph), and different assignment techniques.

on the percentage of collisions at high loads. As a result, the difference between the saturation throughput achieved for different assignments increases with the number of channels.

### B. Number of Nodes

Next, we comment on the performance of BRS and token passing for an increasing number of nodes, with  $N_c = 4$ . The results are shown in the left charts of Fig. 5 and Fig. 6.

**Latency.** BRS has a much lower latency than token passing due to its ability to transmit when the channel is idle. The span of the latency values differs across number of nodes and assignments, but in general are restrained to similar values because in the end, the same aggregated load ends up being distributed over more nodes. Static assignment of channels (AS2) works worse than the other alternatives. On the other hand, from the plot of token passing, it is clear that more nodes lead to much higher latency due to the increase of the token turnaround time. In fact, the low-load latency is proportional to the number of nodes in all cases. The span of the latency values is similar across the different system sizes.

**Throughput.** In general, saturation throughput is slightly higher for a lower number of nodes. In our protocols, having more nodes means having a higher population and, hence, a higher chance of collisions even for the same load for BRS, and a higher waiting time (or lower probability of having all nodes backlogged) in token passing. It seems, in any case, that BRS is more resilient to the change in the number of nodes as the drop is more subtle, except for AS3, where possibly the load balancing algorithm is not performing well when such a large number of nodes has to be classified. Finally,

all three assignments have very similar throughput in all cases for token passing, whereas AS1 (random channel assignment to individual packets) works better in BRS.

### C. Hotspot Traffic

We next discuss the results shown in the middle plots of Fig. 5 and Fig. 6, which illustrate the impact of uneven spatial injection distribution on performance. We remind that low/high values of  $\sigma$  mean that traffic is hotspot/evenly distributed [26].

**Latency.** In BRS, the hotspot behavior of traffic does not seem to have a large influence on the performance of the different assignment methods. The outlier, third quartile, and maximum values within the distribution seem to be mildly impacted by the hotspot nature of traffic. In general, BRS is resilient to such variations and actually could benefit from having a lower amount of nodes contending for the available channels. Still, the results show a small tendency to worse results when traffic is concentrated around a few nodes, possibly because of the nodes with higher load reaching higher backoff values. In AS3, this situation is avoided by proactively placing high-load nodes in different channels. Similarly, in token passing, latency is affected by the concentration of traffic around a given set of nodes mostly because the different assignment methods are able to provide tokens quickly to nodes that need it, even if they are spaced apart within the ring. This is clearly visible in the extreme case of  $\sigma = 0.05$ . Similarly, outlier values seem to be larger when traffic is more hotspot. We also observe how AS2 fails to provide a good performance at low loads, and this behavior is exacerbated for very hotspot traffic.

**Throughput.** The throughput of BRS in its different implementations does not vary significantly with the type of

spatial distribution of traffic, except for AS3, where a higher concentration of traffic around a few nodes seems to have a positive effect on the throughput. One reason could be that the most active nodes are distributed over the different channels so that contention is minimized. That does not happen in other assignment methods. Different behavior is observed in token passing, where the hotspot behavior of traffic modifies the throughput of the different assignment methods, with AS3 being affected a bit less. This is because if the load is concentrated around a small set of nodes, a large portion of the airtime is wasted while passing the token among these nodes.

#### D. Bursty Traffic

Finally, we present the latency and throughput results for an increasingly bursty traffic. The results are shown in the right plots of Fig. 5 and Fig. 6 for BRS and token passing, respectively. Temporal injection of traffic is modeled through the Hurst exponent [26], with higher values indicating more bursty behavior, i.e. longer bursts followed by longer silences.

**Latency.** In BRS, it can be seen that the higher the value of  $H$ , the higher the latency in average and also the more unpredictable. This is because with an  $H$  of 0.5, the packets are injected following a random Poisson process, which minimizes the probability of collisions. However, when increasingly bursty traffic is considered, the probability of packets being injected (and nodes trying to transmit) in the same exact cycle increases. The effect is multiplicative with the burstiness, as the effect of cascading collisions leads to an exponential increase of the backoff time. This affects the system at all loads. On the other hand, token passing also suffers when bursty traffic is served, leading to very high latency especially for high values of  $H$ . The latency is a bit more stable than in the case of BRS, mainly because the protocol does not react with exponential backoffs, but rather with linear token passings to bursts of traffic. Still, the latency is much higher than that of BRS, discouraging its use for large number of nodes.

