Horizon 2020 Program (2014-2020)
FET-Open – Novel ideas for radically new technologies
FETOPEN-01-2018-2019-2020

WiPLASH

Architecting More than Moore – Wireless Plasticity for Massive Heterogeneous Computer Architectures [†]

D3.3: Adaptive protocol stack

WP3 - Wireless Communications within Package

Contractual Date of Delivery	30/09/2022
Actual Date of Delivery	03/10/2022
Deliverable Dissemination Level	Public
Editor	Sergi Abadal (UPC)
Contributors	UPC (leader), RWTH, UNIBO, EPFL
Quality Assurance	Davide Rossi (UNIBO) Giovanni Ansaloni (EPFL)

[†]This project is supported by the European Commission under the Horizon 2020 Program with Grant agreement no: 863337.

Document Revisions & Quality Assurance

Deliverable Number	D3.3
Deliverable Responsible	UPC
Work Package	WP3
Main Editor	Sergi Abadal

Internal Reviewers

1. Davide Rossi (UNIBO)
2. Giovanni Ansaloni (EPFL)

Revisions

Version	Date	By	Overview
1.4.0	03/10/2022	*Editor*	Synthesized final version.
1.3.0	29/09/2022	*Reviewers*	New version including comments from internal reviewers.
1.2.0	03/09/2022	*UPC Team*	Third draft for review only missing results from Chapter 5.
1.1.0	02/08/2022	*Editor*	Second draft with structure and text.
1.0.0	02/07/2022	*Editor*	First draft. Outline.

Legal Disclaimer

Executive Summary

The WiPLASH project aims to develop wireless-enabled architectures that deliver an improvement of 10$\times$ over existing architectures in multi-chip environments. To that end, wireless communications promises to provide a powerful, flexible, low-latency, and broadcast-capable interconnection layer. However, fulfilling that promise within the stringent resource and bandwidth constraints of the scenario remains a challenge. Aware of this fact, in this deliverable we propose a protocol stack able to deliver the required communication performance while being flexible enough to adapt to the multiple workloads that the wireless on-chip network may need to serve. To combat a possible lack of bandwidth for data-intensive applications, the proposed protocol stack also provides support for multiple frequency and space channels, which may be enabled by the use of tunable graphene antenna arrays. In this context, the main contributions of this deliverable are: (i) the demonstration that flip-chip computing packages support the existence of multiple frequency channels between 60 GHz and 240 GHz, as well as multiple spatial channels with compact antenna arrays; (ii) the analysis of the traffic generated by the architectures and workloads assumed in the architecture work packages, demonstrating that such traffic is inherently bursty and hotspot; (iii) the proposal of a PHY capable of handling the multiple channels via a controller that tunes the frequency and gain of different elements of the RF chain, including the tunable graphene antennas; (iv) the analysis of whether it is preferable to use the multiple channels to reduce the latency of the MAC protocol or the latency of the transmission itself; (v) the proposal and thorough evaluation of multi-channel extensions of two widespread MAC protocols for on-chip networks; and (vi) a preliminary discussion motivating the need for bridging the architecture with the protocol stack in an attempt to maximize the system performance through appropriate reconfiguration policies at the lower layers of the communications stack.

Abbreviations and Acronyms

WNoC Wireless Network-on-Chip

WNiP Wireless Network-in-Package

THz terahertz

EM Electromagnetic

AlN Aluminum nitride

SiP System-in-Package

BER Bit Error Rate

SNR Signal-to-Noise Ratio

SoC System-on-Chip

MAC Medium Access Protocol

CSMA Carrier-Sensing Multiple Access

PHY Physical Layer

OOK On-Off Keying

SPP Surface Plasmon Polariton

SIR Signal-to-Interference Ratio

The WiPLASH consortium is composed by

UPC	Coordinator	Spain
IBM	Beneficiary	Switzerland
UNIBO	Beneficiary	Italy
EPFL	Beneficiary	Switzerland
AMO	Beneficiary	Germany
UoS	Beneficiary	Germany
RWTH	Beneficiary	Germany

Contents

List of Figures

List of Tables

1. Introduction

Efficient integrated networks at the chip scale for data exchange between the processing elements of a multicore processor or System-on-Chip (SoC) is a prerequisite for high performance in such computing systems. Currently, most systems incorporate a Network-on-Chip (NoC) consisting of a set of on-chip routers and intra-chip wired links [1]. However, recent trends in computer architecture are leading to extreme scaling (using many processor cores), specialization (using hardware accelerators), and disintegration (interconnecting multiple small chiplets in a System-in-Package (SiP) instead of building large monolithic SoCs). This casts unprecedented bandwidth and reconfigurability requirements on the interconnect fabric, which now has to also extend beyond the limits of a single chip [2, 3]. New paradigms are thus required in the manycore era, which is the hypothesis over which the WiPLASH project unfolds.

Among the different emerging alternatives, wireless in-package communications stand as a promising contender as advocated by WiPLASH [4,5]. This communication paradigm relies the use of Electromagnetic (EM) waves for data transmission using the chip package as communications medium. The resulting *wireless links* provide low latency, inherent broadcast capabilities, and global reconfigurability; three unique features that wired alternatives, including nanophotonics, cannot offer because of need of a path infrastructure (and possibly many hops) to reach distant locations [4–6]. By integrating such wireless links within and across chips, the concepts of Wireless Network-on-Chip (WNoC) and Wireless Network-in-Package (WNiP) are born.

Figure 1.1 illustrates the wireless paradigm of WiPLASH through an example of an heterogeneous architecture with multiple wireless links within and across chips. Also, without loss of generality, the figure presents a simplified protocol stack that defines the communication process. Information coming from the processors or memory modules go through a network interface, which routes the data towards the wired or wireless network (network layer); once entering the wireless network, the Medium Access Protocol (MAC) protocol determines the channel that shall be used for transmission and the right instant for transmission (link layer). Upon transmission, data is serialized and modulated by the transceiver (physical layer). Modulated signals are radiated and propagate through the computing package until they reach the receiver, which demodulates the signals, deserializes the bits, checks that there are no errors on the packet, and then passes the information towards network interface, which delivers it to the processor core or memory module if they are the intended receivers.

As demonstrated in the literature and within WiPLASH, the unique features of WNoC and WNiP networks can become key enablers of radically new architectures capable of pushing the scalability limits of nowadays SoCs and SiPs [7–9]. However, wireless communications also have disadvantages, namely: (i) the moderate or low energy efficiency stemming from the need to compensate for the detrimental effects of the wireless channel; (ii) the non-negligible chip area required to lay out the transceivers

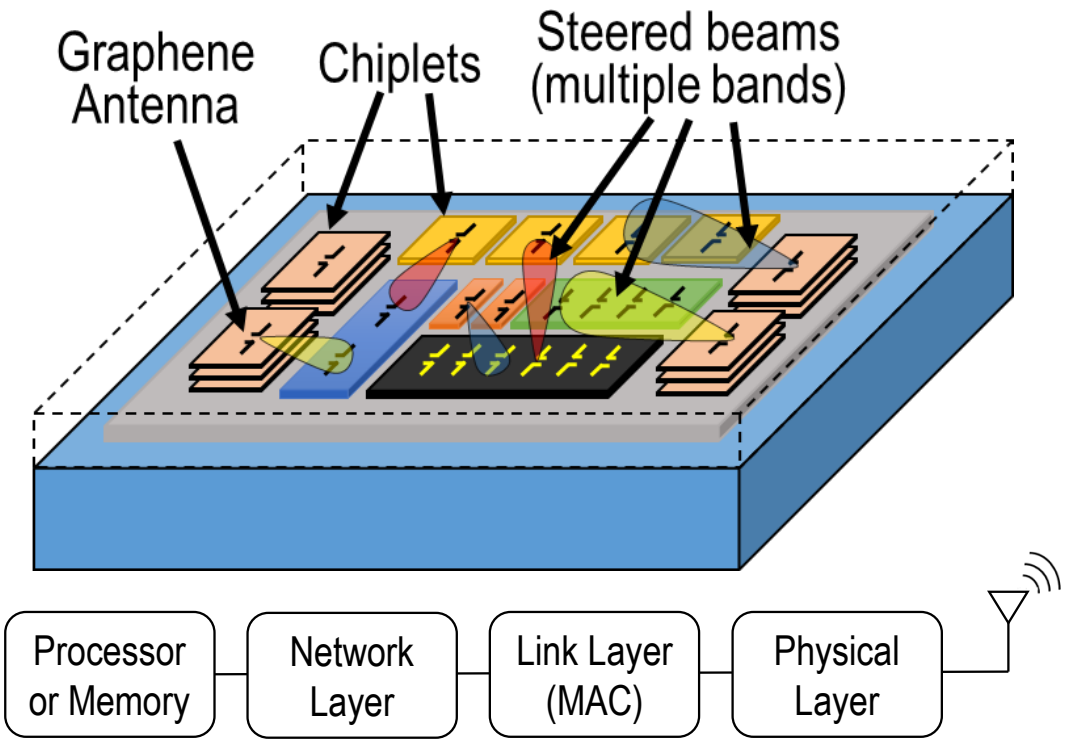


Figure 1.1: A general view of the WiPLASH vision on wireless communications at the chip scale within a heterogeneous computer architecture and multiple frequency-tunable, beam-steerable antennas. At the bottom, we show the logical structure of the wireless network with its network, link, and physical layer protocols.

that serialize, modulate and radiate the information; and (iii) the low aggregate bandwidth resulting from the need to share a few channels among all antennas. WiPLASH aims to address these three disadvantages, with this deliverable focusing on the lack of bandwidth specifically.

Thus far, radiation in WNoC and WNiP environments is typically assumed to be omnidirectional and within a single fixed frequency band in concordance with the required simplicity and limited availability of directive antennas in this scenario. However, WiPLASH proposes to use miniaturized tunable antenna arrays to produce field concentrations both (1) in certain areas of the chip leading, possibly, to spatial multiplexing, and (2) in different frequency channels. By implementing such multi-channel directive schemes, we can build architectures that leverage the advantages of the wireless approach while alleviating some of the downturns such as the low bandwidth or low efficiency.

In this direction, the main aims of this deliverable are, first, to demonstrate that the chip scenario can support such multiple frequency and space channels, and second, to outline a protocol stack that builds on those multiple channels to implement a wireless network that can adapt to different architectures and workloads. To these ends, the main contributions reported in this document are:

- A model of a typical chip package from an electromagnetic perspective and show, via full-wave simulations, that this scenario can sustain multiple frequency and spatial channels. This demonstration is made assuming vertical monopoles, but could be easily extended to planar antennas similar to graphene patches.

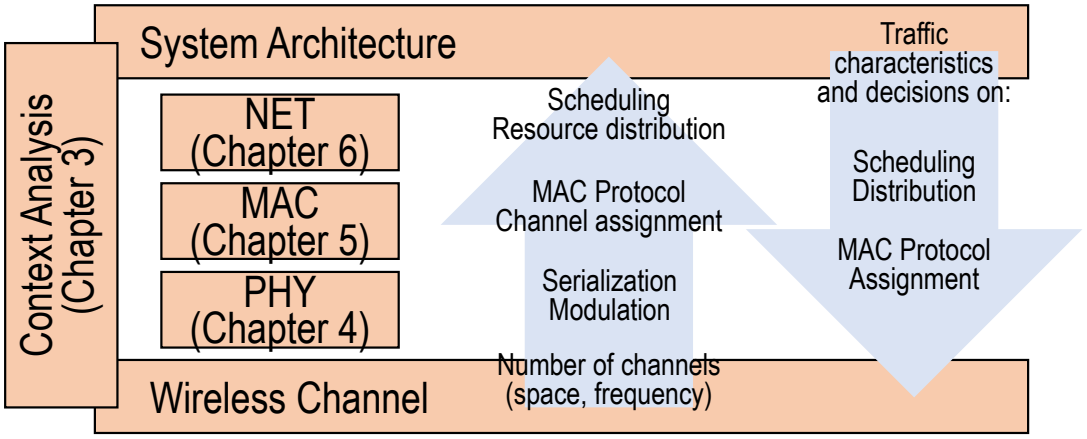


Figure 1.2: Graphical abstract of this deliverable. We start by a context analysis, motivating the need for a protocol stack handling multiple channels and bursty/hotspot traffic. Subsequent chapters discuss the implications of that at the physical, link, and network layers.

- A study of the communication requirements of legacy general-purpose and novel AI-oriented processor architectures to dimension the requirements cast on the wireless network and justify the need for reconfigurable architectures.
- A description and an evaluation of the techniques and tradeoffs that become necessary when considering multiple channels within the network at compressed protocol stack of on-chip environments. In particular, we discuss the management of multiple channels at the physical, medium access, and network levels.

In the current state of the art, wireless-enabled architectures have been either simplified to account for the single-channel limitations of currently achievable wireless transceivers [10–12], or assumed to account for multiple space, frequency or code channels, often neglecting the difficulty of implementing such channels at the antenna or transceiver side [13–16]. Works trying to substantiate the multi-channel assumption have been limited to directive antenna designs [17–19] and have not touched upon tunable radiators or transceivers, nor the behavior of the on-chip wireless channel in those cases. Hence, the results contained in this deliverable advance beyond the state of the art in the field.

The remainder of this deliverable is organized as summarized in Figure 1.2. In Chapter 2, we first lay down the methodological details for the results shown in the subsequent chapters. In Chapter 3, we present a brief analysis of the on-chip communications context, touching upon two critical aspects, namely, (1) the support for multiple wireless channels, and (2) the communication requirements of different architectures. In Chapter 4, we analyze the trade-offs of using multiple channels at the physical layer and propose a controller to manage the different modes of a tunable graphene antenna array. In Chapter 5, we study different methods to assign channels to packets with the aim to maximize performance. To that end, we extend the protocols evaluated in previous deliverables with their respective multi-channel versions. In Chapter 6, we briefly describe the possible ways that the network layer could work with the architecture to maximize the performance of the system. Finally, the deliverable is concluded in Chapter 7.

2. Methodology

This chapter summarizes the methods employed in subsequent chapters to model the different aspects impacting the performance and resource consumption of wireless links within a computing package. Figure 2.1 shows a graphical schematic of the methodology, which details inputs and outputs, how the different parts interact with each other, as well as the software used in each step.

In essence, we model the wireless channel using a methodology based on the one used in Deliverable D3.1 [20] together with some extra processing for the simulation of arrays and the treatment of the results. In particular, we model phased arrays of antennas instead of single antennas and use simultaneous excitation from multiple arrays to evaluate potential interference patterns, towards the formation of multiple spatial patterns. More details on the methods to obtain these parameters are given in Section 2.1.

At the physical layer of design, the availability of several channels can be used to increase the bandwidth of a single transmitter or give service to multiple transmitters simultaneously. One can choose to have a design that accommodates both options. In any case, the tradeoffs are analyzed via simulations as depicted in Section 2.2.

Assuming a given data rate and error rate from the physical layer, a link layer analysis delivers the performance of a wireless link in terms of latency and throughput. To that end, event-driven simulations are conducted which consider different types of traffic, different number of antennas sharing a link, and different number of available channels. More details on the simulation methods are depicted in Section 2.3.

Finally, to analyze the communication requirements of different architectures, which will definitely influence the decisions at the protocol stack, or to evaluate the impact of different network design decisions at the system level, we rely on full-system architectural simulations. For spatiotemporal workload analysis, we extract communication traces and then perform some post-processing steps to extract the spatial and temporal characteristics. Further details are given in Section 2.4.

2.1 Wireless Channel

In our previous work in Deliverable D3.1 [20], the electromagnetic wave propagation within three different computing packages (flip-chip, interposer and bondwire) was assessed. The methodology used for this task was based on taking advantage of the static and monolithic nature of the systems in question. All the packages were simulated in the full-wave solver CST Microwave Studio [21], because of its variety of methods for solving computational electromagnetism problems in the frequency and time domains. The packages were modeled based on datasheets and real packages features, and simulated in the frequency and time domains. Several simulations and

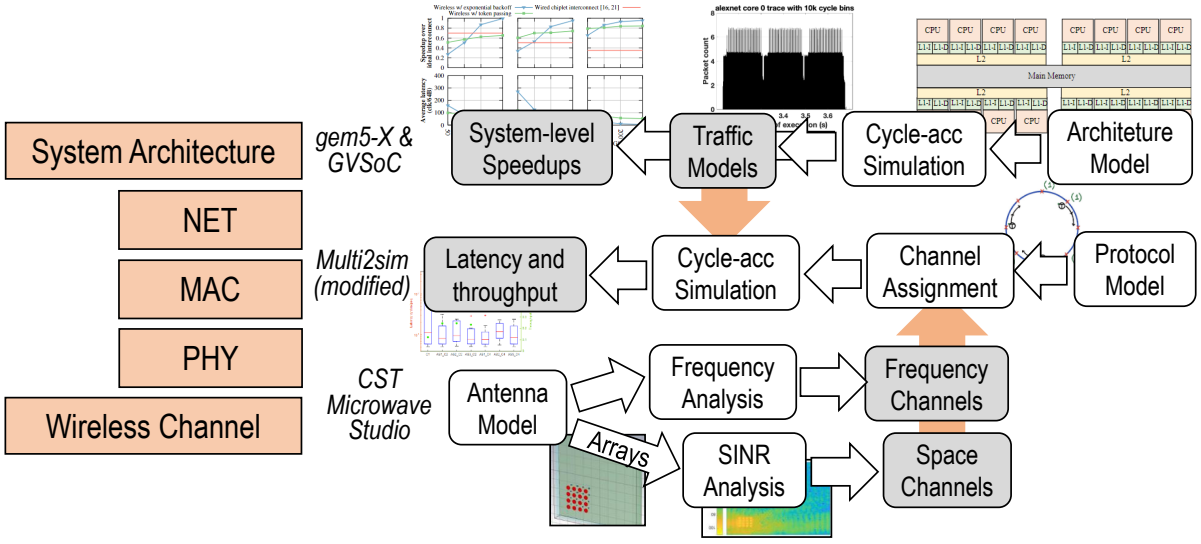


Figure 2.1: General view of the methodology used in this deliverable for the modeling of the wireless channel, the evaluation of link-layer protocols, and the assessment of system architectures and their traffic traces.

parameter sweeps were performed by changing the materials and dimensions of the layers of the package, as well as the frequency ranges. This allowed us to obtain the field distribution and S-parameters of the channel, which were then feed to a MATLAB post-processing to achieve path loss (PL) of the channel.

In this deliverable, the methodology from D3.1 is adapted to study the availability of frequency and space channels in on-chip environments. The methodology is summarized Figure 2.2. In the following, we first describe the physical environment modeled in CST in Section 2.1.1 and the antennas used in our study in Section 2.1.2. Then, we outline the specific methods followed to study the frequency and spatial channels in Sections 2.1.3 and 2.1.4, respectively. All simulations have been performed in two workstations, namely, a quad-core CPU at 3.90 GHz with 32 GB of RAM and a GeForce GTX 1080Ti GPU to accelerate time-domain simulations, and a 16-core CPU at 2.16 GHz with 128 GB of RAM.

2.1.1 Environment Description

The channel modeling simulations considered in this deliverable are only carried out in a flip-chip package model due to its structural simplicity and its better support for wireless communications, as concluded in Deliverable D3.1 [20]. However, we note that the same analysis could be performed in an multi-chiplet interposer-based package, which is very similar to the flip-chip one.

An instance of a complete flip-chip package with solder bumps is shown in Figure 2.3. During the manufacturing process, the solder bumps are deposited on the chip pads and, then, the chip is flipped over and its solder bumps are aligned precisely to the pads of the package carrier external circuit.

The layers are described from top to bottom as summarized in Table 2.1. On top, the heat sink and heat spreader dissipate the heat out of the silicon chip, as they both have good thermal conductivity. Bulk silicon serves as the foundation of the transistors. This layer has low resistivity (10 Ω·cm), which is convenient for the operation

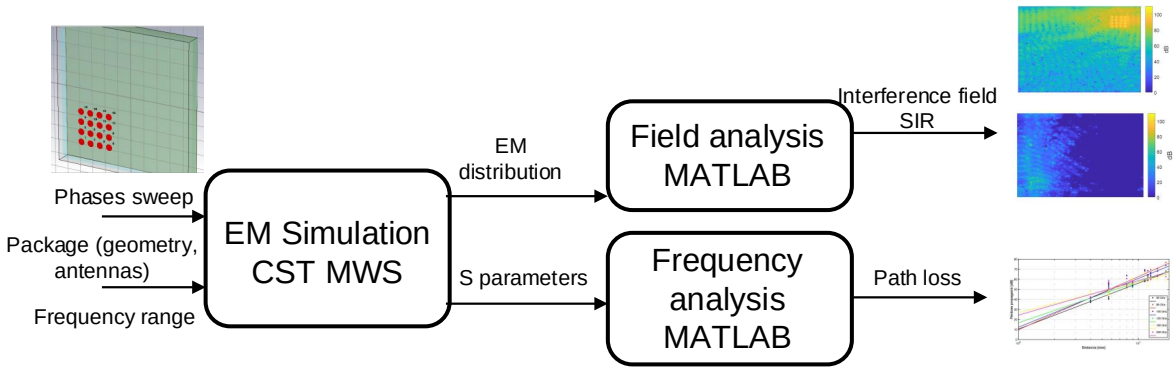


Figure 2.2: Methodology used in the evaluation of spatial and frequency channels in on-chip environments.

of transistors, but not for electromagnetic propagation [22]. The interconnect layers, which occupy the bottom of the silicon die as shown in the inset of Fig. 2.3, are generally made of copper and surrounded by an insulator such as silicon dioxide (SiO_2) [23]. Finally, we find a package substrate or PCB below the bump array. Although the material of the carrier may be alumina or similar, we model it as perfect electrical conductor due to the existence of a dense metallic redistribution layer within it.

The bulk silicon used in the chip substrate generally has low resistivity, and therefore a thin substrate is preferred [24]; whereas materials used as heat spreaders have low electrical losses [22] and rather thick layers are desirable. To evaluate this impact in our simulations, we assume that both the substrate and the heat spreader, Aluminum nitride (AlN) in our case, can have a thickness of either 0.1 or 0.5 mm each. On the sides of the die, we assume an empty space of variable size filled with air or epoxy. The package is laterally enclosed with a metallic lid.

