

**ADVERTIMENT.** La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX ([www.tesisenxarxa.net](http://www.tesisenxarxa.net)) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

**ADVERTENCIA.** La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR ([www.tesisenred.net](http://www.tesisenred.net)) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

**WARNING.** On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX ([www.tesisenxarxa.net](http://www.tesisenxarxa.net)) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author



Departament d'Arquitectura  
de Computadors  
UNIVERSITAT POLITÈCNICA DE CATALUNYA



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

# Broadcast-Oriented Wireless Network-on-Chip: Fundamentals and Feasibility

*A thesis submitted for the degree of*

**Doctor of Philosophy in Computer Architecture**

by

**Sergi Abadal Cavallé**

*Advisors:*

Dr. Albert Cabellos Aparicio

Dr. Eduard Alarcón Cot

June 2016

Nanonetworking Center in Catalunya (N3Cat)  
Departament d'Arquitectura de Computadors  
Universitat Politècnica de Catalunya



## Acknowledgments

This thesis is the culmination of almost five years of work. During this time, I have had the pleasure of living, working, or simply interacting with lots of people; some of these people have played a significant part in my life as they have helped me to grow both technically and personally. I would like to dedicate a few lines to them.

First and foremost, I would like to truly thank my advisors Dr. Albert Cabellos and Dr. Eduard Alarcón for their dedication and support. I genuinely admire their combination of ambition, pragmatism, brilliance, rigor, and optimism. It has been a pleasure to learn from them as we developed the research that has led to this dissertation. I sincerely feel honored to have had the chance to work under their guidance and inspiration, and I hope that this will be the first of many ventures together.

I would also like to take the chance to express my gratitude to my first mentor, Prof. Ian Akyildiz. I have very fond memories of my stay with him and his group in Georgia Tech back in 2009, which opened my eyes to the world of research. The continued support of Prof. Akyildiz during this unique experience and beyond has been one of the main reasons that encouraged me to start my PhD. I am also extremely grateful to Prof. Josep Solé i Pareta, which invited me to join N3Cat just after completing my bachelor's degree and, ever since, has had nothing but kind words for me and the rest of the group.

Although my determination was strong, this PhD would have not been possible without financial support. I acknowledge N3Cat and the Department of Computer Architecture for awarding me with a doctoral grant, as well as for providing generous support for my multiple trips to conferences around the world. I am also extremely thankful to INTEL for honoring me with a fellowship, which provided an extra thrust to develop my investigations. I would like to extend my gratitude to Prof. Antonio González of INTEL Labs for his advice and support since then.

During the five years of PhD, I have had the luck to meet and work with outstanding researchers. Their experience and excellent qualities have been a great inspiration and have helped me to bring this thesis to the next level. I would like to thank Dr. Mario Nemirovsky for the essential discussions on multiprocessor architecture and the hours devoted to shape the whole vision, Prof. José Antonio Lázaro for our discussions on photonic on-chip networks, and Dr. Raúl Martínez for his mentorship and help with traffic characterization issues. My gratitude also goes to the PhD external reviewers and members of the different PhD committees, namely, Dr. Josep Miquel Jornet, Dr. Maurizio Palesi, Dr. Jordi Domingo, Dr. José Núñez, Dr. Davide Careglio, and Dr. Peter Haring Bolívar, for their time and feedback.

Here, I wanted to reserve a special spot to Prof. Josep Torrellas. I was very lucky to be hosted by Josep for a research stay in his group at University of Illinois at Urbana-Champaign in 2015. His honesty, tireless dedication, and vast knowledge in computer architecture are truly inspiring. Also, I felt at home while working and exchanging lots of fruitful discussions with him and his group. I would like to especially thank Antonio Franquès, Tom Shull, and Ronak Buch for their friendship and for turning that stay into an unforgettable experience.

Research can be tough at times, but it is always fun and engaging when you are surrounded by good colleagues. It is hard to find the words to express my gratitude towards Ignacio Llatser, Albert Mestres, Raúl G. Cid-Fuentes, and Mario Iannazzo for the hours

spent together working hard. I am also grateful to them and to Florin Coras and Alberto Rodríguez for the good times we had discussing about research or simply exchanging bad jokes. Special mention is deserved to our lab manager Albert López not only for being part of this group, but also for his ability to make our work –and by extension, our lives– easier.

Last, but not least by any means, I would like to extend my deepest appreciation to the ones that have unconditionally been by my side, cheering me up and walking with me throughout this journey: my parents, my brother Kini, my precious dog Uma, and my wonderful significant other Marta. I love you.

## Abstract

Recent years have seen the emergence and ubiquitous adoption of Chip Multiprocessors (CMPs), which rely on the coordinated operation of multiple execution units or cores. Successive CMP generations integrate a larger number of cores seeking higher performance with a reasonable cost envelope. For this trend to continue, however, important scalability issues need to be solved at different levels of design. Scaling the interconnect fabric is a grand challenge by itself, as new Network-on-Chip (NoC) proposals need to overcome the performance hurdles found when dealing with the increasingly variable and heterogeneous communication demands of manycore processors. Fast and flexible NoC solutions are needed to prevent communication become a performance bottleneck, situation that would severely limit the design space at the architectural level and eventually lead to the use of software frameworks that are slow, inefficient, or less programmable.

The emergence of novel interconnect technologies has opened the door to a plethora of new NoCs promising greater scalability and architectural flexibility. In particular, wireless on-chip communication has garnered considerable attention due to its inherent broadcast capabilities, low latency, and system-level simplicity. Most of the resulting Wireless Network-on-Chip (WNoC) proposals have set the focus on leveraging the latency advantage of this paradigm by creating multiple wireless channels to interconnect far-apart cores. This strategy is effective as the complement of wired NoCs at moderate scales, but is likely to be overshadowed at larger scales by technologies such as nanophotonics unless bandwidth is unrealistically improved.

This dissertation presents the concept of Broadcast-Oriented Wireless Network-on-Chip (BoWNoC), a new approach that attempts to foster the inherent simplicity, flexibility, and broadcast capabilities of the wireless technology by integrating one on-chip antenna and transceiver per processor core. This paradigm is part of a broader hybrid vision where the BoWNoC serves latency-critical and broadcast traffic, tightly coupled to a wired plane oriented to large flows of data. By virtue of its scalable broadcast support, BoWNoC may become the key enabler of a wealth of unconventional hardware architectures and algorithmic approaches, eventually leading to a significant improvement of the performance, energy efficiency, scalability and programmability of manycore chips.

The present work aims not only to lay the fundamentals of the BoWNoC paradigm, but also to demonstrate its viability from the electronic implementation, network design, and multiprocessor architecture perspectives. An exploration at the physical level of design validates the feasibility of the approach at millimeter-wave bands in the short term, and then suggests the use of graphene-based antennas in the terahertz band in the long term. At the link level, this thesis provides an insightful context analysis that is used, afterwards, to drive the design of a lightweight medium access control protocol that reliably serves broadcast traffic with substantial latency improvements over state-of-the-art NoCs. At the network level, our hybrid vision is evaluated putting emphasis on the flexibility provided at the network interface level, showing outstanding speedups for a wide set of traffic patterns. At the architecture level, the potential impact of the BoWNoC paradigm on the design of manycore chips is not only qualitatively discussed in general, but also quantitatively assessed in a particular architecture for fast synchronization. Results demonstrate that the impact of BoWNoC can go beyond simply improving the network performance, thereby representing a possible game changer in the manycore era.



# Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Motivation . . . . .	3
1.3 Vision: Broadcast-oriented Wireless Network-on-Chip . . . . .	7
1.4 Scope and Contributions of the Thesis . . . . .	8
<b>2 Broadcast in Emerging Interconnect Paradigms</b>	<b>10</b>
2.1 Advanced Network-on-Chip Design . . . . .	11
2.2 3D Integration . . . . .	13
2.3 RF Interconnects . . . . .	15
2.4 Nanophotonics . . . . .	18
<b>3 Wireless On-Chip Communication and Networking</b>	<b>22</b>
3.1 Enablers of the Wireless RF Approach . . . . .	24
3.1.1 Scaling CMOS Technology . . . . .	24
3.1.2 Graphene Technology . . . . .	26
3.1.3 Surface Wave Technology . . . . .	27
3.2 Wireless On-Chip Communication Patterns . . . . .	28
3.3 Design Principles: Up in the Protocol Stack . . . . .	29
3.4 Existing Wireless Network-on-Chip (WNoC) Architectures . . . . .	35
<b>4 PHY: Towards Core-Level Wireless Communication</b>	<b>37</b>
4.1 Design Principles, Objectives and Challenges . . . . .	38
4.1.1 On-chip Channel Modeling . . . . .	39
4.1.2 Modulations . . . . .	41
4.1.3 Coding . . . . .	42
4.2 Scalability Analysis of mmWave Transceivers . . . . .	43
4.2.1 Evaluation Framework . . . . .	44
4.2.2 Performance Evaluation . . . . .	52
4.2.3 Discussion . . . . .	55
4.3 On the Suitability of Graphene-enabled THz Wireless Communication . . . . .	59
4.3.1 Background on Graphene-based Miniaturized Antennas . . . . .	60
4.3.2 Background on Terahertz Propagation . . . . .	63
4.3.3 Evaluation Framework . . . . .	65



4.3.4	Design Space Exploration . . . . .	74
<b>5</b>	<b>MAC: Seeking Reliable and Scalable Broadcast Communication</b>	<b>84</b>
5.1	Design Principles, Objectives and Challenges . . . . .	85
5.2	Context Analysis . . . . .	86
5.2.1	The Chip Scenario . . . . .	87
5.2.2	Workload Characteristics . . . . .	89
5.2.3	Architectural Requirements . . . . .	94
5.3	BRS: A MAC Protocol for Reliable Broadcast in WNoC . . . . .	95
5.3.1	Protocol Description . . . . .	96
5.3.2	Performance Analysis . . . . .	100
5.3.3	Comparative Scalability Exploration . . . . .	106
5.4	Optimization of Existing Protocols . . . . .	114
<b>6</b>	<b>NET: Exploring the Hybrid Wired-Wireless Design Space</b>	<b>120</b>
6.1	Motivation . . . . .	121
6.2	OrthoNoC: A Dual-Plane Wired-Wireless Network Architecture . . . . .	123
6.2.1	Design Decisions . . . . .	124
6.2.2	Evaluation Framework . . . . .	127
6.2.3	Performance Evaluation . . . . .	128
<b>7</b>	<b>ARCH: Broadcast-Enabled Massive Multicore Architectures</b>	<b>133</b>
7.1	Potential Impact of BoWNoC on Future Manycores . . . . .	133
7.1.1	Software Architecture and Algorithms . . . . .	135
7.1.2	Coherence Mechanisms . . . . .	136
7.1.3	Synchronization and Control . . . . .	136
7.1.4	Programming Models . . . . .	137
7.1.5	Novel Computing Systems . . . . .	137
7.2	WiSync: An Architecture for Fast On-Chip Synchronization . . . . .	138
7.2.1	Overview of WiSync . . . . .	139
7.2.2	Design Decisions and Implementation Issues . . . . .	140
7.2.3	Performance Evaluation . . . . .	147
<b>8</b>	<b>Conclusion</b>	<b>155</b>
8.1	Lessons Learned . . . . .	155
8.2	Future Avenues of Research . . . . .	156
<b>A</b>	<b>Synthetic Traffic Generation</b>	<b>158</b>
<b>B</b>	<b>Scientific Production</b>	<b>164</b>
	<b>Bibliography</b>	<b>168</b>

# List of Figures

1.1	Representation of the cache coherence design space. . . . .	3
1.2	Performance of a 2D mesh as a function of the system size. . . . .	4
1.3	Multicast traffic as a function of the number of cores assuming a tiled architecture with private 32-kB L1-D/L1-I caches, 512-kB of shared L2 per core and three coherence protocols. Results are the geometric mean of all the SPLASH-2 and PARSEC benchmarks. . . . .	5
1.4	Methods to serve multicast messages in packet-switched NoCs. . . . .	6
1.5	Performance of a 8×8 mesh with tree-based multicast support as a function of the broadcast intensity. . . . .	6
1.6	Schematic representation of the vision of this dissertation: a hybrid architecture combining a Broadcast-oriented Wireless Network-on-Chip (BoWNoC) for broadcast flows and a throughput-oriented and energy-efficient NoC for the rest of the traffic. . . . .	7
1.7	Roadmap towards the vision. The highlighted region represents the scope of the thesis. . . . .	9
2.1	Future evolution of RC interconnect performance according to the 2013 International Technology Roadmap for Semiconductors (ITRS). . . . .	11
2.2	High connectivity topologies for low latency NoC. . . . .	13
2.3	Sketch of a stacked processor setting with on-chip links and vertical vias. . . . .	14
2.4	Sketch of a frequency-multiplexing scheme using transmission lines or optical waveguides. A similar approach can be applied to perform code multiplexing. To perform time multiplexing, cores simply need to transmit in different time slots. . . . .	16
2.5	Sketch of a NoC overlaid with transmission lines. . . . .	17
2.6	Basic operation principle of ring resonators. . . . .	19
2.7	Sketch of a photonic NoC with snake-shaped waveguides. . . . .	20
3.1	Inductor scalability according to data from the literature. . . . .	23
3.2	Basic WNoC scheme and implementable communication patterns. . . . .	28
3.3	Custom layered representation of the different design levels for BoWNoC. . . . .	30
3.4	Theoretical BER as a function of the SNR for different well-known modulations. . . . .	32
3.5	Coarse classification of multiple access mechanisms. . . . .	34
3.6	Rough classification of the topologies used in existing WNoC architectures. White and black squares represent tiles with routers and wireless interfaces, respectively. . . . .	36

4.1	Channel modeling in the BoWNoC scenario: electromagnetic waves that may potentially reach the receiver. . . . .	40
4.2	Model-based framework for physical layer design space exploration. . . . .	44
4.3	Area and energy of state-of-the-art wireless transceivers as a function of their data rate. . . . .	45
4.4	Maturity factor as a function of the operation frequency of state-of-the-art transceiver proposals. . . . .	46
4.5	Area of state-of-the-art wireless transceivers as a function of the frequency. . . . .	48
4.6	Energy efficiency figure of merit of state-of-the-art wireless transceivers as a function of their central frequency. . . . .	50
4.7	Area scaling for different interconnect technologies and architectures as functions of (a) the number of cores with $C = 80\text{Gbps}$ , and (b) the link capacity with $N = 256$ . . . . .	54
4.8	Energy per bit scaling for different interconnect technologies and architectures as functions of (a) the number of cores with $C = 80\text{Gbps}$ , and (b) the link capacity with $N = 256$ . . . . .	55
4.9	Scaling of the proposed figure of merit (higher is better) for different interconnect technologies and architectures as functions of (a) the number of cores with $C = 80\text{Gbps}$ , and (b) the link capacity with $N = 256$ . . . . .	56
4.10	Schematic representation of a set of graphene-based antennas proposed in the literature. Silicon lenses are used in (c) and (d) but not shown for simplicity. . . . .	60
4.11	Frequency response at $r = 1\text{m}$ of a $5\mu\text{m} \times 1\mu\text{m}$ freestanding graphene patch fed with a $10\text{-k}\Omega$ microstrip for different chemical potential and relaxation time values. The voltage inside the antenna is $1\text{V}$ for all frequencies. We refer the reader to Section 4.3.3 for further methodological details. . . . .	63
4.12	Molecular absorption of the terahertz channel at a distance of $1\text{cm}$ . . . . .	64
4.13	End-to-end channel model of a graphene-enabled wireless link in the time domain. . . . .	66
4.14	Vertical methodology for the time-domain characterization of graphene-enabled wireless links as functions of nanoscale phenomena. . . . .	68
4.15	Envelope of the analytic response of two graphennas with different chemical potential and carrier mobility. . . . .	76
4.16	Impulse Response matrix of $5\mu\text{m} \times 1\mu\text{m}$ graphennas for a design space exploration with different carrier mobility and chemical potential values. The time interval ranges from $0$ to $10$ picoseconds. . . . .	78
4.17	Response amplitude metrics (left) and response length characteristics (right) as functions of the carrier mobility and chemical potential. . . . .	79
4.18	(a) Molecular absorption in dB for transmission distances of $1\text{cm}$ (blue solid line) and $10\text{cm}$ (red dashed line). The top blue (bottom red) background shows the frequency region which determines the available bandwidth for a transmission distance of $1\text{cm}$ ( $10\text{cm}$ ). (b) Available bandwidth in the frequency band up to $50\text{THz}$ . . . . .	80
4.19	Semi-log plots of the different characteristics of the channel impulse response due to molecular absorption. . . . .	81
4.20	Response peak and RMS delay spread as functions of the chemical potential of the antenna and the amount of water vapor in the channel for a distance of $3\text{cm}$ (top) and $1\text{m}$ (bottom). $\mu=4\text{m}^2\text{V}^{-1}\text{s}^{-1}$ . . . . .	83

5.1	Main facets that define the MAC design process in the WNoC scenario. . .	87
5.2	Number of multicast flits per $10^6$ instructions for MESI, HT and TokenB protocols as a function of the processor size (logarithmic scale). Lower charts represent the percentage of multicast flits before and after replication for each respective case. . . . .	90
5.3	Phase behavior exhibited by the traffic generated by the <i>fluidanimate</i> application during 2B cycles. Communication-intensive and computation-intensive phase are interspersed. . . . .	91
5.4	Spatiotemporal characterization of the injection distribution of multicasts (geometric mean). . . . .	92
5.5	Factor of predictability (blue bars, left axis) and degree of correlation (red lines, right axis) with $\tau = 50 \cdot T_{CLK}$ . . . . .	93
5.6	Flowchart of the BRS-MAC protocol for the transmitter (left) and the receiver (right). Transitions due to collisions are made explicit with red labels.	97
5.7	Operation of BRS-MAC both for a collision and a successful transmission. .	101
5.8	Simulated and analytical throughput for worst-case and exact propagation times. . . . .	104
5.9	Throughput and delay characteristics as functions of the propagation time.	105
5.10	Throughput and delay as functions of the position of the preamble. . . . .	105
5.11	Delay as a function of the throughput for different backoff values. . . . .	106
5.12	Latency-throughput curves for representative system sizes for 100% of broadcast traffic and a wireless capacity of one flit per cycle. . . . .	109
5.13	Low-load latency of broadcast transmissions as a function of the number of nodes $N$ for $C = 1$ . . . . .	110
5.14	Throughput of broadcast transmissions at the maximum admissible latency (150 cycles) as a function of the number of nodes $N$ for $C = 1$ . . . . .	111
5.15	Low-load latency as a function of the percentage of broadcast traffic $\beta$ for $N = 256$ and $C = 1$ . . . . .	112
5.16	Throughput at the maximum admissible latency (150 cycles) as a function of the percentage of broadcast traffic $\beta$ for $N = 256$ and $C = 1$ . . . . .	113
5.17	Throughput for different spatial injection distributions, from spread out ( $\sigma = 100$ ) to extremely hotspot ( $\sigma = 0.1$ ), with $N = 64$ , $\beta = 100\%$ and $C = 1$ . . .	114
5.18	Throughput for different temporal injection distributions, from exponential ( $H = 0.5$ ) to extremely bursty ( $H = 0.85$ ), with $N = 64$ , $\beta = 100\%$ and $C = 1$ .	114
5.19	Multicast source prediction scheme, with the detail of a static predictor (left box) and a last value predictor (right box). . . . .	117
5.20	Geometric mean of the prediction accuracy in SPLASH2 and PARSEC for the SP and LVP assuming a 50-cycle observation window. Labels indicate the coverage in LVP (we assume 100% coverage in the static case). Higher is better. . . . .	118
5.21	Multihop ring for token passing. . . . .	118
5.22	Performance of token passing with and without multi-hop capabilities. . . .	119
6.1	Latency speedup of a hybrid NoC with respect to MESH-FT as a function of the broadcast percentage for different system sizes and channel capacities. .	121
6.2	Throughput speedup of a hybrid NoC with respect to MESH-FT as a function of the broadcast percentage for different system sizes and channel capacities.	122
6.3	Schematic diagram of the OrthoNoC architecture. . . . .	124

6.4	Controller methods for load balancing. . . . .	126
6.5	Manhattan distance statistics for Rent traffic in a 8×8 mesh. . . . .	128
6.6	Network performance speedups of the broadcast policy of OrthoNoC for different admission probabilities and percentages of broadcast traffic. . . . .	129
6.7	Network performance speedups of the global policy of OrthoNoC for different distance thresholds and traffic profiles. . . . .	130
6.8	Scatter plot of the performance improvements of the two policies of OrthoNoC for different system sizes, wireless plane capacities, and input traffic characteristics. . . . .	132
7.1	Representation of the many-core design space opened by broadcast between the shared memory and the message passing paradigms. . . . .	134
7.2	Graphical representation of inter-process communication in a DCG iterative solver over time. Each row represents a thread. . . . .	135
7.3	WiSync architecture. The different colors represent different programs running. . . . .	139
7.4	How global writes appear in WiSync. The jagged line means that the non-local BMs are physically far. . . . .	140
7.5	Transmission channels in WiSync. . . . .	141
7.6	Sharing the Tone Channel among several barriers. . . . .	142
7.7	B-memory address translation. . . . .	144
7.8	Examples of code used for synchronization. . . . .	146
7.9	CAS throughput of three kernels on different architectural configurations for several critical section sizes and 64 or 128 cores. Higher is better. . . . .	149
7.10	Execution time of TightLoop on different architectural configurations. Note that the Y-axis is <i>logarithmic</i> . . . . .	150
7.11	Execution time of the Livermore loops on different architectural configurations for several vector sizes and 64 or 128 cores. . . . .	151
7.12	Speedup of the SPLASH-2 and PARSEC applications on different architectural configurations for 64 cores. . . . .	153
7.13	Geometric mean of the execution speedup for the different evaluation profiles. . . . .	154
7.14	Speedup of WiSync for different wireless channel capacities. . . . .	154
A.1	Coefficient of determination ( $R^2$ ) of the spatial injection distribution, including the gaussian fitting in two relevant cases. . . . .	160
A.2	Statistical analysis of the multicast destinations in MESI (perfect tracking). . . . .	161
A.3	Measured Hurst exponent (top) and load (bottom) as functions of the input load for $H = \{0.53, 0.7, 0.9\}$ . Dashed and solid lines represent the theoretical value and geometric mean of the measured values, respectively. . . . .	162

# List of Tables

2.1	Comparative summary of the emerging interconnect technologies for NoC. . .	12
3.1	Characteristics of wireless scenarios as transmission ranges are scaled down.	24
4.1	Summary of the specifications of the analyzed transceivers . . . . .	43
4.2	NoC Parameters . . . . .	53
4.3	Dominant area and energy scalability trends . . . . .	57
4.4	Per-Tile Area and Power Comparison . . . . .	58
4.5	Parameters for graphene-enabled wireless links design space exploration. . .	75
4.6	Performance comparison of different metallic and graphene-based antennas.	77
5.1	Wireless Manycore Scenario Requirements . . . . .	86
5.2	Differences between traditional wireless networks and the WNoC scenario in terms of MAC protocol design. . . . .	88
5.3	Notation for BRS-MAC performance analysis. . . . .	100
5.4	Simulation Parameters . . . . .	107
6.1	Simulation Parameters . . . . .	127
6.2	Synthetic traffic profiles . . . . .	128
6.3	Explored Controller Policies . . . . .	129
7.1	Architecture modeled. RT means round trip. . . . .	148
7.2	Architecture configurations compared. . . . .	148
7.3	Kernels and applications executed. . . . .	149
7.4	Lock (L) and barrier (B) calls in SPLASH-2 and PARSEC for 64 cores. . .	152
7.5	Utilization of the Data Channel in WiSyncNoT and WiSync in % of the total cycles for the most demanding applications. . . . .	153
7.6	Memory and network configuration variants. . . . .	154



# Chapter 1

## Introduction

The relentless march of technology scaling has forced a widespread transition in processor design from *single core* to *multicore* due to power and complexity reasons. The adoption of multicore processors is now ubiquitous, extending its influence from supercomputers and high-end servers to commodity computers, embedded systems, and handheld devices. In successive generations, processors have been steadily increasing in the number of integrated cores to continue with current upward trends in terms of performance and efficiency. We are thus at the *manycore era*.