**Throughput.** On one hand, it can be verified that in BRS, the saturation throughput remains rather constant across all assignments regardless of the value of the Hurst exponent. A possible reason could stem from the behavior of the backoff mechanism; bursty traffic leads to a large number of collisions which increases latency even for low loads, but the protocol may converge to a large backoff value that can accommodate the load even if it comes in bursts. In other works, the backoff mechanisms spreads out the bursts of traffic over time, until all nodes are backlogged. On the other hand, it can be seen that in the case of token passing, the saturation throughput seems to drop significantly for higher numbers of  $H$ , to a point that the achieved throughput becomes comparable with that of BRS. A potential reason for this behavior is the lack of an adaptive mechanism to react to bursts; the token has to still move around the ring even if bursts of traffic lead to the generation of multiple packets in a given node, leading to gaps where the wireless channel remains silent. When traffic is less bursty, the probability of such events is lower.

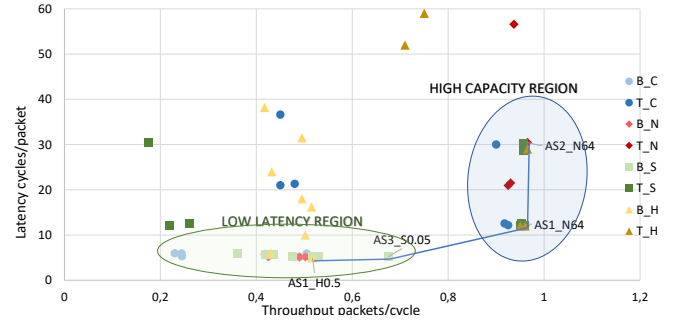


Fig. 7. Summary of the latency and throughput results over all the protocols, assignment methods, and traffic conditions.  $B$  and  $T$  stand for BRS (random access) and token passing,  $C$  and  $N$  denote number of channels and nodes, whereas  $S$  and  $H$  represent the different spatial and temporal injection distributions. For instance, the  $B_C$  symbols represent the latency-throughput of all the assignment methods for BRS for different number of channels. Two desirable design spaces and a Pareto frontier are also given.

#### IV. DISCUSSION

Figure 7 plots the performance of all the compared protocols and assignments representing the zero-load latency (X axis) and saturation throughput (Y axis) of a particular protocol for a given number of channels and assignment method.

In general terms, BRS is preferred over token in terms of zero-load latency given its ability to transmit immediately when the channels are idle. Hence, we see most BRS points located in a *low latency region*. Among the assignment techniques, AS1 achieves similar results than AS3 and would probably be preferred as it does not require prior knowledge of the load of each node to assign the channels. On the downside, the throughput is half of that of token passing, at most.

On the other hand, token passing can reach high throughput levels in the *high capacity region*, close to the maximum total bandwidth of the wireless network. However, while putting more channels reduces the latency significantly, the best latency in token passing is still several cycles away from the BRS values. Finally, we observe that it is hard to provide a good channel assignment overall: AS3 requires prior knowledge on the traffic distribution, AS1 does not perform well for hotspot traffic and AS2 has high latency.

#### V. CONCLUSIONS

This paper has explored several techniques to extend random access and token passing MAC protocols to multiple channels for wireless chip-scale networks. In general, more channels alleviate the problems of both types of protocols, increasing the throughput of random access and cutting down the latency of token passing to a few tens of cycles. Additionally, random access is more resilient to hotspot and bursty traffic and more scalable to massive chip-scale networks. However, the higher throughput achievable with token renders the decision of the protocol (and assignment) to choose extremely challenging. Hence, we see a trend similar to that of single-channel protocols: it would be desirable to develop a multi-channel protocol that is able to seamlessly obtain the best of both paradigms. This will be explored in future work.