2.1.2 Antenna Design

Due to its good *a priori* lateral coupling and opportunistic compatibility with conventional chip package designs, we have chosen a vertical monopole antenna as baseline for our study. The monopole antenna is modeled as a thin and long cylindrical metallic structure, placed vertically passing through the silicon and fed from the first metal

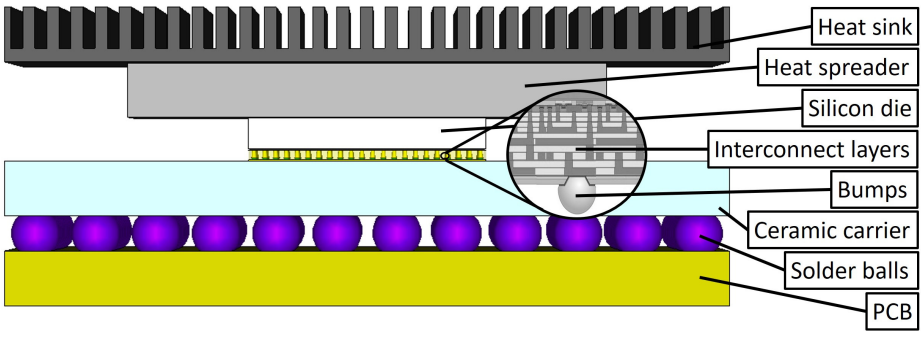


Figure 2.3: Schematic of the layers of a flip-chip package.

Table 2.1: Characteristics of the layers in a flip-chip package and default dimensions. ε_r is the relative permittivity of the material, tan(δ) is the loss tangent, and ρ refers to the conductivity. PEC stands for perfect electrical conductor (lossless material of infinite conductivity).

	Thickness	**Material**	ε_r	tan(δ)	ρ
Heat sink	0.1–0.5 mm	Aluminum	PEC	PEC	PEC
Heat spreader	0.1–0.5 mm	Aluminum Nitride	8.6	$3{\cdot}10^{-4}$	–
Silicon die	0.5 mm	Bulk Silicon	11.9	–	10 Ω·cm
Insulator	10 μm	SiO_2	3.9	0.025	–
Bumps	87.5 μm	Cu and Sn	PEC	PEC	PEC
Redistribution layer	3 μm	Copper	PEC	PEC	PEC
PCB	0.5 mm	Epoxy resin	4	–	–

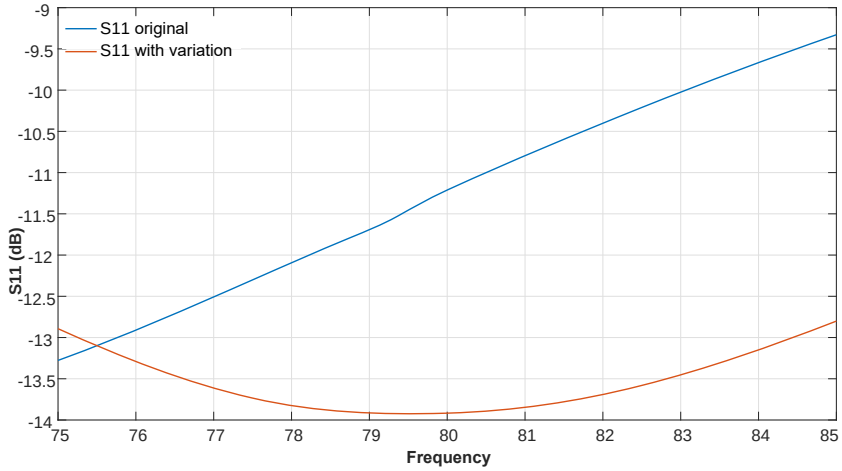


Figure 2.4: Reflection coefficient of a single monopole antenna within a chip at 80 GHz before and after the length tuning.

layers. Practically, this can be implemented by fabricating TSVs that emerge from the metallization layers and prematurely stopping the fabrication before reaching the heat spreader. Since the bumps layer is seen as a solid metallic block of metal at 60 GHz, due to the small bump pitch, this layer acts as a sort of ground plane for the monopole, increasing the effective antenna length due to image theory [25].

$$L = \frac{\lambda}{4} = \frac{v_p}{4 \cdot f} = \frac{c_0}{4 \cdot \sqrt{\varepsilon_{Si}} \cdot f} \tag{2.1}$$

where c_0 is the speed of light, f is the target frequency, and ε_{Si} is the permittivity of silicon in that frequency region.

While the monopole will, in principle, fit entirely within the silicon layer, its proximity to the interface with the heat spreader material with a different permittivity may lead to a shift in the resonance. Taking this into account, to adjust the dimensions of our antenna, we first model a simple scenario with a quarter-wave monopole sized using (2.1). Afterwards, we introduced the monopole in the chip environment and we fine-tuned the length with multiple simulations to get a good reflection coefficient close to the desired central frequency. See Figure 2.4 for an example at 80 GHz.

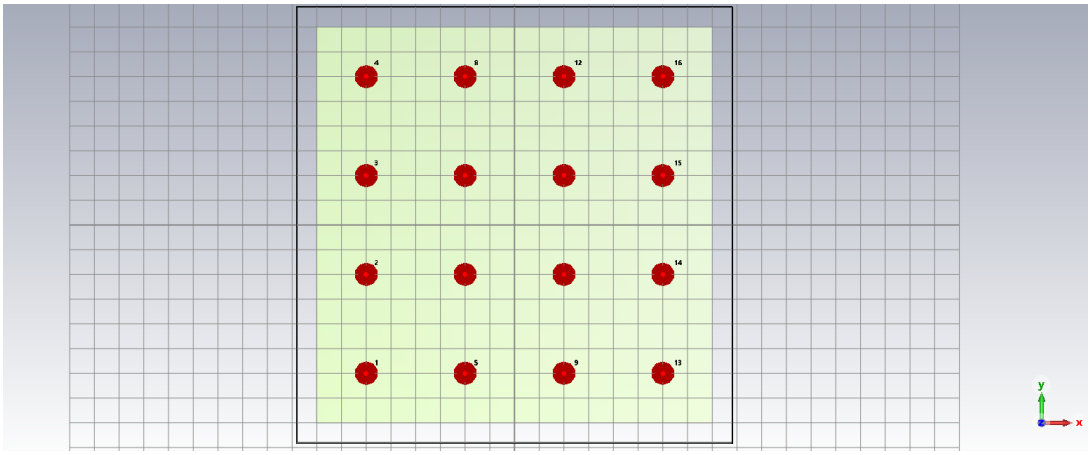


Figure 2.5: Top view of monopole antennas in the chip

2.1.3 Frequency Channels Methodology

To test frequency diversity, the flip-chip is subjected to several simulations in different frequency ranges similarly to in Deliverable D3.1 [20], but in this case, having actual antennas modeled in the channel. The chip will have dimensions of 16×16 mm^2, while the layers are as described in Section 2.1.1. Figure 2.5 shows the distribution of the monopoles in the chip.

The outcome of each simulation is a set of S-parameters relating the output at the receiving antenna as a function of the input at the transmitting one. Once the S-parameters are obtained, the channel frequency response $H_{ij}(f)$ is evaluated for each antenna pair as

$$G_i G_j |H_{ij}(f)|^2 = \frac{|S_{ji}(f)|^2}{(1-|S_{ii}(f)|^2)\cdot(1-|S_{jj}(f)|^2)}, \tag{2.2}$$

where G_i and G_j are the transmitter and receiver antenna gains, S_{ji} is the coupling between transmitter i and receiver j, whereas S_{ii} and S_{jj} are the reflection coefficients at both ends as obtained from the simulations.

Once the study for a particular chip package is done, we sweep the dimensions of the different materials to understand whether the availability of different frequency channels is maintained while engineering the package to reduce losses. To this end, we apply the variations described in Table 2.2. Each scenario that is simulated with the variations of the materials is also simulated for several frequencies (60, 80, 100, 120, 200), so the monopole length must be tuned to ensure resonance at the desired frequency. An important matter to consider here is that in certain combinations of silicon thickness and frequency range, the monopole will not fit entirely within the silicon layer, which leads to a manufacturing-complex design. For instance, we study the channel for silicon thickness of 0.2 mm, while at frequencies of 60 GHz and 80 GHz, the monopole would be 0.3285 mm and 0.2429 mm long, respectively. These design points are avoided in our simulations. In this manner we can size up the behaviour of the chip on different frequencies.

2.1.4 Spatial Channels Methodology

To complement the support of multiple frequency channels, we explore spatial multiplexing. There are some limitations regarding the number of non-overlapping fre-

Table 2.2: Flip-chip variations for frequency channels support.

	Default Value	Variations	Units
Heat spreader	0.5	0.2, 0.4	mm
Silicon die	0.5	0.12, 0.2, 0.4	mm
Frequency	60	80, 100, 120, 160, 200	GHz

quency channels achievable [26] as well as an inherent increase in hardware complexity, not affordable in this resource-constraint scenario. Hence, we instead aim to create field distributions concentrated in different parts of the chip using antenna arrays, and explore the phase distributions that result in field hotspots. To carry out the feasibility analysis of the spatial channels, we start from the flip-chip design with the default frequency and widths shown in the Table 2.2. In this case, there will be no variations in the characteristics of the materials, but rather in the position of the antennas, the distances among them and their excitation phases. Also, in order to reduce the computational cost of the simulations, the chip size has been reduced to 10×10 mm^2.

For this experiment, we also select monopole antennas. When placing a single antenna in the chip and performing some phase sweeps to its excitation to create or direct the field concentration to any part of the chip, none of the sweeps yield any controllability or appreciable differences in the field concentration, as expected. Therefore, to be able to direct the beam or create concentration of energy in any part of the chip, we will need to use antenna arrays.

This entails a larger study because, although the bulk silicon characteristics allows to create compact arrays, one must take caution with the antenna position in the chip, the number of elements we can afford to place, as well as the minimal distance among them to avoid mutual coupling. The coupling issue was studied placing two monopoles in the corner of the chip, then exciting them with the same phase at the same time and monitoring the S-parameters to get a measure of the coupling in each case. We simulate λ/20, λ/10, λ/8, λ/5, λ/4 and λ/2 where $\lambda = \frac{c_0}{\sqrt{\varepsilon_{Si}}\cdot f}$ is the wavelength in silicon.

The results derived from this experiment show that for minimum distances between elements the coupling seems to remain low. The experiment id duplicated but for a 16 element array placed in the corner of the chip. We use two antennas (antennas 6 and 7) placed in the center of the array to measure the influence of the increase in the number of elements.

The results are seen in Figure 2.6 and prove the harmful effect of adding more antennas and lowering the distance among them, leading to an inter-element coupling worse than -10 dB when the distance is smaller than $\lambda/4$. With these results, in terms of manufacturing simplicity and area constrains, the best compromise is to explore the energy concentration with a 16-element array with distances of λ/4 among them.

After we settle in the optimal array configuration, we are set to find a combination of excitation phases that can provide a clear beam and certain controllability. Instead of using an analytic approach, we use the post-processing combine results tool offered by CST, to study the changes in the energy patterns. This makes a sweep of the phases re-using and combining the fields provided by the solver's field monitor at 60 GHz. To simplify the simulations further, we built a smaller environment with one monopole and employed the array factor tool offered by CST to recreate the pattern of an actual array. Hence, we manipulated the radiation pattern of the antenna creating a virtual array after checking that this did not decrease the accuracy of the simulation.

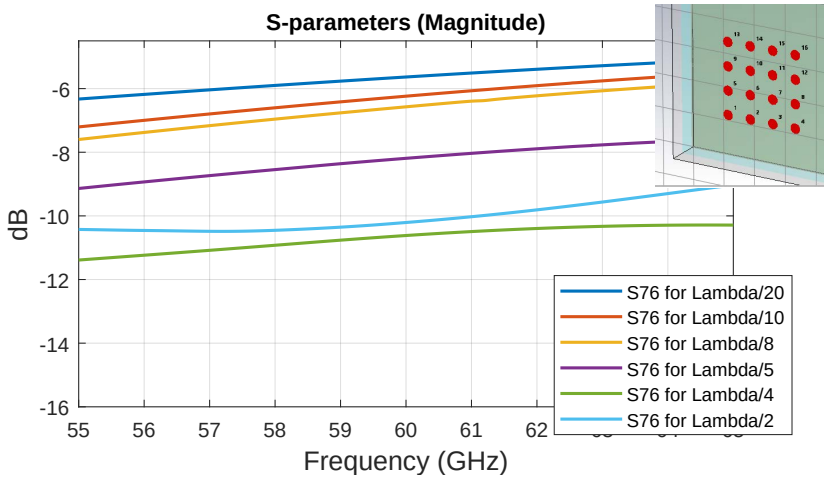


Figure 2.6: Landscape of the array and test coupling results for different distances among the elements.

2.2 Physical Layer

At the physical layer, we focus on the evaluation of the area and power required to implement a wireless interface capable of transmitting at R bits per second, including the analog front-end, data converters, and the serialization scheme. Moreover, we may be interested in calculating the area and power overhead of certain aspects related to the management of multiple channels, such as an adaptive controller at the PHY layer or the MAC circuits.

For the serialization circuits, data converters, and analog front-end we use the methodology developed in Deliverable D3.2 [27]. In particular, the serialization circuits take base on area and power values from related work [28–30] adapted to the speeds required in our systems, assuming that area scales linearly with data rate due to the need to employ larger multiplexers and demultiplexers (or more stages) while power consumption scales linearly with data rate as it implies faster switching of the multiplexers and demultiplexers. The data converters area and power are assessed based on extrapolations from models that summarize the data from Murmann's survey [31] and its multiple figures of merit. Finally, the analog front-end models depend on the sub-system (e.g. power amplifier models come from existing surveys [32]) and are also based on a preliminary link budget to estimate the required gains and the effect of noise. We refer the interested reader to Deliverable D3.2 [27] for more details.

To assess the potential overhead of certain digital circuits used in the controllers and other circuits proposed in this deliverable, we use either comparisons with state-of-the-art designs that provide a similar functionality, or specific simulators such as CACTI [33] for integrated memory systems.

2.3 Link Layer

The link layer of design assumes the existence of a physical layer providing a raw transmission rate R through a number of channels. These channels may be shared among a number N of wireless interfaces, whose MAC protocol will determine when to send and how to manage collisions, if any.

In this deliverable, we propose several possible implementations of multi-channel versions of known MAC protocols for on-chip communications described in Section 2.3.1. In particular, we propose ways to assign a number N_C of channels (which may be frequency or space channels) to the different packets that need to be transmitted.

To assess the performance of these MAC protocols, one can resort to analytical models like those of the original works on Carrier-Sensing Multiple Access (CSMA) [34]. However, these depend on multiple assumptions that do not necessarily hold in the WNoC scenario, e.g. Poisson arrivals [26, 35]. Instead, we resort of event-driven simulation of the WNoC to obtain the performance metrics as we depict in Section 2.3.2.

2.3.1 Baseline Protocols

As discussed in Deliverable D3.2 [27] as well as other review articles [26, 36], related works on MAC for WNoC can be divided into various groups, namely multiplexing in space, time, frequency, or code [14, 17, 37], different variants of token passing [14, 38–40], random access protocols [41, 41, 42], and hybrid approaches [12, 42–44]. Seeking to model different types of MAC protocols, our baselines consider a random access protocol resembling CSMA and a token passing variant as collision-free protocol. These protocols are initially modeled as single-channel protocols, but then extended to support multiple channels via different assignment alternatives described in Section 5.1. More specifically, our simulations consider these protocols:

Carrier Sensing (`BRS`) with which we aim to represent contention-based protocols. We model the slotted version of the BRS-MAC protocol [41], using non-persistence and adopting the NACK burst mechanism to reduce the control overhead. The preamble size is fixed to 20 bits, which implies that the preamble accounts for a variable portion of the transmission. A packet will be considered lost in the unlikely case that it exceeds the maximum number of retries (8). Note that the network will most likely be saturated when this happens.

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 [45]. 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 implicitly.

2.3.2 Simulation

The characterization of link-level performance requires obtaining the latency and throughput of a link shared by a variable number of nodes, for different types of traffic, and increasing the load from a few packets per second up to levels where the saturation of the link is expected. This requires implementing the MAC protocols and traffic generators within a network or architecture simulator that replicates the WNoC scenario.

In our case, evaluations are carried out in the cycle-accurate architecture simulator Multi2sim [46]. Multi2sim has been augmented with wireless on-chip communication modules that model collisions and multiple MAC protocols, on top of which we implemented multi-channel versions of the token passing and random access protocols

described above. Multi2sim admits synthetic traffic and multithreaded applications. In this deliverable, we use the synthetic traffic generator described below.

2.3.2.1 Traffic Patterns

Typically, NoCs are evaluated with synthetic traffic models that have, as main parameter, the injection rate λ in packets/cycle. Widespread simple models assume a Poisson process with the same average injection rate for all cores. However, as we will see in Section 3.1, traffic shows a clear self-similarity caused by the data dependencies within the applications. Moreover, common memory patterns such as producer-consumer lead to some cores transmitting more often than others. Our traffic model takes these aspects into account as follows.

Temporal burstiness: We model a heavy-tailed distribution of traffic via a Pareto distribution [35]. In more detail, injection is composed by bursts of length t_{ON} followed by periods of silence of length t_{OFF}. Bursts and silences are expressed as

$$\begin{aligned} t_{ON} &= \frac{b_{ON}}{(1-U)^{1/a}} \\ t_{OFF} &= \frac{b_{OFF}}{(1-U)^{1/a}} \end{aligned} \tag{2.3}$$

where $b_{ON} = 1$, $b_{OFF} = b_{ON}(\frac{1}{\lambda} - 1)$, U is a random generator following a uniform probability distribution of values between 0 and 1, and $a = 3 - 2H$. The value of $H \in [0.5, 1)$, the Hurst exponent, leads to increasing degrees of self-similarity as $H \to 1$.

Spatial hotspotness: To model an uneven injection of traffic across nodes, we make use of the hotspotness parameter σ proposed in [47], where σ represents the standard deviation of the spatial injection distribution. In particular, we generate a Gaussian probability distribution with σ^2 variance and sample it with N points corresponding to the number of cores. Then, each point is randomly assigned to a core ID. That value, normalized to 1, is used as probability of being assigned a new packet. Hence, low values of σ will lead to higher concentrations of traffic around a few cores.

2.3.2.2 Performance Metrics

Two metrics are generally employed to evaluate the performance of a given MAC protocol. On the one hand, the *latency* of the protocol τ_{MAC} measures the time spent by a packet in the MAC queue, this is, from the instant the message is queued until the transmission is successful. Multi2sim calculates such delay and adds it to the transmission and propagation delays, which are deterministic and dependent on the packet length, transmission speed, and distance among antennas. On the other hand, an equally important metric is the MAC throughput M, which is calculated as average rate of correct transmissions, in bits per second or any derivative (e.g. packets per clock cycle). Throughput is typically reported at saturation, this is, at the load after which throughput does not increase anymore.

In this work, we perform a complete evaluation by increasing the load from very small values to saturation. To provide a better view of the statistics of latency and throughput across protocols, we visualize the results as box-plots containing a complete suite of statistics across 15 load points corresponding to evenly spaced loads, from negligible to saturation.

2.4 Network-Architecture

For the context analysis, at the architecture layer, we are interested in assessing which is the communication load that a certain architecture and application generates and that the wireless network may need to serve. To this end, we use architecture simulators to assess a set of architectures and applications, obtaining traces and then parsing them for a spatiotemporal analysis. Section 2.4.1 describes the simulators, architectures, and applications for general purpose and accelerator-oriented systems. Then, Section 2.4.2 describes the methods and metrics obtained from the spatiotemporal analysis.

2.4.1 Simulation

The communication patterns and the induced pressure on the wireless network is dependent on the architectural organization of the target system-in-package. Striving to provide and unbiased view of the capabilities of in-package wireless technology, we herein consider two different scenarios. First, we investigate a state-of-the-art general purpose system, featuring multiple cores interfaced via a multi-level cache hierarchy, supporting off-the-shelf operating systems (Section 2.4.1.1). Second, we target a massively parallel, specialized system featuring clusters of ultra-low-power cores and dedicated scratchpad memories (Section 2.4.1.2). Both architectures are detailed in the following.

2.4.1.1 General Purpose System

The general purpose platform was modelled by the project partner EPFL using the gem5-X full system simulation environment [48]. The modelled system is organized in 4 clusters, where each cluster comprises 4 ARMv8 out-of-order cores running at 2 GHz. Each core has private 32 kB L1 instruction and data caches, and each cluster has a shared 1 MB L2 cache. The main memory is characterized is a DDR4 DRAM of 4 GB running at 2400 MHz.

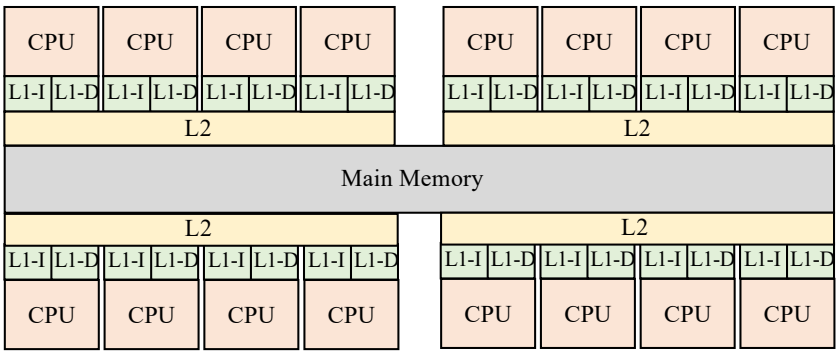


Figure 2.7: Target general purpose system.