Such scaling tendency has profound implications in a wide variety of design aspects, in ways that sharply depend on the considered computing scenario. The core microarchitecture, the memory hierarchy, the interconnect, or the software itself, are some examples of the facets impacted by the increase in core density suggested by both academic and industrial experts. This thesis is ambitious in breadth, yet necessarily narrows down its influence to aspects at the frontier between the interconnect and the multiprocessor architecture. In particular, the research presented here delves into the scalability issues of current interconnects, analyzes the architectural restrictions imposed by these issues, and then proposes a novel interconnect solution that aims to overcome them.

To put the reader in context, this chapter first provides essential background on multiprocessor architecture and on the importance of on-chip communication. In Section 1.2, the challenges that on-chip communication faces at the time of this writing are outlined, setting the focus on the outstanding issue of broadcast communication in the manycore era. This is the main motivation of the dissertation, a challenge that could be addressed with the research vision outlined in Section 1.3. The core of this vision is the proposal of the Broadcast-oriented Wireless Network-on-Chip (BoWNoC) paradigm, briefly explained in Section 1.4 along with the specific contributions made at the different levels of design.

### 1.1 Background

Parallelization has been the natural trend in microprocessor architecture design for the last decades and it is expected to continue in the near future. Parallelism can be found and exploited at different granularities, from the instruction level to the thread level. Instruction-level parallelism, which takes advantage of the potential overlap among simple instructions, has been used since the dawn of microprocessors. However, fundamental limits led to diminishing returns in its exploitation and finally caused power consumption in uniprocessors to



grow way faster than its performance [1,2]. This power bottleneck suggested the exploration of higher levels of parallelism.

Multiprocessor architectures have recently emerged aiming to keep pace with the performance trends predicted by Moore's Law while maintaining an acceptable power envelope. Parallelism is exploited either by simultaneously executing instances of independent applications or by dividing the application in a set of tasks and processing them in a collaborative manner. To do so, multiprocessors consist of the interconnection of a given number of independent *processor cores* and a memory system within a single die.

The on-chip interconnect is a central element of a multicore processor since it implements the communication between cores and, regardless of the programming model of choice, has a profound impact on performance. In message passing systems, communication is explicit and needs to be carefully orchestrated by the programmer [3]. In shared memory, which will be the programming model considered by default throughout this dissertation, communication between cores occurs implicitly as a result of reads and writes to the cache hierarchy. More specifically, communication is a consequence of the need to maintain the on-chip caches coherent. Since multiple copies of the same shared data may be distributed in a plurality of caches, different cores may be seeing different values if this data is modified locally. Cache coherence methods prevent this by ensuring that a read to a shared variable always returns the last written value [1,2]. Communication between caches is required to enforce this.

Given the direct relation between memory architecture, communication, and overall performance, the research focus in multiprocessors has gradually shifted from how cores compute to how cores communicate. Buses were first widely considered for the implementation of the on-chip interconnect, but their use is restricted to small-scale architectures given the limited scalability of buses beyond a few cores [4]. Network-on-Chip (NoC), or the application of networking theory and methods to on-chip communication, has been subsequently adopted as the paradigm of choice for on-chip interconnection networks [5,6]. In NoCs, routers and point-to-point links are arranged forming a certain topology. The network is packet-switched, which means that each core turns request and response messages into packets that will need to traverse one or multiple routers towards the destination. With an appropriate design of aspects such as the routing protocol or the flow control mechanisms, NoCs offer substantial improvements in the fault tolerance, the modularity and, most importantly, the scalability properties of the on-chip interconnect [7].

The paradigm shift in interconnect design has occurred in parallel to a paradigm shift in terms of architecture. Snooping protocols are the more common cache coherence mechanism in bus-based processors due to its simplicity and effectiveness. However, snooping protocols generally require support for global communication and a certain consistency in the order of message delivery, demands that are costly to guarantee in denser processors with modern NoCs. In response to this, coherence schemes relying on directories that keep track of the location and sharing status of memory blocks have been employed instead. The directory acts as ordering point, allowing its use in unordered networks. Unfortunately, maintaining the directory incurs certain area and power overheads, increases the architectural complexity, and lowers performance due to indirection.

As shown in Figure 1.1, inspired by the work in [8], snooping and full-map directory mechanisms represent the two extremes of the cache coherence design space, which trade off architectural cost against interconnect cost. These two solutions are not suitable for manycore processors; therefore, the coherence design space needs to be explored in depth as future chips are expected to integrate a thousand cores [9].

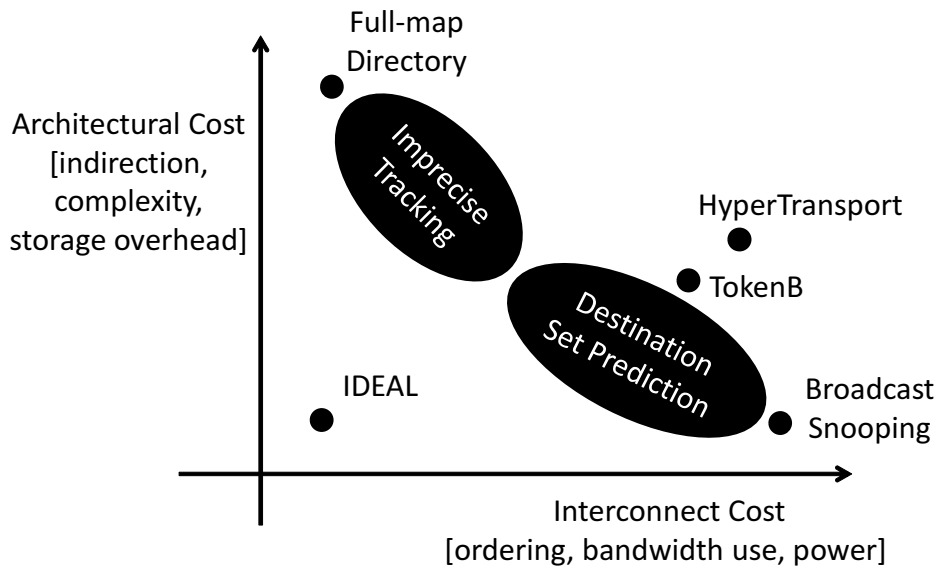


Figure 1.1: Representation of the cache coherence design space.

## 1.2 Motivation

### On-Chip Communication Challenges in the Manycore Era

Scaling parallel applications generally implies an increase in terms of the communication-to-computation ratio given that the computation is distributed among a larger number of cores [10, 11]. This implies not only a higher amount of communication, but also a higher traffic heterogeneity given by the wider selection in terms of possible destinations. As we discuss briefly in the following paragraphs, and in depth in Chapter 5, this is translated into a higher importance of the global and multicast communication transactions. Another consequence of scaling applications and architectures is that traffic becomes more dynamic as multiprogramming, i.e. mapping several applications within different subsets of cores, is likely to be leveraged more frequently [12].

The increase in core density also implies that the increment in the intensity, heterogeneity, and variability of the load needs to be absorbed with higher energy efficiency and placing a stronger emphasis on latency. Energy consumption is important because while the power envelope keeps rather constant as the number of cores grows [13], the traffic intensity does not; whereas latency is critical because the communication delay in operations that are in the critical path of the processor will directly impact upon the execution speed.

Within this context, it remains unclear whether conventional NoCs alone will be able to meet the increasingly stringent requirements of next-generation multiprocessors. One of the main issues is the growing number of routers that a packet needs to traverse in average, which impacts on the time and energy required to move one bit across the die. To partially quantify this, Figure 1.2 shows how the performance of the widely considered mesh topology scales with the system size for uniform random traffic. The latency at low loads increases sharply, whereas the per-core throughput suffers a severe drop in spite of the much higher bisection bandwidth of the scaled topology. To solve these issues, one could use alternative topologies with lower distance between processors as suggested in Section 2.1; however, there are different performance-cost tradeoffs behind this decision, and it has

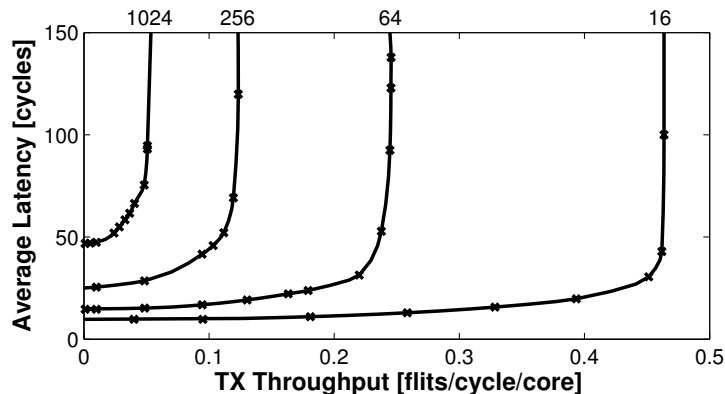


Figure 1.2: Performance of a 2D mesh as a function of the system size.

been demonstrated that there is no topology that can fit to all situations [14]. Manycore processors, thus, call for either architectural approaches that enforce the *locality* of traffic or flexible solutions at the on-chip interconnection network level.

### The Case of Multicast and Broadcast

Thus far, we have taken the case of traffic *locality* to briefly explain the main challenges that future on-chip networks need to address. A similar reasoning can be performed for multicast and broadcast, but with an added particularity: these types of communication are much more detrimental to performance of conventional NoCs than global traffic. We subsequently discuss the contrast between the increasing presence of collective communication in manycore architectures and the reduction of performance that the underlying NoCs suffer. Although we assume cache-coherent processors here, the analysis could be extended to message passing or other programming models.

Scaling parallel architectures and applications increases their multicast requirements basically because data is shared among larger groups of cores [10, 11]. Therefore, transactions managing the sharing of data between groups of cores become more frequent and involve a larger destination set. To exemplify this trend, Figure 1.3(a) shows how the ratio of multicast transactions per instruction scales with the number of cores in processors with three different cache coherence protocols. A consistent growth of the traffic intensity is observed in protocols that actively avoid multicast traffic (MESI), as well as in broadcast-intensive alternatives, i.e. HyperTransport (HT, [15]) and TokenB [16]. In the latter cases, multicast transactions actually become accountable for a potentially huge percentage of all the traffic. Figures 1.3(b) and 1.3(c) exemplify this by showing the percentage of multicast flits that are injected to and ejected from the NoC in the HT and TokenB schemes. Note that in TokenB, assuming 64 cores, 25% of the injected traffic is broadcast and generates 80% of the flits served by the NoC.

There are different ways to serve multicast messages in conventional NoCs, as depicted in Figure 1.4. The most simple mechanism, generally known as *unicast-based*, consists in breaking down the multicast packet into as many replicas as the number of destinations and serving each replica independently. Intuitively, this method is highly inefficient when the destination set is large [17]. Consequently, techniques used in multicomputer networks such as the *path-based* or *tree-based* multicast have been proposed instead. In the former case, several copies are sent to separate chip partitions. Each copy is in turn replicated when

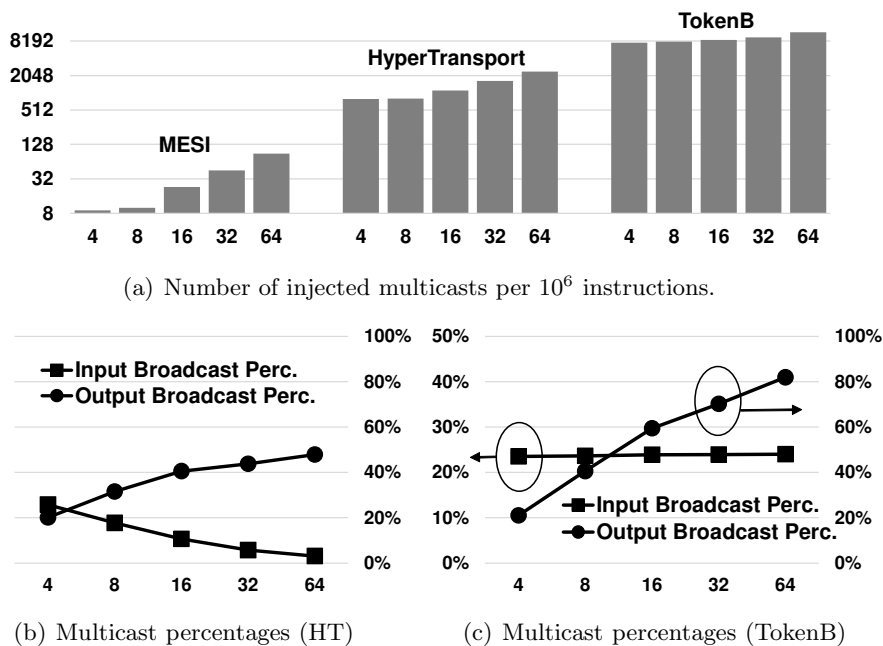


Figure 1.3: Multicast traffic as a function of the number of cores assuming a tiled architecture with private 32-kB L1-D/L1-I caches, 512-kB of shared L2 per core and three coherence protocols. Results are the geometric mean of all the SPLASH-2 and PARSEC benchmarks.

reaching each destination within its partition: the copy is delivered and the original flit is forwarded to the next destination through a determined path [18]. In the latter case, the source injects a single message, which is replicated at intermediate routers and delivered to the intended destinations following a spanning tree [19].

Despite recent research efforts [17, 20, 21], the multicast performance of NoCs does not scale well for two reasons. As mentioned above, the sheer addition of more cores causes the average logical distance between processors to increase. This affects the latency of broadcasts and dense multicasts, which always need to reach distant destinations. On top of this, message replication within the network increases contention and aggravates the performance degradation when the load increases. These two issues are clearly observed in Figure 1.5, which plots the latency-throughput characteristic of a 64-core mesh with tree-based multicast as a function of the percentage of broadcast traffic. Even considering an aggressive design with fast routers, the low-load latency increases substantially and the throughput drops as the core density increases. Here, it is worth noting that the latency would further increase if path-based or unicast-based multicast methods are considered.

As we will see in Section 2.1, there have been several proposals trying to bridge the existing gap between multicast requirements and performance in conventional NoCs. Some works have achieved outstanding improvements, but there are two issues that still remain unclear. First, how will these designs be scaled so that the latency is kept low without incurring into unaffordable overheads? Second, how can broadcast bursts be efficiently served without affecting unicast traffic?

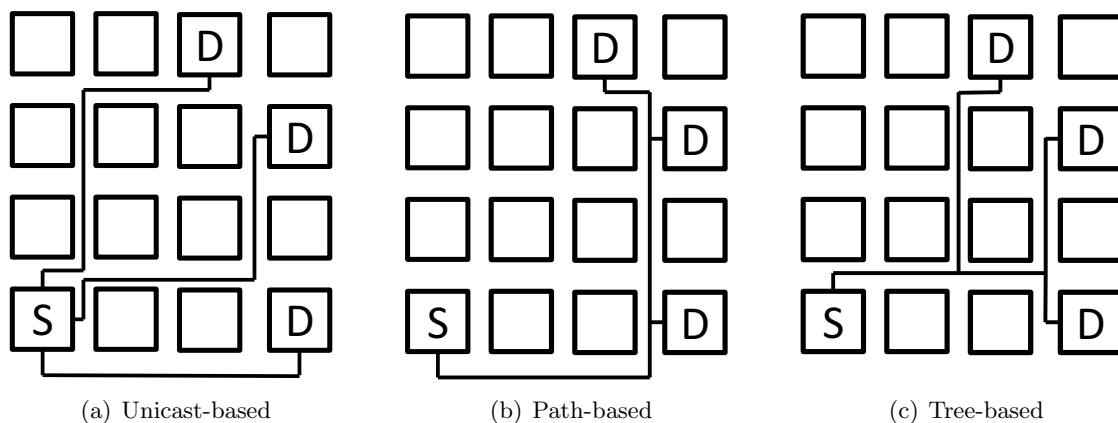


Figure 1.4: Methods to serve multicast messages in packet-switched NoCs.

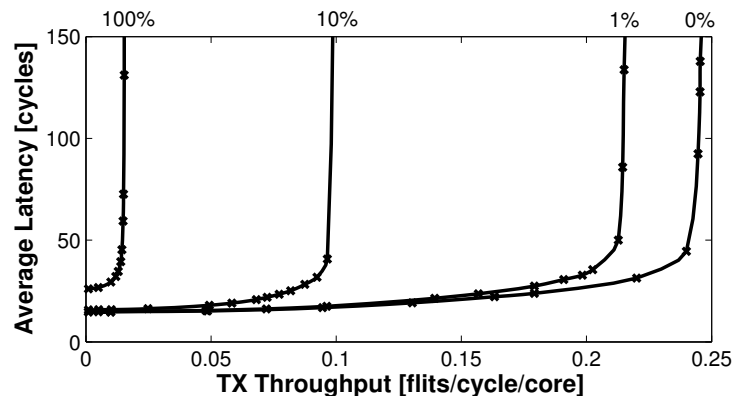


Figure 1.5: Performance of a  $8 \times 8$  mesh with tree-based multicast support as a function of the broadcast intensity.

## Impact of Multicast Performance on the Manycore Architecture

The scaling limitations of conventional NoCs force manycore architects to minimize the use of methods that generate global and multicast traffic. This has two direct consequences.

On the one hand, the poor multicast support causes the performance of current architectures to drop significantly as they are scaled. Some works have shown that limited broadcast support can lead to a remarkable slowdown of the execution speed of a cache-coherent multiprocessor [20]. Functions that are inherently global or collective, e.g. synchronization, become expensive by default and can degrade performance of benchmarks that use it. Liang *et al* have demonstrated that poorly implementing synchronization can reduce performance by approximately 40% in widely used benchmarks despite representing a small fraction of the code [22]. In message passing systems, algorithms that make frequent use of collective communication patterns will see their performance significantly reduced when scaled.

On the other hand, the active avoidance of multicast traffic increases the complexity of architectures and applications. To avoid slowing down coherence, architects have to resort to hefty directory coherence protocols or latency hiding optimizations that are often hard to debug and verify. In accordance with Figure 1.1, these methodologies push architectures towards a suboptimal solution space close to the full-map directory corner. The use

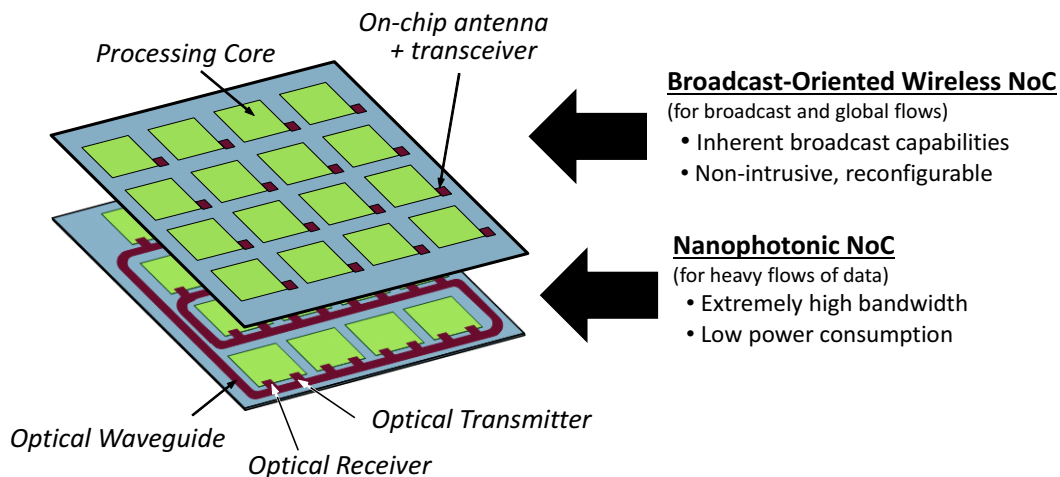


Figure 1.6: Schematic representation of the vision of this dissertation: a hybrid architecture combining a BoWNoC for broadcast flows and a throughput-oriented and energy-efficient NoC for the rest of the traffic.

of techniques such as imprecise tracking [23] is strongly limited, whereas simpler architectures employing token coherence [16] or snooping protocols for unordered networks [24, 25] are completely prohibited. These constraints also affect message passing systems, where programmers are encouraged to minimize the use of collective communication primitives, complicating the task of parallel programming even further.

All these evidences suggest that an effective broadcast plane may be not only valuable, but even necessary, to avoid slowing down progress in the manycore era. The potential benefits of such approach would be substantial since, first, the performance of a wide set of architectures and programming models would be greatly improved. Additionally, the constraints relative to the use of broadcast would be relaxed, thus significantly reducing the cost of operations based on broadcast. This would enable the adoption of architectural and algorithmic techniques overlooked at the manycore scale, opening a large set of unexplored possibilities.

### 1.3 Vision: Broadcast-oriented Wireless Network-on-Chip

A plethora of different research lines have emerged as a response to the wide range of challenges that have arisen in the manycore era. In most cases and as summarized in Chapter 2, the use of novel interconnect technologies has been envisaged as the main path towards overcoming the issues of conventional NoC designs at the technological, microarchitecture and network architecture levels. Yet, at the time of this writing, a review of such proposals reveals a notable lack of effective and scalable multicast communication mechanisms [26].

The vision stated on this thesis originates from this deficit in terms of broadcast communication support within the chip context, and is motivated by the potential consequences at the architectural level. The vision is graphically represented in Figure 1.6 as the combination of a flexible and truly scalable broadcast platform, and a more conventional throughput-oriented network plane. *We aim to provide flexibility, versatility, and inherent broadcast capabilities where the architecture shows variability, heterogeneity, and demands broadcast support.*

The core of this hybrid architecture represents the main contribution of this thesis: the concept of BoWNoC. Unlike in most WNoC architectures [26], the main design objective of this novel paradigm is to provide outstanding broadcast capabilities even in manycore scenarios. This is ideally achieved by integrating one wireless communication unit per processing tile and by allowing all processors to share a small set of broadband channels. With these design guidelines, broadcast communication can be performed in a few processor cycles regardless of the number of cores, leading to significant network performance improvements even with modest bandwidths. Moreover, the wireless channel naturally enforces certain ordering rules, which is highly desirable from the –often overlooked– architecture perspective. Since no path infrastructure is required between cores, this approach is non-intrusive and flexible from a system-level standpoint. For all this, the BoWNoC embodies one of the network planes of our hybrid architecture, which would be leveraged for the transmission of global and broadcast traffic.