## REFERENCES

- [1] R. Marculescu, U. Ogras, L.-S. Peh, N. Enright Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 3–21, 2009.
- [2] D. Bertozzi, G. Dimitrakopoulos, J. Flich, and S. Sonntag, "The fast evolving landscape of on-chip communication," *Design Automation for Embedded Systems*, vol. 19, no. 1, pp. 59–76, 2015.
- [3] S. Salahuddin, K. Ni, and S. Datta, "The era of hyper-scaling in electronics," *Nature Electronics*, vol. 1, no. 8, pp. 442–450, 2018.
- [4] C. Mayr, S. Hoepfner, and S. Furber, "Spinnaker 2: A 10 million core processor system for brain simulation and machine learning," *arXiv preprint arXiv:1911.02385*, 2019.
- [5] T. Krishna, H. Kwon, A. Parashar, M. Pellauer, and A. Samajdar, "Data orchestration in deep learning accelerators," *Synthesis Lectures on Computer Architecture*, vol. 15, no. 3, pp. 1–164, 2020.
- [6] T. Wang, F. Feng, S. Xiang, Q. Li, and J. Xia, "Application defined on-chip networks for heterogeneous chiplets: An implementation perspective," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 1198–1210, IEEE, 2022.
- [7] A. Kannan, N. E. Jerger, and G. H. Loh, "Exploiting interposer technologies to disintegrate and reintegrate multicore processors," *Ieee Micro*, vol. 36, no. 3, pp. 84–93, 2016.
- [8] S. Naffziger, N. Beck, T. Burd, K. Lepak, G. H. Loh, M. Subramony, and S. White, "Pioneering chiplet technology and design for the amd epyc™ and ryzen™ processor families: Industrial product," in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pp. 57–70, IEEE, 2021.
- [9] 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.
- [10] A. Ganguly, S. Abadal, I. Thakkar, N. E. Jerger, M. Riedel, M. Babaie, R. Balasubramanian, A. Sebastian, S. Pasricha, and B. Taskin, "Interconnects for dna, quantum, in-memory, and optical computing: Insights from a panel discussion," *IEEE micro*, vol. 42, no. 3, pp. 40–49, 2022.
- [11] D. W. 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, pp. 58–65, 2012.
- [12] S. Laha, S. Kaya, D. W. Matolak, W. Rayess, D. DiTomaso, and A. Kodi, "A new frontier in ultralow power wireless links: Network-on-chip and chip-to-chip interconnects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 186–198, 2014.
- [13] S. Abadal, R. Guirado, H. Taghvaei, A. Jain, E. P. de Santana, P. H. Bolívar, M. Saeed, R. Negra, Z. Wang, K.-T. Wang, *et al.*, "Graphene-based wireless agile interconnects for massive heterogeneous multi-chip processors," *IEEE Wireless Communications*, 2022.
- [14] R. S. Narde, J. Venkataraman, A. Ganguly, and I. Puchades, "Intra-and inter-chip transmission of millimeter-wave interconnects in noc-based multi-chip systems," *IEEE Access*, vol. 7, pp. 112200–112215, 2019.
- [15] D. DiTomaso, A. Kodi, D. Matolak, S. Kaya, S. Laha, and W. Rayess, "A-WiNoC: Adaptive wireless network-on-chip architecture for chip multiprocessors," vol. 26, no. 12, pp. 3289–3302, 2015.
- [16] N. Mansoor and A. Ganguly, "Reconfigurable wireless network-on-chip with a dynamic medium access mechanism," in *Proceedings of the 9th International Symposium on Networks-on-Chip*, pp. 1–8, 2015.
- [17] K. Duraisamy, R. G. Kim, and P. P. Pande, "Enhancing Performance of Wireless NoCs with Distributed MAC Protocols," in *Proceedings of the ISQED '15*, pp. 406–411, 2015.
- [18] A. Mestres, S. Abadal, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "A MAC protocol for Reliable Broadcast Communications in Wireless Network-on-Chip," in *Proceedings of the NoCArc '16*, pp. 21–26, 2016.
- [19] S. Jog, Z. Liu, A. Franques, V. Fernando, S. Abadal, J. Torrellas, and H. Hassanieh, "One protocol to rule them all: Wireless network-on-chip using deep reinforcement learning," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pp. 973–989, 2021.
- [20] A. Franques, S. Abadal, H. Hassanieh, and J. Torrellas, "Fuzzy-Token: An adaptive mac protocol for wireless-enabled manycores," in *Design, Automation and Test in Europe Conference (DATE)*, 2021.
- [21] Y. Ouyang, C. Sun, B. Jia, Q. Wang, and H. Liang, "Architecting a priority-based dynamic media access control mechanism in wireless network-on-chip," *Microelectronics Journal*, vol. 116, p. 105218, 2021.
- [22] W. Rayess, D. W. Matolak, S. Kaya, and A. K. Kodi, "Antennas and Channel Characteristics for Wireless Networks on Chips," *Wireless Personal Communications*, vol. 95, no. 4, pp. 5039–5056, 2017.
- [23] X. Timoneda, A. Cabellos-Aparicio, D. Manassis, E. Alarcón, and S. Abadal, "Channel characterization for chip-scale wireless communications within computing packages," in *2018 Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pp. 1–8, IEEE, 2018.
- [24] P. Baniya, A. Bisognin, K. L. Melde, and C. Luxey, "Chip-to-Chip Switched Beam 60 GHz Circular Patch Planar Antenna Array and Pattern Considerations," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 4, pp. 1776–1787, 2018.
- [25] F. Rodríguez-Galán, E. P. de Santana, P. H. Bolívar, S. Abadal, and E. Alarcón, "Towards spatial multiplexing in wireless networks within computing packages," in *Proceedings of the 9th ACM International Conference on Nanoscale Computing and Communication*, pp. 1–6, 2022.
- [26] V. Soteriou, H. Wang, and L. Peh, "A statistical traffic model for on-chip interconnection networks," in *14th IEEE International Symposium on Modeling, Analysis, and Simulation*, pp. 104–116, IEEE, 2006.
- [27] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 236–251, 2001.
- [28] R. Ubal, P. Mistry, D. Schaa, H. Ave, and D. Kaeli, "Multi2Sim: A Simulation Framework for CPU-GPU Computing," in *Proceedings of the PACT'12*, 2012.