We targeted a collection of workloads:

- 3 applications from the SPLASH-2 suite (OceanCP, Radiosity and Raytrace) [49]
- 2 CNNs mapped with intra-layer parallelism i.e., in which the computation of each layer is distributed among cores (MobileNetV1 and MobileNetV2 [50])

- 4 CNNs with inter-layer parallelism i.e., in which each CNN layer is mapped to a core (AlexNet [51] and the three Chatfield variations VGG8F, VGG8M and VGG8S [52])

Communication traces were acquired using dedicate gem5 components termed "communication monitors", which were instantiated in-between each L2 cache and main memory, hence logging all transactions at the cluster level. Each entry in the trace indicates the processor originating the data transfer, the transfer direction (read or write), its size in bytes, and the transfer time stamp. Logging was performed emulating an ideal interconnect, where any transaction consumes one clock cycle only. In all cases, the virtual system ran Ubuntu 16.04.

2.4.1.2 Massively Parallel Accelerators

The massively parallel accelerator platform was modelled by the project partner UNIBO using the GVSoC full system simulation environment [53]. The modelled system is organized in up to 512 PULP clusters, where each cluster comprises up to 16 PULP RISC-V cores (CORES), 1 In-Memory Accelerator (IMA), up to 1 MB shared scratchpad data memory (L1) and a DMA subsystem to move data between memory hierarchy autonomously. The main memory (L2), of up to 1.5 GB, is globally shared by all the clusters. The entire system frequency is set to 1 GHz. Clusters and L2 are interconnected via a scalable and parameterizable hierarchical network, whereas every level has its own latency and bandwidth.

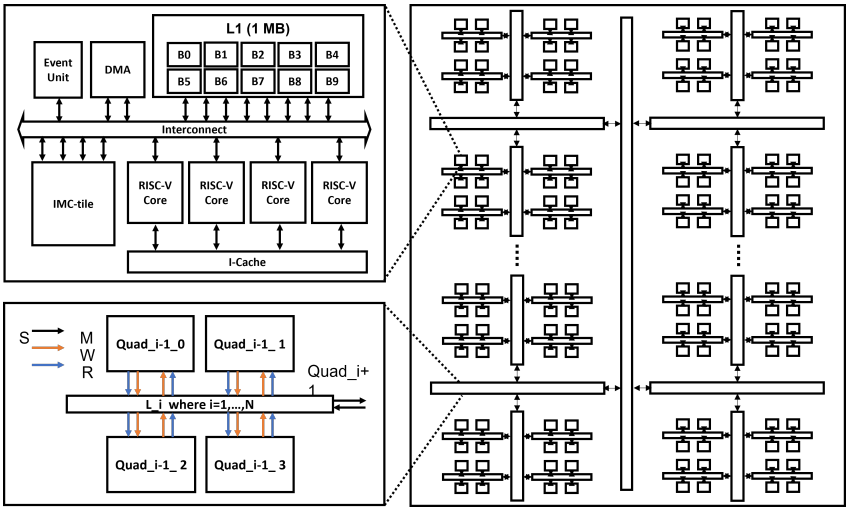


Figure 2.8: Target massively parallel accelerator system.

We targeted two workloads:

- ResNet-like CNN inference mapped with inter- and intra-layer parallelism, i.e., layers are distributed among groups of clusters, while the single group shares the layer execution [54]
- ResNet-like CNN training mapped with distributed data-parallel computational model, i.e., every cluster performs training passes (forward and backward) independently on different batches [55].

Communication traces were acquired using GVSoC's system traces, in particular the ones related to the DMA movements between scratchpad memories (cluster-to-

cluster communications) or between scratchpad memory and global memory (cluster-to-L2 communications). The system traces are non-intrusive special events used by the simulator to keep track of the execution state of every module in the simulator with text messages. Each entry in the DMA traces indicates the source and destination of the data transfer, the transfer direction (read or write), its size in bytes, and the transfer time stamp. GVSoC emulates an ideal interconnect to extract traces, where transactions consume one clock cycle.

2.4.2 Trace Parsing and Analysis

Trace files obtained for each of the above mentioned architectures are pre-processed, cleaned and analysed. Method of analysis and graphs obtained for both the architectures are similar for common parameters and also customised for each. MATLAB and unix shell tools are used to process and analyse in conjunction with each other.

The communication traces for the system described in Section 2.4.1.1 are over 14 GB. There are 10 workloads in total. For each workload, there are four trace files detailing the packets that were sent and received at every of the core clusters (specifically, the memory-side ports of each L2). The files have been imported in MATLAB using multiple techniques to handle huge files, and then cleaned and analysed. Raw memory addresses are mapped to corresponding cores to identify source and destination of the packets. Timestamp of each packet is used, in the form of inter-arrival time, for temporal analysis. Ideal interconnect speed (1 GHz) was taken for calculating cycle time and cycle bin size.

For the architecture described in Section 2.4.1.2, data for 2, 4, 8, 16 and 32 clusters is obtained. The traces have been cleaned and the memory addresses mapped to corresponding clusters and the L2 memory for both source and destination. Similar metrics as above architecture were obtained with the same methods to compare both the interconnects. Each calculated value corresponds to a particular system, and no averages have been taken.

With regards to the trace analysis, we first note that communication data traversing an interconnection network exhibits temporal variance. In other words, the level of burstiness or packets per unit time is not constant and varies across applications. We thus parameterize this level of burstiness using a single parameter, the Hurst exponent H. Self-similarity is exhibited where $0.5 < H \leq 1$. There are a number of tests to measure H as self-similarity manifests itself in a number of ways. Here we use time-domain analysis based on the re-scaled adjusted range statistic, known as the R/S statistic. To obtain H, one plots $log_{10}\frac{R(s)}{S(n)}$ versus $log_{10}n$. This is called an R/S *pox plot*, where the slope of the R/S line is H. This slope is calculated using an inverse-variance-weighted least-squares curve fit.

Another interesting aspect to investigate is the spatial distribution of the traffic injection over multiple clusters or cores. Results in this regard may be useful for the identification of potential hotspots. To evaluate the spatial distribution, we calculated the coefficient of variation (CoV) as $c_v = \sigma/\mu$, where σ and μ are the standard deviation and mean of the spatial injection distribution. We chose this metric in order to measure dispersion while filtering out the dependence of the standard deviation with the overall number of injected messages. A higher CoV means a higher concentration of the packet injection over given cores.

3. Context Analysis

Wireless chip-scale communications are among the different candidates for interconnecting processing elements and memory within complex computing packages. Specifically, the wireless paradigm has been proposed as a complement to the wired interconnects to (i) reduce the latency in communication between distant processors, possibly across chip boundaries, (ii) alleviate existing bandwidth bottlenecks caused by IO pin limitations, and (iii) establish global and reconfigurable links. These are possible thanks to the inherent low latency, broadcast capability, and lack of path infrastructure of the wireless technology.

Designing such wireless networks *on chips* requires, as for any other network, understanding the resources (e.g. channels, bandwidth, power) available as well as the communication requirements (e.g. load, deadlines) to be satisfied. In prior works such as [26], such a context analysis has been initiated. In the framework of the WiPLASH project, however, the rules of the game change by the introduction of graphene antenna arrays, which could provide multiple channels through frequency tuning and reconfigurable beaming. Also, the introduction of AI workloads and accelerators and the potential need for adaptive architectures also pose new interesting challenges.

In light of the above, in this chapter we provide an update of the context analysis for wireless on-chip networks. First, in Section 3.1, we qualitatively state the main characteristics of the scenario as already underlined in prior works [26]. Then, we add two specific contributions to the existing analyses. In particular, we prove that computing packages can potentially support multiple frequency and spatial channels in Section 3.2. Then, we quantify the communication requirements of the architectures and AI applications considered in the WiPLASH project, as a relevant subset of the applications that may run in multi-chiplet systems in the future, in Section 3.3.

3.1 General Considerations

Wireless in-package communications gather a distinctive mixture of requirements and constraints, which will likely define the feasible solution space in protocol design. These are summarized quantitatively in Figure 3.1 and qualitatively discussed in the following subsections.

3.1.1 High Performance

Computing systems demand ultra-fast and reliable communications mainly because the communication latency slows down the computation and minor errors may corrupt a full program execution. Generally, systems are architected to hide latencies as much as possible, but at the cost of indirection and complex memory hierarchies [56]. Also,

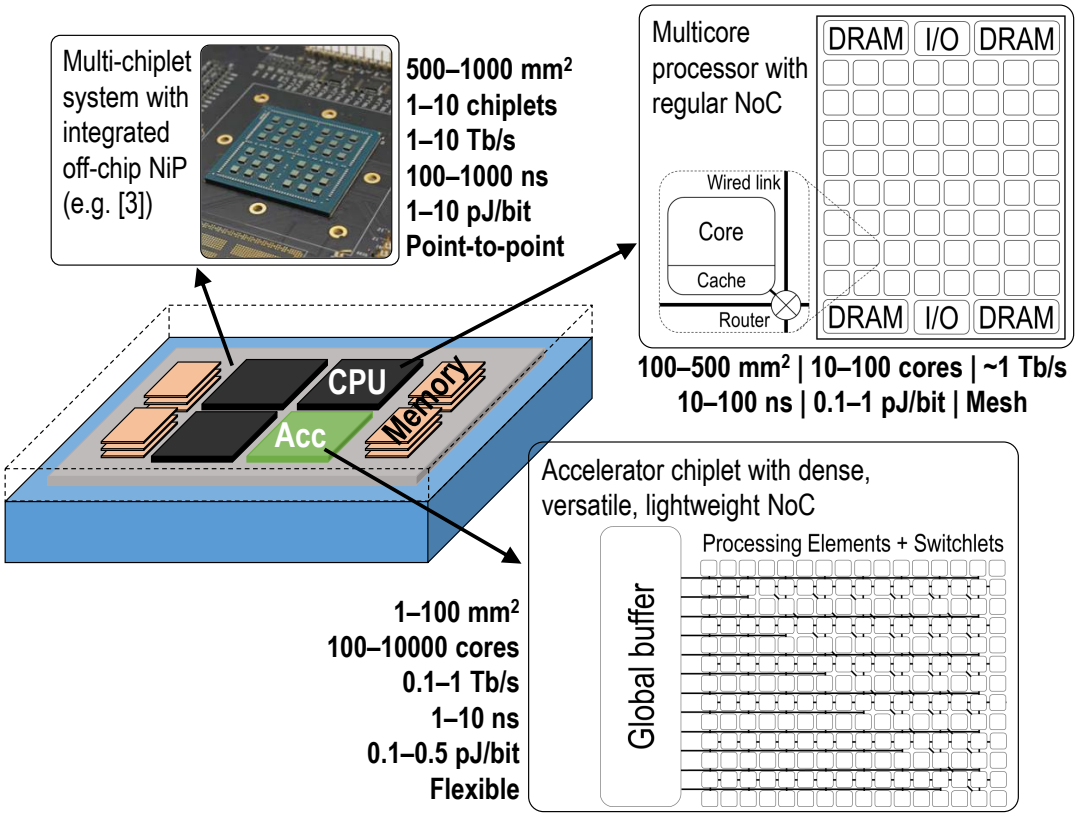


Figure 3.1: The chip-scale communication landscape in the heterogeneous chiplet era: Network-in-Package (NiP) to interconnect chiplets, Network-on-Chip (NoC) for multicore processors, and dense fabrics for accelerators. For the three scenarios, we list popular system sizes, number of nodes, bisection bandwidth, latency, energy per transmitted bit, and topology.

some applications may tolerate certain types of errors [57], but certainly cannot be considered a general-purpose technique.

Most WNoC proposals consider wireless in the order of 10–100 Gb/s to achieve system-wide latencies around or below 10 ns, hence being applicable (at least in terms of latency) to all the scenarios summarized in Figure 3.1. It is also widely accepted that the error rate should be comparable to that of wired interconnects, i.e. $\sim 10^{-15}$ [58]. This has several implications, namely: (i) the channel needs to support such a high bandwidth, (ii) the physical layer of design needs to use a modulation that either has a very high spectral efficiency in bit/s/Hz or a low order modulation with a very high modulation rate, and (iii) the MAC protocol needs to ensure high throughput with low delays, not only in average but also in the worst case.

3.1.2 Resource Awareness

Energy supply is generally guaranteed in computing systems, yet limited by heat dissipation constraints. Currently, power has actually become a first-class driver of processor design, relying on the use of power-gating among other techniques [59]. Similarly, chip real state is a precious resource due to cost reasons relative to fabrication and yield, i.e. larger chips have a higher probability of fabrication faults.

Taking into consideration that wireless in-package communications imply the integration of a large number of transceivers and antennas on a small number of chips, an effort must be made to minimize their area and energy footprint. This generally implies that simple low-order modulations are preferred as they do not require bulky or power-hungry components [60]. From the perspective of the wireless channel, resource awareness forces architects to minimize path loss while increasing the frequency, looking for wide spectral bandwidths to satisfy the high data rate requirements.

3.1.3 Monolithic and Static System

A multicore processor is basically a monolithic system from the designers' point of view and often a proprietary solution. The design team has a certain control over the architecture and the physical landscape of the system. This represents one of the main uniquenesses of the WNoC scenario, as in traditional wireless systems, the network stack and the applications are designed and developed by different teams. Additionally, communication takes place in a confined space that is static and known beforehand [61].

The implications are manifold. For instance, the chip-scale channels become quasi-deterministic at the physical and link layers [61]. Such a static property can be exploited to streamline the performance of encoders and decoders, which now do not have to depend on signal statistics [62]. Also, protocols can be optimized (or even co-designed) given the knowledge and control exerted over the architecture and applications [26].

3.2 Multiple Wireless Channels within Package

Signals radiated by the transmitting antenna suffer losses and dispersion as they propagate through the channel, which affect the ability of the receiver to correctly demodulate the transmitted information. Moreover, it is generally held that two overlapping transmissions through the same channel (frequency, space, and time) would create a *collision* and be lost. Due to the resource constraints of the scenario stated above, a research line considers that WNoCs/WNiPs may make use of a single broadband and broadcast channel and let a MAC protocol either avoid or resolve collisions. However, this requires not only that antennas be broadband, but also that the chip package supports very wide channels, which is difficult in light of the results of our previous reports [20, 27].

Graphene is able, at least theoretically, to change this landscape by providing an affordable way to enable frequency and space channels even within a chip. This is due to two key properties enabled by their plasmonic behavior at terahertz (THz) frequencies, namely, miniaturization and tunability. In a nutshell, plasmonic effects allow graphene antennas to be smaller, by even orders of magnitude, than metallic antennas of the same frequency. Moreover, the resonance frequency can be tuned over relatively wide margins by changing the electrostatic bias of the antenna. For more details, we refer the reader to WP1-WP2 deliverables or to related works in the literature [63–66].

The properties of graphene antennas would in principle enable the development of very compact and tunable antenna arrays, possibly opening the door to working

with multiple frequency and space channels. However, this capability does not have value unless it is demonstrated that the in-package wireless environment supports such channels. In this section, we aim to prove that a flip-chip package can indeed support multiple frequency (Section 3.2.1) and space channels (Section 3.2.2).

3.2.1 Support for Multiple Frequency Channels

To prove the support of multiple frequency channels, we perform multiple simulations over a given flip-chip package, modeling a vertical monopole antenna tuned to a particular frequency band each time. Figure 3.2 shows the S11 parameters obtained for one of the material thicknesses sweeps for the modeled antennas, which were required to operate at 60-80-100-120-160-200 GHz. In each simulation, all S-parameters were seen in a wide 10 GHz bandwidth.

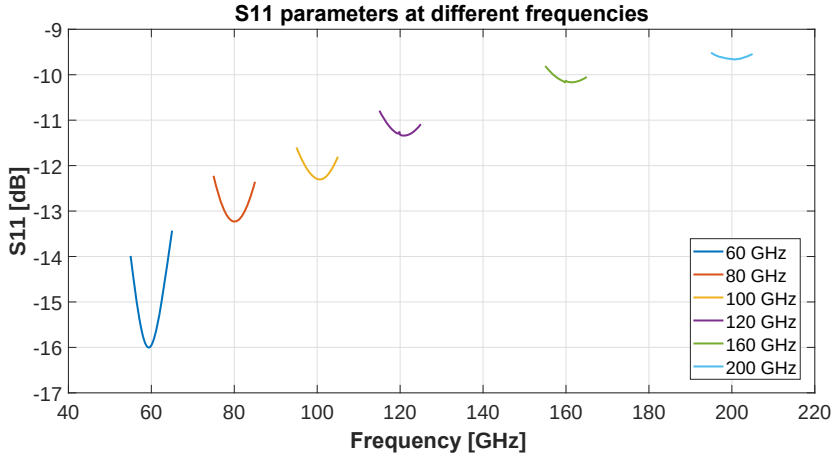


Figure 3.2: S11 parameter of the monopole antennas at different frequencies with Si=0.4mm and AIN=0.2mm.

From the simulations it was concluded that in general, we were able to tune the antennas to the required frequencies. Also, the higher the operating frequency, the larger and wider the S11 parameter. In our experiments, we could also see (not shown in the results for the sake of brevity) that by reducing the thicknesses of the materials, both Aluminum Nitride and Silicon, the values of the S11 parameters increase a little. Either way, the numbers obtained in all cases are around -10 dB for all frequencies, which makes them acceptable.

Figures 3.3 and 3.4 show the values of mean and maximum path loss across all antenna pairs, respectively. These values have been obtained for each frequency with different combinations of material thicknesses. Hence, the figures provide both (1) an assessment of the variations across frequencies and (2) a measure of the improvements achieved when making variations to the scenarios.

We can observe that the mean and maximum path losses do not vary appreciably when changing the frequency of operation. In the average, there is less than 10 dB variation across frequencies, whereas the worst-case path loss varies by around 13 dB at most. Even though there is not a formal definition on which variations are acceptable to consider the scenario to support multiple frequencies, it seems that the lossy nature of silicon leads to a non-resonant cavity without frequency-selective behavior.

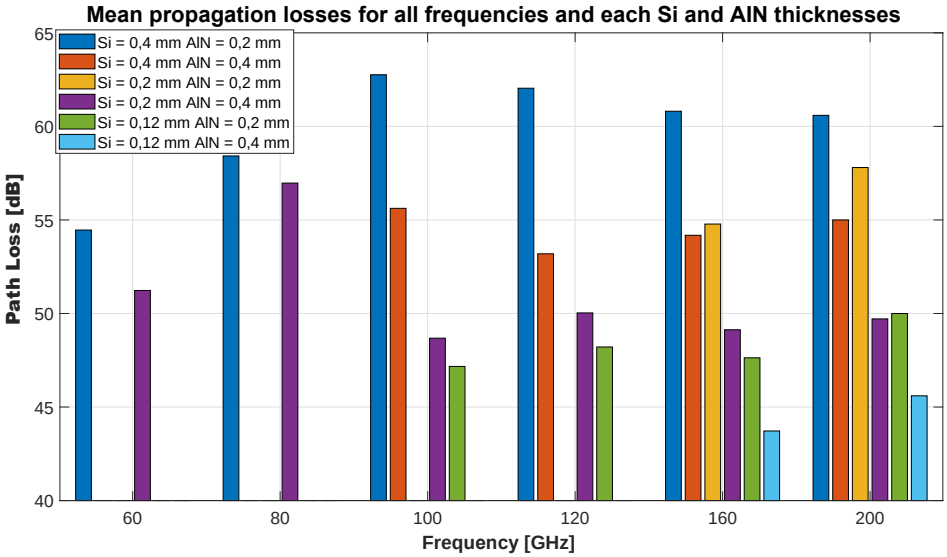


Figure 3.3: Mean path loss across different frequency bands and variations of the thickness of the silicon and heat spreader layers.

With regard to the dimensions choices, we observe how the results vary when the AIN thickness is increased and we decrease that of silicon. The best path losses numbers are obtained for the case where AIN=0.4mm and Si=0.2mm or 0.12mm where the operation frequencies allow this thickness to perform the simulations. With this combination, we achieved acceptable values of 45 dB and 53 dB for the mean and maximum path losses, respectively, for the 160 GHz and 200 GHz channels. The worst values where obtained for the combination AIN=0.2 and Si=0.4, with maximum losses of more than 80 dB and mean losses of over 60 dB.

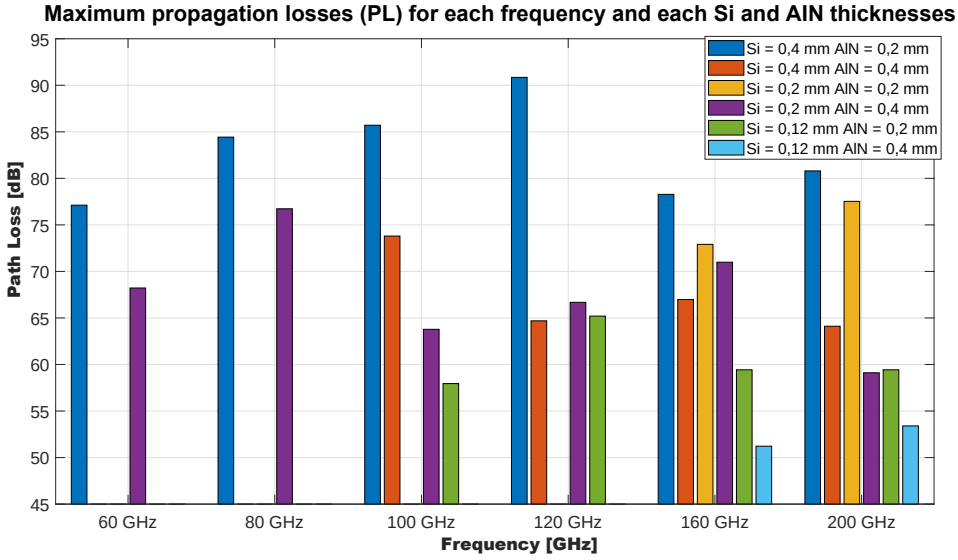


Figure 3.4: Maximum path loss across different frequency bands and variations of the thickness of the silicon and heat spreader layers.