Overlaying a BoWNoC over a conventional NoC not only would provide specialized support for multicasts, but also would offload the main NoC and thus increase performance and efficiency for the rest of communication flows. Unlike the existing hybrid proposals that combine wired and wireless technologies, our vision implements two independent network planes. Interaction between planes is not performed at the routers in each hop, but rather at the network interfaces, which implies that messages will not bounce back and forth between planes. The network interface includes a hybrid controller that performs traffic steering as a function of a policy that can be easily reconfigured at runtime. These decisions contribute to make the network architecture simpler, more flexible, and easier to integrate in architectures that might need the ordering properties of BoWNoC to implement certain functions.

The vision is ambitious and extends beyond the horizon of the next generation of processors. To provide enough bandwidth and energy efficiency even in thousand-core architectures [9], we envisage the use of graphene-based RF components in the wireless plane and of nanophotonic technologies in the wired plane. Yet still, this dissertation has more immediate objectives as we will see next.

## 1.4 Scope and Contributions of the Thesis

The main objective of this thesis is twofold. First, we aim to set the foundations of the novel BoWNoC approach by analyzing how its unique traits impact on a wide set of design aspects. Second, we aim to prove that BoWNoC is feasible by studying the theoretical scalability of most of the actors involved in this paradigm, seeking to eventually provide a starting point of design from which different investigations can evolve.

Figure 1.7 shows a tentative roadmap describing this research area from different design perspectives and stretching the temporal horizon towards the long term. The highlighted region within this figure represents the scope of this dissertation which, albeit prospective, remains temporally close to the state of the art in the field. As mentioned above, the objectives of this work are in consonance with this temporal frame and do not point towards a complete and optimized design. Instead, we aim to validate the feasibility of the approach and showcase its future potential. The roadmap figure also graphically represents the conceptual breadth of this dissertation, which cuts through different levels of abstraction. We made an effort to provide the fundamentals and feasibility from most of the perspectives involved in the field of on-chip communication: electronics, communications, networking, and computer architecture.

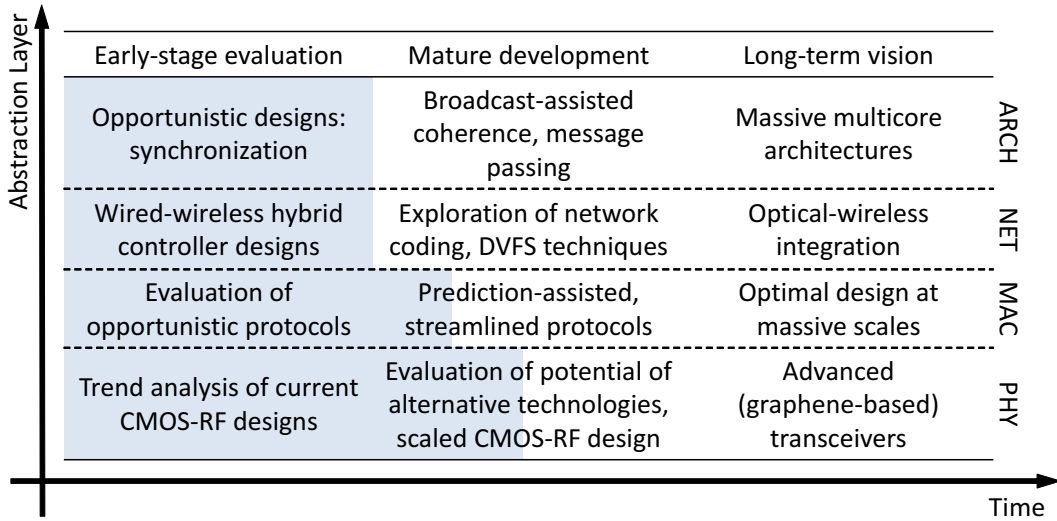


Figure 1.7: Roadmap towards the vision. The highlighted region represents the scope of the thesis.

At the physical layer, this work studies the main uniquenesses of the BoWNoC scenario in terms of antenna and transceiver design, channel modeling, coding, and modulations. Then, the feasibility of the BoWNoC paradigm is investigated from the implementation perspective by analyzing how RF designs scale with frequency and by assessing the applicability of graphene technology within our vision. At the link layer, medium access control mechanisms are key for the correct operation of a BoWNoC. This work provides a thorough context analysis, from which a set of design guidelines is extracted. A lightweight and scalable medium access control protocol is proposed, modeled, and compared with a set of representative implementations. At the network layer, the interplay between the wired and wireless planes of the hybrid architecture is evaluated, seeking to prove the flexibility and higher performance of our approach. Finally, at the architecture level, it is demonstrated that the integration of a BoWNoC within the memory hierarchy can provide substantial benefits in terms of execution speed of multiprocessors.

The remainder of the thesis is as follows. Before detailing the different contributions of our work, this writing provides a taxonomy of recent works that propose ways to improve the broadcast support in NoCs. This is presented in Chapter 2 which, for the sake of fairness, puts emphasis on emerging interconnect technologies. Chapter 3 delves into the fundamentals of wireless on-chip communication, detailing the main enabling technologies and surveying existing works in the area. Thereafter, the contributions at each of the design layers are detailed following a bottom-up approach: we start from the physical layer in Chapter 4, up to the layer level in Chapter 5, to then address the network layer in Chapter 6, and finish with the multiprocessor architecture in Chapter 7. Chapter 8 concludes the dissertation with a summary of the lessons learned and paths that could be explored in future research endeavors.



## Chapter 2

# Broadcast in Emerging Interconnect Paradigms

Current interconnects present intrinsic limitations in terms of delay and energy efficiency that suggest the use of alternative solutions in next-generation on-chip networks. The width of the on-chip wires is reduced in each technology generation, increasing the resistance per unit of length. This, in turn, causes an exponential increment of the Resistive-Capacitive (RC) delay, as shown in the log plot of Figure 2.1(a). Traditionally, technology downscaling also implied faster system clocks, which reduce the symbol time and thus rendering the charging and discharging the wire within the allotted times a challenging task. All in all, this means that while processors may be able to work at frequencies as high as 5 GHz, moving data across the chip would be likely done much slower [27].

Although parallel and low-power computing trends seen in recent years strongly discourage the use of high clock frequencies, the interconnect problem still persists. The energy efficiency issue for global interconnects is especially concerning given that the power density is expected to increase with the trend shown in Figure 2.1(b). Therefore, in an era where the technology downscaling keeps narrowing the performance bottleneck imposed by traditional wireline on-chip networks, research efforts have been recently focused on both (1) pushing the performance limits of electrical interconnects while hiding their limitations in upper layers of design, and (2) finding scalable alternatives in the physical layer of design and conceiving on-chip networks around them [28].

By extending the original NoC paradigm to new interconnect technologies with enhanced characteristics, the performance scalability of the NoCs could be greatly improved. However, this is not the only way new technologies impact the design of next-generation processors. In fact, we will see throughout this dissertation that unique capabilities of new interconnect technologies will likely lead to a rethinking of current network and multiprocessor architectures.

For all this, and with the aim to provide the necessary context for the dissertation, this chapter provides a brief yet comprehensive survey on the state of the art of on-chip networking. We review advances in NoC design based on conventional RC signaling, three-dimensional NoCs enabled by vertical stacking, RF interconnects, and nanophotonics [29]. As summarized in Table 2.1, we compare their principles of operation, technological readiness, and bandwidth and energy efficiency characteristics. Finally, we devote a few lines to describe the broadcast capabilities of each analyzed contender in order to clarify their potential as possible alternatives for the implementation of broadcast-oriented NoCs [26].

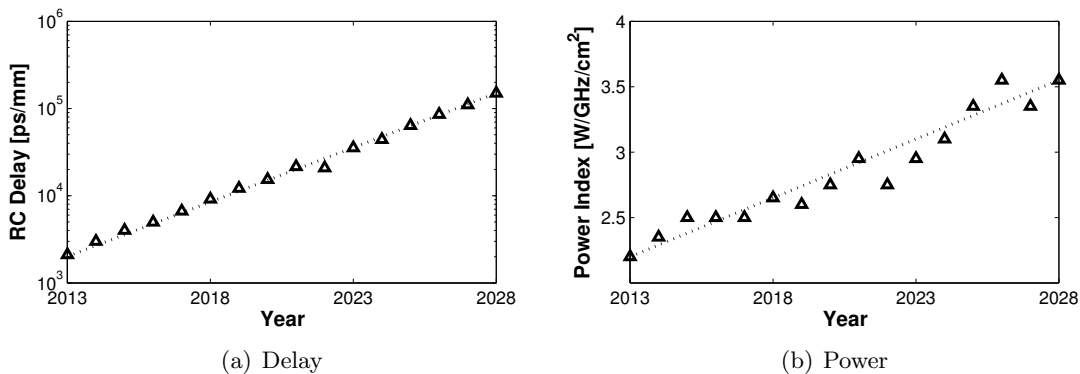


Figure 2.1: Future evolution of RC interconnect performance according to the 2013 International Technology Roadmap for Semiconductors (ITRS).

## 2.1 Advanced Network-on-Chip Design

The delay and energy problems of conventional signaling schemes are the main motivation behind the quest to find new interconnect technologies for the short or mid term. However, this does not stop research to keep improving the performance and efficiency of traditional RC wires and networks based on them. A plethora of works have been pushing the limits at the physical layer of design, whereas upper levels have been attempting to alleviate the problems inherent to the technology [12].

A first and understandable step has been to try to reduce the area footprint and power consumption of on-chip links without compromising their speed. This has been accomplished with new transceivers and line drivers that are capable of operating with reduced voltage swings (i.e. difference between the highest and lowest voltage levels) maintaining the signal integrity at acceptable levels [21,30]. These works use different techniques which are out of the scope of this dissertation, but that accomplish significant the power savings. For instance, the chip fabricated in [21] has a power consumption 40% lower than its baseline.

New approaches have also seen on-chip links as something more than a mere point-to-point interconnection between two routers. The work by Grot *et al*, for instance, proposes the use of multidrop links where signals reach multiple routers at once [31]. This strongly reduces the diameter of the network without the need of very high radix routers. Another interesting research line proposes the use of asynchronous links that can traverse multiple routers within the same clock cycle [32]. These links are built up and broken down on a per-cycle basis by sending bypass signals towards the destination.

The next step is to improve the routers. Continuous efforts have been achieving a progressive reduction of the router pipeline delay from around 4-5 cycles to 1 cycle in contention-free situations, or 2 cycles at most when there is contention. Speculation has been the main path towards such improvement. A widespread speculation technique consists in assuming that the switch will be correctly allocated if an output virtual channel is successfully granted [33]. Other predictive schemes are less common and generally less accurate, like assuming that all requests will be granted under low load [34] or predicting which will be the output port required by the next flit without having yet read its header [35].

At the network architecture level, the need for lower latency suggests the use of topologies with higher connectivity than the all-pervasive 2D mesh. In these cases, there is a trade-off between connectivity and the radix of the router, which is often indicative of the

Table 2.1: Comparative summary of the emerging interconnect technologies for NoC.

	Baseline	3D NoC	Nanophotonics	RF-I	Wireless
Wiring	RC Wires	Vertical Vias	Waveguides	Transmission Lines	None
Principle	Wire charge	Wire charge	Optical signals	Guided RF signals	Radiated RF signals
Propagation Speed	Large	Short	Speed of light	Speed of light	Speed of light
Technological Availability	Now	Now	Mid term	Short term	Short term
Integration Complexity	Low	Low	Very high	High	High
Bandwidth Density	Good	Better	Best	Better	Modest
Intrinsic Efficiency	Good	Better	Best	Better	Modest
Architectural Flexibility	Average	Average	Low	Low	High
Issues	Wire delay	Thermal Effects	Laser power, buffering	Signal reflections	Multiple access

cost of a given topology. Examples of some new topologies are depicted in Figure 2.2. The flattened butterfly [36,37], which interconnects a core with the rest of cores of its row and column; the express cube topology [31], which achieves a similar connectivity with lower radix by using multidrop links and only allowing one node to transmit through the same row or column; or the small-world NoC [38], which lays express links between strategic routers over a mesh topology to reduce the network diameter without significantly increasing the cost. Finally, there has been a recent proposal that advocates for the design of very high radix switches, i.e. up to 64 ports per switch, providing global connectivity with allegedly affordable area and power footprints [39].

### Multicast and Broadcast

Most of the link, router, and topology improvements summarized above reduce the latency and power to some extent without making any assumption over the traffic typology. Obviously, topologies that shorten the diameter of the network will most likely reduce the latency of multicast traffic in a similar proportion than the latency of global traffic. Yet still, the network remains point-to-point, fact that prevents the design of truly scalable multicast and broadcast schemes without affecting other traffic flows.

As we have seen in Section 1.2, there are three ways to serve multicast messages in conventional NoCs: unicast-based, path-based, and tree-based. On the one hand, unicast-based multicast has been widely disregarded in recent efforts due to its high inefficiency; on the other hand, path-based and tree-based multicast have received constant improvements both at the link and network levels. At the link level, several works have striven to reduce the delay of the flit replication that takes place within the routers. Initial designs with single-port allocation and simple crossbars [19, 40] have evolved into routers that combine (a) allocators that can associate multiple outputs to the same input at the same clock cycle with (b) crossbars that instantly replicate the message at the allocated outputs [20, 41, 42]. This reduces the hop delay from a quantity proportional to the packet length and the number of allocated output ports, to a quantity proportional only to the packet length. At

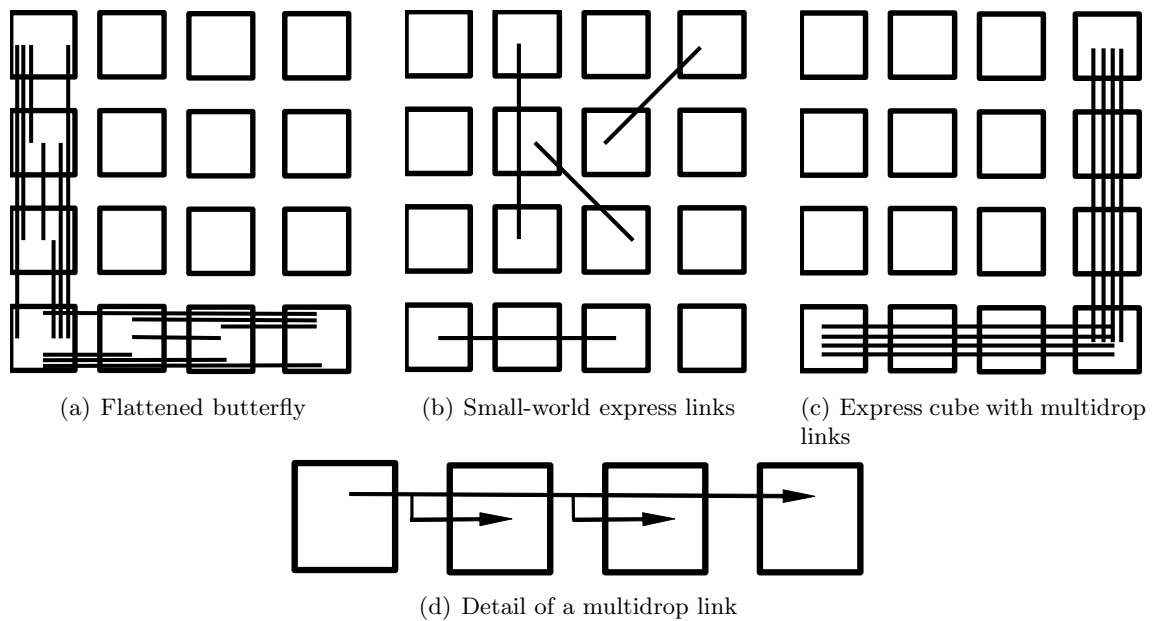


Figure 2.2: High connectivity topologies for low latency NoC.

the network level, successive path-based and tree-based NoCs have been augmented with adaptive routing that balances the load [17, 20, 21, 43] to further enhance their performance.

MIT led several of the aforementioned proposals regarding tree-based routing and multicast router microarchitecture. These works culminated into SCORPIO [25], a 36-core prototype that implements them together with a broadcast ordering mechanism. The authors demonstrated that snooping coherence can be achieved at this scale using an unordered network, and achieved a 25% average execution speedup with respect to directory-based schemes. Even though this work is a huge step towards demonstrating the feasibility and benefits of broadcast-based manycore architectures, it also recognizes important scalability concerns beyond 100 cores.

To achieve scalable broadcast in manycore environments, the use of multidrop links [31] seems a feasible option given that messages are automatically replicated in each node of the same row and column. Similarly, the high-radix switch approach proposed in [39] claims to be able to provide multicast and broadcast with affordable cost. Unfortunately, the paper does not demonstrate this fact. Finally, the multi-hop link approach by Krishna *et al* has been also augmented to allow the fast transmission of multicasts and broadcasts as analyzed in [44]. All these options could provide broadcast communication within at least two hops, which is a great improvement with respect to the path-based and tree-based schemes outlined above. Yet still, two aspects need to be resolved: the scaling complexity of the required routers and links, and how the *flooding* effects caused by broadcast traffic would impact on the unicast traffic.

## 2.2 3D Integration

Three-dimensional stacking consists on the superposition of different layers of active devices. These layers are separated by just a few tens of micrometers and are vertically interconnected generally by means of through-silicon vias [45] as sketched in Figure 2.3.

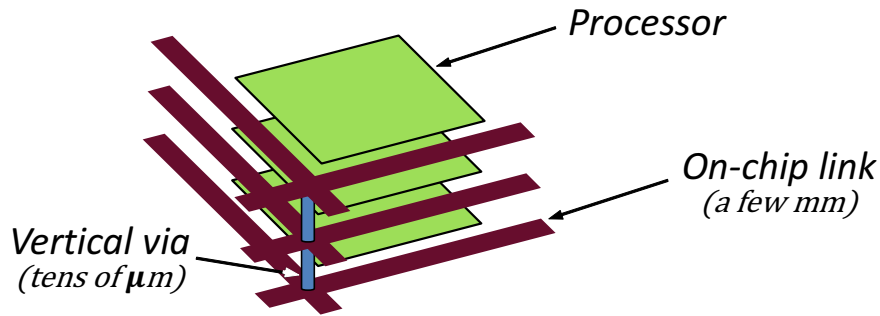


Figure 2.3: Sketch of a stacked processor setting with on-chip links and vertical vias.

The creation of 3D integrated circuits has proved to be a promising paradigm, since it has shown to imply significant benefits such as higher packing density, improved noise immunity and overall superior performance [46].

Latest advancements in this field focus on improving the performance of vertical interconnects. The work in [47] proposes the employment of vertical slit field-effect transistors, which consume less area and power than CMOS through-silicon vias. Near-field coupling schemes have been also proposed, aiming to virtually eliminate the need for a physical interconnection between layers [48, 49]. Proposals in this regard are divided according to the coupling type. Inductive coupling schemes generally offer longer transmission ranges than capacitive ones for similar data rates [49], whereas the area overhead introduced by capacitive coupling transceivers is approximately one order of magnitude lower than the inductive options [48]. In both cases, the energy required for transmission is still substantially higher than state-of-the-art vias.

### Pros and Cons

From a communications perspective, the advent of 3D stacking interconnect technologies has allowed to break the traditionally planar nature of NoC, opening a set of new design opportunities. The main advantage of this approach is the significant reduction in terms of average propagation delay and energy per bit given by the short distance between layers of processors, in the order of a few tens of micrometers. The work by [50] provides a thorough analysis on this respect.

More significantly, the use of 3D stacking enables the design of topologies that would be unfeasible in the 2D design space, potentially yielding reduced multihop latency results [51]. Since the advantages of this approach are mainly at the network level, the potential improvements are compatible with and practically independent of the underlying interconnection technology. Additionally, 3D stacking is an effective way to intuitively interface different technologies in hybrid approaches, facilitating modularity by avoiding the integration of different technologies within the same layer.

It is important to note that 3D stacking presents considerable challenges. The superposition of active layers produces an increase in the heat density that must be circumvented in order to avoid thermal effects. In this case, the work by Nandakumar *et al* reduces the power of through-silicon vias, alleviating the heat density issues to some extent [47]. Another downturn of this approach is that refined techniques are needed for the manufacture of such tridimensional integrated circuits and networks; in particular, alignment method-

ologies are required for the precise positioning of the vias. The use of wireless coupling schemes could eliminate this issue, yet at the cost of higher power consumption. These issues still need to be further researched in order to validate the applicability of 3D stacking in the manycore era.

### 3D Network-on-Chip

The intrinsic bandwidth density and technological readiness of 3D stacking have been the main reasons of its adoption in several works. Due to this, communication capabilities in the third dimension are being introduced within NoC design tools and as part of a multicore processor design flows [52].

The topology exploration performed in [51] has shown that 3D extensions of the mesh topology can achieve significant latency and throughput improvements. This is due to the reduction in terms of both physical and logical distance between nodes, which leads to a lower network diameter and overshadows the area and power overheads resulting from the increase of network radix. The work also studies new topologies where vias do not always directly connect two nodes located directly above or below each other, disposition that enables the 3D extension of butterfly fat tree topologies.

The stringent heat issues of 3D NoCs have also affected the design of the routing schemes. A relevant example of this is presented in [53], which proposes a routing mechanism that obtains thermal profile information at runtime and adapts the routing table to avoid traversing *hot areas*.

### Multicast and Broadcast

The application of 3D techniques has obvious implications on the design of multicast and broadcast strategies. Due to the reduction in the average distance between nodes, multicast and broadcast techniques become less expensive than in 2D environments. Yet their application is not straightforward and requires the extension of the routing schemes in the third dimension. This has been the object of different works [54, 55], which explored different routing options encompassing both path-based and tree-based alternatives, as well as fixed or adaptive routing. In any case, and even though the three-dimensional paradigm improves the performance of multicast and broadcast flows, contention in tree-based techniques and latency scalability problems in path-based techniques are still issues inherent to the wired interconnects.

## 2.3 RF Interconnects

The RF interconnection paradigm is presented as an alternative to traditional voltage and current signaling through metallic wires. Precisely, RF interconnection accounts for the transmission of electromagnetic waves over microstrip transmission lines or co-planar waveguides printed within the metal layers of the chip [27, 56]. The original baseband signals are modulated using carrier waves at frequencies up to several GHz and then guided through the transmission lines. Signals propagate at the speed of light instead of at the charging and discharging speed of RC wires, and need to be taken back at the baseband frequency and demodulated at the receiving end.

Transmission lines can have multiple inlets and outlets, allowing to design multi-receiver schemes similar to the multidrop links mentioned in Section 2.1. Also, shared transmission

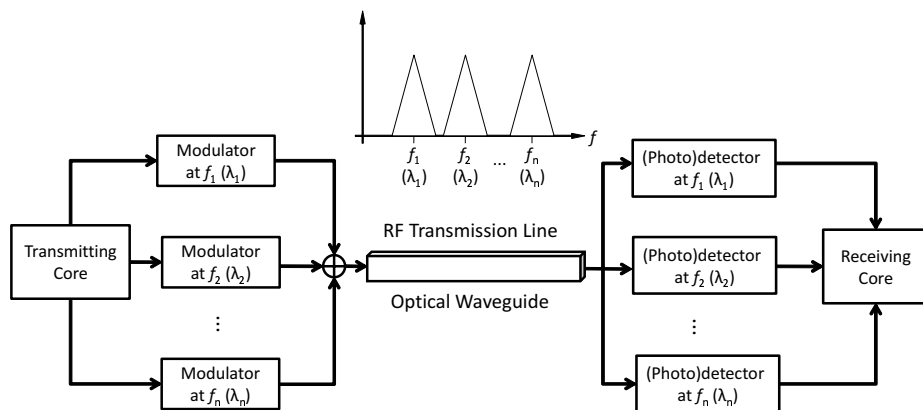


Figure 2.4: Sketch of a frequency-multiplexing scheme using transmission lines or optical waveguides. A similar approach can be applied to perform code multiplexing. To perform time multiplexing, cores simply need to transmit in different time slots.

lines enable the use of multiplexing techniques, this is, the simultaneous transmission of signals modulated at different frequencies (as represented in Fig. 2.4), codes, or time slots. This can be used to improve the bandwidth of a single transmission or to address transmissions coming from different nodes. As we will see, the RF interconnect paradigm shares the same basic principles with wireless on-chip communication. Thus, we refer the reader to Chapter 3 for further details.

### Pros and Cons

The speed-of-light propagation is one of the main advantages of the RF interconnects. In long-range links, the improvement in terms of propagation time is very large with respect to the delay introduced by the modulation process and, therefore, the communication latency can be effectively reduced and be made virtually independent of the link length [27]. A second advantage stems from the possibility of transmitting several frequency-orthogonal signals through the same transmission line. Each core can be assigned a set of channels, so that several cores are interconnected using the same transmission line, thereby reducing the number of wires. Further, the bandwidth for each core could be dynamically assigned at runtime according to application demands.

The use of RF interconnects also entails several issues that reduce its scalability and practicality from a system-level perspective, specially when considering multicast or broadcast mechanisms. For instance, the circuital implementation of frequency multiplexing transceivers produces both an area and power overhead that must be reduced before the whole design can be scaled. Also, the physical topology must be carefully designed as impedance mismatch reflections at the terminations of the transmission line may generate interferences. This strongly limits the number of nodes connected through the same transmission line, and forces the use of amplifiers that bridge different transmission line segments. The work in [57] limits the number of connections per segment to five, and then discusses the cost of introducing amplifiers and the need to avoid positive feedback loops at the topology.

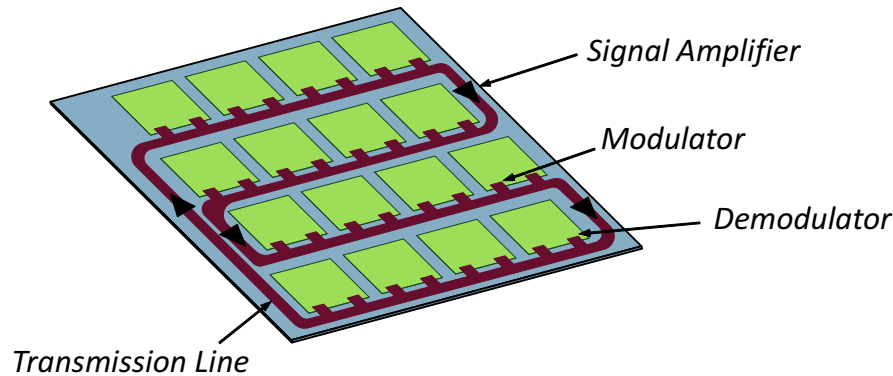


Figure 2.5: Sketch of a NoC overlaid with transmission lines.

### Networks-on-Chip with RF-I

Perhaps due to its scalability limitations, the use of the RF interconnect technology has been somewhat limited despite its technological readiness. Opportunistic applications have been proposed over this approach, such as communication among very large caches with bounded latency [58] or the use of carrier sensing techniques within the transmission lines in order to implement very fast synchronization barriers [57]. Transmission lines as global multidrop interconnects have been also opportunistically proposed to support fast locks [59] and barriers [60].

In the more traditional sense of NoC, the RF interconnect has been proposed in unswitched schemes and as a complement of an underlying conventional NoC. For instance, the work in [61] considers a global multiband transmission line laid over a mesh. The transmission line, which has a shape similar to the one sketched in Figure 2.5, is considered as a *shortcut* between far apart cores and used only for long-range communication. This reduces the overall latency and execution time of SPLASH-2 benchmarks. A similar approach is that of [62], which combines a mesh with a transmission line ring that goes through all nodes. The decision on whether to use the mesh or the ring is taken at the network interface and depends on the load and the potential latency gain at each network plane.

NoCs implemented with RF interconnects only have been also proposed. The work by Carpenter *et al* takes a bus-based approach by interconnecting all cores to the same transmission line and using a centralized arbiter to provide multiple access [63]. Their subsequent paper improves the throughput of the network by means of techniques applied in other contexts to transmission lines or buses [64]. Another work that is worth noting is presented in [65]. It proposes a global transmission line scheme that implements Orthogonal Frequency Division Multiple Access (OFDMA), promising dynamic reconfiguration and better performance than baseline NoCs for 32 cores and above. The approach is scaled by using the transmission line to interconnect clusters of cores, yet the complexity of the signal processing required to implement OFDMA seems to remain as a roadblock for its actual implementation.

### Multicast and Broadcast

Most of the NoC proposals with RF interconnects have a global transmission line that passes through almost all nodes. If all nodes share the same frequency channels or a single-writer-multiple-reader scheme is replicated, the transmission line will inherently support multicast



and broadcast. However, it is not straightforward to scale a system with these features due to the high number of reflections present at the transmission line inlets and outlets. In fact, the bus-based scheme by [63] does not support broadcast to reduce the design complexity. Only the works in [62, 65] put emphasis in the broadcast capabilities of the scheme, but do not provide details on their performance. Related to this, the work in [57] takes advantage of the shared medium nature of RF interconnects to implement an unconventional reduction scheme used, in this case, to implement barrier synchronization. Basically, the presence of signal in the waveguide is interpreted as a collective OR, as it appears when one or more nodes transmit.

## 2.4 Nanophotonics

At the time of writing this thesis, Nature published an article that reveals the implementation details of a single-chip multiprocessor that internally communicates using light [66]. This represents a very important milestone in the application of nanophotonics in the on-chip communication field, idea that has been around since the first creation of CMOS-compatible optical building blocks [28]. The scaling and generalization of the idea implemented in [66] embody the concept of *photonic NoC*, where light is coupled on chip, modulated at the transmitter, and guided to the receiver through integrated waveguides. Other works propose to reach the receiver through freespace optics and steerable mirrors, which would add a certain flexibility at the cost of efficiency [67]. In either case, nanophotonics support the transmission of signals at different frequencies, or better referred here to as *wavelengths*, but at a much higher density than in RF interconnects.

Intense research efforts in this field have been directed towards developing low-loss and low-footprint building blocks for the creation of these networks. The objective is to achieve transmission speeds in excess of 50 Gbps in integrated platforms [68] with energy figures as low as 1 fJ/bit [69]. Couplers, graters, light splitters, modulators, switches, waveguides and photodetectors are the components that need to be developed and optimized to this end [70–74]. Some of them can be implemented using microring resonators, the behavior of which is summarized in Figure 2.6. Microring resonators basically divert light of a certain wavelength when a voltage is applied over them or let light pass through otherwise. The resonant wavelength depends on the size of the microring, and needs to be tuned with fine granularity by applying localized and stable heat. One can develop a modulator by using a digital source to drive the resonator, whereas a switch can be devised by combining a set of these microrings [72]. Switching can be also wavelength-selective [73].

The work in [66] previously mentioned is of essential importance as it demonstrated the CMOS-compatibility of the paradigm. The authors report that a *zero-change* policy was achieved when manufacturing the prototype, meaning that only standard fabrication processes were used to implement and assemble the different components. A total circuit efficiency of 1.3 pJ per bit was achieved while operating at 2.5 Gbps, outstanding numbers considering that this is the first complete prototype implemented with standard processes.

### Pros and Cons

The transmission and reception principles of nanophotonics can be somehow seen as a particular and extreme case of RF interconnect, as optical signals are basically electromagnetic waves at much higher frequency than RF signals. Consequently, they share some of the

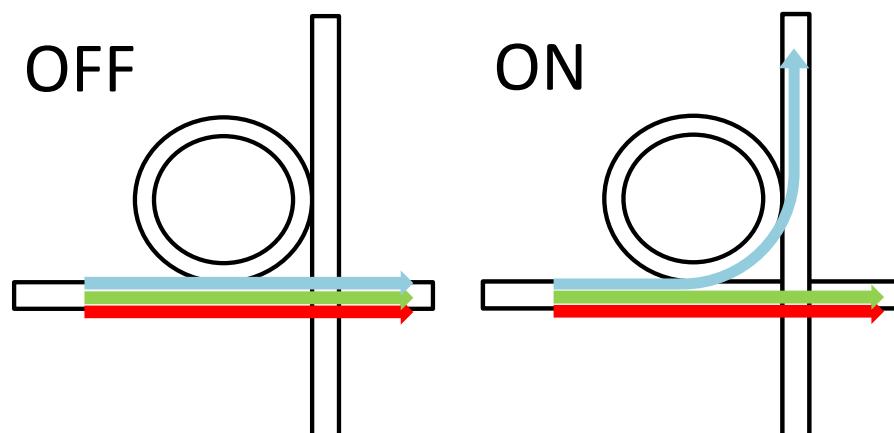


Figure 2.6: Basic operation principle of ring resonators.

advantages, like the relative flexibility provided by multiplexing or the low latency potential given by the speed-of-light propagation. Also, the bandwidth density delivered by the nanophotonics technology is much higher than in other emerging alternatives due to the very small wavelengths involved in the communication. Additionally, the intrinsic energy efficiency of the approach is orders of magnitude better than that of RC wires.

The main limitation of the nanophotonic paradigm is that existing laser sources are bulky and technically complex, which force designers to place them off the chip. The lack of effective on-chip lasers imposes the use of static power allocation techniques, which reduces the flexibility and energy efficiency of the network. Research on laser management adaptable at runtime [75], predictive strategies [76], or freespace optics solutions favoring the use of on-chip vertical-cavity surface-emitting lasers [67] may alleviate these issues though. Finally, it is worth noting that the impracticality of light buffering and header processing prevents the creation of optical routers for on-chip communication, hampering the use of conventional NoC techniques in a nanophotonics platform. Even though these problems can be overcome with hybrid electro-optical routers [77], converting light into electrical signals (and vice versa) at each hop increases the energy cost and decreases performance, discouraging their use in large-scale systems.

### Photonic Network-on-Chip

The experimental work by [66] represents a very simple design of microprocessor to memory communication, yet the initial demonstration of a future family of photonic NoCs. Actually, ever since the advent of nanophotonic interconnects, considerable efforts have been put into leveraging their outstanding properties to design on-chip networks for next generation multiprocessors. Given that buffering and header processing must be done in the electrical domain, alternatives to routed mesh designs were sought towards the idea of completely optical NoCs. A wide and growing variety of designs can be found throughout the literature, from simple and conventional to novel and tailor-made topologies, including both arbitration based and contention-free architectures, a selection of which we briefly outline next.

Kirman *et al.* realized one of the first works on nanophotonic NoCs by devising a bus-based architecture for snooping purposes. The architecture consists of a logical crossbar that is implemented by replicating a single-writer multi-reader (SWMR) structure [78]. Each transmitter is tuned to a unique wavelength and broadcasts its messages through a

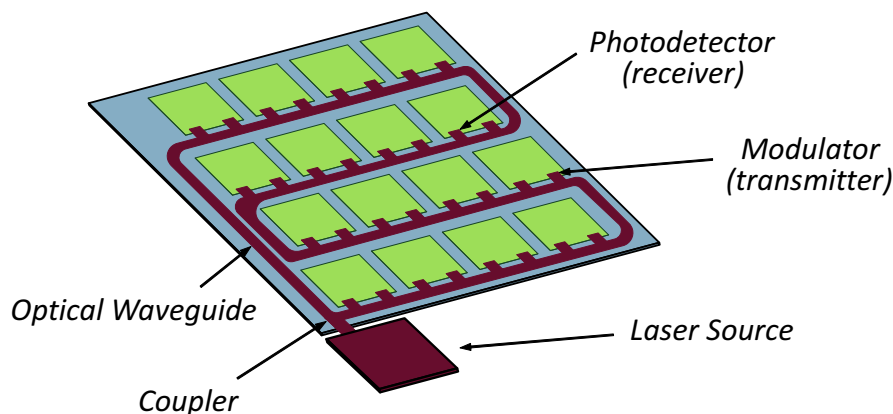


Figure 2.7: Sketch of a photonic NoC with snake-shaped waveguides.

serpentine waveguide (as the sketched in Figure 2.7) that connects all  $N$  cores, each of which must account for  $N - 1$  detectors in the receiving end. Although the transmission medium is shared, contention is avoided through the use of exclusive wavelengths. Multibit transmission can be achieved by means of waveguide replication. The work in [79] extends this architecture for general-purpose chip communication, evaluating its performance in a directory-based coherence protocol.

HP designed a similar architecture named CORONA [80]. This scheme also implements a full crossbar, but this time by replicating a multi-writer single-reader (MWSR) scheme. Each core is associated to a given waveguide and can write to any other waveguide. In this case, two cores will contend if they want to communicate with the same destination. To avoid possible collisions, CORONA implements an all-optical token-ring-based arbitration scheme. WDM techniques are used to implement multibit transmissions.

An alternative to the abovementioned crossbars consists in deploying a set of switches that will guide the information to the intended receiver. The challenge here is to avoid contention by adequately setting up the switches. In a first approach, a parallel electrical NoC was used in order to implement circuit switching [72]: when a node has something to transmit, it sends a path setup packet through the electrical NoC, which prepares the switches as it is routed to its destination. Upon acknowledgment, the transmission starts at the optical plane and the path is maintained until the transmission finishes. Then, the receiver sends a path breakdown packet through the control plane. Unfortunately, this approach strongly limits the number of simultaneous transmissions and, therefore, the achievable throughput. Path multiplicity by means of component replication [72] and time-multiplexed interleaving [81] have been proposed in order to alleviate this issue.

Static switching schemes constitute another important branch of research. In this case, switches try to cover all the transceiver-receiver pairs by means of wavelength-selective switching. The work in [82] implements an alternative topology by employing higher-radius switches and static wavelength assignment. The work in [73] considers a mesh topology instead and assigns the orders in the *switching table* so that oblivious routing is accomplished. Spatial routing, i.e. the wavelength assignment is made as a function of the switch position, has been also proposed [83].

Finally, a set of proposals aim to combine nanophotonics with other technologies for different purposes. For instance, 3D stacking techniques are used in [84] to create a set of partial networks without incurring in an excessive number of waveguide crossings. Also,

the proposals in [85, 86] implement a hybrid topology which performs intra-cluster communication by means of a photonic NoC and inter-cluster communication by means of a conventional or wireless NoC. This structure is also proposed in [78, 79] to scale their designs.

### **Multicast and Broadcast**

As in RF interconnects, nanophotonics is mainly proposed as a platform for global communication in a sort of bus that goes through all nodes. In CORONA [80], a fraction of the resources is devoted to implement a broadcast bus arbitrated by means of a token-passing scheme. On the other hand, SWMR structures like those of [78, 79] inherently deliver the messages to all receivers without the need for arbitration. This would be an ideal scheme for broadcast, but laser power problems strongly limit its scalability: each of the nodes that are interconnected extract a fraction of the light and add losses that are logarithmic in nature. As we will see in Chapter 4, these losses and the fact that laser power allocation is generally static, render impractical the development of scalable broadcast schemes.

Morris *et al* employ a slightly different photonic network to implement broadcast within a snooping coherence architecture [87]. There are two groups of waveguides: a first set that passes through the different cores divided in columns, and a second set that implements a tree connecting all cores. Light passes first through a control center, which redirects it through only one particular column. At that column, token-based arbitration ensures that only one core can intercept the optical signals. That core modulates the light and couples it to the tree waveguides, which distribute the signals to all cores. The authors also propose a multicast scheme that incorporates ring resonators at the splitting points of the tree, so that a message will not enter a *leaf* unless one of its nodes is within the destination set. Albeit complex, these structures reduce the amount of components within the critical path of light (thus reducing the laser power requirements), but does not completely solve the scalability problem associated with this technology.

Finally, note that the unconventional many-to-one communication scheme proposed in [57] for RF interconnects is theoretically applicable to the nanophotonic scenario. A similar scheme has in fact been proposed in [80] to implement a token-based arbitration mechanism and in [88] to implement global locks. In these cases, light at a particular wavelength circulates through a shared waveguide unless one of the nodes diverts it. Therefore, the presence or absence of light can be considered as a collective AND of all the nodes that share the same wavelength.

## Chapter 3

# Wireless On-Chip Communication and Networking

Wireless on-chip communications and networking is a relatively recent field. A first pioneering work appeared in 2002 proposing the use of the novel concept of on-chip antenna to wirelessly distribute the clock signal across a single chip [89]. This approach was validated with a test chip integrating both the transmitter and the receiver with their respective zig-zag antennas at 15 GHz. It was a very simple scheme that did not even need to modulate the clock signal at the transmitting end, and only amplified and frequency-divided the signal at the receiver.

Albeit very simple, this work paved the way for the development of the WNoC paradigm. The approach is possible by virtue of the steady technology scaling trends that, in the pathway to millimeter-wave (mmWave) frequencies, have scaled the antennas and the passive elements of the circuits that drive the antennas down to sizes much smaller than the silicon chips where they are integrated. As a reference, Figure 3.1 shows the scaling trend of inductors according to data from the International Technology Roadmap for Semiconductors (ITRS) and [61]. This behavior can be extrapolated to on-chip antennas which, for instance, would have an quarter-wave axial length of around 0.38 mm and a width on the order of 0.01 mm if designed in zig-zag shape for 60-GHz operation [90]. In the face of this prospect, intense research has revolved around the development of efficient and effective on-chip antennas, transmitters, and receivers for wireless chip-scale communication [91, 92].

Wireless on-chip communication shares the basic working principles with the RF interconnect paradigm outlined in the previous chapter, yet with different propagation medium. Basically, a transmitting antenna modulates information coming from a processor core and radiates the resulting RF signals at a given frequency. These waves propagate within the chip package following different propagation mechanisms [93] and may be received by any other antenna tuned to the same frequency and within the range of the transmitter. In other words, all antennas tuned to the same frequency channel share the medium similarly to in a bus or a transmission line. Also, different channels can be implemented by using frequency or code multiplexing either to increase the bandwidth or for addressing purposes.

The advantages and downturns of wireless on-chip communication subtly differ from those of RF interconnects mostly due to the fact that energy is radiated instead of guided through transmission lines. The latency advantage with respect to RC signaling still applies, as wave propagation occurs at the speed of light. Wireless on-chip communication also shows improved simplicity, flexibility, and reconfigurability potential as compared with the rest

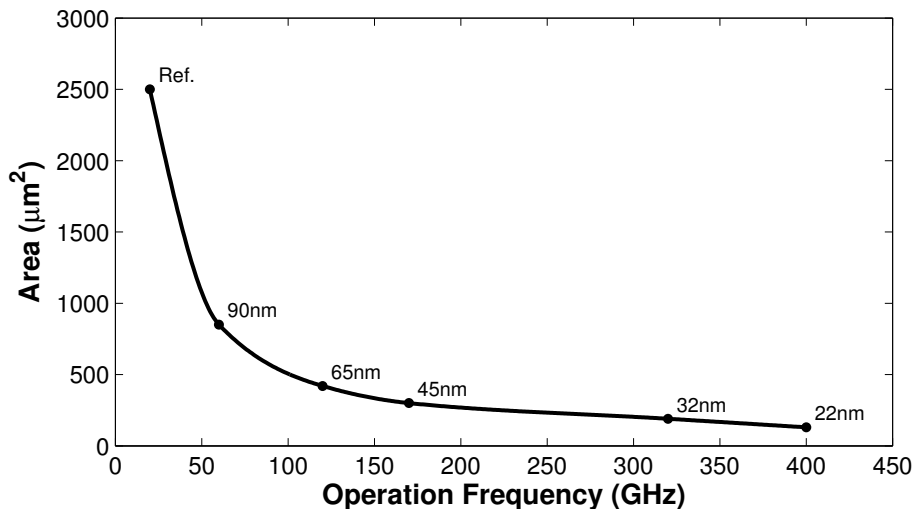


Figure 3.1: Inductor scalability according to data from the literature.

of interconnect technologies since no path infrastructure is required between nodes. Thus, a WNoC can modify the logical topology or other transmission parameters without the need of any physical modification. This paradigm can even become modular by virtue of the *wireless core* concept, wherein an antenna and a transceiver are integrated within each core. A library of general-purpose or specific wireless cores could be created, allowing the building of custom multicore processors by the integration and initial configuration of a set of such pre-designed cores. Last but not least, WNoC offers native broadcast capabilities as information may potentially reach any core regardless of its location.

The WNoC paradigm also shows two main downturns with respect to the rest of alternatives: energy efficiency and bandwidth. Since energy is radiated through space rather than guided in through planar 2D structure, wireless communication is intrinsically less efficient. The lack of an isolated medium for propagation also affects bandwidth, as all antennas tuned to the same frequency need to either *compete* or be scheduled to access to the medium. There are no means to increase bandwidth by replicating a structure such as a transmission line or a waveguide. Adding frequency channels does increment the network bandwidth, yet with a non-scalable cost in terms of implementation.

The inherent broadcast capabilities of WNoC are arguably its most appealing advantage. This implies that multicast messages may actually be conveyed to the receivers in a few clock cycles. As we will see in Chapter 5, this directly contrasts with the broadcast performance of conventional NoCs where the latency increases proportionally with the system size. The inherent simplicity and potential modularity of WNoC also favors its use for broadcast schemes rather than transmission lines or optical waveguides. As discussed in the previous chapter, the scalability of RF interconnects is limited by the complexity of transmission line design for point-to-multipoint communication, whereas the scalability in nanophotonic options is limited by issues related to the laser power. It is worth noting that, as we will see in Chapter 7, the scalable implementation of a broadcast platform opens a vast design space at the architectural level and heavily alleviates the constraints of parallel architecture design, therefore reducing the complexity of parallel programming.

As its name implies, the BoWNoC paradigm envisaged in this dissertation builds upon the unique broadcast advantage provided by the wireless technology. Before delving into

Table 3.1: Characteristics of wireless scenarios as transmission ranges are scaled down.