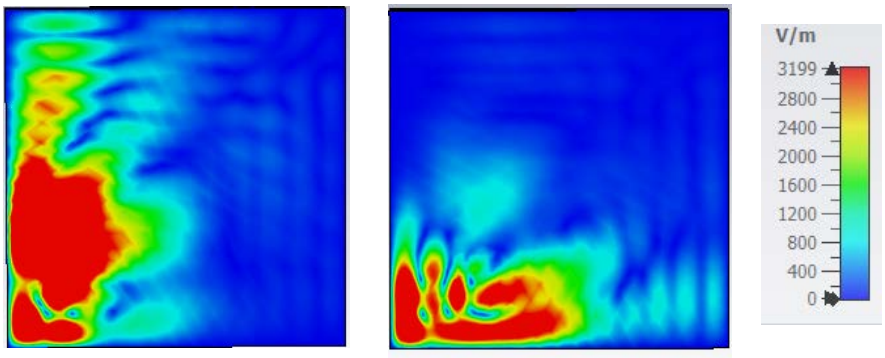


Figure 3.5: Field distributions of a phased array with configurations to steer the field along (a) the Y axis and (b) the X axis of the coordinate system.

3.2.2 Support for Multiple Spatial Channels

To demonstrate the support for multiple spatial channels, we now consider monopole arrays within the same exact package than in the previous section. We refer the reader to Section 2.1.4 for details on the array design methodology. As a result of this methodology, we are able to show that a 4×4 array, placed on the bottom left corner of the chip, can radiate towards the opposite corners with a clear and well shaped field distribution (see Figure 3.5). To that end, appropriate phases need to be applied to the signals fed to each antenna of the array. In Table 3.1, we list the specific excitation phases applied on each antenna to get the parallel channels.

Table 3.1: Phase distributions leading to the field distribution in Figure 3.5

Vertical Beam								
Port	1	2	3	4	5	6	7	8
Phase	90	120	150	180	0	30	60	90
Port	9	10	11	12	13	14	15	16
Phase	-90	-60	-30	0	-180	-150	-120	-90
Horizontal Beam								
Port	1	2	3	4	5	6	7	8
Phase	0	-330	-300	-270	-150	-120	-90	-60
Port	9	10	11	12	13	14	15	16
Phase	60	90	120	150	270	300	330	0

These results imply that, although conventional array theory may not be fully applicable within a chip due to some typical free-space assumptions not holding, we can still gain some control over the field concentration and create different spatial patterns by carefully applying the excitation phases to the antennas.

The next step is to come up with a combination that leads to two parallel channels radiating at the same time without interfering with each other. To do this, we place an identical array in the upper right corner (array 2), and perform phase sweep procedure described in Section 2.1.4. This sweep is based on the results obtained with the

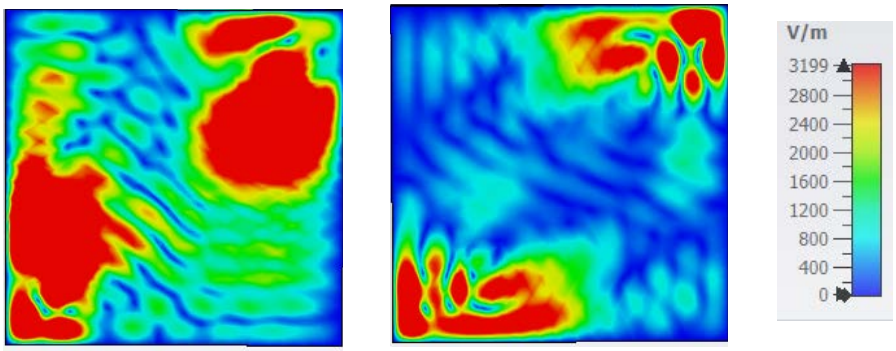


Figure 3.6: Field distributions of two phased arrays with configurations to steer the field towards the opposite corner along (a) the Y axis and (b) the X axis of the coordinate system.

array factor in the one monopole scenario. Figure 3.6 presents the results of our simulations. At first sight, when visually comparing the resulting field concentrations with those of Figure 3.5, it seems that we manage to create two parallel concentrations of energy that radiate at the same time, in principle without interfering with each other. This already proves one of our main goals, which was to create non-interfering beams inside a chip using antenna arrays.

To verify that the channels seen in Figure 3.6 radiate as independently as they appear to do, another post-processing step is performed by using the resulting fields of the phase manipulation of the two parallel vertical channels. We take the field created when only the array on the bottom left corner (array 1) radiates and subtract from it the field produced when both arrays radiate. This gives us the level of interference on array 1 when array 2 radiates. From the left plot of Figure 3.7, it is observed that the space where array 1 dominates is clearly along its Y axis, with bright colors, whereas the other side of the chip clearly shows dominance from array 2. Overall, both channels are separated by way more than 20 dB of interference, hence they are isolated from each other. To further quantify the possibility of having multiple channels, one can resort to a Signal-to-Interference Ratio (SIR) calculation as a measure of the reliability of the channel in this case. From the right plot of Figure 3.7, we see that the radiation from array 1 arrives to the intended opposite corner with a SIR of more than 40 dB, meaning that the interference level at the intended receiver is very low when both arrays radiate simultaneously.

The current analysis can be repeated at different frequencies and for different positions of the transmitting and receiving arrays. This is precisely one of the results of our extended work presented in [67]: the same methodology is followed to demonstrate that the spatial channels are also achievable at higher frequencies, i.e. 110 GHz, and at a higher granularity, this is, three channels in three rows/columns.

3.3 Traffic Analysis of Multiprocessor Architectures

Understanding the traffic to be served is a fundamental step towards the design of any network. As a result, in this context analysis, we aim to deliver some insight from the analysis of workloads derived from a range of architectures from general-purpose,

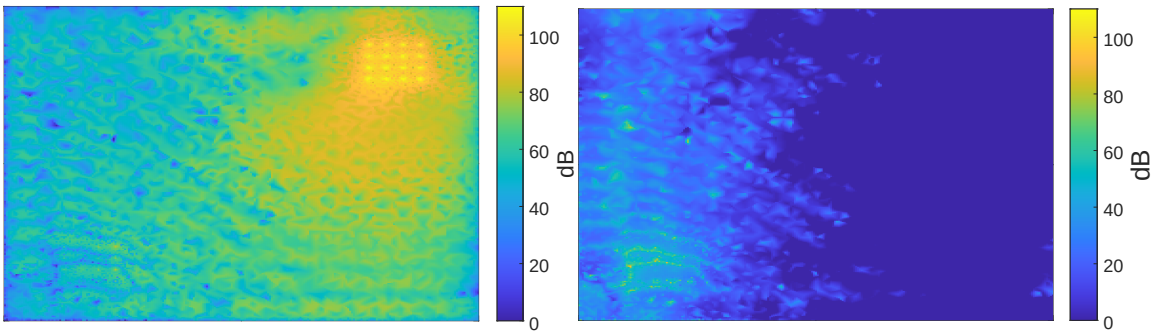


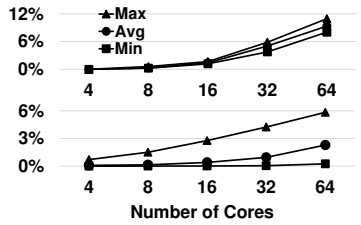
Figure 3.7: Interference field (left) and Signal-to-Interference Ratio (SIR, right) at 60 GHz in the case of vertical corner-to-corner communication.

single-chip, homogeneous processors to chiplet-based, accelerator-oriented and possibly heterogeneous systems. We summarize the outcomes of prior work [26] in Section 3.3.1 with respect to the former case, and shed new light on the communication requirements of the latter type of systems in Section 3.3.2.

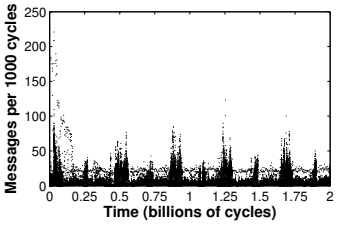
3.3.1 Analysis of Legacy Systems

The main characteristics of the workloads traditionally exhibited by multithreaded applications in single-chip multiprocessors are heterogeneity, variability, spatial hotspot behavior, and temporal bursty behavior. In more detail:

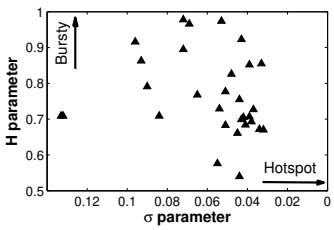
- **Heterogeneity:** Manycore processors need to support heterogeneous traffic profiles coming from diverse applications. Traditionally, local and unicast communications have dominated, but it has been shown that global and multicast flows can also become significant in manycore processors [7, 47]. Figure 3.8(a) illustrates this by plotting the percentage of long-range and multicast traffic as a function of the number of cores in legacy benchmarks [68]. The rise of the chiplet paradigm might exacerbate this aspect, as specialization may lead to different chiplets generating completely different intra-/inter-chiplet traffic patterns.
- **Variability:** The existence of a wide range of programming models and application domains may cause large changes in terms of communication demands from one application to another. Moreover, the particular chiplet combination in a heterogeneous architecture influence in such a variability, as different applications may require to use a particular accelerator chiplet intensely while others may not. Within each particular application, phase behavior also leads to wild variations on the traffic characteristics over time [69]. Such a behavior is exemplified in Figure 3.8(b), which clearly shows how the application *fluidanimate* alternates between communication-intensive and computation-intensive phases.
- **Spatial hotspot behavior:** Soteriou *et al.* revealed that most applications generate traffic unevenly across the processor cores [47]. Similarly, in our prior work [68], we confirmed that multicast flows can be concentrated around a few cores. Figure 3.8(c) reproduces some of these results by plotting the standard deviation $\sigma \in [0, \infty)$ of the injection distribution, where small values represent hotspot traffic.
- **Temporal bursty behavior:** Soteriou *et al.* also demonstrated that on-chip traffic is self-similar. Hence, packets are injected in bursts followed by relatively long



(a) Percentage of long-range traffic (3 hops or more, top chart) and multicast traffic (bottom chart) for the SPLASH2 suite over MESI coherence [68].



(b) Phase behavior exhibited by the traffic generated by a 64-threaded instance of the *fluidanimate* application over MESI coherence [68].



(c) Spatiotemporal characteristics of the applications analyzed in [47]. Each mark represents a specific application.

Figure 3.8: Workload characterization of different multiprocessor architectures and applications exhibiting (a) increasing heterogeneity, (b) intra-application variability, and (c) inter-application variability with bursty and hotspot traffic.

silences. The Hurst exponent $H \in [0.5, 1]$ evaluates this behavior, where large values indicate bursty behavior. As shown in Figure 3.8(c) and also in [68], traffic in multicore processors tends to be busty ($H > 0.7$), a trend that is likely to continue to hold in chiplet architectures as shown next.

There are at least two important lessons to take away from this analysis. First, the heterogeneity and variability of traffic suggests that protocols should be reconfigurable to adapt to large-scale changes with a reasonable cost. However, slow reconfigurability may not be enough to cover the needs of modern computing systems, since the harmful hotspot/bursty characteristics of traffic call for fast and fine-grained adaptivity.

3.3.2 Analysis of WiPLASH Architectures

Next, we provide the results of a similar analysis made on the architectures that WiPLASH explores, from general-purpose chiplet-based systems executing both traditional and AI workloads, to massively parallel accelerators. The methodology and architectures are described in Section 2.4.

3.3.2.1 General Purpose System

The communication traces for 10 applications were obtained. For each application, the communication at L2 port for each core was analysed. The main characteristics of such workloads are heterogeneity, variability, spatial hotspot behavior, and temporal bursty behavior captured as following:

- **Temporal Burstiness:** As the results of Figure 3.9(a) show, in all the traffic traces, injected packet arrivals (packets per unit time) are self- similar as H is well above 0.7 except *raytrace* and *oceancp*. Besides this, Figure 3.9(b) shows aggregated time series with 10k cycle bins for core 0 of *alexnet* which is more intuitive to understand and justifies its H value indicating highly bursty and self-similar traffic. The traffic pattern for all other applications is similar to Figure 3.9(b) over corresponding cores and is not shown here for conciseness. Figure 3.9(a) also provides us with bandwidth of data averaged over all cores for each application. The applications seem to behave iterative over the interconnect as

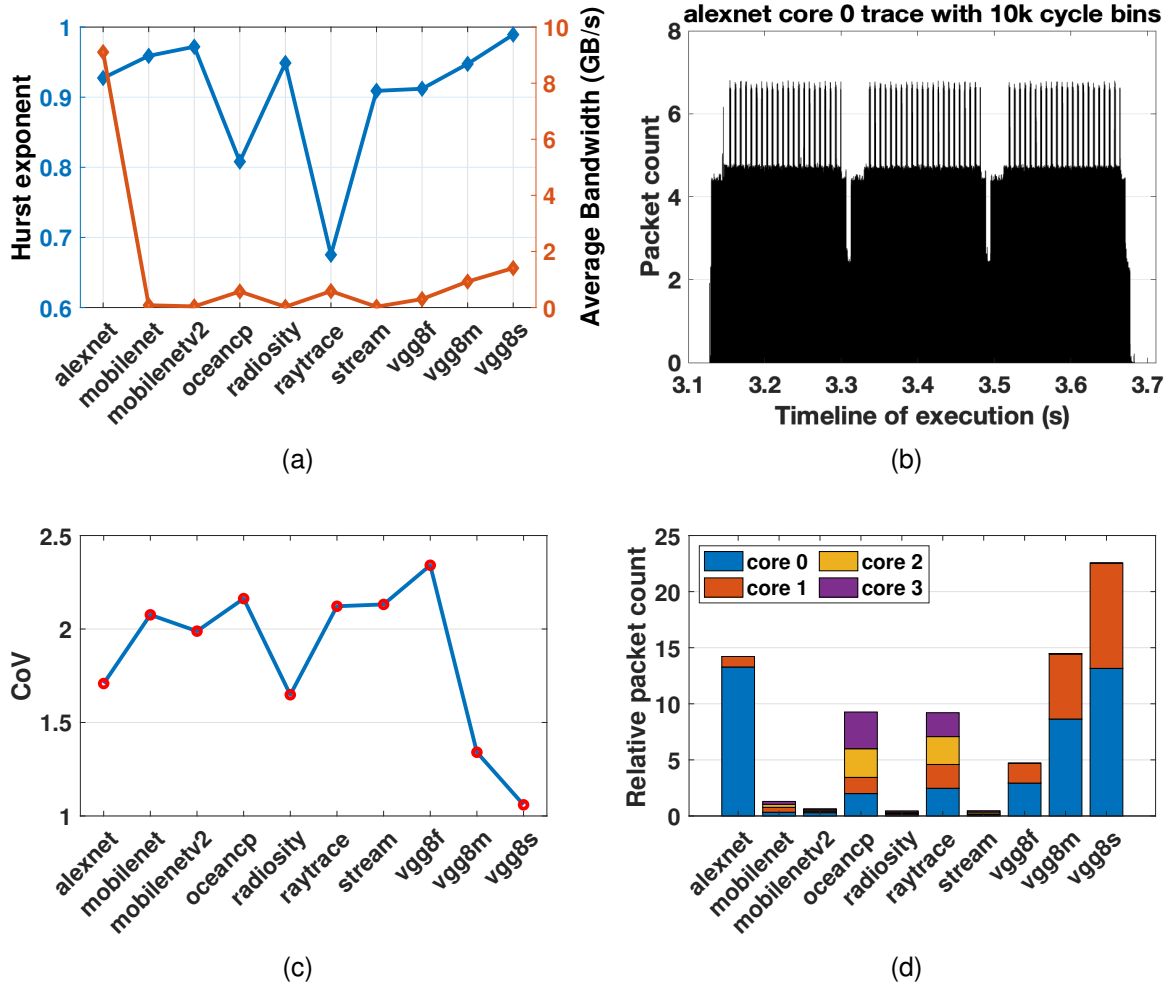


Figure 3.9: Traffic characterization of applications running on a general-purpose chiplet-based architecture showing (a) temporal burstiness and average bandwidth of each application, (b) pictorial proof of self-similarity and iterative behavior for the injected traffic of the alexnet application at core 0 aggregated over 10000-cycle bins, (c) Coefficient of Variation (CoV) of the spatial injection distribution of packets, and (d) relative packet count at each core for all applications.

their bandwidth wraps around 1 GB/s. The only exception is alexnet which produces huge amount of traffic at core 0 also confirmed by Figure 3.9(b).

- **Spatial Distribution:** The CoV for all applications is greater than 1, showing a high variance of traffic averaged over all the cores. More results from Figure 3.9(d) can be drawn. It shows the normalized packet count at each core for each application. It is clear that core 0 and core 1 receive the highest amount of traffic for most applications, exceptions being *oceancp* and *raytrace*. The reason is that these applications produce more random traffic in the network as shown by their hurst exponents. Distribution of traffic over the cores is accounted by the protocol under use than the underlying architecture, latter being uniform.
- **Variability and Heterogeneity:** The results also illustrate how applications lead to very distinctive behaviors, showing iterative behavior within the application and also large variations in the spatiotemporal behavior across applications.

3.3.2.2 Massively Parallel Accelerators

The communication traces were obtained for ResNet training for 2, 4, 8, 16 and 32 clusters. The main characteristics of such workloads are heterogeneity, variability, spatial hotspot behavior, and temporal bursty behavior captured as following:

- **Temporal Burstiness:** Figure 3.10(a) shows the values of Hurst exponents for each cluster size running ResNet training. For each cluster size, values above 0.8 meaning the traffic is highly bursty and self-similar showing a long-term memory effect. Also the exponent increases with cluster size meaning more burstiness with increased components. To get an overview of temporal traffic, Figure 3.10(b) shows the packet count in bins of 100 cycles for the system with 32 clusters. The traffic is dense at some points and almost non-existent at other. The latter can be explained by the algorithm implementation of ResNet distributed training where in the mentioned timeframe the clusters do not talk to each other and compute within themselves. The more bursty part is during the middle where the distribution is close to random and at the end where the aggregation part of the algorithm is processed. The average bandwidth of traffic communication remains under 1 GB/s until 32 clusters are used as shown in Figure 3.10(a).
- **Spatial Distribution:** Figure 3.10(c) plots the CoV of each system. Two observations can be noted here: first, CoV is higher than 1 for each of the systems, indicating the high spatial variance of injected traffic. Second, that CoV rises with the cluster size. This indicated more variation with increased components in a system due to cross-talk between clusters. An interesting view is provided by Figure 3.10(d). The x-axis represents the source of packet generation in a 32 cluster system and the y-axis represents the destinations. Here the memory is represented by L2. The pattern observed describes the selection of clusters for the ResNet training with distributed pattern in a binary fashion. The dense top row indicated that most traffic is destined towards the memory.

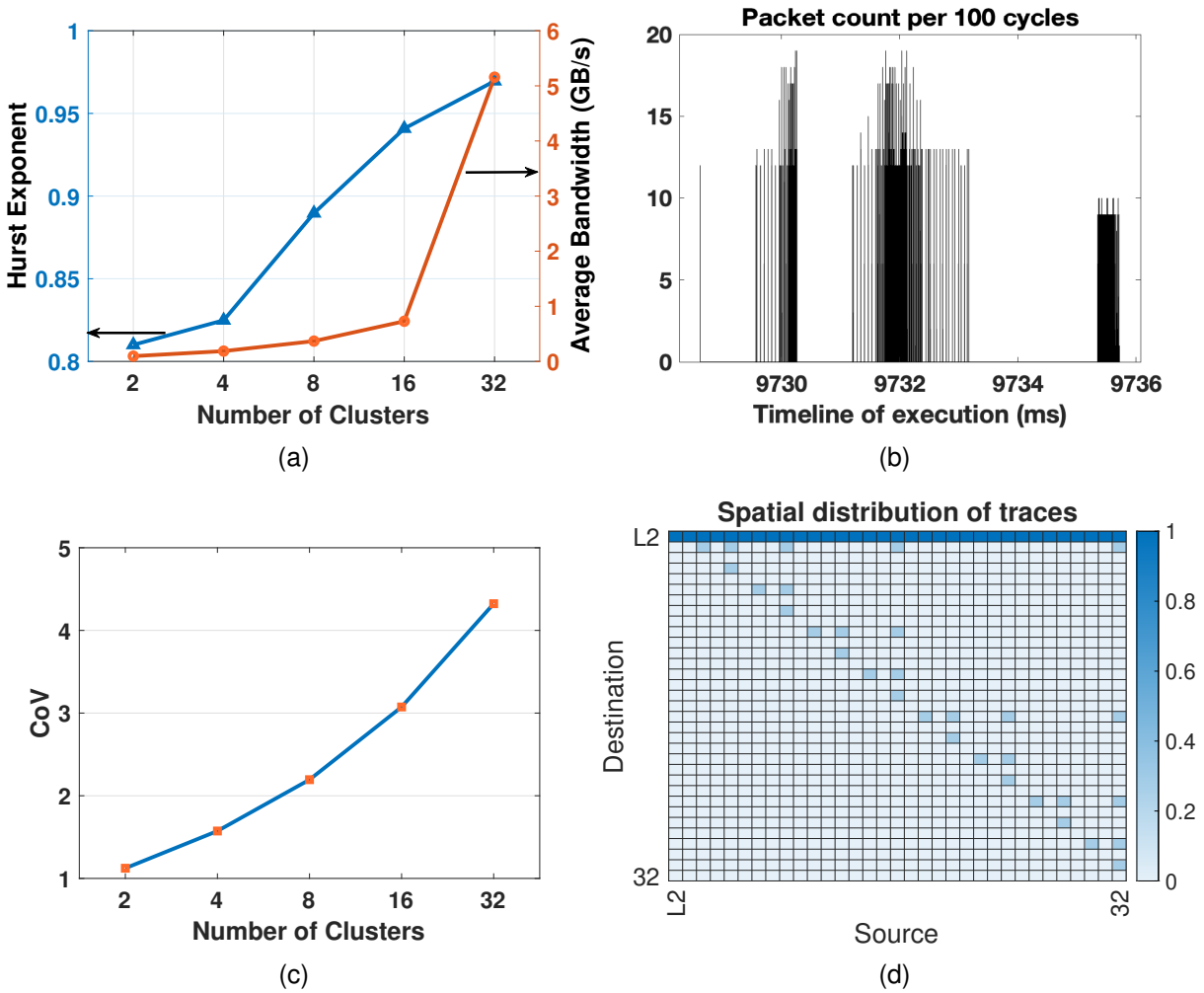


Figure 3.10: Traffic characterization of ResNet training on different cluster sizes of Massively Parallel Architecture (a) Hurst exponent and average bandwidth for each cluster size, (b) Pictorial "proof" of self-similarity in the "burstiness" preservation sense for injected traffic on 32 clusters aggregated over 100-cycle bins, (c) Coefficient of Variation (CoV) of the spatial injection distribution of packets for each cluster size, and (d) heatmap showing traffic movement from source to destination with 32 clusters.