	Indoor	Chip-to-chip	Intra-chip	NoC scale
Transmission Range	100 cm	10 cm	1 cm	0.1 cm
Power	100 mW	10 mW	1 mW	0.1 mW
Data Rate	0.1 Gbps	1 Gbps	10 Gbps	100 Gbps

the details of BoWNoC design and the study of its feasibility, this chapter provides the necessary background to understand the fundamentals of wireless on-chip communication and networking. We first detail the main enabling technologies of the wireless RF approach in Section 3.1. We then describe, in Section 3.2, how to use the technology to implement individual and collective communication patterns. In Section 3.3, we outline the main design guidelines of WNoC using a custom layered approach. Finally, Section 3.4 provides a brief summary of the related work in WNoC architectures.

### 3.1 Enablers of the Wireless RF Approach

The downscaling of CMOS-RF technologies has been the main enabler of the WNoC paradigm thus far. The availability of on-chip antennas and transceivers commensurate to the dimensions of a processor core, providing bandwidths of a few Gbps and consuming several picojoules per bit have been enough for first WNoC proposals [94–97]. However, as the core density increases, the size of current antennas may restrict the scope of WNoC to hybrid architectures wherein the wireless plane is employed to communicate a reduced set of cores. Although such wireless backbone approach allows a reduction of the network diameter and has been shown to outperform conventional NoCs, its potential for broadcast-based communications is limited by the performance of the electrical edges of the network.

To take full advantage of the BoWNoC vision outlined in Section 1.3, wireless communication must be implemented at the core level through a single broadband channel with a capacity of around one flit per cycle. This implies that the wireless unit should:

- Have a size between 0.01 and 0.1mm<sup>2</sup> to be commensurate in size with future cores,
- Offer a data rate of several tens of Gbps with reasonably low Bit Error Rate (BER) to ensure the usefulness of the system, and
- Consume less than 1 pJ/bit/core to provide an at least similar energy efficiency than that of related work in broadcast support for NoCs [25].

These goals are consistent with the vision of the research area given by Laha *et al* [92] and contextualized in Table 3.1 based on their assumptions. As the authors point out, meeting these objectives compels the use of antennas and transceivers able to operate at high frequencies. This theoretically (and empirically, as we will see in Chapter 4) reduces both the area occupation and energy per bit and increases the available bandwidth [57, 61]. Assuming a clock frequency of 1 GHz, an operation frequency of around 100 GHz and above seems an appropriate target. Next, we outline some technologies that could allow to reach the area, power, and performance figures mentioned above.

#### 3.1.1 Scaling CMOS Technology

A growing number of publications report advancements in the design of on-chip antennas [91, 98–101] and transceivers [102–104] working at the 60-GHz band for different applications.

As CMOS technology evolves and advanced devices such as FinFETs and III-V on silicon are implemented, it becomes possible to further raise the carrier frequency into the mmWave region (up to 300 GHz), thereby significantly increasing the available signal bandwidth and decreasing the energy required per bit of data. However, translating this potential into effective performance requires advancing beyond the state of the art in the design of both on-chip antennas and circuits for transceivers, as briefly detailed next.

**On-chip antennas:** At 60 GHz, the literature presents a wide variety of integrated antenna designs with gains ranging between -20 dBi and 15 dBi for different applications [100]. A design relevant for the WNoC application is presented in [105]. It is a rectangular patch antenna that is carefully integrated on chip to meet all the restrictions of the 130nm CMOS process, and resonates at 60 GHz with a peak gain of -3.32 dBi and enough isolation from the surrounding circuits.

For antenna designs beyond 60 GHz, Markish *et al* review design trends and specific implementations found in the literature [101]. The work by Hou *et al* present two designs: a monopole with a peak gain around 6 dBi and a bandwidth of 13% around 130 GHz, and a Vivaldi antenna with a peak gain of 5.5 dBi and 30% bandwidth around 150 GHz [98]. Pushing further in frequency, Park *et al* achieve a 4.9 dBi with more than 26% bandwidth around 245 GHz through a microstrip leaky-wave antenna. Approaching the terahertz band, classically used for imaging applications, Markish *et al* provide a skirt-shaped design with a peak gain of 7.1 dBi and a huge 65% bandwidth at 1 THz.

Designing an on-chip mmWave antenna close to the transceiver requires a high level of integration and area reuse. This also implies a need to combat the coupling effects between the antennas and inductors or waveguides that may be used in the transceiver [106], the loss of efficiency due to power absorption in the doped silicon chip, or the high interconnect metal density in the metallization layers above the substrate. Actually, the works mentioned above achieve high performance through different methods that minimize these impairments [101]. Some of these methods require augmenting standard fabrication processes to include elements such as lenses or dielectric resonators, which may not be compatible with the WNoC application. However, antenna types that use only the standard fabrication methods are naturally desired. In this regard, meta-surfaces or periodic structures could be used to improve antenna efficiency [101].

Finally, it is important to note that the use of on-chip antennas inside the multicore environment brings up additional challenges. Reflections between cores could cause channel degradation due to destructive interference and dispersion and must be taken into consideration. Fortunately, the WNoC application occurs within an environment that is static and can be potentially known beforehand; thus, one may try to harness the reflections in order to actually enhance the antenna performance.

**Transceiver circuits:** Current transceiver implementations for the WNoC scenario modulate data on a high frequency carrier in the V band range (40-75 GHz) using simple schemes that reduce the area and power footprint. A representative example is that proposed by Yu *et al*, a 60-GHz transceiver implemented with 65nm CMOS that performs close to the objectives pointed out in Table 3.1 by providing 16 Gbps with a BER of  $10^{-15}$  at a few centimeters, while occupying  $0.3\text{mm}^2$  and consuming 30mW ( $\sim 2\text{pJ/bit}$ ) [102]. Another example is the transceiver presented in [103], which is implemented in 65nm CMOS and operates in the 85-90 GHz band, achieving an impressive energy efficiency of 1.5 pJ/bit when transmitting at 6 Gbps and occupying  $0.09\text{mm}^2$ .



Other proposals have pushed the frequency up to the 135-140 GHz band, yet with more modest performance figures due to the early stages of development of works at those frequencies. A demonstration of the feasibility of integrated transceivers at this band was presented in [107], which successfully transmitted signals at 4 Gbps over a few meters. A design closer to the WNoC requirements is presented in [108] in 40nm CMOS, delivering a data rate of 10 Gbps at around 10 pJ/bit with a BER of  $10^{-11}$  at a distance of 10 cm.

Following these advancements in the mmWave bands, recent years have seen a surge in research for components and circuits for subTHz and THz band wireless communication [109–111]. First circuits for multigigabit communication at frequencies up to 400 GHz have been already proposed [112–114]. Additionally, components reaching frequencies of 0.8 THz are under intense research [115–117] for their application in terahertz imaging and sensing systems [118–120]. These implementations are still highly exploratory, but provide a look into the potential of such high frequency bands in the future.

Most of the research efforts look to overcome the main challenges of ultra-high frequency transceiver design. Although technology scaling indeed allows using CMOS devices at mmWave or even THz frequencies, aspects such as the device parasitics, low supply voltage, device dimension limitations, and the complex metal stack (which adds parasitic intrinsic inductance) limit the transceiver operation frequency and performance. The transceiver design is further complicated by the small available area, power budgets, substrate noise, and the use of a technology optimized for digital operation rather than RF-oriented. These challenges can be addressed by the use of forward body biasing to lower the threshold voltage, the inclusion of dedicated CMOS devices with improved RF performance, or the adoption of improved technologies such as III-V semiconductors and 3D integration.

### 3.1.2 Graphene Technology

Graphene, a flat monolayer of carbon atoms tightly packed in a two-dimensional honeycomb lattice, has recently attracted the attention of the research community due to its extraordinary mechanical, electronic, and optical properties [121]. Graphene allows to utilize novel physics in a plethora of potential applications, ranging from ultra-high-speed transistors [122] to meta-materials [123].

Graphene technology could have a strong impact in the WNoC scenario given the unique characteristics of graphene-based antennas or *graphennas*. These antennas have garnered substantial attention since their subwavelength plasmonic effects allow a patch of a few micrometers to radiate in the lower part of the THz band [124–126] with reasonable efficiency, as we will see in Chapter 4. Therefore, graphene antennas could help to reduce the area footprint of a WNoC, as their dimensions are actually between one and two orders of magnitude below than those of metallic antennas for the same frequency. Additionally, the use of metallic antennas of a few micrometers is unfeasible as the infrared resonant frequencies lead to insurmountable channel attenuation and transceiver complexity issues.

Another interesting property of graphene-based antennas is their inherent *tunability*. Tunable components can change their operation frequency as a reaction to a change in terms of voltage or current. While conventional antennas need to resort to complicated mechanisms to achieve a limited tunability [127], graphene-based antennas can completely change their frequency of operation by simply modifying its chemical potential [126, 128]. This phenomenon is further studied in Chapter 4 and opens the door to a considerable set of opportunities at the multiprocessor architecture level.

Graphene could also impact on the design of future transceivers. The same plasmonic phenomena that drive the operation of graphene-based antennas could also be used to implement a nanotransceiver [129, 130]. But more importantly, it has been shown that graphene-based components are excellent candidates for ultra-high-frequency applications due to the high carrier mobility in the nanomaterial and its ambipolar properties. As summarized in [131, 132], these outstanding characteristics have spawned significant research on graphene-based RF devices and circuits. The concept of Graphene Field-Effect Transistor (GFET), first demonstrated by Lemme *et al* [133], has been implemented reaching impressive cut-off frequencies in excess of 350 GHz [132]. The inherent ambipolarity of graphene also opens the door to non-linear graphene-based components: frequency multipliers [134] and mixers [135] have been already demonstrated.

Besides proving the usefulness of graphene in such isolated circuits, a few works have successfully integrated them into operational CMOS transceivers. The mixer concept, including GFETs and other components, was monolithically integrated on a single silicon wafer as reported in [136]. The subsequent work presented in [137] improves the integration and uses up to three GFETs to build an operational RF receiver front-end.

Finally, note that the possible adoption of graphene as the basis of a new generation of antennas and RF circuits brings up a set of challenges. Precise, efficient and replicable production of graphene is, at the time of this writing, a grand challenge that has sparked huge investments by major companies and research agencies. Another on-going research issue is the identification of heterogeneous integration techniques enabling the integration of graphene into a semiconductor circuit environment. Although graphene device technology is compatible with silicon technology, graphene-dielectric interfaces and metal-graphene contacts need to be optimized as they limit the overall RF performance by reducing the carrier mobility [133] and introducing parasitic resistances [138].

### 3.1.3 Surface Wave Technology

A recent work has proposed the use of engineered surfaces that support the propagation of Zenneck surface waves [139]. Early-stage results confirm that, instead of propagating in all directions, radiated signals are bound to the surface and propagate along it. This shrinks the spreading loss from  $O(1/d^2)$  to approximately  $O(1/d)$ , implying a strong reduction on the path loss and, therefore, on the energy required obtain an acceptable bit error rate at the receivers. Moreover, surfaces can be designed to support the bounding effect for a broad range of frequencies, which overcomes the technical challenge of designing broadband antennas. This paradigm also practically eliminates the propagation of signals at the vertical direction, opening the door to future stacked designs with different orthogonal ultra-broadband channels enabled by a sort of *surface multiplexing*. If feasible, this technology could represent an important breakthrough towards the implementation of energy-efficient and high-throughput shared media [140].

The application of the surface wave technology in on-chip environments shares a challenge with the 3D NoC paradigm explained in 2.2. Signals from the transceiver are coupled into the designed surface using a transducer and a through-silicon via that are stacked on top of the surface. The correct integration of all the elements requires the use of accurate alignment techniques, which poses significant manufacturing challenges.

The characteristics and performance of surface wave wireless interconnects has been compared with those of other emerging interconnects in [26]. The study highlights the combination of advantages inherent to wireless communication and advantages related to

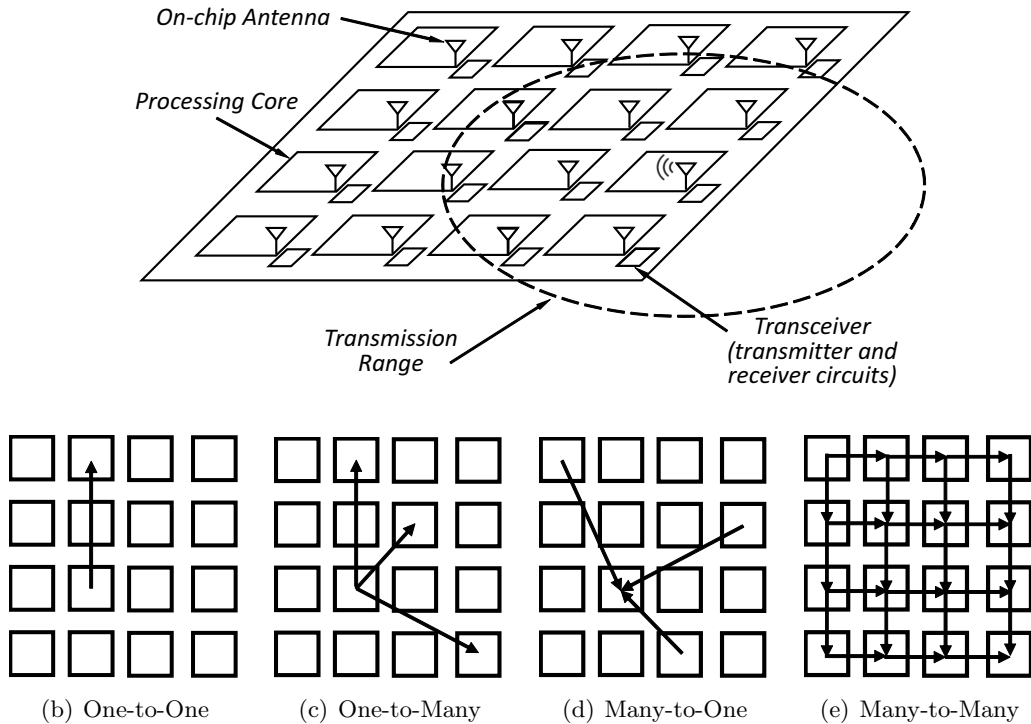


Figure 3.2: Basic WNoC scheme and implementable communication patterns.

wired or guided alternatives. In recent works, the same research group has proposed to lay a network based on surface wave interconnects over a conventional mesh and to use the surface wave planes to assist in the transmission of multicast messages [140].

### 3.2 Wireless On-Chip Communication Patterns

Figure 3.2 summarizes the basic elements present in a WNoC, where communication occurs as follows. First, the sender core serializes information and translates bits into RF signals at a given frequency. These signals are then radiated by an antenna, and propagate throughout the chip until reaching the antennas of other cores. If the receiving antennas are within the transmission range of the source and are tuned to its same frequency, the information can be received and demodulated; otherwise, the receiver remains idle. This opens the door to frequency multiplexing schemes as the sketched in Figure 2.4, where transmissions occurring at different frequencies can take place at the same time.

These features allow the use of the wireless technology to implement different types of communication using both conventional and unconventional design approaches. As summarized in charts 3.2(b)–3.2(d), we can basically distinguish between four typologies depending on the number of transmitters and receivers involved in the communication. Although they can be achieved using any of the interconnect technologies outlined in Chapter 2, we focus on the wireless case next.

**One-to-One (unicast)** - this is the basic type of communication, from a single source to a single destination. Actually, all other communication transactions can be decomposed into a number of unicast messages. In the wireless case, unicast communication can be

implemented by either assigning a unique channel for each pair of nodes, which is impractical, or checking the destination address at the receiving Network Interface (NIF). Due to its potential latency advantage, most of the related work has focused on the application of wireless on-chip communication to serve this kind of traffic between distant cores [94, 141].

**One-to-Many (multicast)** - multicast communication involves a single source and multiple destinations. If the destination set is the whole network, then the message is *broadcast*. Wireless RF shows natural broadcast capabilities since all nodes located within the transmission range of the source and tuned to the same channel are potential receivers. However, this is an often overlooked capability in related WNoC works despite the wide variety of multiprocessor functionalities that need multicast communication to operate. We refer the reader to Chapter 7 for more details on the applicability of multicast in manycore chips.

**Many-to-One (reduction)** - reductions are the opposite to multicast communication transactions, as they involve a plurality of transmitters and a single receiver. In multiprocessor environments, reductions are typically employed to acknowledge the reception of a multicast message, e.g. an invalidation of a shared cache line, or to perform a global operations across all members of a given group of processors [20]. The direct implementation of reductions in wireless communications requires every source to transmit through a unique channel and the receiver to be able to receive from all those channels. This approach is not scalable. However, as mentioned in Chapter 2, different works in RF and nanophotonic interconnects have proposed the use of unconventional *in-flight combination* of signals to perform simple and scalable one-bit reductions [57]. The wireless version of this mechanism requires all transmitters to be tuned to the same channel and the use of a modulation where the presence or absence of signal is interpreted as a zero or one. Under these conditions, the receiver will see a bitwise OR combination of all transmissions, as the absence/presence of signal means that none/at least one of the sources did transmit. As we will see in Chapters 5 and Chapter 7, this opportunistic reduction scheme can be used to notify the presence of collisions in a wireless transmission or to implement lightweight barriers.

**Many-to-Many** - this last type of communication can be defined as the simultaneous occurrence of multiple transfers from several sources to several destinations. This definition includes different operations often found in multiprocessor environments, such as all-to-all broadcasts and reductions where several cores produce a unique broadcast or reduction transactions, or *allscatter* and *allgather* routines where each core sends/receives  $N$  unique pieces of information to/from  $N$  destinations. These operations usually have their explicit communication primitive in message passing systems, as they are common techniques used in parallel algorithms that, for instance, partition data into blocks and maps each data block to a core [142]. The use of a wireless network alone may not be suitable to implement such communication patterns, as the involvement of different sources may require the deployment of a potentially large of channels. Instead, integrating a wireless network on top of a conventional NoC can be beneficial since the wireless plane can efficiently deal with the broadcast and global fractions of such patterns.

### 3.3 Design Principles: Up in the Protocol Stack

In conventional networks, the stacked model seeks the interoperability of diverse communication systems through the use of standard protocols at several levels of abstraction.

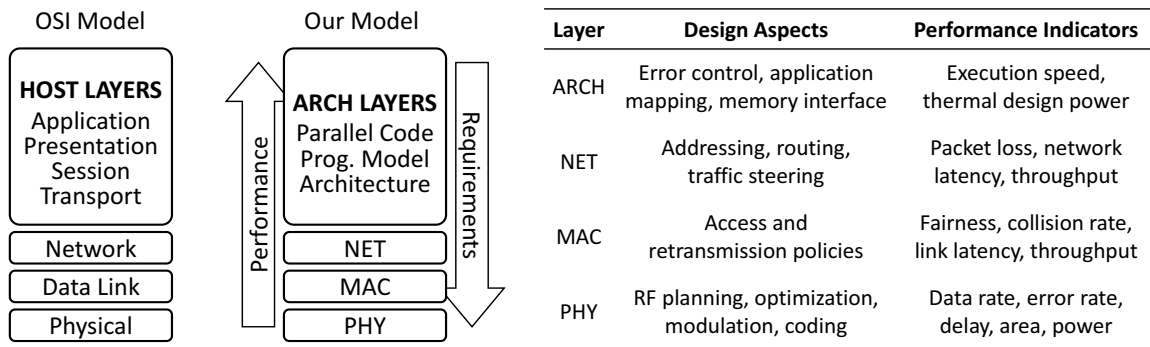


Figure 3.3: Custom layered representation of the different design levels for BoWNoC.

On-chip communication occurs, instead, in a monolithic system where a single vendor may have control over the whole system. Yet within this context, the stack-like organization may still be useful to compartmentalize research, clarify the challenges and objectives present at different levels of design, and expose transversal design opportunities not available in conventional networks.

Figure 3.3 draws an analogy between the classical Open Systems Interconnection (OSI) model and a simplified stack for wireless on-chip networking design. We maintain the physical layer (PHY), use the Medium Access Control (MAC) nomenclature to refer to the data link layer. The main reason is that, as we will see through the work, the MAC protocol is perhaps the most critical design aspect within the BoWNoC paradigm. For simplicity, we also collapse the network and transport functions in the network layer (NET). In our representation, the three lower layers of our model describe the communication side of the multiprocessor. On top of them, the architecture (ARCH) layer describes the computation side of the multiprocessor and encapsulates the higher levels of design, including the multiprocessor architecture or the programming model.

The peculiarities of the WNoC scenario in general (and of the BoWNoC vision in particular) require the design and development of a novel network architecture. Protocols for classical wireless networks cannot be applied to on-chip communications due to the unique blend of physical limitations, stringent performance demands, and unprecedented optimization opportunities of multiprocessor settings. Next, we detail the main design aspects, objectives and outcomes of each of the layers as summarized in Figure 3.3, in a depth proportional to the contribution of this thesis to that particular layer.

### Physical Layer (PHY)

The physical layer is the foundation over which wireless networks are developed and the main focus of Chapter 4. It defines how bits are transmitted over the wireless links and, thus, the design of the antenna and the transceiver. In a WNoC, the PHY module will basically serialize processor messages, modulate the resulting bits at a given frequency much higher than the processor clock, and deliver the modulated signal to the antenna. The inverse operation is performed at reception.

The decisions taken at the PHY level basically concern the design of the antenna and the transceiver, which are affected by decisions such as the operation frequency or the chosen modulation and provide a given transmission rate, as well as the area and power footprint of the solution.

**Frequency Bandwidth and Antenna Design:** one of the first decisions is the frequency of operation, which largely determines the dimensions of the on-chip antenna. Due to the planar nature of a chip, it is reasonable to consider the employment of a patch antenna. Assuming this particular case, the width ( $W$ ) is comparable to a wavelength  $\lambda$ , while the length  $L$  must be approximately  $\lambda/2$ . Therefore, for a given operation frequency  $f$  we have an on-chip antenna of area

$$A_{ant} \approx \frac{c_0^2}{2\epsilon_{eff} f^2}, \quad (3.3.1)$$

where  $c_0$  is the speed of light and  $\epsilon_{eff}$  is the effective permittivity of the substrate. Although recent works have proposed to reuse the ground supply metallizations as the radiating elements of the antenna [143], we will count the area occupied by the antenna as an overhead.

In order to fulfill the bandwidth requirements  $B$  at such resonance frequency  $f_c$ , a conventional resonant antenna must yield a quality factor of  $Q \approx \frac{f_c}{B}$ . A high quality factor implies a sharper resonance, which leads to a better efficiency but a lower bandwidth overall. Also, maintaining a certain  $Q$  at higher frequencies leads to an equally higher bandwidth, reason for which it is generally held that it is easier to obtain high bandwidths at high frequencies. The use of printed dipole antennas and its different derivatives have been also widely proposed in the literature [93] and have distinct relations, albeit similar than for a patch antenna, between antenna size and radiation frequency.

The geometry of the antenna is another important design aspect as it determines, apart from the radiation frequency, the bandwidth and radiation pattern of the antenna. The WNoC scenario suggests the use of wideband antennas to achieve a very large transmission speed. This can be achieved by overlapping multiple resonances, as in a planar inverted-F antenna, or directly employing shapes that generate a wide frequency response, e.g. a bow-tie antenna. Finally, to leverage the inherent broadcast capabilities of the BoWNoC approach, we also need the antenna to generate an omnidirectional radiation pattern around the antenna. Patch antennas in general achieve a rather omnidirectional response, whereas dipoles can only radiate well perpendicularly to the antenna. To fix this, antenna arrays could be used as proposed in [144], yet at the cost of higher area.