4. Physical Layer

The Physical Layer (PHY) defines how bits are transmitted over the wireless links and, thus, plays a fundamental role in determining the requirements of the associated transceivers. 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.

There are two fundamental decisions that affect the physical layer deeply. The first one is the frequency of operation, which determines the dimensions of the required antenna, commensurate to the classical definition of wavelength λ in metallic antennas or to the Surface Plasmon Polariton (SPP) wavelength λ_{SPP} in graphene antennas. It also determines the available bandwidth and the complexity of designing an accompanying transceiver, especially if the frequency approaches f_T or f_{MAX} of the technology. The second decision relates to the modulation to use, which not only defines the transmission speed in bits-per-second for a given bandwidth in Hertz, but also largely impacts on the complexity of the required transceiver to implement it. Finally, the modulation also determines the Signal-to-Noise Ratio (SNR) required to achieve a given Bit Error Rate (BER) at the receiver end.

As further explained in Deliverable D3.2 [27], in the wireless on-chip scenario, one tends to increase the frequency of the system as much as the technology allows while choosing a simple but fast modulation to minimize the area required by the antenna and the transceiver. One may also resort to a single broadband channel in an attempt to minimize the amount of filters or other components required to implement a set of narrower channels. **However, such a trend may change if designers are able to provide multiple channels which are broadband and separated enough, either in frequency or space, to minimize inter-channel interference.** This could seem the case in WiPLASH, since graphene antennas could provide wide tunability [70] and, as shown in Section 3.2, the in-package environment might be able to support such widely-spaced and weakly-interfering channels.

Due to the introduction of multiple on-chip channels, the PHY will be impacted in ways that are summarized in Figure 4.1. A first decision to make is whether the transmitter is equipped with means to split a single packet and transmit multiple parts of it at multiple channels simultaneously. As discussed in Section 4.1, this would help reduce the transmission delay which is useful at low loads, while also alleviating the requirements in the serialization and data conversion stages. Then, depending on the channel that is assigned (and hence the possible changes in path loss or even antenna gain that may ensue), one may need to tune not only the antenna frequency, but also the modulator frequency and the power amplifier frequency and gain. This is discussed in Section 4.2. Finally, one would need to develop a controller which, given a channel assignment, is capable of configuring the transceiver so that the appropriate

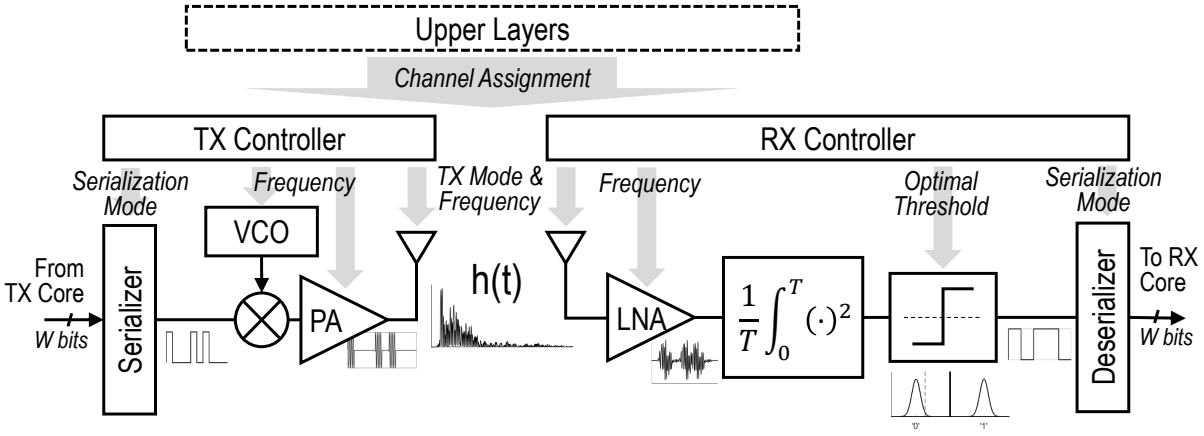


Figure 4.1: Impact of upper layers to PHY. Example with OOK modulation and energy detection at the receiver. Channel assignments and MAC policies may affect the serialization rate, the frequency of modulation, as well as the mode and resonance frequency of the antennas. At the receiver, the decider may be optimized based on the link budget.

modes of serialization, data conversion, modulation, amplification, and antenna tuning are activated. A design for such a controller is given in Section 4.3.

4.1 Handling Multiple Channels

The availability of multiple space and frequency channels in principle allows to increase the overall capacity of the network, increasing the throughput and reducing the delay. These channels can be used to either (1) transmit a single message from a single transmitter faster using multiple channels in parallel, (2) transmit multiple messages from multiple transmitters in parallel. The first choice reduces the transmission latency due to the higher data rate, whereas the second one generally alleviates the pressure on the MAC protocol.

Next, we discuss the different alternatives in terms of transceiver and antenna design to accommodate the capacity of handling multiple channels, to then describe the serialization considerations stemming from the use of multiple channels in a single transmitter. The discussion on how to use multiple channels at the MAC level is given in Chapter 5.

4.1.1 Impact on Transceiver Design

The accommodation of multiple channels at the physical layer depends on multiple aspects, which include the following:

- Whether the antenna (arrays) are fixed or tunable.
- Whether arrays can transmit multiple independent beams simultaneously.
- The modulation used.
- Whether the transceiver is fixed or tunable in frequency.
- Whether we can afford to have multiple modulators in parallel.

While our results in previous sections show that antenna arrays leading to a sort of beaming are possible on-chip, the realization of multi-beam patterns is less likely. Moreover, transmitting multiple (potentially different) messages over multiple beams as it is done in multi-user MIMO [71] requires signal processing techniques that are prohibitive given the tight area and power budget constraints of the application at hand. Therefore, we will limit our discussion to the handling of multiple frequency channels at the physical layer. Also, for simplicity and agreeing with the short introduction made at the beginning of this chapter, we assume that the modulation is On-Off Keying (OOK).

Among all the different possibilities stemming from the list of decisions above, we will start from a single-channel fixed design and progressively add tunability to the different components towards having a fully tunable RF front-end. We will not discuss alternatives that are unfeasible, such as employing multiple tunable transceivers with a single fixed non-tunable antenna. The different options are graphically represented in Figure 4.2. Next, we discuss the pros and cons of the different options, as summarized in Table 4.1.

(a) Fixed single RF-chain and antenna: this is the case by default where both antenna and RF-chain are fixed to a single broad frequency band. This leads to minimum complexity and overhead, but does not offer any flexibility to switch to another frequency channel, nor transmit via multiple channels simultaneously. If the antenna is replaced by a phased array, one would be able to transmit through a specific spatial channel.

(b) Fixed multiple RF-chains and antennas: this option would lead to maximum flexibility as the information can be transmitted to any of the frequency channels or even multiple of them simultaneously as indicated in Figure 4.3. Also, one can receive from all the frequencies at the same time if required. Thanks to this, one could do very quick broadcasts using all channels for a single message, or run *scatter* primitives efficiently by sending multiple different pieces of information to multiple destinations in parallel. Of course, this option leads to the maximum overhead as the space-consuming RF-chain is replicated several times. Finally, this can be combined with spatial channels by having multiple arrays instead of single antennas.

(c) Fixed multiple RF-chains and single tunable antenna: to reduce the overhead of the previous option and considering that graphene could lead to the building of widely tunable antennas, one could feed multiple transceivers to a single tunable antenna through a selector. However, this would lead to only being able to transmit through a single (yet choosable) channel still at the expense of a considerable area overhead, unless aggressive RF component reuse techniques are implemented or ultra-broadband sub-systems are employed [72]. Similarly, one cannot receive from all channels at the same time, meaning that the antenna should be tuned to the appropriate channel – otherwise information may be lost. In any case, access to multiple channels can be leveraged at the MAC layer to reduce the latency. If the tunable antenna is replaced by a tunable phased array, one would be able to transmit through a specific spatial channel as well.

(d) Tunable RF-chain and antenna: in this final case, the entire RF-chain including the transceiver and the antenna are tunable. In reality, not all the components of the transceiver need to be tunable, as the baseband remains unchanged and some sub-systems such as the power amplifiers can be extremely wideband [72]. One would achieve a modest overhead with respect to the fixed solution while still retaining flexibil-

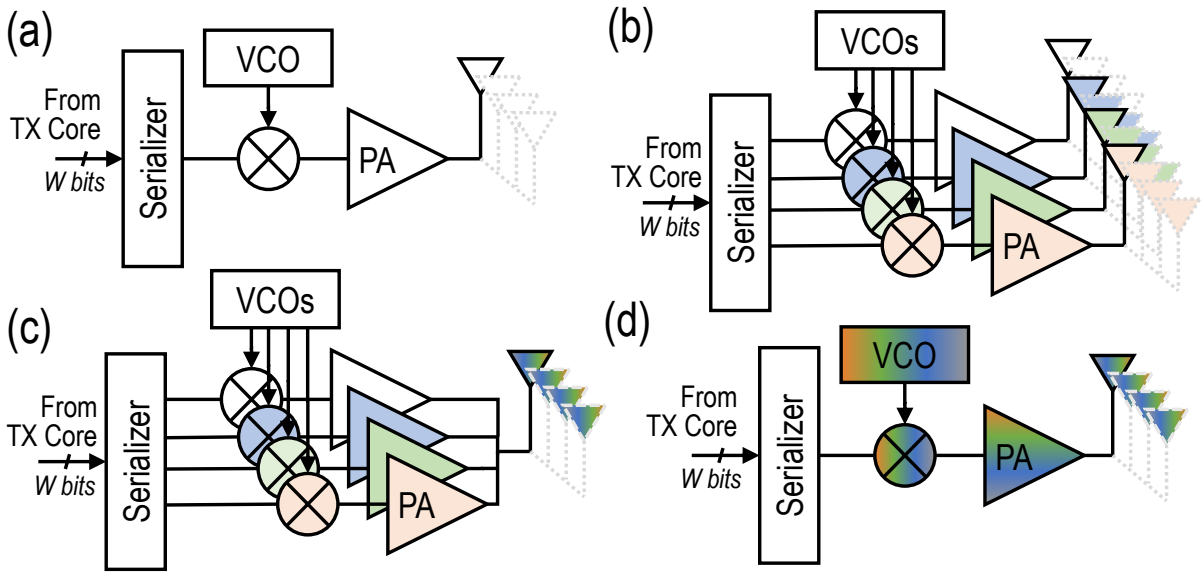


Figure 4.2: Evolution of a multi-channel PHY implementation from (a) single fixed RF-chain, to (b) multiple fixed RF-chains, (c) multiple chains with tunable antenna, and (d) single tunable RF-chain. Grey antennas denote the possibility of having an array instead of a single antenna, regardless of whether it is tunable or not.

Table 4.1: Summary of different PHY schemes.

Antenna	Transceiver	Channel Selection	Parallel Transmission	Overhead	Crosstalk
Single Fixed	Single Fixed	No	No	Low	No
Multiple Fixed	Multiple Fixed	Yes	Yes	Very High	High
Single Tunable	Multiple Fixed	Yes	No	High	Medium
Single Tunable	Single Tunable	Yes	No	Moderate	No

ity to transmit to a selected frequency channel. Also, if the tunable antenna is replaced by a tunable phased array, it would be possible to transmit through a specific spatial channel as well. However, this renders unfeasible the transmission to or reception from multiple channels at the same time.

4.1.2 Assessing Multi-channel PHY

From Table 4.1, we reiterate the qualitative observation that the multi-chain and multi-antenna option provides the highest level of flexibility. Indeed, this option not only allows to choose among a set of channels to transmit, which can be leverage at the MAC layer as we will see in Chapter 5, but also to use multiple channels concurrently to transmit a single message. Hence, the schemes shown in Figure 4.3 could be implemented, where two or even four channels are used to transmit fractions of a packet hence cutting the transmission delay in half or by four times.

To implement a multi-channel PHY, besides having multiple RF front-ends with their RF chains and antennas, the serialization step should be adapted so that bits are fed at the right speed to the correct transceiver. A possible implementation of such adaptive serializer is shown in Figure 4.3. A first serialization needs to be done to serialize the W bits coming from the processor or memory into 4 high-speed lanes. Then,

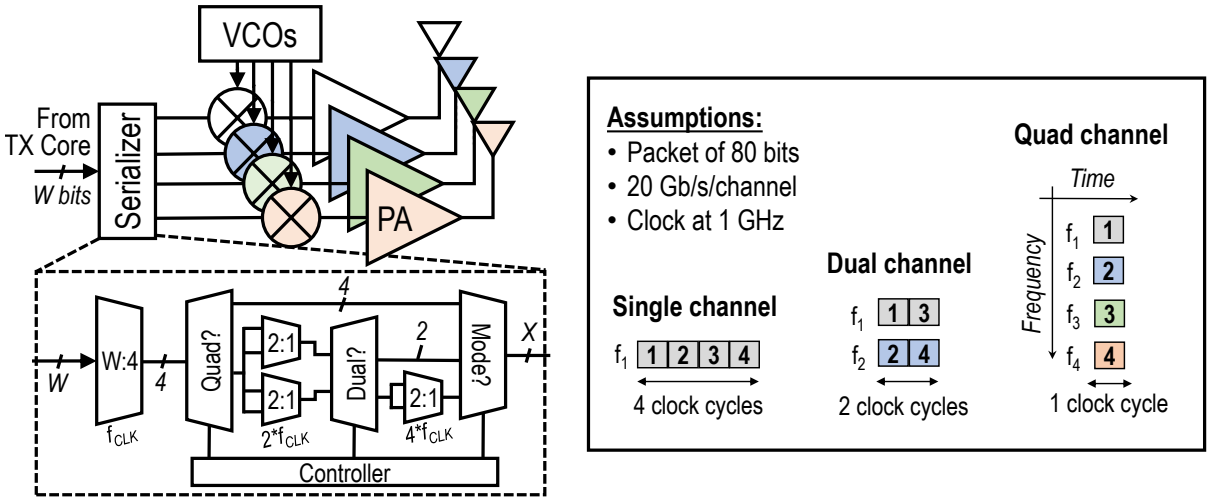


Figure 4.3: Schematic of a multi-configuration serializer feeding a multi-chain transmitter and the modes of transmission that can result of it.

two DEMUX-MUX stages serve to choose between quad, dual, or single channel. In between these stages, 2:1 multiplexers allow to translate between the quad-channel configuration and the other two.

To understand the area and power overhead of this option, we first refer the reader to Deliverable D3.2 [27]. The models therein indicate that, in general, the transceiver area and power increase with the frequency. In the case of OOK modulation, we estimated an area and energy between 0.45 mm^2 and 1.7pJ/bit, and 0.63 mm^2 and 5 pJ/bit in the range between 60 GHz and 240 GHz. Compared to these figures, the overhead of the serialization stage would be modest given that only a few extra multiplexers and demultiplexers are needed. In our example here, we would need to accommodate four channels of 20 GHz each, which we assume centered in 60-90-120-150 GHz with 10 GHz guard band in between. Hence, the cost of a quad-RF transceiver would skyrocket from 0.45 mm^2 and 1.7pJ/bit to 2.1 mm^2 and 2.5 pJ/bit approximately, respectively. In comparison, and according to the models in D3.2 [27], achieving a similar data rate than the quad channel option using a single ultra-broadband transceiver (assuming that the wireless in-package channel allows that) would require half the area and a similar power budget.

Another pertinent question is whether the improvement in terms of transmission delay has an impact on the overall performance of the network. In this respect, even though reducing the transmission delay by a given factor is generally beneficial, we observed in Deliverable D3.2 [27] and later in Chapter 5 that the overall latency of a transmission is largely impacted by the time spent at the MAC layer (either waiting to transmit or resolving collisions). In fact, the benefit of reducing the transmission delay from, say, eight cycles to two cycles may be masked by the overhead of the MAC protocol. To illustrate this, we take the baseline MAC protocols described in Section 2.3 in a link shared by 64 nodes and modify the transmission delay from eight cycles to two cycles (that could correspond, for instance, to the transmission of a short 160-bit packet using a 20 Gb/s and a 4$\times$20 Gb/s channel).

As shown in Figures 4.4 and 4.5, the improvement in the raw speed of the physical layer has an impact on the overall latency of the wireless link, especially in the case of

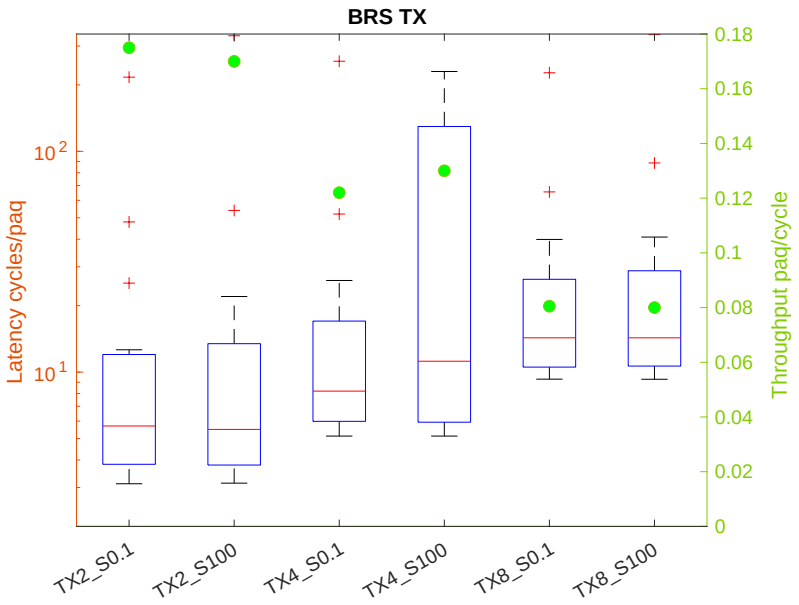


Figure 4.4: Performance of a transceiver link in a 64-node network with CSMA-like MAC protocol with hotspot (_S0.1) and spread out traffic (S100) from 40 Gb/s to 10 Gb/s of raw speed.

CSMA-like protocols where the low-load latency is mostly determined by the PHY rate. The average latency is higher for slow links, but the worst-case (outlier) latency points are similar regardless of the link speed. In terms of throughput, we observe that the decrease in throughput for slower links is not directly proportional to the loss of speed. As an example, the ratio between the throughput achieved with 40 Gb/s and 10 Gb/s links, which could be expected to be close to $4\times$, is actually $0.18/0.08 \sim 2.25\times$. In the case of token passing, the impact of transmission speed in latency is small because the latency is dominated by the overhead of passing the token. In terms of throughput, the difference can be proportional to the difference in raw speed, but it is noted that, when the traffic is hotspot, token passing loses much of its appeal at high speeds.

In conclusion, the overhead of having multiple RF-chains to increase the speed of a single transmission will probably not compensate for the overall gain in performance due to the large impact of the MAC protocol. Instead, as we will see in Chapter 5, it is desirable to use the multiple channels to alleviate the impact of the MAC layer by reducing either the waiting time or the likelihood of a collision.

4.2 Link Budget Considerations

Regardless of the choice to handle the diversity of frequency and space channels potentially offered by having multiple RF-chains and/or using tunable antenna arrays, one needs to consider the different channels may not have the same power budget and, hence, the output power may need to be tuned accordingly.

From link budget theory, one can evaluate the power that needs to be radiated at the transmitter to ensure a given signal strength at the receiver and, hence, a specific error rate boundary. In short, the SNR can be expressed as [73]

$$SNR = \frac{P_t \cdot G_t \cdot G_r \cdot B}{N \cdot L_{RX} \cdot PL \cdot R} \tag{4.1}$$

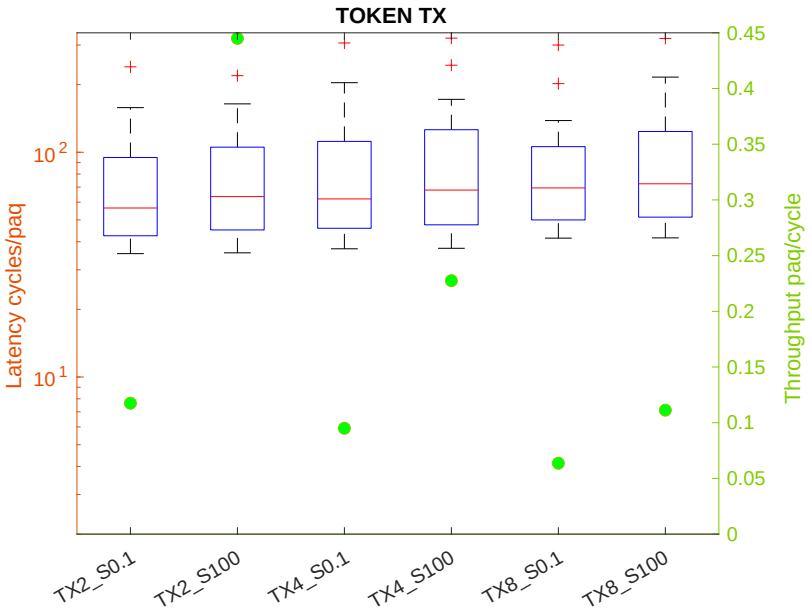


Figure 4.5: Performance of a transceiver link in a 64-node network with a token passing MAC protocol with hotspot (_S0.1) and spread out traffic (S100) from 40 Gb/s to 10 Gb/s of raw speed.

where P_t is the power at the output of the transmitter, G_t and G_r are the gains of the transmitting and receiving antennas, L_{RX} is the loss of the receiver, and $N = k{\cdot}T_0{\cdot}B{\cdot}F$ is the input noise power that depends on the Boltzmann constant k, the receiver noise temperature T and the noise figure of the receiver F. Other known terms discussed above are B for bandwidth, R for transmission rate, and PL for path loss.

Let us assume, for simplicity and without loss of generality, a fixed bandwidth B, data rate R, noise floor N, receiver loss L_{RX}. Then, when we select a given frequency or space channel, we need to take into consideration that:

- Not all frequency channels undergo the same losses, meaning that we can model PL as a function of the central frequency of the selected channel, $PL(f_c)$. As observed in Section 3.2.1, there may be a variation of $\sim$10 dB across channels.
- If tunable antenna arrays are used, one may need to distinguish between an omnidirectional mode used for broadcasting and a directional mode for spatial multiplexing [67]. Also, all spatial channels might not lead to the same directive gain. In these cases, the gain of the transmitting antenna G_t will vary possibly about a few dB.
- Again if tunable arrays are used, one may need to account for potential losses stemming from the fact that the array is not pointing towards the direction of the transmitter, which may results in the loss of a few dB in G_r.

Assuming that our objective is to adjust to the required SNR to the exact value required attain the target error rate (so as to not waste energy), then there is a need to adjust the transmitted power P_t to compensate for the variations of path loss and antenna gains as functions of the spatial and frequency channels used. In particular, we can employ a variable gain amplifier [74, 75] to select the gain required, with the advantage that since our channel is static and known beforehand, we can exactly apply

the required gain for any specific spatial-frequency channel combination. This could be done using simple models or a look-up table, as described next.

Assuming a linear increase of the power consumption of the amplifiers with respect to their gain, not having the amplifiers perfectly tuned can result in a power consumption penalty between 3.16$\times$ and 10$\times$ for a 5-dB and 10-dB mismatch, respectively.

4.3 A Controller for Adaptive PHY

As summarized in Figure 4.1 and based on the explorations done above in this chapter, an adaptive physical layer would require a digital controller capable of exposing the reconfigurability properties of the transceiver to the upper layers of design or, ultimately, to the architect. The controller could be in charge of multiple functions, namely:

- Determining the serialization/deserialization ratio and speed, and then configuring the corresponding circuits (see Figure 4.3).
- Waking up the specific transceivers required for transmission or tuning them to the appropriate channel, depending on the structure of the PHY layer (see Figure 4.2).
- Tuning the amplifiers to the appropriate gain configuration based on the spatial or frequency channel to be employed.
- Setting the decision threshold of the receiver to appropriate values based on the expected constellation at the particular frequency and space channel.

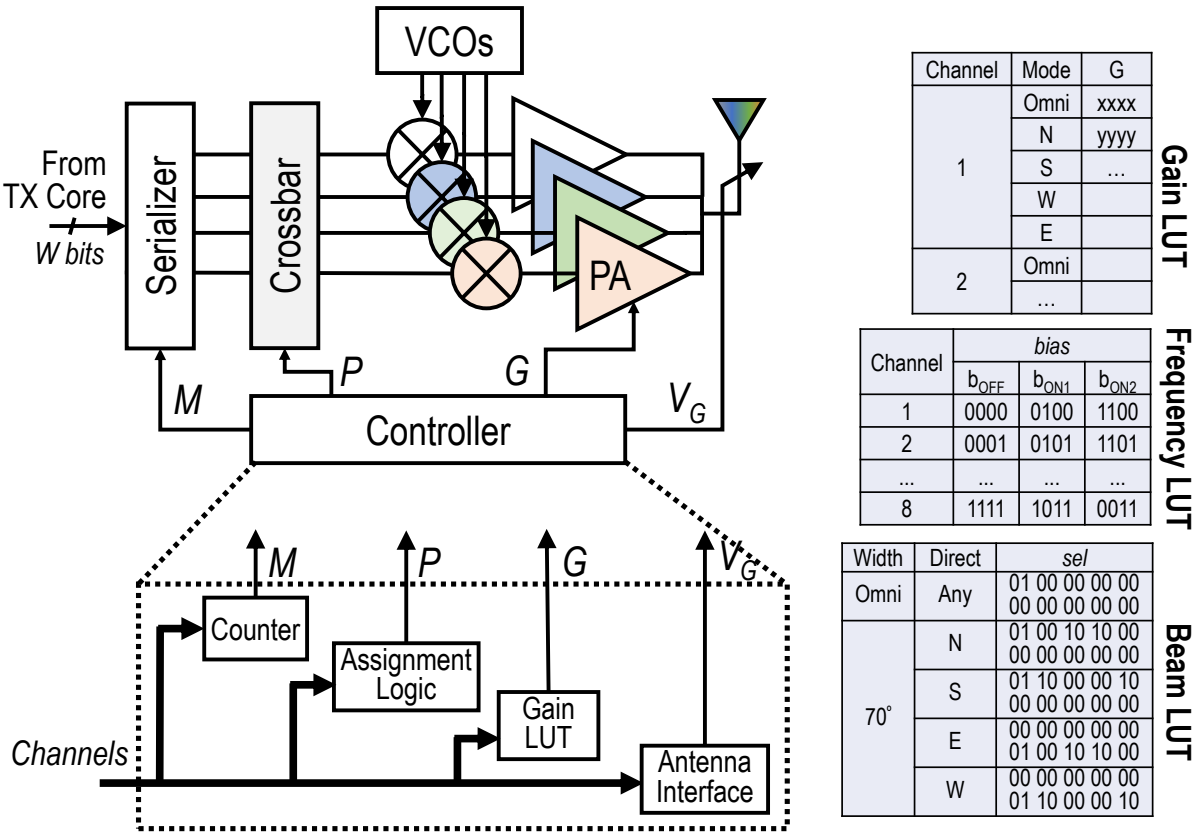
Although the decisions can be taken anywhere in the communications stack, we will here assume that decisions come from the MAC layer. The MAC circuitry is in charge of determining when to transmit a packet and, as evaluated in Chapter 5, through which particular channel(s). The choice of channel determines the frequency to use and the spatial mode of transmission, hence tuning the amplifier, transceiver and antenna. In case multi-channel transmission is allowed, then the serialization ratio is modified accordingly.

Figure 4.6 outlines a possible implementation of a controller that would take as input a code that represents the channel(s) assigned by the MAC protocol and outputs the signals M, P, G, and V_G that denote control signals to choose the serialization mode (quad, dual, single), the channel to use, the gain to apply to the power amplifier, and the voltages to apply to the graphene elements of the antenna. The receiver side could also have a controller, as depicted in Figure 4.1. Next, we qualitatively discuss the sub-blocks that generate the different signals in transmission.

Counter. This block is assume to count the channels from the binary code coming from the MAC circuitry. The output M is a one-hot vector encoding the mode of transmission, from single-channel to modes with more simultaneous channels being used.

Assignment Logic. This block provides a permutation vector P which will drive a $N_C \times N_C$ crossbar, where N_C is the maximum number of channels. This vector describes which stream of bits needs to go to which channel.

Gain LUT. This block distinguishes the frequency channel from the mode of transmission (omnidirectional or directional), which address a Look-Up Table (LUT) that has, as output, a string of bits G. This string of bits encodes the gain that a variable-gain



Channel	Mode	G
1	Omni	xxxx
	N	yyyy
	S	…
	W	
	E	
2	Omni	
	…	

Gain LUT

Channel	*bias*		
	b_{OFF}	b_{ON1}	b_{ON2}
1	0000	0100	1100
2	0001	0101	1101
...	...	...	...
8	1111	1011	0011

Frequency LUT

Width	Direct	*sel*
Omni	Any	01 00 00 00 00 00 00 00 00 00
70°	N	01 00 10 10 00 00 00 00 00 00
	S	01 10 00 00 10 00 00 00 00 00
	E	00 00 00 00 00 01 00 10 10 00
	W	00 00 00 00 00 01 10 00 00 10

Beam LUT

Figure 4.6: Controller structure in an example containing multiple RF-chains and a tunable graphene antenna.

amplifier should apply to the analog signals of that particular channel, based on a link budget analysis performed a priori.

Graphene Antenna Interface. Following the assumption made throughout the WiPLASH project, which is the existence of graphene antenna arrays providing frequency-beam tunability, here we summarize a controller design presented in [76] that exposes the reconfigurability capabilities of the antenna to the PHY and MAC layers. The controller translates directives coming from the MAC protocol to a set of voltages that determine the antenna state (frequency and beam).

The controller could be composed by an array of actuators driven by a digital interface, similar to the design of [77] at microwave frequencies. The interface of the antenna controller is composed by two LUTs that translate the antenna state requirements into digital signals that drive the actuator. The reconfigurability process has two steps:

1. **Frequency state:** In the first step, the frequency channel is translated into codes representing various levels of *bias*. For the sake of example, we assume three biasing levels: b_{ON1} for 0 phase, b_{ON2} for π phase, and b_{OFF} for completely tuning off the element. However, the scheme can be generalized to any number of phase–amplitude levels. Such bias levels need to be calculated for all frequency channels for the particular graphene antenna at hand (see Deliverable D1.2 [78] for some design examples). The number of bits of the *bias* signal depends on

the range of voltages and the resolution required to implement a given number of channels to a given graphene antenna.

2. **Beam state:** In the second step, the beam width and direction are translated into a code *sel* that identifies the *bias* level to apply to each antenna element. In the example of Figure 4.6, we assume the existence of five states, one for omnidirectional radiation and the other for directional radiation towards four sides following the results from Section 3.2.2. However, this could be generalized to any number of beam-states, depending on the size of the antenna and its capability to create spatial channels.

Finally, the signals *sel* and *bias* are input to a set of actuators which turn these digital signals into the voltage levels required to electrostatically bias the graphene elements to the states leading to the desired frequency tuning and transmission mode.

Overhead Evaluation. In the proposed controller, the main sources of area and power consumption are the LUTs and the data converters. To estimate the former, we model the LUTs as a single cache of 2KB with a line size of 32 bits and use CACTI [33] to derive their cost. At 32nm, this memory would be 90×40 μm^2 and consume less than 0.02 nJ to read a configuration, with a leakage power of less than 10 μW. Despite of the small overhead, this antenna interface would be capable of retaining as many as 512 antenna states (our examples have a few dozens) with fairly large resolution. To model the data converters necessary to drive the level shifters, we assume a conservative design with 4 bits (16 possible voltage levels) and a speed of 1 GS/s, enough to provide multiple channels and reconfigurability at a packet granularity. With the models of existing surveys [31], such a design would be below 0.01 mm^2 and 0.15 mW at 32nm. Compared with the area and power of an RF transceiver or the complexity of phase shifters or delay lines required for MIMO, these overheads are negligible. Finally, note that all the components of the antenna controller can work at the nanosecond scale. The change of graphene conductivity required to reconfigure the antenna also occurs fundamentally in a sub-ns scale [63], hence not posing a latency bottleneck in the tuning of the antenna.

5. Link Layer

Although this deliverable demonstrates that multiple space and frequency channels would be supported in an on-chip environment and tunable graphene antenna arrays, the number of links (understood as transmitter-receiver pairs) is very likely going to be higher than the number of channels. Since two overlapping transmissions in the same channel will fail with high probability, it is important to have link layer protocols that ensure that channels are shared fairly and efficiently.

In Deliverable D3.2 [27], we presented models of the latency and throughput of three different MAC protocols, i.e. CSMA, token, and fuzzy token, for different traffic conditions and a single-channel configuration. However, as multiple channels become available, a pertinent question is how should MAC protocols be extended to share the multiple channels efficiently. One could consider a strategy proposed in several works [13, 14]: a static channel assignment, this is, a node is assigned a single fixed channel and its transceiver and antenna are statically tuned to that channel. Even though this solution is simple and easy to reason about, it might not work well when the workload is variable in space and time – which is the case of several workloads as exemplified in Section 3.3. Hence, here we consider multi-channel extensions of CSMA and token passing as two representative MAC protocols for wireless on-chip networks.

In the following, we first explain the different ways we can extend CSMA and token passing with multiple channels in Section 5.1. Then, we evaluate these different options with traffic of different spatiotemporal characteristics in Section 5.2. Such an analysis will shed light on the impact of channel assignment on the protocol performance. In future work, we aim to expand the analysis by developing a multi-channel version of *fuzzy token* [44].

5.1 Multi-Channel MAC Protocols

In general, given a single-channel MAC protocol, there are several ways of adding support for multiple channels. Channels can be mapped statically or dynamically, and such a mapping can be done on a per-packet basis (channels are assigned to each packet individually regardless of the node that will transmit) or on a per-node basis (the same channel is assigned to all packets coming from a given node). In this work, we describe three possible distinct channel assignment strategies to CSMA and token passing, which adapt to each protocol particularities. The strategies presented here may not be optimal, but they are simple and representative of the possible channel assignment techniques that can be used.

5.1.1 Assignment Methods for CSMA/BRS

In CSMA-like protocols such as BRS [41], nodes contend for channel access and backoff when a collision occurs. When more than one channel is available, channel assignment can be done at an individual packet level (AS_1) or at a node level (AS_2 and AS_3) seekign to reduce the likelihood of collisions. Here, we test the following three possible methods:

- **AS1:** Channels are assigned to packets individually and randomly. When a node has a packet to transmit, the packet is assigned a random channel among the entire set of channels. When there is a collision, the affected nodes apply a random backoff while they also change their channel to a random one, so that the probability of the same nodes having another collision is minimized.
- **AS2:** In this strategy, nodes are statically linked to a given channel following a uniform distribution, this is, assuming that all nodes have the same probability of transmitting. Hence, if we have sixty-four nodes and four channels, the first sixteen nodes will be assigned to the first channel, the next sixteen nodes to the second channel, and so on. While this is not optimal for spatially unbalanced traffic, it serves as a baseline.
- **AS3:** In this strategy, nodes are statically assigned to a given channel following a distribution that tries to balance the load in each channel. To that end, nodes are ordered based on the expected load (normalized to the total load) and assigned to each channel in order, picking one from the top of the list and then from the bottom of the list. When the load assigned to a channel exceeds $1/N_c$ where N_c is the number of channels, that channel does not add any other node to it.

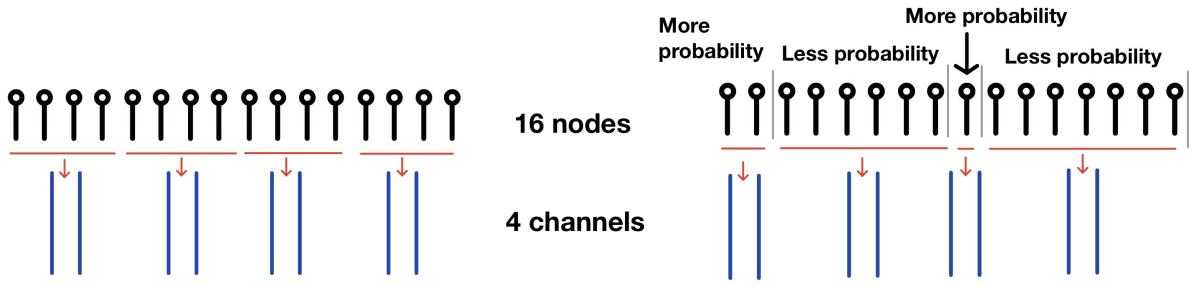


Figure 5.1: Graphical representations of assignment techniques AS_2 (left) and AS_3 (right) for BRS/CSMA assuming 16 nodes and 4 channels.

5.1.2 Assignment Methods for Token Passing

In token passing with a single channel, all nodes are sorted forming a virtual ring and the token is passed in order through that ring. In a multi-channel version of the protocol, each channel can be a token. The design decisions then lie on the number of rings to be deployed and the set of nodes that forms each ring, both seeking to reduce the average waiting time of the protocol. We study three alternatives:

- **AS1:** A possible strategy consists of having as many rings as there are channels, and mapping nodes uniformly to each ring. In other words, having N nodes and N_c channels, we distribute nodes in N_c rings of N/N_c nodes each regardless of their expected load.
- **AS2:** A second strategy would be to have a single virtual ring with multiple tokens circulating on it. We assume that tokens can jump over other tokens: e.g. when node i holds a token for multiple cycles during a transmission, then idle tokens that arrive at $i-1$ can directly jump to $i+1$.
- **AS3:** Finally, the third tested strategy is similar to the first one, but nodes are mapped to rings following a load distribution protocol, i.e. based on their expected load. This may lead to rings of different sizes, but in principle similar in the expected overall load.

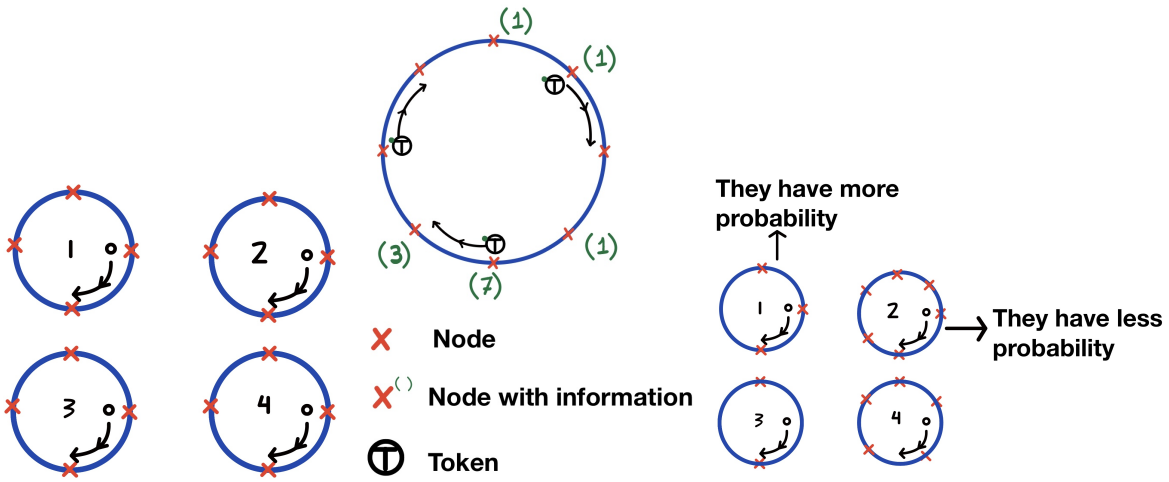


Figure 5.2: Graphical representations of the different assignment techniques for token passing. Note that in the third method, not all rings have the same amount of nodes as AS_3 maps nodes to rings based on their expected transmission probability.

5.2 Performance Evaluation

The architecture and application parameters are summarized in Table 5.1. To evaluate the performance of the different assignment options, we implement them within Multi2sim [46] where single-channel BRS and token passing were already implemented from [44], and compare the packet latency and throughput of the different options. Unless otherwise noted, the default values for the different parameters are $N = 64$ nodes, $N_c = 4$ channels, $H = 0.5$ (Poisson traffic) and $\sigma = 1$.

In all cases, the protocols are evaluated with multiple simulations starting with a low load and finishing with a load that saturates the network. Given the high number of protocol strategies and traffic types, instead of plotting the classical latency–throughput curve for each option, here we make use of box plots that summarize the latency and throughput statistics.

Table 5.1: Characteristics of simulated protocols and applications.

Wireless NoC Parameters	
Application	Synthetic traffic, H=0.5–0.9, σ=0.1–100
System	N=16–1024 cores, one antenna/core, N_c=1–4 channels
Network	80-bit (4-cycle) packets (preamble: 20 bits, 1 cycle)
Link	BRS [41], Token passing
Physical	OOK, 20 Gb/s

In our plots, the X axis shows the parameters under study, e.g. the type of assignment, the Hurst exponent for bursty traffic, or the σ parameter for hotspot traffic. The plots have two Y axis: the left axis represents the latency and corresponds to the box plot values, whereas the right axis represents the throughput and corresponds to single-value markers of saturation throughput. Note that the latency axis is generally in logarithmic scale, whereas the throughput axis is in linear scale. The throughput is expressed in packets per cycle and since a single packet takes 4 cycles in a single channel to be transmitted, the maximum throughput is 0.25 packets/cycle per channel.

Box plots summarize the latency statistics in the following way. We generate the latency distribution by sampling the results of the simulations between zero load and saturation load uniformly. Bottom and top black T markers of the plot describe the maximum and minimum values of the distribution, assuming that there are no outliers. The blue box describes the span of the latency between the first quartile (bottom of the box) and third quartile (top of the box). Also, the red line within the box indicates the average of all the different values of latency between the minimum and the maximum one. This red line can be useful to describe, in general terms, how the latency evolves with respect to the different evaluation parameters, namely, the type of assignment, the Hurst exponent, and the σ parameter. Finally, we note that the points that are represented with a red cross marker are outliers, which in this case corresponds to simulations with an unexpectedly large latency, either due simulating a very large load beyond saturation or the protocol becoming unstable beyond certain loads.

A box graph can be useful to study the stability of the system. A wide box indicates that the latency slope moves away from a stable system, which happens when the function slope approaches infinity. Otherwise, a narrow box indicates that the system is stable because the latency function does not increase significantly with the load, through all its points.

5.2.1 Number of Channels

In this section, we discuss the results shown in Figure 5.3 and Figure 5.4 for BRS and token passing, respectively, for an increasing number of channels.