**Modulation and Transceiver Design:** the modulation choice is the main determinant of the components required at the transmitter and receiver circuits, and basically defines the *spectral efficiency*  $S_E$  of the system, this is, how many *bps* are transmitted for each *Hz* of frequency bandwidth. Thus, the transmission rate  $R$  is:

$$R = B \cdot S_E, \quad (3.3.2)$$

where  $B$  is the frequency bandwidth of the link. Hence, transmission rates of a system can be scaled by either (a) increasing  $B$  at the expense of an area and power cost that is roughly linear up to a certain limit imposed by the technology, or (b) using a modulation with higher  $S_E$ , which may come with non-linear area and power costs due to the need of a complex devices. In all cases, RF circuit engineering optimization can help to reduce area and power without impacting upon the bandwidth. Technology evolution pushes the achievable frequencies, which basically reduces the area and increases  $B$ , but has more complex implications in power consumption mainly related to the maturity of the technology. Section 4.2.1 provides more details on this.

There are two broad families of modulations: Continuous Wave (CW) and Impulse Radio (IR). Conventional wireless communications mostly employ CW schemes, which

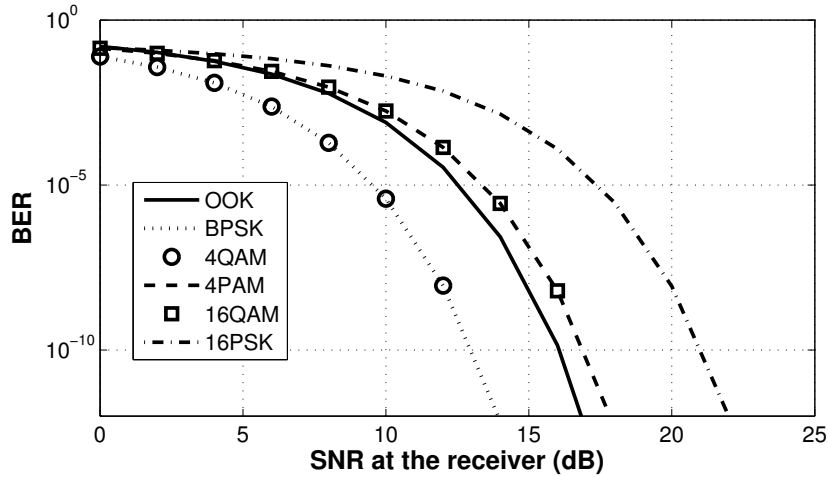


Figure 3.4: Theoretical BER as a function of the SNR for different well-known modulations.

basically change the characteristics of a carrier wave depending on the input. The amplitude, the phase or the frequency of the carrier can be modulated at different orders so as to achieve a given spectral efficiency. The alternative to CW is the IR paradigm consisting in the transmission of very short baseband pulses not bound to any carrier wave [145]. Information is modulated by means of the presence or absence of a pulse, or its position, whereas the length of the pulse determines the bandwidth of the resulting instantaneous signal. As we will see in Section 4.1.2, the choice between CW and IR has direct implications on the implementation complexity and power consumption of the scheme.

From the receiver perspective, the modulation has also a direct impact on the error rate of a given wireless communication system. One can distinguish between coherent and non-coherent receivers depending on its ability to detect the phase of the signal. Coherent receivers are typically more complex and power-hungry due to the need for sophisticated components such as the Phase-Locked Loop (PLL), but admit a wider variety of modulations that are typically robust against noise. To exemplify this, consider that wireless signals are received with a given Signal-to-Noise Ratio (SNR). The BER at the receiver will be:

$$BER \propto \mathcal{F}(1/SNR), \quad (3.3.3)$$

where  $\mathcal{F}$  is a function specific to each modulation and that does not have a closed form. As shown in Figure 3.4, coherent modulations like Phase-Shift Keying (PSK) or Quadrature Amplitude Modulation (QAM) will generally require a lower  $SNR$  to reach an objective  $BER$  than non-coherent modulations like On-Off Keying (OOK) or Pulse-Amplitude Modulation (PAM). Fixing the modulation family, one can increase the spectral efficiency at the expense of requiring a larger  $SNR$  to comply with a given error rate. In the end, however, the power consumption will be a combination of the  $SNR$  requirements and the power consumption of the transceiver components. We refer the reader to Section 4.1.2 and the literature [146] for more details.

**Link Budget and Channel Modeling:** as outlined above, there is a fixed statistical relationship between the power received and the BER. In order to know the power that needs to be radiated at the transmitter to ensure a given signal strength at the receiver and, thus, to guarantee a given error rate, we need to model the channel. A *link budget*

takes as inputs the antenna design, the distance between transmitter and receiver, and the operation frequency to evaluate the total attenuation introduced by the link. Thus, power can be safely allocated at the transmitter.

A very simple model is represented by means of the Friis equation, which relates the received power  $P_r$  with the transmitted power  $P_t$  with:

$$\frac{P_r}{P_t} = G_t G_r \left( \frac{c_0}{4\pi d f} \right)^2, \quad (3.3.4)$$

where  $G_t$  and  $G_r$  are the transmitting and receiving antenna gains,  $f$  is the transmission frequency,  $d$  is the distance, and  $c_0$  is the speed of light. Thus, one can coarsely assume that higher distances and higher frequencies negatively affect power. When scaling the frequency of a transceiver, this power deficit can be compensated by the inherently higher bandwidth of the scaled components. Note, in any case, that the on-chip scenario needs a new channel model accounting for its unique characteristics as described in Section 4.1.1.

Another aspect that greatly impacts on the link budget is the design of the antenna, which determine the gains  $G_t$  and  $G_r$  and their dependence with respect to the radiation direction. As mentioned above, broadcast-oriented designs advocate for omnidirectional patterns and, thus, rather constant gains over all directions. Other designs with dipole antennas or directive designs with higher gains at certain directions, though, need to take into consideration the orientation of antennas when calculating the link budget. The work in [147] proposes a methodology to optimize the link budget in this scenario.

## Medium Access Control (MAC)

The medium access control layer implements mechanisms to ensure that all nodes can access to the medium in a reliable manner. As we will study in Chapter 5, this plays a decisive role in determining the performance of any network as two simultaneous accesses to the same channel will fail and result into a waste of resources.

Figure 3.5 shows a rough classification of medium access mechanisms. *Channelization* generally consists of the fixed assignment of a number of channels to a number of nodes, whereas the rest of MAC protocols grant access to shared channels dynamically through the *coordinated* or *random* action of nodes. In static channelization and dynamic coordination schemes, collisions are avoided at the cost of a rigid and non-scalable channelization or the need of an –oftentimes centralized– arbitration mechanism. In random access cases, flexibility and simplicity are higher, yet at the cost of performance. The protocol needs to resolve collisions and make sure that transmissions are successfully completed, aspects that are addressed by means of acknowledging and retransmission policies.

Three metrics are generally employed to evaluate the performance of a given MAC protocol. First, the *latency* of the protocol 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. Adding all factors, the transmission latency  $t_L$  of a packet of length  $L$  through a link of data rate  $R$ , as calculated with Eq. (3.3.2), is:

$$t_L = \frac{d}{c_0} + \frac{L}{R} + t_{MAC}, \quad (3.3.5)$$

where the term  $\frac{d}{c_0}$  is the propagation latency,  $\frac{L}{R}$  is the transmission latency, and  $t_{MAC}$  is the MAC delay, including retransmissions, timeouts and acknowledgments. As we will see



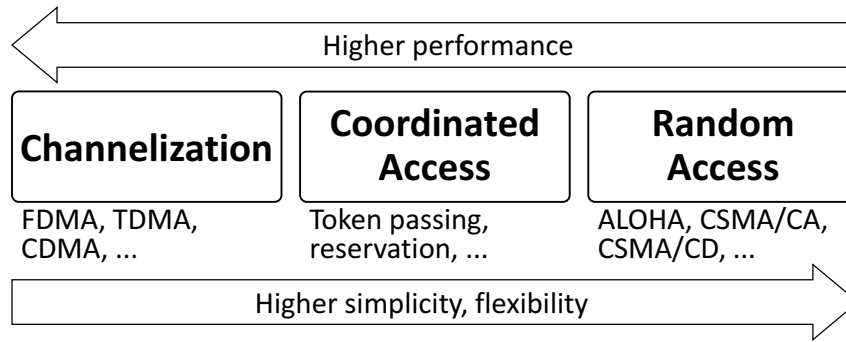


Figure 3.5: Coarse classification of multiple access mechanisms.

in Chapter 5, the MAC delay will be the main contributor to the overall delay in BoWNoC. Reducing this term is critical to maintain the latency advantage of the approach.

A no less important metric is the MAC throughput, which is calculated as the fraction of channel capacity used for transmissions:

$$\nu = \frac{\sum L_i}{T \cdot R}, \quad (3.3.6)$$

this is, the sum of the lengths of the successfully transmitted packets, divided by the total elapsed time  $T$  and normalized by the data rate of the channel. This metric does not have units and  $\nu \leq 1$ . Given that bandwidth is a precious resource in WNoC, the MAC protocol should maximize this metric without significantly increasing the latency.

Finally, a more subtle performance indicator of the MAC protocol is its *fairness*. Ideally, all packets should be treated equally leading to a uniform latency for the entire traffic. However, some protocols may not comply with this rule by giving preference to a certain type of traffic. In other cases, some nodes may unintentionally capture the channel by constantly transmitting before others can even attempt to transmit. Extreme unfairness leads to *starvation*, which generate huge delays and may cause deadlock situations.

### Network Layer (NET)

The network layer mainly deals with addressing, forwarding, and other aspects concerning the path from source to destination in a network. Note that, in our custom stack, we also consider transport layer functions such as congestion avoidance and end-to-end reliability to take place in the network level. In the particular on-chip scenario, the network layer functions impact directly on three basic concepts: load balancing, Quality of Service (QoS), and deadlock-freedom.

Load balancing concerns the distribution of the injected packets throughout the network according to the capacity of its components to maximize the throughput. In a NoC, load balancing is generally performed at the routers, carefully selecting the path to distribute the load. QoS refers to a set of priority rules enforced to achieve a certain latency or reliability requirements of certain communication flows. Finally, mechanisms at the network level are required to avoid deadlock conditions within the NoC, i.e. packets that form a cyclic dependency of resources so that none of them can advance.

Since a WNoC is generally intertwined with a conventional NoC to reduce the hop length, the NET layer of design modifies the routing algorithms taking into consideration the new wireless paths and re-evaluates load balancing, QoS, and deadlock conditions ac-

cordingly. In the BoWNoC vision proposed in this dissertation, though, network planes are separated and the wireless plane implements one-hop communication. This plane separation greatly simplifies the NET level procedures. There is no need to modify the existing rules at the wired plane; whereas, as we will see in Chapter 6, wireless plane rules can be directly implemented at the NIF that bridges the processor with the network. Load balancing and QoS boil down to a problem of traffic steering at the NIF, while deadlock freedom can be guaranteed as long as it is demonstrated at both planes separately.

### 3.4 Existing WNoC Architectures

Thus far, WNoC has been generally regarded as a valid option to complement a traditional wired NoC since the size of current on-chip antennas, i.e. in the millimeter range, does not allow to include one antenna per core. In this hybrid approach, wireless links have been generally employed for communicating distant locations within the chip in order to significantly decrease the average hop count of traditional NoC topologies. This is in contrast with the vision presented in this dissertation which, as detailed in Section 1.3, proposed the use of wireless on-chip communication to implement a scalable broadcast platform. Next, we provide a brief survey on WNoC design proposals found in the literature, explaining the main differences from the link, network, and architecture perspectives. We refer the reader to subsequent chapters for deeper analyses on each of the design layers.

Related works can be taxonomized considering the employed Medium Access Control (MAC) mechanism as main distinguishing aspect. First designs heavily relied on frequency multiplexing [96], which were later enhanced with basic time multiplexing or even code multiplexing schemes to scale to a larger number of cores and still avoid collisions [94, 97, 148]. Subsequent works also proposed the use of token passing arbitration [90], which represents a more flexible variant of time multiplexing and where only the core that possesses the token can transmit. Another mechanism that ensures an operation free of collisions has been recently proposed in [149], consisting of distributed protocol where nodes request access using tightly-synchronized code-multiplexed packets. Yet another arbitration strategy consists of a handshaking protocol that uses wired connections between neighboring nodes to arbitrated access [150]. Finally, it is worth noting that random access mechanisms have received less attention than the aforementioned collision-free approaches due to their poor performance when high loads cause recurrent collisions. To address this issue, Dai *et al* propose a slotted carrier sensing protocol with theoretically optimal persistence calculated *a priori* [151]. Others have proposed an adaptive scheme that switches between a carrier sensing protocol and a token passing scheme depending on the level of contention [152]. As we will see in Chapter 5, our vision differs from all related works by proposing the use of random access protocols augmented with knowledge of the traffic characteristics for unprecedented performance and flexibility.

From a network design perspective, we have seen that most existing proposals advocate for the use of wireless RF to implement long-range links within large NoCs. Within this approach, one can distinguish between two main groups depending on the positioning of the wireless links. Regular positioning –see Fig. 3.6(a)– has been extensively inspected [86, 95–97, 153] due to its apparent simplicity and potential modularity. Other works have relied on positioning algorithms that follow the principles of small-world networks –see Fig. 3.6(b)–, which minimize the hop count [90, 94] or maximize the beneficial use of the wireless network [154]. Regardless of the positioning strategy, all these works rely on the use of a

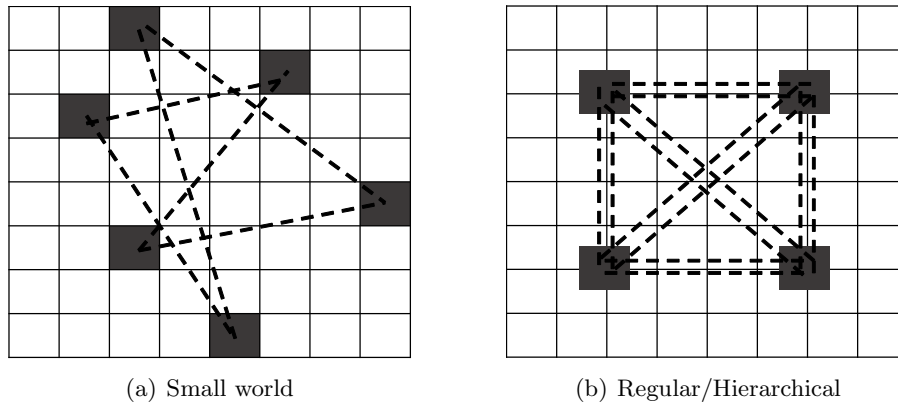


Figure 3.6: Rough classification of the topologies used in existing WNoC architectures. White and black squares represent tiles with routers and wireless interfaces, respectively.

conventional NoC for local communication, with the notable exception of the architecture proposed in [86], which employs a nanophotonic local network instead. Another set of proposals breaks away from what the aforementioned designs have in common, which is that a packet will perform a single wireless hop at most. A few efforts [144, 155, 156] have explored the multi-hop wireless design space both in standalone and overlaid configurations, investigating the trade-off between throughput and delay arising from the use of variable transmission ranges: shorter ranges allow for a greater frequency reuse but require more hops to reach the destination. Works in [144, 155] consider an homogeneous topology, whereas in [156] unequal RF nodes lead to an irregular topology. It is worth remarking that the emphasis on lower latency is shared by almost all works, oftentimes overlooking the inherent broadcast capabilities of WNoC. As it is stressed during the dissertation, our vision inherits the one-hop approach of most of the related works, but aims to emphasize the broadcast advantage of WNoC by integrating, if feasible, one antenna per core. This also allows the designer to completely separate the wired and wireless planes, simplifying backbone functionalities like load balancing. These aspects are discussed in Chapter 6.

From an architectural or application perspective, the different WNoC approaches have been evaluated for a wide variety of applications and configurations. Due to their technical simplicity and apparent completeness, unicast synthetic traffic patterns have been present in most of the WNoC evaluation frameworks [94–96, 150]. The execution time of benchmark suites has been also evaluated in papers that integrate the WNoC design within shared memory [153, 154] or message passing systems [157] in an architecture-agnostic manner. Other efforts have considered less widespread benchmark suites and investigated the impact of WNoC when executing applications that would better benefit from its uniquenesses. Specifically, graph analysis [158], bioinformatics [159] and the MapReduce framework [160] have been modified and re-evaluated taking the WNoC paradigm into consideration. Finally, the impact of WNoC upon the design of Dynamic Voltage and Frequency Scaling (DVFS) policies has been also investigated [154, 161], putting emphasis on the thermal hotspot and power reduction capabilities. As we will see in Chapters 6 and 7, our evaluations consider both synthetic traffic and the SPLASH-2 and PARSEC benchmarks suites, integrating our BoWNoC within both architecture-agnostic and architecture-oriented configurations.

## Chapter 4

# PHY: Towards Core-Level Wireless Communication

The physical layer of design defines the means of transmitting raw bits over a physical link interconnecting two nodes. In the specific case of wireless communications, the physical layer is concerned with the modulation, the coding, error control, and other methods that determine the data rate and error rate of the solution, as well as its area and power.

Given that area and power are two precious resources in the manycore chip scenario, the development of appropriate methods at the physical layer is essential to guarantee the viability of the BoWNoC paradigm. These physical constraints suggest the use of simple solutions leading to small and low-power transceivers. However, the scenario casts strong bandwidth and reliability requirements that point towards the opposite direction. The challenge is to find a graceful compromise between both extremes while exploiting the unique optimization opportunities given by the static and controlled on-chip landscape.

At the time of this writing, the wireless on-chip communication paradigm is yet in its nascent stages from the implementation perspective as few on-chip antenna, devices, and full transceiver proposals have been reported [101, 102]. These designs can be considered conservative, as they represent first iterations of antennas and transceivers conceived for the WNoC application. Developments for applications with similar requirements such as Wireless Sensor Networks (WSNs) [162] or Wireless Personal Area Networks (WPANs) [163] may be considered to find a scaling trend and extrapolate design guidelines for the BoWNoC paradigm. Similarly, few research works have modeled the on-chip wireless channel [93, 164] or discussed the use of coding techniques for reliability in WNoC [165].

Following the contribution of Chapter 3, which outlined the technologies and fundamental breakthroughs that have led to consider the WNoC paradigm in the manycore era, this chapter aims to demonstrate the viability of the approach by means of a scalability analysis of antenna and transceiver proposals at different frequency bands. We first review existing implementations at the mmWave band for wireless communications applications with similar requirements than WNoC, and use the data to create a scalability model that may predict the area and power of future chip-scale implementations. Then, we perform a more exploratory work at terahertz frequencies that considers the use of graphene-based antennas. Specifically, we create a theoretical framework that links the performance of graphene-enabled wireless communications with the nanoscale phenomena occurring at the graphene antennas and the terahertz channel. This methodology is then used to provide a first assessment of the potential of graphene antennas within the on-chip communication

context. We expect that, eventually, the investigations contained in this chapter will provide qualitative and quantitative guidelines for the design of future transceivers for wireless on-chip communication.

The remainder of this chapter is as follows. Section 4.1 lays the foundations of the PHY design for wireless on-chip communication by discussing the main differences between the chip scenario and traditional wireless networks, as well as how these impact upon aspects such as the wireless channel modeling or the development of coding and modulations. Section 4.2 delivers the analysis of the scalability of the area, power, and performance of chip-scale wireless communication in the mmWave band. Finally, Section 4.2 explores the viability of terahertz chip-scale communications using graphene antennas.

## 4.1 Design Principles, Objectives and Challenges

The PHY layer of a WNoC defines how messages coming from the processor are serialized, modulated, coded, and delivered to the on-chip antenna in transmission, and decoded, demodulated and deserialized in reception. Additionally, as we will see in Chapter 5, the PHY layer should also provide carrier sense or any other method to detect collisions, information that may be useful for the MAC protocol.

Due to evident physical constraints, the multiprocessor scenario shares a few design objectives with cellular or sensor networks. For instance, simplicity and low-power techniques are key to comply with the area and average power limitations that are found in conventional Chip Multiprocessors (CMPs) and that will be aggravated in the manycore era. However, the data rate requirements of such a communication-intensive environment are far more demanding from those of the aforementioned cellular or sensor networks. Reliability-wise, existing works advocate to providing a BER commensurate to that of RC wires to ensure the correctness of computations. Such requirement sets the BER around  $10^{-15}$ , orders of magnitude lower than in other applications. However, the emergence of new paradigms like approximate computing [166] may relax these constraints by allowing results to have a certain error margin.

All in all, the main challenge here is to provide a large bandwidth density and low error rate without having to resort to complex and power-hungry devices. This trade-off between complexity and performance drives the whole design and demands highly streamlined solutions. Fortunately, the chip scenario also offers favorable conditions for wireless communication mainly stemming from the static and controlled nature of the system. Also, since the chip is enclosed within a potentially isolated package, external interference impairments or regulatory limitations in terms of transmission power can be overlooked.

The BoWNoC paradigm, which embodies the main vision of this thesis, introduces subtle additions to the design objectives and challenges of the PHY layer. Remind that BoWNoC emphasizes the broadcast nature of wireless communication by potentially integrating one wireless unit within each core and sharing the same channel. This suggests the use of single broadband transceivers and omnidirectional antennas. Notwithstanding this, the availability of a small set of channels would be desirable from an architectural perspective. Also, variable-gain amplifier schemes have been proposed [167] to dynamically allocate power so as to save energy.

In what follows, we detail the fundamental design aspects of BoWNoC at the physical layer, including related work in the areas of channel modeling, modulations and coding.

### 4.1.1 On-chip Channel Modeling

A channel model that takes into consideration the peculiarities of the BoWNoC scenario is fundamental in order to evaluate the available on-chip communication bandwidth and to properly allocate power. The enclosed nature of chip processors causes the apparition of a large number of reflections that must be taken into consideration at the receiver. Additionally, the physical landscape of a multiprocessor involves multiple dielectric/metallization layers and components printed on the chip surface that may challenge the propagation of EM waves within the package [164].

One of the main uniquenesses of the scenario is that all these elements are fixed and known beforehand. This allows to have a better control of the propagation and leads to a time-invariant and quasi-deterministic channel model. Thus, processes depending on this channel model, such as power allocation or signal equalization, can be performed with unprecedented precision and effectiveness. Eventually, the analysis of the channel characteristics will help in the election and optimization of the modulation scheme.

At the time of this writing, few works have explored the on-chip wireless channel. First papers characterized the propagation within computing chassis below 10 GHz [168, 169], observing a path loss lower than in freespace propagation and with small variance despite the large number of components found inside a computer. They also observed that the orientation of antennas hardly affects the results. These conclusions are mostly due to the high multipath density, which might also ultimately limit the maximum data rate of the system unless they are sorted out with equalization techniques. Results from the analysis in [168] were used to reduce the interference caused by static multipath, thereby experimentally improving the BER of an IR modulation [170].

Some of the expertise gained in within-chassis modeling can be applied to the on-chip scenario. Few works have explored this case experimentally [93] or by means of simulation [171] considering printed dipole sources. Zhang *et al* confirmed that on-chip propagation has a similar path loss exponent that those of within-chassis environments, suggesting that propagation sums up the contribution of several paths. Their subsequent work in [172] used such channel characterization to demonstrate that Binary Phase-Shift Keying (BPSK) signals at the 22-29 GHz band could be transmitted at 3.33 Gbps with a BER lower than  $10^{-20}$  at a four centimeter distance.

Considering the general setting shown in Figure 4.1, radiated signals reach the receiver via three different paths [93]:

- First, surface waves propagate at the interface of the chip and the package medium. These waves show particularly low attenuation per unit of distance due to their cylindrical characteristics and are affected by the circuits printed on the chip surface. When using printed dipoles in a high permittivity substrate, surface waves are reported to become the dominant propagation mode as long as the ratio between the substrate thickness and the free-space wavelength is greater than 0.045 [93, 172]. However, the scenario also suggests the use of patch antennas with very low radiation efficiency in the coplanar direction [101]. In this particular case, surface waves are still generated but act more as an interference due to their reduced strength. Other works [140], in turn, advocate for the use of custom materials that favor the propagation of surface waves and reduce their attenuation (see Section 3.1.3 for more details).
- Second, part of the energy of patch antennas is radiated into the substrate. These waves are guided within the substrate and reach the receiver after repeated reflections

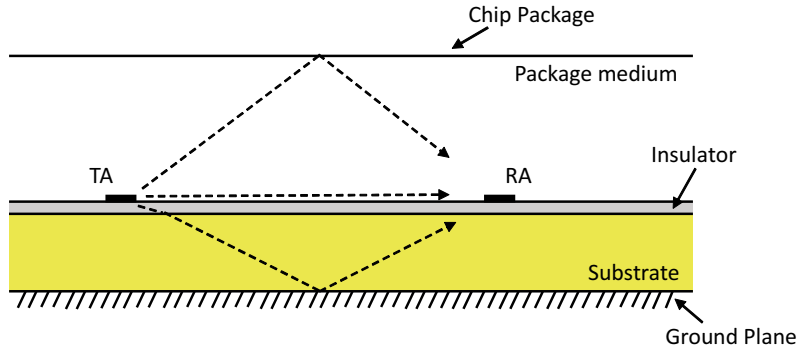


Figure 4.1: Channel modeling in the BoWNoC scenario: electromagnetic waves that may potentially reach the receiver.

upon the ground plane of the chip and the insulating layer. However, the substrate is generally lossy and introduces a very high attenuation per unit of distance that needs to be taken into account in the analysis.

- Given that surface and guided waves may be highly attenuated by the antenna and the substrate, respectively, we can also consider that communication occurs by means of a third mechanism: space waves that propagate through the medium and reflect upon the chip package and surface. This is specially true when using patch antennas.

Due to the potentially high bandwidth requirements of the scenario, which could lead to the use of wideband devices, channel modeling techniques in the time domain are more appropriate. This approach allows the use of ray tracing as a computationally reasonable methodology. This implies evaluating all possible rays reaching the receiving antenna for each and every pair of antennas within the chip [164]. The result in the time domain will then be a sum of channel impulse responses:

$$h(t) = \sum_i \alpha_i e^{j\phi_i} g_i(\tau - \tau_i), \quad (4.1.1)$$

where  $\alpha_i$ ,  $\phi_i$  and  $\tau_i$  are the amplitude, phase shift and delay of the  $i$ -th ray. These parameters will differ depending on the distance and the particular propagation mechanism on that path. This model should account for propagation losses, reflections, and other phenomena affecting the electromagnetic waves, and is reasonably accurate for frequencies up to the mmWave bands. If the analysis includes the terahertz band, though, we need to consider that propagation and reflections are frequency-dependent.

**Propagation:** models generally account for far-field propagation. To ensure that this condition holds, we need to check that the transmission distance is of at least a few wavelengths. Since this may not be true in some cases, the work in [171] advocates the need to account for near- or intermediate-field problems. Assuming that far-field conditions are met, propagation losses are mainly due to spreading and attenuation due to the interaction of the wave with the propagation medium. The first factor depends on the type of propagation mechanism, as the electrical field strength decay varies between  $O(1/\sqrt{d})$  and  $O(1/d)$  depending on the nature of the wave. The second term depends on the medium and its complex permittivity, as dielectrics generally introduce additional losses. Finally, note that as we approach to terahertz frequencies, additional effects such as molecular absorption or

particle scattering may need to be taken into consideration. We refer the reader to Section 4.3.2 for more details on this.

**Reflections:** the characteristics of reflected waves depend both on the roughness of the surface and on the reflective material. The effects of the former can be neglected for conventional metallic materials in the frequency range of interest [111], whereas the latter is polarization-sensitive and given by the Fresnel coefficients of the different media as outlined in Section 4.3.2. Working at mmWave bands, the refractive indexes are independent of frequency and, thus, the reflection can be considered specular. See Eqs. (4.3.10) and (4.3.11) in Section 4.3.2 for more details.

**Interferences:** the use of a chip package ensures that external interferences can be blocked with a high probability. This allows the use of frequency bands that may be licensed and used by other equipment nearby. Interference sources, however, can be internal as it has been proved that parallel and normal metal lines located between transmitter and receiver can affect propagation, mostly in cases where surface waves dominate [93]. Results show that, on the one hand, interferences can be constructive in cases where metal lines are laid out in a periodic manner, as this enhances the band pass characteristic of the propagation channel. On the other hand, antennas still need to be either placed far enough from inductors and coplanar waveguides or protected with isolation methods to minimize crosstalk [106]. Finally, care must be taken on avoiding that antennas pick up noise coming from the operation of nearby circuitry [173]. This is ensured by using a radiation frequency much higher than the frequency of operation of the processor.

### 4.1.2 Modulations

The area and energy figures of a transceiver not only strongly depend on the implemented modulation, but are also generally traded off against performance. Therefore, modulations are an important design step in the BoWNoC scenario, as a balance between area, energy and performance is sought.

Assuming the area overhead limitation as the most stringent design constraint, simple modulations are desirable since they often admit the use of low area and low power components. Yu *et al* exemplify this by comparing different CW transceivers for on-chip communication that implement coherent and non-coherent modulations. Coherent options, which require a complex PLL to work, take around 0.4 mm<sup>2</sup> of silicon area and consume around 3 pJ/bit; whereas the non-coherent alternative relaxes the complexity, achieving overheads of 0.25 mm<sup>2</sup> and 2 pJ/bit, at the cost of lower potential data rates. Besides demonstrating the difference between simple and complex schemes, these results also suggest that improvements are still needed to meet the predicted requirements of the scenario (see Table 3.1 in Chapter 3).

Another reason to advocate for simplicity is that scaling simple schemes in frequency (to reduce the size of passives and to increase the available bandwidth) is generally more feasible than with complex components. In this respect, several interesting features suggest that IR may be a strong contender in the BoWNoC scenario [174]. As in CW, simple modulations and non-coherent reception can be employed. To further reduce the complexity of the system, the integration of an on-chip oscillator can be totally avoided since most IR alternatives do not require any carrier wave. Moreover, IR allows to perform initial signal processing tasks in the analog domain, leading to sub-Nyquist sampling rates [175]. This aspect is critical since Nyquist sampling rates imply a need for power demanding



analog-to-digital converters able to operate at very high speeds. As the BoWNoC design is scaled towards the terahertz band to comply with the potential communication demands of manycore processors, the added simplicity features of IR are highly compelling. IR signals in the terahertz band could be achieved by using femtosecond pulses [176].

### 4.1.3 Coding

Due to the stringent area and power budgets, redundancy in the NoC context is introduced within the transmitted information rather than embedded directly in hardware. Thus, Error Control Coding (ECC) techniques have been recently explored to combat the increase of errors produced by power saving techniques in wired links [177]. This approach reduces the error probability by using detection or correction codes, in a process that increases the power consumption and reduces the effective data rate as a fraction of each message is devoted to the code. The coding and decoding of information also adds a certain latency. The choice of coding scheme will depend on its overheads, as well as on the maximum number of errors and their position, i.e., whether they occur in bursts or not.

As mentioned in previous sections, on-chip interconnects are designed to operate with a BER on the order of  $10^{-15}$ . This is a rather stringent requirement that may not be achieved by means of increasing the SNR only. Assuming a transceiver capable of distinguishing between collisions and errors due to propagation losses, Forward Error Correction (FEC) could provide an effective way to reduce the error probability. As in the case of modulations, simplicity should drive the design of the coding scheme.

In spite of the importance of this issue, research on coding for wireless on-chip communication has been scarce. Rahaman *et al* evaluate the use of simple Hamming codes over an intra-chip link using a Rician model for fading, demonstrating that a substantial coding gain can be achieved for high SNR values [178]. Ganguly *et al* propose the use of Hamming product codes instead, which basically code information twice to also detect bursty errors [165]. However, its use is poorly justified by the hypothetically high number of defects in the antennas leading to bursts of errors.

In light of the few proposals in this area, the BoWNoC field might benefit from the expertise gained in other wireless networks. For instance, previous works in wireless sensor networks have used knowledge on the channel, the modulation, and the overhead of the coding implementation to determine under which conditions coding makes sense [179]. A similar analysis could be performed in the on-chip scenario given the unprecedented level accuracy that can be achieved in the modeling of the channel. Other interesting works have culminated into the designation of Reed-Solomon (RS) codes with low-density parity check (LDPC) as the coding schemes of the 802.15.3c standard [163], which defines the physical layer of mmWave radio for high-rate WPAN networks. LDPC codes may be relevant to the BoWNoC interests as they are expected to provide the required simplicity and correction capabilities.  $RS(n, k)$  codes build codewords of length  $n$ ,  $k$  of which are data; the remaining  $2t$  bits are for parity check, allowing the correction of up to  $t$  erroneous symbols within the codeword. Assuming  $p$  to be the bit error probability considering a raw channel, the use of  $RS(n, k)$  codes reduces the BER to:

$$BER = 1 - \frac{(1-p)^n}{k} - \frac{n}{k}p(1-p)^{n-1}. \quad (4.1.2)$$

In addition to codes for ECC, BoWNoC systems employing IR modulations may benefit from the use of low-weight codes. When combined with the OOK modulation, such

technique reduces the average power consumption as it favors the transmission of zeroes, which are represented by the absence of signal [176].

Another interesting aspect to explore would be codes with *variable gain*. This would be well-suited to BoWNoC architectures that adapt to varying reliability scenarios. For instance, the use of DVFS within a multiprocessor may change the performance of the BoWNoC, forcing it to reconfigure the coding gain to avoid wasting energy or to compensate for higher error rates. This approach would also benefit approximate computing [166], wherein different applications may have different reliability requirements.

## 4.2 Scalability Analysis of mmWave Transceivers

Having stated the fundamentals of PHY layer design in the BoWNoC paradigm, the next step is to evaluate the validity of the approach from an implementation perspective. To this end, we need to analyze the area and power consumption of the antennas and transceivers required to realize the wireless on-chip communication. Given the novelty of the scenario and the application, the analysis should not focus on specific characteristics such as a given modulation or frequency band of operation. Instead, there is a need for a comprehensive study that explores a large solution space towards understanding under which conditions the BoWNoC paradigm is feasible.

In this section, we base our analysis on the scaling trends given in RF design for wireless communication and mentioned by related works in RF interconnects [61]. In a nutshell, component scaling leads to higher frequency of operation, which intrinsically implies a higher bandwidth density and energy efficiency. However, circuit design in frequencies beyond 60 GHz is exploratory and needs to be improved before it can be applied to the BoWNoC scenario. Also, higher attenuation due to spreading loss needs to be considered.

To assess whether the theoretical scaling trends can be extrapolated to the BoWNoC paradigm, we perform a circuit-oriented design space exploration that considers a heterogeneous set of transceiver proposals summarized in Table 4.1. Due to the scarcity of specific developments for the chip-scale communication field, we consider transceivers used in other applications and transmission ranges yet maintaining a high data rate as a primary requirement. The area and power results are then compared to those of representative electrical and nanophotonic alternatives. We expect that this benchmarked exploration will allow for a first identification of design *sweet spots* at different scales of core integration, and will provide design guidelines for the design of future transceivers for BoWNoC.

Table 4.1: Summary of the specifications of the analyzed transceivers

Technology	40 - 130 nm CMOS 130 - 250 nm SiGe BiCMOS
Transceiver Architecture	Impulse Radio (IR), Continuous Wave (CW)
Modulation	On-Off Keying (OOK), Amplitude Shift Keying (ASK), Phase Shift Keying (BPSK, QPSK), Frequency Shift Keying (FSK), Quadrature Amplitude Modulation (QAM)
Operation Frequency ( $f_c$ )	8 - 820 GHz
Transmission Range ( $d_{max}$ )	1.4 - 210 cm
Data Rate ( $R$ )	2 - 18 Gbps

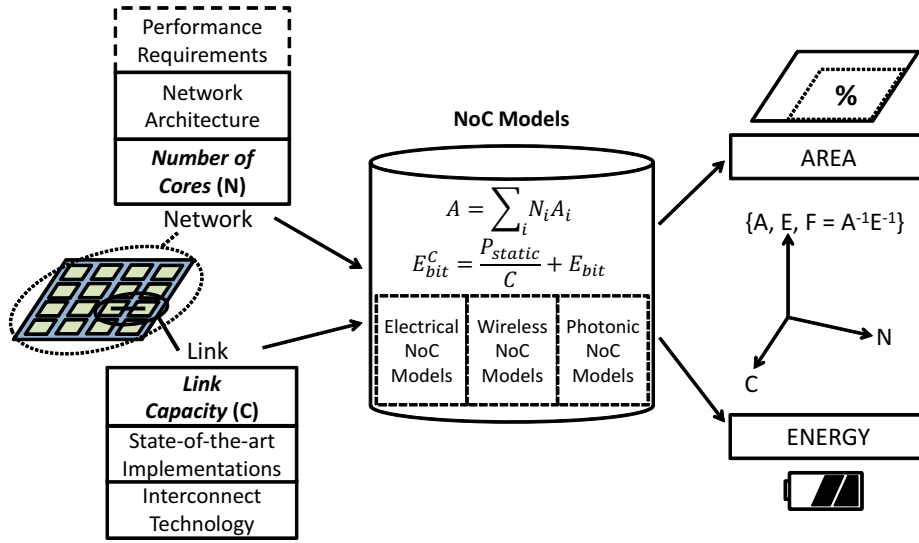


Figure 4.2: Model-based framework for physical layer design space exploration.

In what follows, we first give methodological details regarding the area and power models in Section 4.2.1 and then provide the results of the evaluation in Section 4.2.2.

#### 4.2.1 Evaluation Framework

Figure 4.2 outlines the methodology employed throughout this section. The investigation is entirely based on analytical models and compares how the area and energy consumption of a BoWNoC scales as a function of the size and bandwidth requirements of the network for a given architecture. To this end, we consider two variables, namely the number of effective receivers  $N$  and the capacity of a wireless link  $C$ .

We build a hypothetical network with  $N$  nodes and a certain topology depending on the chosen interconnect technology. In all cases, each link in the topology will have a bandwidth  $C$ . The exploration is aware of the huge difference in terms of network bandwidth between a BoWNoC, where all nodes share the same channel, and any conventional NoC where links are point-to-point. This difference is actually compensated by the use given to each network, as we expect the BoWNoC to carry broadcast traffic and the wired plane to carry unicast flows. Under this assumption and as we will see in Chapter 5, the throughput of a BoWNoC becomes commensurate to that of any wired NoC for a similar link capacity  $C$ .

The area and power models employed in the analysis, detailed in Sections 4.2.1.1 and 4.2.1.2, use state-of-the-art implementations to define the performance curves. In the specific case of BoWNoC, the area and energy can be easily expressed as a function of the number of involved nodes. However, it is not straightforward to assess how the area and energy of a wireless transceiver scale with its maximum achievable data rate (noted as  $C$  here) due to the number of factors involved. As detailed in Chapter 3, Eq. (3.3.2), the data rate depends on the transceiver bandwidth and the spectral efficiency of the selected modulation. The frequency band wherein the communication will take place imposes additional requirements on some components of the transceiver, the performance of which also depend on the maturity of the employed technology.

In order to extract a trend from the state of the art, a generally accepted approach is to represent the area or energy efficiency as a function of the data rate of different transceiver

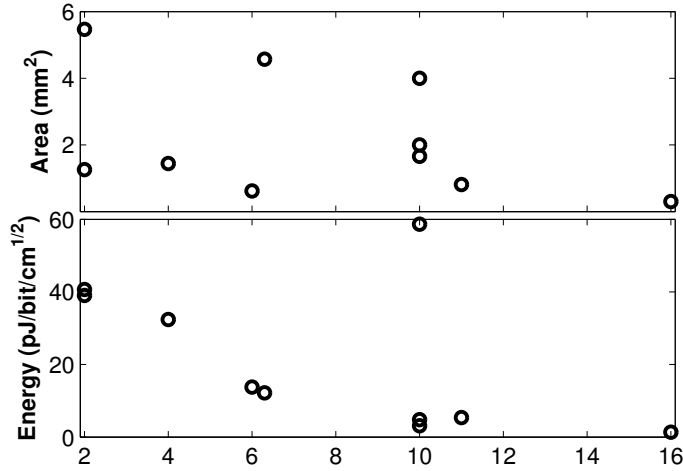


Figure 4.3: Area and energy of state-of-the-art wireless transceivers as a function of their data rate.

implementations, as done in Figure 4.3 for the works analyzed in [180]. However, the tendency shown by such scatter plots is unclear and only covers a range between 2 and 18 Gbps, rendering its extrapolation inadequate for the purpose of this work.

Thus, in light of the heterogeneity of the state of the art in the field summarized in Table 4.1, we will consider the following reasoning. Let us define the *Maturity Factor* as

$$M = S_E \cdot Q \text{ [bps/Hz]}, \quad (4.2.1)$$

where  $S_E = \frac{R}{B}$  and  $Q = \frac{B}{f_c}$  are the spectral efficiency of the modulation and the transceiver quality factor, respectively.  $B$  and  $f_c$  represent the frequency bandwidth and center frequency, whereas  $R = C$  is the data rate. Therefore:

$$M = \frac{R}{f_c}. \quad (4.2.2)$$

In summary, the maturity factor tries to evaluate the efficiency of implementing a given modulation and bandwidth in order to yield a target data rate operating at a target frequency band. As technology matures, we expect highly optimized transceivers leading to increasing maturity factors, this is, higher data rates for similar area and energy values. For a transceiver at a given operation frequency and with certain area and energy efficiency figures, we will *a priori* assume a maturity value in order to extract a projected data rate. This way, a rough estimate of the area and energy efficiency of future wireless transceivers can be obtained.

Figure 4.4 shows the maturity factor of several state-of-the-art transceivers (see [180] and references therein) as a function of their frequency. We observe factors of up to 35% at the 60 GHz band followed by a decrease below 5% when reaching sub-THz frequencies. These values will be used throughout this work as reference guidelines indicating the maturity of wireless transceivers at a given frequency. We will consider that initial designs could achieve a maturity factor of up to 10%, while refined implementations may reach a 20% and well-established transceivers could provide a 30%.

Eventually, the feasibility of the BoWNoC approach will be determined by the data rate requirements of the system. These could be met with current designs as transceivers

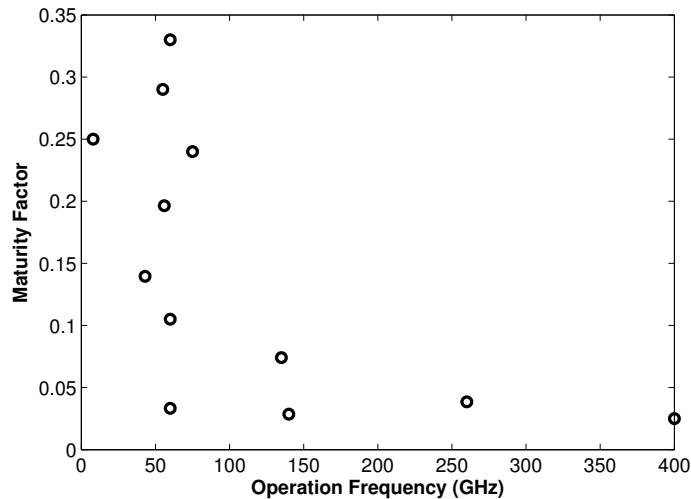


Figure 4.4: Maturity factor as a function of the operation frequency of state-of-the-art transceiver proposals.

with such performance have been already proposed [181]. Data rates up until 50 Gbps may be achievable in the near future provided that either technologies at 100-300 GHz mature and reach a reasonable factor of 20%, or initial designs appear in the terahertz band. In order to reach speeds above 50 Gbps, mid term efforts are required in order to raise the maturity of transceivers at in the terahertz band close to well-established levels. Section 4.3 delves in the possibilities of terahertz chip-scale wireless communication.

Next, we detail the models used for each of the considered interconnect technologies.

#### 4.2.1.1 Area Models

In order to calculate the area overhead of an on-chip interconnection network, we will use the following general expression:

$$A = N_{TX}A_{TX} + N_{RX}A_{RX} + N_LA_L + N_RA_R, \quad (4.2.3)$$

where  $N_i$  and  $A_i$  indicate the number of components of type  $i$  and its mean area occupancy, being the types divided in transmitters ( $TX$ ), receivers ( $RX$ ), links ( $L$ ), as well as routers, switches or other arbitration mechanisms ( $R$ ). To evaluate this equation, we need to detail the analytical models that relate the number of components and their area to the number of nodes of the network and the targeted link capacity in wireless, electrical and photonic NoCs. Note that the area figures will be independent of the traffic typology, as the considered NoCs are designed to support both unicast and multicast.

**Wireless NoC Area Models:** in the case of wireless on-chip communication, physical links are not needed in order to convey the information from the transmitter to the receiver. Moreover, switches or routers are not required if we assume one-hop communication. Therefore, the only components that occupy chip area are the antennas and the transceivers needed to modulate the data and to drive the signals to the antenna. We will assume one antenna and one transceiver per node, even though configurations with multiple antennas could be devised. Also, the analysis does not consider the area occupied by the logic required for the MAC protocol. For all this, Equation (4.2.3) can be reduced to:

$$A = N_{TX}A_{TX} + N_{RX}A_{RX} = N(A_{ant} + A_{txrx}), \quad (4.2.4)$$

where  $N$  is the number of nodes in the network,  $A_{ant}$  is the antenna area and  $A_{txrx}$  is the transceiver area. The antenna and transceiver area will be mainly determined by the on-chip communication requirements. In order to achieve a given goal, the wireless plane must provide a certain effective network throughput which depends on the MAC protocol that arbitrates medium access and, more importantly, the data rate of each transceiver. Generally, higher data rates require higher bandwidths which, in turn, require communication in higher frequency bands.