Throughput. The results for token passing, shown in Figure 5.4, 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 shown in Figure 5.3 depict a different behavior than in token passing. Firstly, BRS cannot reach a saturation throughput as high as token protocol. The main reasons are that the traffic patterns evaluated here suit token passing well; whereas in BRS, channel contention and multiple collisions

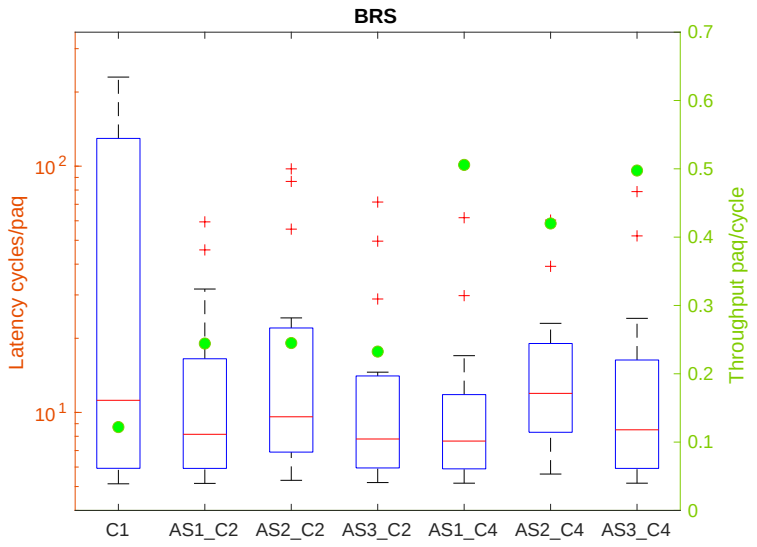


Figure 5.3: Performance of multi-channel BRS protocol for an increasing number of channels, $C1$ to $C4$, and different assignment techniques.

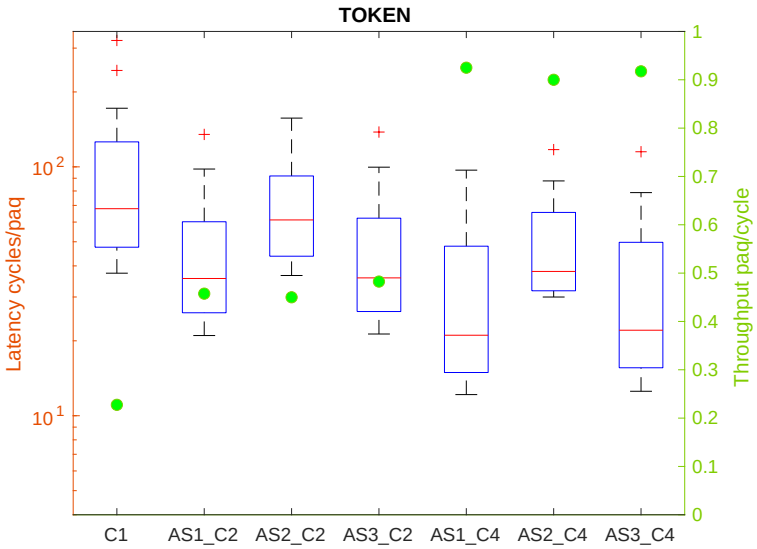


Figure 5.4: Performance of multi-channel token protocol for an increasing number of channels, $C1$ to $C4$, and different assignment techniques.

lead to channel waste and, hence, to a reduced saturation throughput. Furthermore, BRS is more irregular than token passing in terms of saturation throughput as it depends on the percentage of collisions at high loads. In fact, the difference between the saturation throughput achieved for different assignments increases with the number of channels. For example, using two channels, the difference between the limit of saturation on different assignments is negligible, whereas with four channels, assignment $AS2$ does not seem to capture subtle differences between the loads of different nodes. In the other cases, the increase of throughput is clearly proportional to the number of channels.

Latency. In general, it can be observed from the box plots that BRS is less stable than token in terms of latency as the difference between minimum and maximum values

is larger, with a higher number of outlier points. However, BRS has a much better zero-load latency than token as 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 number of collisions is less as expected even at high loads.

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. More specifically, the minimum and average latency appears to be dependent on:

1. **Number of channels:** more channels mean more tokens and/or rings, which means that one can distribute the tokens/rings evenly to reduce the waiting time between two token arrivals to a certain node. Intuitively, the minimum latency is reduced inversely proportionally to the number of channels.

2. **The assignment:** It can be observed that AS1 and AS3 have a similar zero-load latency. That is because both strategies assign channels to different rings, effectively reducing the size of each ring and, hence, the waiting time at low loads. On the other hand, AS2 does not achieve the same reduction in latency, most likely because tokens tend to accumulate instead of maintaining the same hop distance among tokens.

5.2.2 Number of Nodes

Next, we discuss the latency and throughput results for an increasing number of nodes, while fixing the number of channels to $N_c = 4$. The results are shown in Figure 5.5 and Figure 5.6 for BRS and token passing, respectively.

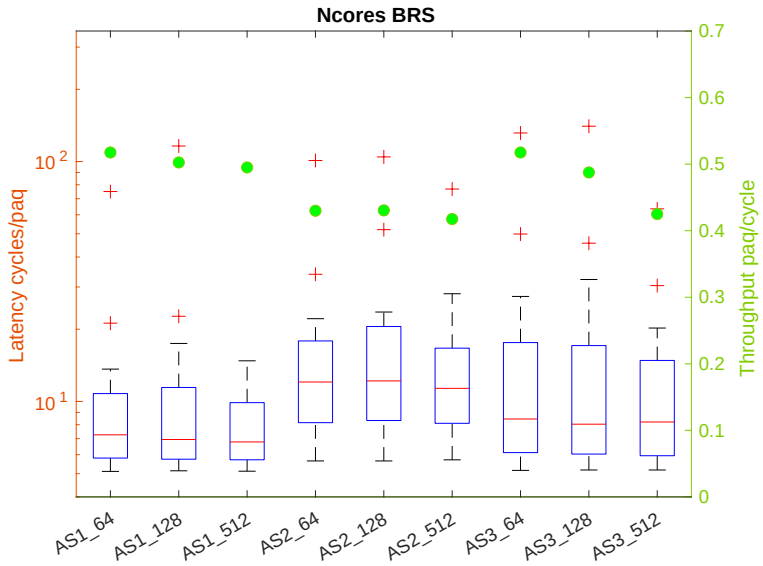


Figure 5.5: Performance of multi-channel BRS protocol for an increasing number of nodes, N=64–512, and different assignment techniques.

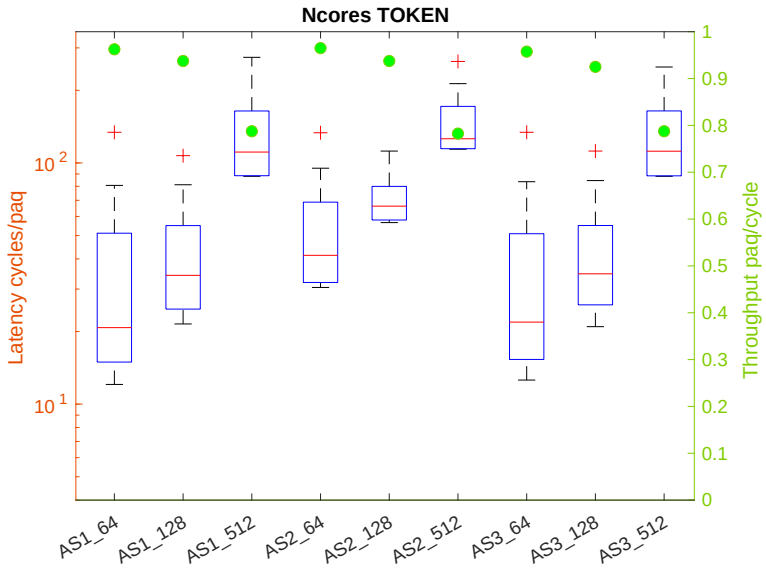


Figure 5.6: Performance of multi-channel token protocol for an increasing number of nodes, N=64–512, and different assignment techniques.

Throughput. In general, saturation throughput is higher for a lower number of nodes. Saturation can be reached either because the protocol is not able to handle a higher load or because, intrinsically, the protocol is slower. 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.

Latency. In terms of latency, BRS has a much better performance 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. Again, 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 in terms of 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 maintained across the different system sizes (note that the y-axis scale is logarithmic and the box plots tend to be compressed as they move to higher values).

5.2.3 Hotspot Traffic

In this section, we discuss the results shown in Figure 5.7 and Figure 5.8 that illustrate the behavior of BRS and token passing, respectively, and their different assignment

techniques for an increasingly hotspot traffic. We remind that a lower value of σ means that traffic is more concentrated around a smaller set of nodes.

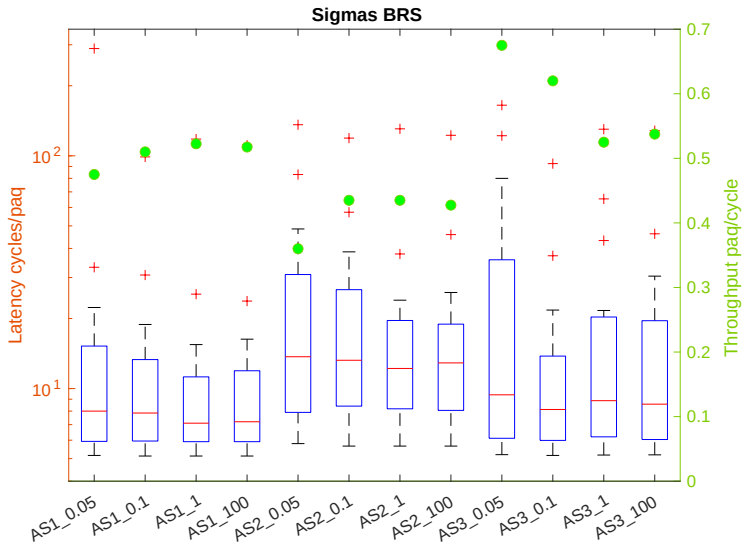


Figure 5.7: Performance of multi-channel BRS protocol for different spatial concentration levels, σ=0.1–100 (lower is more hotspot), and different assignment techniques.

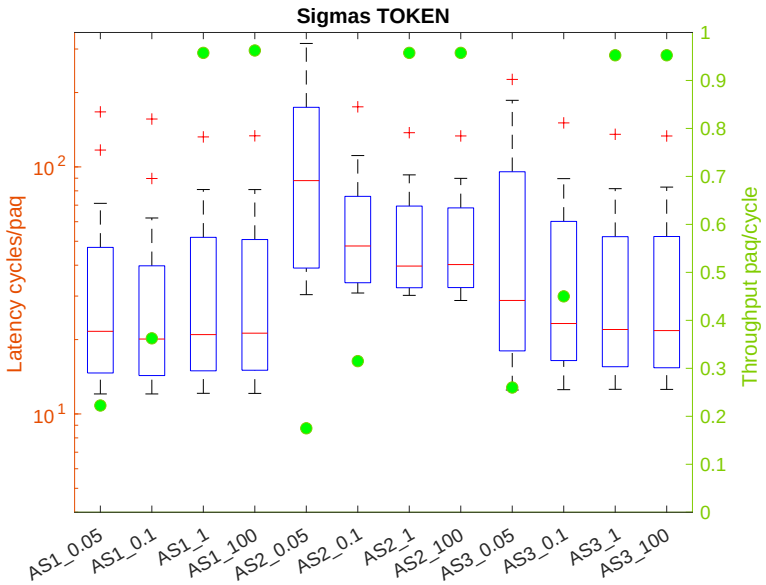


Figure 5.8: Performance of multi-channel token passing protocol for different spatial concentration levels, σ=0.1–100 (lower is more hotspot), and different assignment techniques.

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. In other words, a single node per channel is responsible for most of its load, and therefore the probability of colliding with

another node is very low. That may not happen in other assignment methods, as AS1 has all nodes contending in equal conditions, and AS2 does not distinguish between high-load and low-load nodes, assigning them a channel randomly and possibly putting several high-load nodes in the same channel.

A different behavior is observed in token passing, where the hotspot behavior of traffic clearly 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 between these nodes. In extreme cases, the throughput is reduced down to 20% of the maximum achievable throughput.

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 values follow a very similar distribution in all cases, and only the 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.

5.2.4 Bursty Traffic

Next, we discuss the latency and throughput results for an increasingly bursty traffic. The results are shown in Figure 5.9 and Figure 5.10 for BRS and token passing, respectively.

Throughput. On one hand, it can be verified that in BRS, on each assignment, the saturation throughput remains rather constant 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.

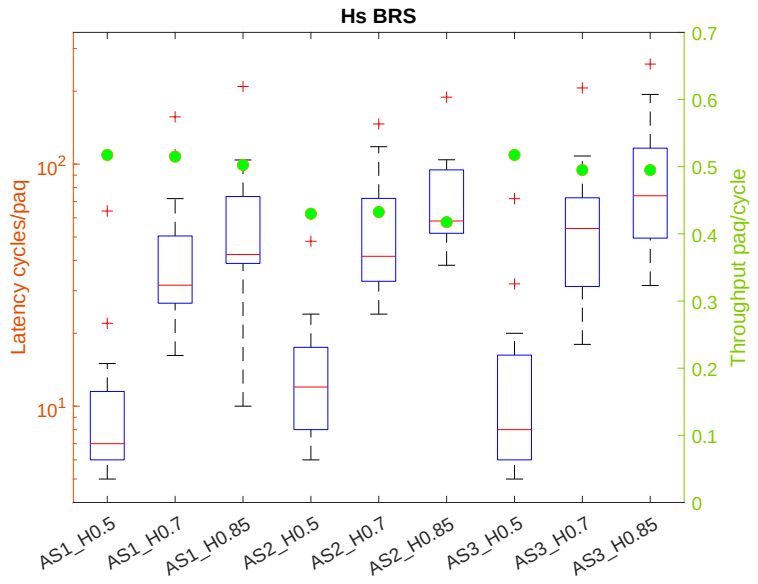


Figure 5.9: Performance of multi-channel BRS protocol for different temporal burstiness levels, H=0.5–0.85, and different assignment techniques.

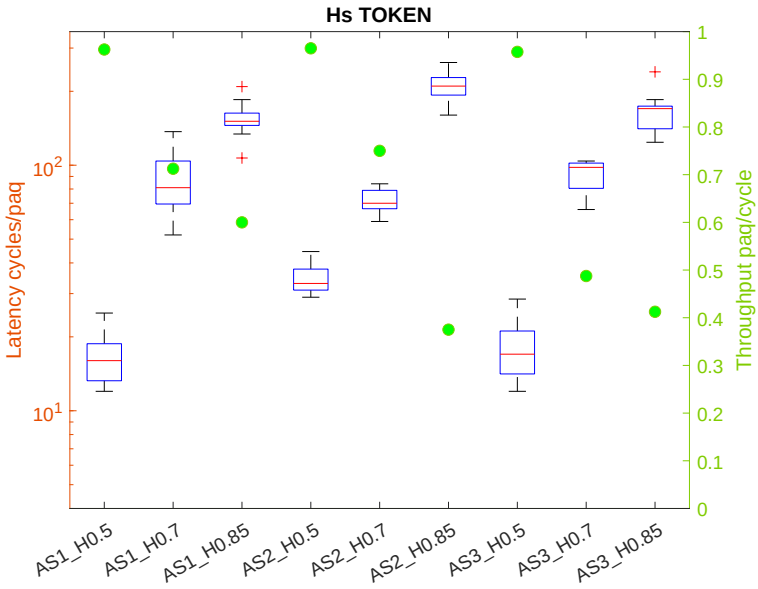


Figure 5.10: Performance of multi-channel token passing protocol for different temporal burstiness levels, H=0.5–0.85, and different assignment techniques.

Latency. In the BRS graph, 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.

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.

5.3 Discussion

Figure 5.11 summarizes the explorations made in this chapter by plotting the performance of all the compared protocols and assignments. Each point in the scatter plot represents the zero-load latency (X axis) and saturation throughput (Y axis) of a particular protocol for a given number of channels and assignment method. Hence, points in the top-left area are preferred. The simulations assume Poisson and spatially evenly distributed traffic.

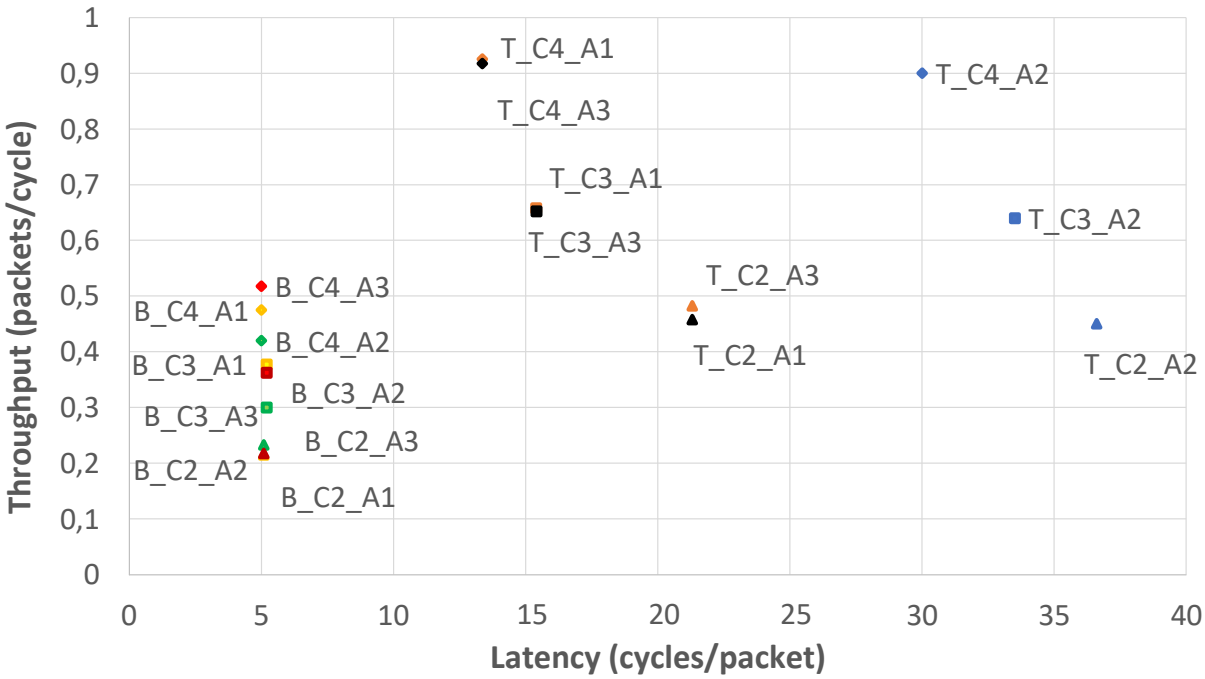


Figure 5.11: Summary of the latency-throughput results over all the protocols, assignment methods, and traffic conditions. Code is *Protocol _ Channels _ Assignment-Method* with B=BRS and T=Token.

A number of conclusions can be drawn from this plot. First, BRS is preferred over token in terms of zero-load latency regardless of the number of channels used in the protocol, given its ability to transmit immediately when the channels are idle. The throughput is inferior to that of token, but can reach reasonably high levels around 0.5 packets/cycle when four channels are employed in the assignments AS1 and AS3. AS1 would probably be preferred over AS3, since in AS1 the channel assignment is performed randomly without prior knowledge of the traffic distribution, which in contrast is required in AS3. If higher throughput is required, token passing can achieve close to double the bandwidth yet at the cost of sensibly larger low-load latency. Still, depending on the application, such latency of around 13 cycles can be acceptable. In

the case of token, however, it is harder to provide a good channel assignment: AS3 requires prior knowledge on the traffic distribution while AS1 will probably not perform well for hotspot traffic. The simpler alternative AS2, which assigns channels to tokens in a single ring (instead of channels to different rings), shows very poor performance in terms of latency.

Beyond the results of this summary shown in Figure 5.11, the rest of the chapter has unveiled results under hotspot and bursty traffic. In general, the conclusion is that BRS is more resilient to challenging 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. Like in the case of *fuzzy token* demonstrated in [44] for single-channel networks, it would be desirable to develop a multi-channel protocol that is able to seamlessly obtain the best of both worlds: the resilience and low latency of BRS with the throughput and fairness of token passing.

6. Network Layer

Several of the multi-channel MAC protocols that have been evaluated in the previous chapter consider a channel assignment method that can be either completely random, hoping it will balance the load across the channels in an adaptive way, or use some information from the upper layers to balance the load across channels in a more proactive way. Similarly, we have found that there is not a clear winner between CSMA-like and token passing protocols because that depends on the amount of nodes that might be contending for the channel, or the pressure applied to the link. In fact, the goal of some of our recent works such as *Fuzzy Token* has been to try to achieve the best of both types of protocols without any knowledge of the traffic via a set of simple rules. However, these works assume that there is only one channel and that all nodes listen to all transmissions, which may not hold in multi-channel configurations.

This chapter, rather than striving to present a complete solution, will try to further motivate the need for the network-architecture to drive the layers below using results from the architectural explorations made within the project. In particular, here we present a summary of results of the EPFL partner published in ASP-DAC 2023 [79] where multi-chiplet systems with wireless links are evaluated. The system configurations and applications considered are those outlined in Section 2.4. In this case, applications where profiled using gem5-X, with custom extensions emulating wireless links and protocols, as described in Deliverable D5.3 [80].

Figure 6.1 shows the speedup of the proposed multi-chiplet system with four clusters of cores, each of which is placed within a chiplet, for multiple legacy and AI workloads and assuming different wireless channel bandwidths and protocols. Similarly, Figure 6.2 repeats the analysis with a subset of the AI workloads. From these results, we can extract some conclusions:

- Not all applications achieve an appreciable speedup with respect to a wired chiplet interconnect, which are in this case modeled after commercial versions of the AMD EPYC processor.
- The best access control protocol depends on the wireless channel bandwidth and on the particular application being executed.
- The break-even point between wired and wireless connectivity very much depends on the application being executed.

Let us now assume that wireless interfaces hold the necessary circuitry to apply token passing and CSMA-like protocols, and further assume that large chiplet-based systems could be executing multiple applications simultaneously while sharing multiple wireless channels. Based on the results above, then, a pertinent question to be solved at the network-architecture frontier would be *how should the wireless resources be mapped to different applications?*.

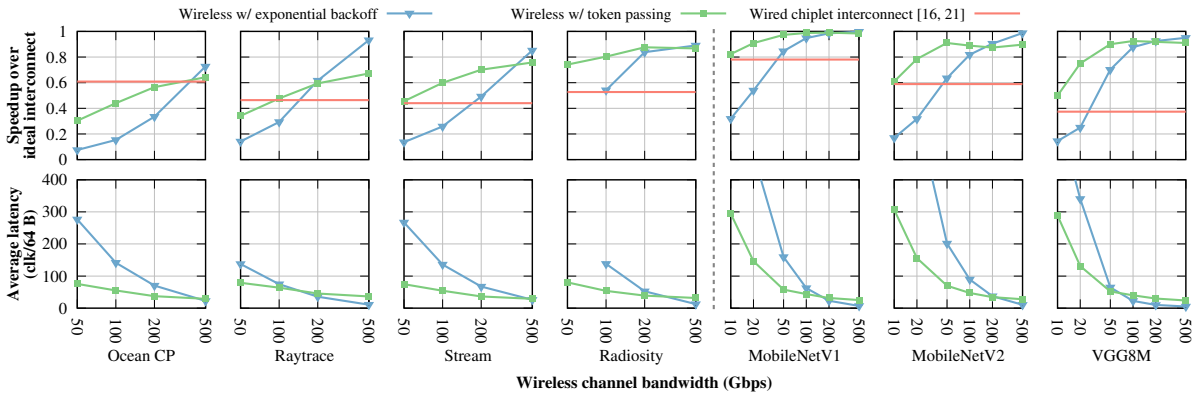


Figure 6.1: Runtime speed-up over ideal interconnect (top) and average inter-chiplet link latency (bottom) of different chiplet interconnects for representative workloads executed on a 4-cluster system. The dashed line divides the applications of the communication-intensive set (left) and the CNN workloads (right).

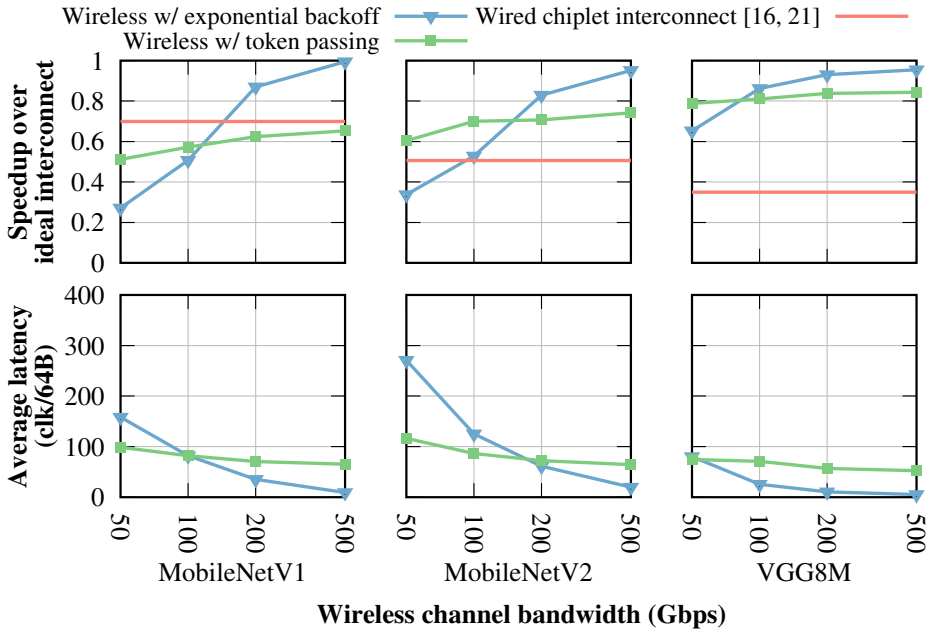


Figure 6.2: Runtime speed-up over ideal interconnect (top) and average inter-chiplet link latency (bottom) of different chiplet interconnects for representative CNN workloads executed on a 16-cluster system.

Our cross-layer vision could allow a central controller with system-level privileges to impact the network, link, and even physical layers so that the performance of the system can be maximized. Based on either hints provided by the compiler or offline analysis of applications, and given a set of workloads to be executed concurrently, the controller could perform a *resource mapping* procedure deciding, for instance, whether an application should make use of the wireless network or not, e.g. based on Figure 6.1 above the controller could conclude that using the wireless network is inadvisable. Once this decision is made, then the controller could decide the protocol to use and the number of frequency channels to assign to it. We plan to explore this option in future work.

7. Concluding Remarks

This deliverable has focused on the development of a protocol stack for wireless in-package communications that could potentially adapt to the requirements of modern multi-chiplet computing systems. To this end, we have taken base on the fundamentals of Deliverables D3.1 and D3.2, which provided models for the wireless channel as well as the performance and cost of certain choices in the protocol stack, and have taken them one step further. In particular, we posit that wireless in-package networks could exploit several spatial and frequency channels thanks to the unique tunability delivered by the graphene-based antennas being actively researched in the project. Consequently, the protocol stack needs expose these channels to the architecture and manage their access.

Building a protocol stack that can generalize to multiple channels has been the main aim of this deliverable. We have first conducted, in Chapter 3, electromagnetic simulations that prove that multiple frequency and space channels are possible in a flip-chip. Due to the lossy nature of the silicon layer, a flip-chip package can support multiple broadband channels between 60 GHz and 240 GHz, with a variation of path loss of less than 10 dB among them. Furthermore, we have also shown that two or more (three in a 10×10 mm^2 area) spatial channels can co-exist in parallel directions, thanks to the use of compact antenna arrays at 60 GHz and, again, to the lossy nature of silicon. In our extended work [67], we show that the same array can also create spatial channels at 110 GHz. Therefore, we have indirectly demonstrated the existence of six channels (three spatial ones at two different frequencies). We also argue that in larger packages, more channels could be created.

In subsequent chapters, we have discussed how the different layers could be extended to support multiple channels in a way compatible with the resource constraints and performance requisites of the wireless in-package networking scenario. At the physical layer, we first described the elements of the RF-chain that need to be tuned (in frequency, gain, or mode of operation) to implement a multi-channel scheme, and then compared various possible implementations considering either multiple fixed RF chains or a single tunable one. The former delivers maximum flexibility and allows to use the multiple channels for reducing the latency of both the transmission itself and the MAC protocol, yet at the expense of very large area ($4\times$ with four channels) and energy overheads (double the energy per bit). However, when the network is large, the reduction of the MAC delay takes precedence –an option that does not require multiple RF chains. This argument is substantiated through the proposal and thorough evaluation of several multi-channel versions of CSMA-like and token passing protocols. With four channels, throughput can be practically multiplied by four and, in the case of token, the zero-load latency can be cut significantly. Finally, at the network layer, we analyzed traffic traces coming from the architectures and AI workloads that WiPLASH aims to implement, and also discussed recent work by the EPFL partner describing

the speedups obtained in such workloads when wireless links are used. Those results have led to the conclusion that the best wireless network configuration depends on the available bandwidth and the application that is being run, further supporting the idea that a sort of controller could overview the operation of the network and make modifications in an adaptive protocol stack based on the applications that are executed in the system.

The deliverable also leaves some open questions and aspects that shall be investigated in future work. For instance, the lossy nature of silicon allows to create multiple broadband channels, but at the expense of a non-negligible path loss that impacts on the efficiency of wireless links. Hence, a natural question arises on whether a similar multi-channel support can be assumed in other packages where losses could be minimized, but where a much more reverberant behavior can be expected. In such packages, perhaps custom-made on top of the conventional flip-chip stack, delay spread is a huge issue and spatial multiplexing is very challenging. One may need to resort to techniques such as the use of programmable metasurfaces [81] or time-reversal [82] to address this issue. A second aspect to consider is that, while the use of multiple channels alleviates most of the issues of BRS and token passing, there is still a window of opportunity for protocols that combine both techniques. Hence, in future work, we aim to create a multi-channel version of fuzzy token [44], which could multiply the throughput while conserving the simplicity, low-latency, and resilience of the original protocol. Last but not least, some aspects remain unclear as we climb through the protocol stack. In particular, future work will need to address (i) the need for a lightweight scheduling policy that ensures that receivers are listening to the right channel at the right instant, and (ii) the possibility, hinted in this deliverable, of having a system-level controller that, through the analysis of the architecture and applications being executed, determines the best configuration in terms of number of channels, MAC protocol, or scheduling policy.

Bibliography

[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. Abadal, R. Guirado, H. Taghvaee, A. Jain, E. P. de Santana, P. Haring Bolívar, M. Saeed, R. Negra, Z. Wang, K.-T. Wang, *et al.*, "Graphene-based wireless agile interconnects for massive heterogeneous multi-chip processors," *arXiv preprint arXiv:2011.04107*, 2020.

[4] J. Kim, K. Choi, and G. Loh, "Exploiting new interconnect technologies in on-chip communication," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 124–136, 2012.

[5] 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.

[6] 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. 112200–15, 2019.

[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] R. G. Kim, W. Choi, Z. Chen, P. P. Pande, D. Marculescu, and R. Marculescu, "Wireless NoC and Dynamic VFI Codesign: Energy Efficiency Without Performance Penalty," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 7, pp. 2488–2501, 2016.

[9] M. A. I. Sikder, A. Kodi, W. Rayess, D. Ditomaso, D. Matolak, and S. Kaya, "Exploring wireless technology for off-chip memory access," in *Proceedings of the HOTI '16*, pp. 92–99, 2016.

[10] 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*, pp. 3–17, 2016.

[11] 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.

[12] V. Fernando, A. Franques, S. Abadal, S. Misailovic, and J. Torrellas, "Replica: A Wireless Manycore for Communication-Intensive and Approximate Data," in *Proceedings of the ASPLOS '19*, pp. 849–863, 2019.

[13] D. DiTomaso, A. Kodi, S. Kaya, and D. Matolak, "iWISE: Inter-router wireless scalable express channels for network-on-chips (NoCs) architecture," in *Proc. of IEEE 19th Annu. Symp. High Perform. Interconnects*, pp. 11–18, 2011.

[14] D. DiTomaso, A. Kodi, D. Matolak, S. Kaya, S. Laha, and W. Rayess, "A-WiNoC: Adaptive Wireless Network-on-Chip Architecture for Chip Multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3289–3302, 2015.

[15] 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 (JETC)*, vol. 10, no. 4, pp. 1–20, 2014.

[16] S. Liu, T. F. Canan, H. Chenji, S. Laha, S. Kaya, and A. Karanth, "Exploiting wireless technology for energy-efficient accelerators with multiple dataflows and precision," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2022.

[17] H. Mondal, S. Gade, M. Shamim, S. Deb, and A. Ganguly, "Interference-Aware Wireless Network-on-Chip Architecture using Directional Antennas," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 3, pp. 193–205, 2017.

[18] 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.

[19] R. S. Narde, J. Venkataraman, A. Ganguly, and I. Puchades, "Antenna Arrays as Millimeter-Wave Wireless Interconnects in Multichip Systems," *IEEE Antennas and Wireless Propagation Letters*, vol. 19, no. 11, pp. 1973–1977, 2020.

[20] WiPLASH consortium, "Wireless Channel Modeling," in *European Commission, H2020-FETOPEN, Project WiPLASH: Accepted Public Deliverable D3.1, 17-Jan-2021*, 2021.

[21] "CST Microwave Studio."

[22] K. Kimoto, N. Sasaki, S. Kubota, W. Moriyama, and T. Kikkawa, "High-Gain On-Chip Antennas for LSI Intra- / Inter-Chip Wireless Interconnection," *Proceedings of the EuCAP '09*, pp. 278–282, 2009.

[23] O. Markish, B. Sheinman, O. Katz, D. Corcos, and D. Elad, "On-chip mmWave Antennas and Transceivers," in *Proceedings of the NoCS '15*, p. Art. 11, 2015.

[24] X. Timoneda, S. Abadal, A. Cabellos-Aparicio, D. Manessis, J. Zhou, A. Franques, J. Torrellas, and E. Alarcón, "Millimeter-Wave Propagation within a Computer Chip Package," in *Proceedings of the ISCAS '18*, 2018.

[25] X. Timoneda, A. Cabellos-Aparicio, D. Manessis, 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.

[26] S. Abadal, A. Mestres, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "Medium access control in wireless network-on-chip: A context analysis," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 172–178, 2018.

[27] WiPLASH consortium, "Wireless Communication Performance and Cost Models," in *European Commission, H2020-FETOPEN, Project WiPLASH: Submitted Public Deliverable D3.2, 02-Oct-2022*, 2022.

[28] F. T. Chen, J. M. Wu, and M. C. F. Chang, "40-Gb/s 0.7-V 2:1 MUX and 1:2 DEMUX with Transformer-Coupled Technique for SerDes Interface," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 4, pp. 1042–1051, 2015.

[29] S. Saxena, G. Shu, R. K. Nandwana, M. Talegaonkar, A. Elkholy, T. Anand, W.-S. Choi, and P. K. Hanumolu, "A 2.8 mw/gb/s, 14 gb/s serial link transceiver," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 5, pp. 1399–1411, 2017.

[30] A. A. S. SH, K. S. Reddy, *et al.*, "A 20 gb/s latency optimized serdes transmitter for data centre applications," in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp. 1–4, IEEE, 2020.

[31] B. Murmann, "ADC Performance Survey 1997-2021," 2021.

[32] H. Wang, T.-Y. Huang, N. Sasikanth, *et al.*, "Power Amplifiers Performance Survey 2000-2021," 2021.

[33] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee, and V. Srinivas, "Cacti 7: New tools for interconnect exploration in innovative off-chip memories," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 14, no. 2, pp. 1–25, 2017.

[34] 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.

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

[36] 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.

[37] 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*, p. 217, 2009.

[38] N. Mansoor, S. Shamim, and A. Ganguly, "A Demand-Aware Predictive Dynamic Bandwidth Allocation Mechanism for Wireless Network-on-Chip," in *Proceedings of the SLIP '16*, 2016.

[39] S. Abadal, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "OrthoNoC: A Broadcast-Oriented Dual-Plane Wireless Network-on-Chip Architecture," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 3, pp. 628–641, 2018.

[40] 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. 406–11, 2015.

[41] 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.

[42] N. Mansoor and A. Ganguly, "Reconfigurable Wireless Network-on-Chip with a Dynamic Medium Access Mechanism," in *Proceedings of the NoCS '15*, p. Article 13, 2015.

[43] N. Mansoor, A. Vashist, M. M. Ahmed, M. S. Shamim, S. A. Mamun, and A. Ganguly, "A traffic-aware medium access control mechanism for energy-efficient wireless network-on-chip architectures," *arXiv preprint arXiv:1809.07862*, 2018.

[44] 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.

[45] 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.

[46] 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.

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

[48] Y. M. Qureshi, W. A. Simon, M. Zapater, D. Atienza, and K. Olcoz, "Gem5-x: A gem5-based system level simulation framework to optimize many-core platforms," in *2019 Spring Simulation Conference (SpringSim)*, pp. 1–12, IEEE, 2019.

[49] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The splash-2 programs: Characterization and methodological considerations," *ACM SIGARCH computer architecture news*, vol. 23, no. 2, pp. 24–36, 1995.

[50] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[51] A. Krizhevsky *et al.*, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, 2017.

[52] K. Chatfield *et al.*, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," 2014.

[53] N. Bruschi, G. Haugou, G. Tagliavini, F. Conti, L. Benini, and D. Rossi, "Gvsoc: A highly configurable, fast and accurate full-platform simulator for risc-v based iot processors," in *2021 IEEE 39th International Conference on Computer Design (ICCD)*, pp. 409–416, 2021.

[54] N. Bruschi, G. Tagliavini, F. Conti, S. Abadal, A. Cabellos-Aparicio, E. Alarcón, G. Karunaratne, I. Boybat, L. Benini, and D. Rossi, "Scale up your in-memory accelerator: Leveraging wireless-on-chip communication for aimc-based cnn inference," 2022.

[55] "Distributed Neural Network Training in Pytorch."

[56] D. Culler, J. P. Singh, and A. Gupta, *Parallel computer architecture: a hardware/software approach*. Gulf Professional Publishing, 1999.

[57] S. Mittal, "A survey of techniques for approximate computing," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, pp. 1–33, 2016.

[58] 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.

[59] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Improving Energy Efficiency in Wireless Network-on-Chip Architectures," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 14, no. 1, p. Art. 9, 2018.

[60] X. Yu, J. Baylon, P. Wettin, D. Heo, P. P. Pande, and S. Mirabbasi, "Architecture and design of multichannel millimeter-wave wireless NoC," *IEEE Design & Test*, vol. 31, no. 6, pp. 19–28, 2014.

[61] D. Matolak, S. Kaya, and A. Kodi, "Channel modeling for wireless networks-on-chips," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 180–186, 2013.

[62] X. Timoneda, S. Abadal, A. Franques, D. Manessis, J. Zhou, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "Engineer the Channel and Adapt to it: Enabling Wireless Intra-Chip Communication," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3247–3258, 2020.

[63] Y. Huang, L. Wu, M. Tang, and J. Mao, "Design of a beam reconfigurable THz antenna with graphene-based switchable high-impedance surface," *IEEE Transactions on Nanotechnology*, vol. 11, no. 4, pp. 836–842, 2012.

[64] J. M. Jornet and I. F. Akyildiz, "Graphene-based plasmonic nano-antenna for terahertz band communication in nanonetworks," *IEEE Journal on selected areas in communications*, vol. 31, no. 12, pp. 685–694, 2013. U.S. Patent No. 9,643,841, May 9, 2017 (Priority Date: Dec. 6, 2013).

[65] P. A. D. Gonçalves and N. M. Peres, *An introduction to graphene plasmonics*. World Scientific, 2016.

[66] A. Singh, M. Andrello, N. Thawdar, and J. M. Jornet, "Design and operation of a graphene-based plasmonic nano-antenna array for communication in the terahertz band," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 9, pp. 2104–2117, 2020.

[67] F. Rodríguez-Galán, E. Pereira de Santana, P. Haring Bolívar, S. Abadal, and E. Alarcón, "Towards spatial multiplexing in wireless networks within computing packages," in *Proceedings of the Ninth Annual ACM International Conference on Nanoscale Computing and Communication*, pp. 1–6, 2022.

[68] S. Abadal, R. Martínez, J. Solé-Pareta, E. Alarcón, and A. Cabellos-Aparicio, "Characterization and Modeling of Multicast Communication in Cache-Coherent Manycore Processors," *Computers and Electrical Engineering (Elsevier)*, vol. 51, no. April, pp. 168–183, 2016.

[69] T. Sherwood, E. Perelman, G. Hamerly, S. Sair, and B. Calder, "Discovering and Exploiting Program Phases," *IEEE Micro*, vol. 23, no. 6, pp. 84–93, 2003.

[70] J. M. Jornet and I. F. Akyildiz, "Graphene-based Plasmonic Nano-antenna for Terahertz Band Communication in Nanonetworks," *IEEE JSAC, Special Issue on Emerging Technologies for Communications*, vol. 31, no. 12, pp. 685 – 694, 2013.

[71] E. Castaneda, A. Silva, A. Gameiro, and M. Kountouris, "An overview on resource allocation techniques for multi-user mimo systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 239–284, 2016.

[72] Y. Mehta, S. Thomas, and A. Babakhani, "A 140–220-ghz low-noise amplifier with 6-db minimum noise figure and 80-ghz bandwidth in 130-nm sige bicmos," *IEEE Microwave and Wireless Components Letters*, pp. 1–4, 2022.

[73] C. Yi, D. Kim, S. Solanki, J. H. Kwon, M. Kim, S. Jeon, Y. C. Ko, and I. Lee, "Design and Performance Analysis of THz Wireless Communication Systems for Chip-to-Chip and Personal Area Networks Applications," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 6, pp. 1785–1796, 2021.

[74] S. Laha, S. Kaya, D. W. Matolak, W. Rayess, D. DiTomaso, and A. Kodi, "A New Frontier in Ultralow Power Wireless Links: Network-on-Chip and Chip-to-Chip Interconnects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 186–198, 2015.

[75] M. H. Eissa, A. Malignaggi, R. Wang, M. Elkhouly, K. Schmalz, A. C. Ulusoy, and D. Kissinger, "Wideband 240-GHz Transmitter and Receiver in BiCMOS Technology With 25-Gbit / s Data Rate," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 9, pp. 2532–2542, 2018.

[76] S. E. Hosseininejad, S. Abadal, M. Neshat, R. Faraji-Dana, M. C. Lemme, C. Suessmeier, P. H. Bolívar, E. Alarcón, and A. Cabellos-Aparicio, "Mac-oriented programmable terahertz phy via graphene-based yagi-uda antennas," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2018.

[77] D. Rodrigo, J. Romeu, B. A. Cetiner, and L. Jofre, "Pixel Reconfigurable Antennas: Towards Low-Complexity Full Reconfiguration," in *Proceedings of the EuCAP '16*, pp. 2–6, 2016.

[78] WiPLASH consortium, "Graphene integration design report," in *European Commission, H2020-FETOPEN, Project WiPLASH: Submitted Public Deliverable D1.2, 31-Dec-2021*, 2021.

[79] R. Medina, J. Klein, G. Ansaloni, M. Zapater, S. Abadal, E. Alarcón, and D. Atienza, "System-level exploration of in-package wireless communication for multi-chiplet platforms," in *2023 28th Asia and South Pacific Design Automation Conference (ASP-DAC)*, IEEE, 2023.

[80] WiPLASH consortium, "Package-level exploration," in *European Commission, H2020-FETOPEN, Project WiPLASH: Submitted Public Deliverable D5.3, 31-Mar-2022*, 2022.

[81] M. F. Imani, S. Abadal, and P. Del Hougne, "Metasurface-programmable wireless network-on-chip," *Advanced Science*, vol. 9, no. 26, p. 2201458, 2022.

[82] Y. Chen, Y.-H. Yang, F. Han, and K. R. Liu, "Time-reversal wideband communications," *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1219–1222, 2013.