Such tendency fortunately imposes a downscale on the antenna size. Due to the planar nature of a chip, we will consider the employment of patch antennas. The area occupation of such antennas is inversely proportional to the square of the resonance frequency as expressed in Eq. (3.3.1) in Chapter 3. Although recent works have proposed to reuse the ground supply metallizations as the radiating elements of the antenna [143], we will assume that the area occupied by the antenna is an overhead.

The relation between the area and peak data rate of a transceiver is calculated as discussed in Section 4.2.1: a given maturity factor  $M$  is assumed so that the data rate requirement  $R$  can be achieved by operating at least at a frequency  $f_c = \frac{R}{M}$ . The area for such transceiver can be extrapolated with data from the state of the art, which points towards a decrease in area when the frequency is upscaled as shown in Figure 4.5 for the transceivers analyzed in [180]. The reasons for the observed tendency may stem from the strong downsizing that is applied to the passive RF components of a transceiver when the operation frequency is increased. On the other hand, the scaling of active RF components remains unclear and should be inspected in future work with the aim of obtaining an accurate area scaling model for wireless on-chip transceivers. In this work, we will use a model obtained by applying fitting methods to the data represented in Figure 4.5, which yielded the following equation:

$$A_{txrx} = \frac{206.1}{f_c + 27.22} [\text{mm}^2], \quad (4.2.5)$$

wherein  $f_c$  is expressed in GHz. Rational fitting was chosen on the grounds that it delivers the most accurate result among the possible fittings and that it does not yield negative values for high frequencies. The weight of each data point is assigned in inverse proportion to the operation frequency, implying that implementations for well-established technologies at low frequencies are more representative than initial designs at the terahertz band. The resulting coefficient of determination, which evaluates the goodness of fit, is 0.68 (with 1 being an exact fit).

**Conventional NoC Area Models:** two steps have been performed in order to calculate the area of an electronic NoC. First, the number of elements that constitute a given architecture can be easily derived by observing how its topology scales with the number of nodes. Once the topology is fixed, the area of each element can be calculated by means of simulation taking into consideration the topology and the target capacity. Our analysis has been performed by means of ORION, a widely recognized power-area simulator for on-chip interconnection networks [182].

Let us assume that each node has two line drivers, one for transmission ( $TX$ ) and one for reception ( $RX$ ). A typical line driver accounts for an inverter and a D flip-flop, and ORION allows the user to calculate their area occupancy for a given technology node. In

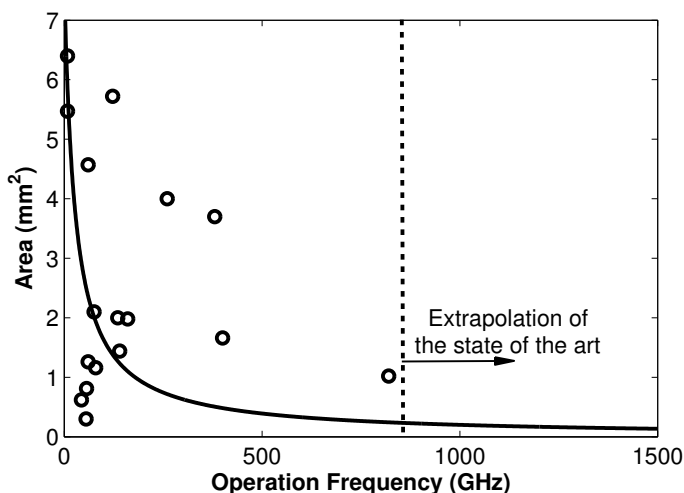


Figure 4.5: Area of state-of-the-art wireless transceivers as a function of the frequency.

the case of the on-chip wires, ORION evaluates the number of repeaters needed for each link ( $L$ ) based on its length (which is determined by the topology) and technology node. The area of each repeater is then calculated, added to the physical area of the wire and multiplied by the number of parallel wires in a link, i.e. datapath width. Finally, the chip area of each router ( $R$ ) is assessed by breaking the router down to the transistor level, calculating the number of transistors needed and multiplying it by the size of a transistor for a given technology node. The final result will depend on parameters such as the number of ports, the size of the buffers or the datapath width.

**Photonic NoC Area Models:** a photonic on-chip network essentially includes modulators, waveguides, switches, filters and photodetectors. In the transmitting side ( $TX$ ), we will assume that modulators are made of one active ring resonator, whereas receivers ( $RX$ ) consist of a passive ring resonator-based filter and a photodetector. Switches can also be devised by employing ring resonators as building blocks [72]. Finally, we also consider that all ring resonators are of the same size. Given these assumptions, the area of a given architecture can be approximated as:

$$A \approx N_{ring}A_{ring} + N_{det}A_{det} + \sum_i A_{wg,i}, \quad (4.2.6)$$

where  $N_{ring}$  and  $N_{det}$  are the number of ring resonators and photodetectors, respectively.  $A_{ring} = W_{ring}^2$  is the area of each ring, or the square of its pitch,  $A_{det}$  is the photodetector area, and  $A_{wg,i}$  is the area of waveguide  $i$ . As in conventional electronic NoCs, the specific network architecture will determine the exact number of components as a function of the number of nodes and the target link capacity. For instance, links might require the employment of  $W_D$  parallel waveguides to accommodate a given number of wavelengths

$$W_D = \left\lceil \frac{N_\lambda}{N_{\lambda,MAX}} \right\rceil, \quad (4.2.7)$$

where  $N_\lambda$  is the number of wavelengths of the network and  $N_{\lambda,MAX}$  is the maximum number of wavelengths. The  $\lceil \cdot \rceil$  operator rounds the result upwards to the nearest integer. It is important to note that  $N_\lambda$  may depend on the number of cores of the network [79] or the targeted network capacity [80].

### 4.2.1.2 Energy Models

The power consumed by any communication network can be classified in two main groups: static and dynamic. The static or zero-load power is the energy consumed independently of the traffic being served, whereas the dynamic power is a load-dependent component. Due to their distinct nature, static and dynamic powers are usually expressed in different units. Static power  $P_{static}$  is expressed in Watts and gives insight about the energy that is consumed invariably through time to, for instance, maintain the circuitry active; whereas dynamic power  $E_{bit}$  is expressed in Joules per bit and gives insight about the energy required to physically transmit one bit of data without errors from the transmitter to the intended receivers for a given interconnect technology.

As a rule of thumb, we will calculate the power consumed by a given on-chip network by using the following formula:

$$P = P_{static} + E_{bit} \cdot T, \quad (4.2.8)$$

where  $T$  is the network throughput in bits per second. In a reverse process, we can also calculate the energy required to convey one bit of information from the transmitter to the intended receivers, operating at a given throughput

$$E_{bit}^T = \frac{P_{static}}{T} + E_{bit}, \quad (4.2.9)$$

where the throughput  $T$  is ideally equivalent to the link capacity considering one transmission flow and no packet loss.

**Wireless NoC Energy Models:** unlike in traditional wireless networks, the network nodes in a WNoC are integrated within the same platform and share the same power supply. Moreover, we will assume one shared channel and enough transmission power so that each wireless message is received by all the processing cores. In this context, the energy consumed in the transmission and reception of one bit is independent of whether the message is unicast or multicast and can be expressed as:

$$E_{bit,W}^T = E_{bit}^{tx} + N \cdot E_{bit}^{rx}, \quad (4.2.10)$$

where  $E_{bit}^{tx}$  and  $E_{bit}^{rx}$  are the mean energy consumption in transmission and reception, respectively. Leakage currents of the  $N - 1$  inactive transmitters, as well as the power consumed by the logic required to implement the MAC protocol are neglected. For a transceiver implementation with measured power in transmission  $P_{tx}$  and measured power in reception  $P_{rx}$ , both for a data rate  $R$  and a given transmission range, the equation above can be also expressed as

$$E_{bit,W}^T = \frac{P_{tx} + N \cdot P_{rx}}{R}. \quad (4.2.11)$$

It is important to remark that Equation (4.2.11) expresses the energy per bit of a specific wireless transceiver yielding a data rate  $R$ . As discussed above, we will assume a maturity factor  $M$  so that a target peak data rate  $R$  can be achieved by operating at least at a frequency  $f_c = \frac{R}{M}$ . This way, a generic trend can be extracted from the state of the art in wireless transceivers. Authors in [104] propose and discuss a figure of merit for wireless transceivers which encompasses both their energy efficiency  $E_{bit}$  and transmission range  $d_{max}$  by means of the following expression:  $\Phi = \frac{E_{bit}}{\sqrt{d_{max}}}$ . Figure 4.6 shows how this



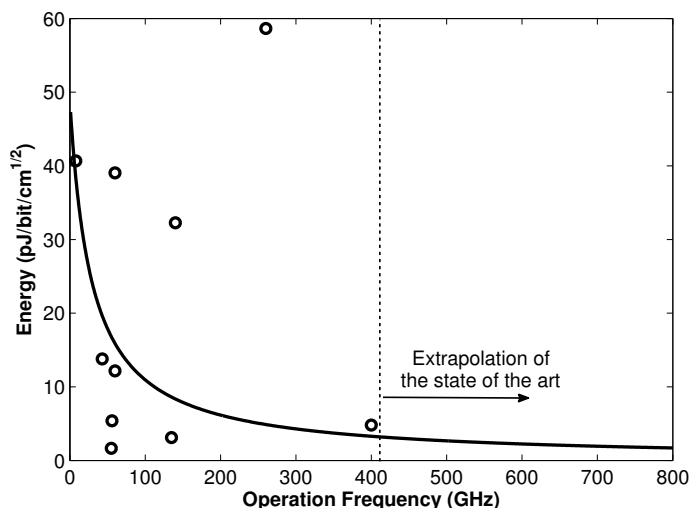


Figure 4.6: Energy efficiency figure of merit of state-of-the-art wireless transceivers as a function of their central frequency.

figure of merit scales as a function of the frequency for implementations analyzed in [180]. A similar fitting approach than the used in Section 4.2.1.1 provided the following relation

$$\frac{E_{bit}}{\sqrt{d_{max}}} = \frac{1.41 \cdot 10^3}{f_c + 28.81} \text{ [pJ/bit/cm}^{1/2}] \quad (4.2.12)$$

with a coefficient of determination of 0.65. In this case,  $E_{bit} = E_{bit}^{tx} + E_{bit}^{rx}$  and  $f_c$  is expressed in GHz. Energy values can be extrapolated for frequencies beyond 400 GHz using the equation above.

The dependence on the transmission range is an important aspect to consider since, under the assumption that any transmitter should be able to reach any receiver, the nodes located at the chip edges will need a higher range than that of more centric nodes. This has two main implications: on the one hand, centric nodes need less transmission power to fulfill the sensitivity requirements at the chip edges. Therefore, the power amplifier can be tuned to consume less power. On the other hand, centric nodes receive transmissions with high power since the link budget is performed considering the worst case, this is, to reach the chip edges. In this case, the requirements for the low noise amplifiers are significantly relaxed. In our analysis, we will calculate which is the average energy per bit over all the on-chip transmitters following the aforementioned considerations with static power allocation.

Finally and unless noted, we will assume  $E_{bit}^{tx} = E_{bit}^{rx} = E_{bit}/2$ .

**Conventional NoC Energy Models:** again, ORION is employed to determine both the static and dynamic power of an electronic NoC. In the former case, we will consider the power due to leakage currents in wires and routers. ORION breaks down these digital circuits to the transistor level and uses experimentally-validated values for quiescent currents. In the latter case, ORION provides means to calculate the energy required to perform one hop within the network, which includes the energy required to (1) transmit one bit of data through an on-chip wire of fixed length and (2) read one bit of data from a router buffer, route it and write it into the next router buffer.

Assuming a throughput  $T$  equal to the link capacity  $C$ , the energy per bit in an electronic NoC is:

$$E_{bit,E}^T = \frac{P_{leakage}}{C} + H \cdot E_{b,hop}, \quad (4.2.13)$$

where  $P_{leakage}$  is the power due to leakage currents and  $E_{b,hop}$  is the average energy required for one bit to perform one hop. The  $H$  is the average distance between transmitter and receiver in terms of number of hops and solely depends on the network topology. For a 2D Mesh of  $N$  cores,  $H_{ucast} = \frac{2\sqrt{N}}{3}$ , whereas  $H_{bcast} = N - 1$  considering a routing algorithm that minimizes the number of hops needed to deliver the message once to all the destinations.

**Photonic NoC Energy Models:** the power consumption in a photonic NoC is mainly driven by three components, namely, the laser power, the ring heating and the energy required to perform the electrooptic (E/O) and optoelectric (O/E) conversions at the modulators and photodetectors, respectively.

*Laser Power:* Since integrating individual laser sources on a chip is currently unfeasible, it is generally accepted that light in a photonic NoC is supplied by an external multi-wavelength source. This light is coupled, modulated and then guided within the chip towards the intended receiver. In order to fulfill the sensitivity requirements at the receiver, the laser must transmit enough power to compensate for the losses incurred by the components found in the light path. Moreover and unless practical real-time laser management systems are made available [75], the laser power needs to be statically allocated to the worst case scenario. In this context, a power budget analysis is performed for each wavelength  $j$ :

$$P_{lsr,j}(dBW) = \left[ S_{RX}(dBW) + \sum_i L_i(dB) \right]_j, \quad (4.2.14)$$

where  $P_{lsr,j}$  is the on-chip laser power needed for wavelength  $j$ ,  $S_{RX}$  is the receiver sensitivity, and  $L_i$  is the loss of component  $i$  including both the laser and coupling efficiencies. The network size, the topology, and the target link capacity will determine the number of components in the critical path of the light at that particular wavelength. Finally, the laser power is obtained by adding up the laser power for each of the required wavelengths.

*Ring Heating:* Another source of static energy in photonic NoCs is the power needed to maintain ring resonators tuned to the desired frequency. Such components are extremely sensitive to temperature as small variations produce a shift in their resonant frequency. The power needed to keep ring resonators thermally tuned is:

$$P_{heat} = N_{ring} \cdot P_{ring}, \quad (4.2.15)$$

where  $N_{ring}$  is the number of ring modulators in the architecture, and  $P_{ring}$  is the power needed to maintain one ring finely tuned (see Table 4.2). As commented in Section 4.2.1.1, we will assume one ring per modulator and filter in all cases.

*E/O and O/E Conversions:* The dynamic power consumption in a photonic NoC is mainly due to the energy required to convert one electronic bit to light and viceversa. In this case, we will consider fixed values demonstrated in the literature, which are shown in Table 4.2. Similarly to wireless NoC, the energy required for the transmission and reception of one bit will depend on the number of  $k$  simultaneous receivers:

$$E_{bit} = E_{bit}^{tx} + k \cdot E_{bit}^{rx}. \quad (4.2.16)$$

The parameter  $k$  is generally dependent on the photonic NoC architecture. Generally, point-to-point ( $k = 1$ ) optical communication is implemented and a separated broadcast channel ( $k = N$ ) is employed for multi-receiver transmissions [80]. Alternatively, a broadcast-based architecture would deliver any message to all the receivers, which would check the destination address and discard the message if necessary [79].

Assuming a throughput  $T$  equal to the link capacity  $C$  and using Equations (4.2.14)-(4.2.16), the energy per bit in a photonic NoC is:

$$E_{bit,P}^T = \frac{P_{laser} + P_{heat}}{C} + E_{bit}. \quad (4.2.17)$$

### 4.2.1.3 Investigated Architectures

We compare a selection of architectures that attempts to cover a representative design space both in terms of network topologies and interconnect technologies. The architectures are:

**EMesh** We consider a conventional electrical mesh as the baseline due to its simplicity and regularity. We assume one 5-port router per core and bidirectional links connecting neighboring routers. The datapath width of the links and routers is adjusted to comply with the capacity  $C$  requirements. This same architecture is considered as the baseline throughout the dissertation (see Chapters 5 to 7).

**WMesh** The BoWNoC paradigm is represented by an architecture accounting for one communication unit (antenna and transceiver) per core. We assume that all cores share the same broadband channel, wide enough to sustain a capacity  $C$ .

**OBUS** A photonic bus arbitrated by means of an all-optical token-based scheme. This is the scheme used in the CORONA architecture to carry broadcast flows [80] as explained in Section 2.4.

**OXBar1** An optical crossbar, wherein each core is tuned to a unique wavelength in transmission and broadcasts its messages to the rest of cores. This is basically the scheme depicted in [78, 79] as outlined in Section 2.4.

**OXBar2** Another optical crossbar, wherein each core is associated to a unique data waveguide. Through this dedicated channel, a given core is able to receive data modulated by any of the other cores. This corresponds to the main scheme of the CORONA architecture [80] explained in Section 2.4.

Table 4.2 shows a summary of the technological parameters used in the study. Note that the variable *number of cores*  $N$  is swept between 4 and 1024, whereas the *link capacity*  $C$  is scaled up to 250 Gbps. The upper limit corresponds to the link capacity of a link capable of transmitting one 128-bit flit per cycle considering a processor clock of 2 GHz. Reducing the flit size or processor frequency –as expected in manycore environments– would greatly reduce this data rate requirement.

## 4.2.2 Performance Evaluation

### 4.2.2.1 Area

Figure 4.7(a) shows the area-network size plane of the design space, corresponding to fixing the link capacity to a value of 80 Gbps. The electrical and wireless options show a linear

Table 4.2: NoC Parameters

Parameter	Value	Unit	Ref.
System			
System Size	16–1024	cores	-
Chip Area	400	mm <sup>2</sup>	-
CMOS Technology Node	32	nm	-
Operation Frequency	2	GHz	-
Supply Voltage	1	V	-
Link Capacity	10–250	Gbps	-
Photonic NoC			
Ring Pass Loss	0.01	dB	[183]
Ring Drop Loss	1	dB	[184]
Ring Area	64	μm <sup>2</sup>	[183]
Ring Heating Power	26	μW/ring	[185]
Waveguide Pitch	2	μm	[71]
Propagation Loss	0.5	dB/cm	[71]
Bending Loss	0.15	dB	[71]
Wavelengths per Waveguide	64	-	[186]
Datarate per wavelength	10	Gbps	[186]
Splitter Excess Loss	0.04	dB	[70]
E/O Conversion	82	fJ/bit	[73]
O/E Conversion	50	fJ/bit	[73]
Photodetector Area	20	μm <sup>2</sup>	[79]
Photodetector Sensitivity	-30	dBm	[74]

behavior, while photonic NoCs grow with the square of the number of cores due to the quadratic scaling in number of components.

In the WNoC case, three different operation frequencies have been chosen, namely 260, 400 and 800 GHz. Taking into account the targeted link capacity, such frequencies lead to maturity factors not exceeding 30%, in consonance with the values shown in the state of the art (see Fig. 4.4). From an area overhead perspective, high frequencies are beneficial since they entail lower area both for the antenna and the transceiver, according to the tendency pointed out in Section 4.2.1.1. Nevertheless, the area occupation in most cases is higher than that of the electrical and photonic alternatives. Considerable transceiver area optimization is needed in order to enable size compatibility with massive multicore architectures: reducing the area of a 800-GHz transceiver to 0.1 mm<sup>2</sup> would yield an overhead of 27% in a 1000-core processor. By employing graphene-based nano-antennas [124, 125], such area overhead would be further reduced to a 25%.

Figure 4.7(b) shows the area-capacity plane of the design space, corresponding to fixing the network size to a value of 256 nodes. It can be observed that both electronic and photonic NoCs show a linear growth of area with respect to the link capacity, since higher bandwidth requirements are generally fulfilled by means of additional wires and circuitry.

In the wireless case, we consider different preset maturity factors and then scale the operation frequency in accordance with the link capacity objectives. Once the operation frequency is chosen, the area is calculated using the model presented in Section 4.2.1.1. Such approach explains the negative slope of the WNoC area plots: higher bandwidth requirements imply an increase in the operation frequency, which in turn entails a reduction in the size of both the antenna and the transceiver. Due to this, it is expected that WNoC will be able to compete with the electrical and photonic alternatives at high link capacities due to the extremely high operation frequencies required for transmission. It is important

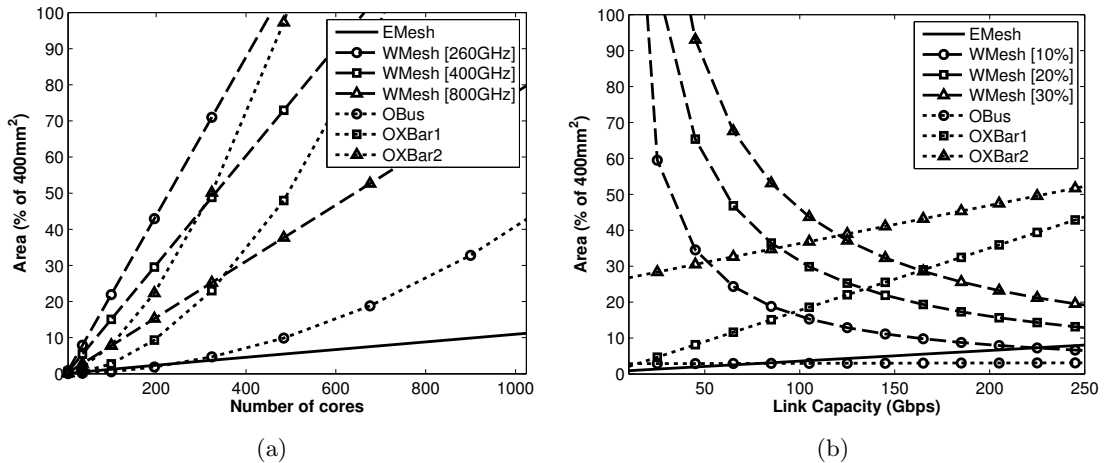


Figure 4.7: Area scaling for different interconnect technologies and architectures as functions of (a) the number of cores with  $C = 80\text{Gbps}$ , and (b) the link capacity with  $N = 256$ .

to note, though, that such possibility is limited by the state of technology as it determines the maximum frequency at which circuits can operate. This may also imply that higher maturity factors may need to be sought in order to increase the link capacity of a WNoC over a given value.

#### 4.2.2.2 Energy

Figure 4.8(a) shows the energy-network size plane of the design space for a fixed link capacity of 80 Gbps. There are several aspects to be noted:

- In a conventional NoC, there is a considerable gap between the energy per bit in a unicast transmission and in a broadcast transmission. In both cases, conventional designs outperform wireless and photonic NoCs.
- WNoCs follow a similar trend than conventional NoCs, being the options working at higher frequencies closer to achieve an energy efficiency comparable to that of conventional NoCs, in accordance to the extrapolation proposed in Figure 4.6.
- In a photonic NoC, the energy figures can be considered independent on whether the transmission is unicast or multicast by virtue of the extremely low energy needed for the O/E conversions. However and despite such potential for low energy transmissions, the photonic NoC configurations scale poorly due to their high laser power requirements, specially at high core counts.

Figure 4.8(b) shows the energy-capacity plane of the design space. On the one hand, it is observed that conventional NoCs yield an energy efficiency which is almost invariant with respect to the link capacity. On the other hand, the energy efficiency of WNoCs not only improves with the link capacity, but also outperforms conventional NoCs at some point, provided that the trend observed in the state of the art continues in future transceivers (see Figure 4.6). Finally, our results confirm that the different photonic NoC options do not scale well, as their efficiency substantially deteriorates for high link capacities. This is mainly due to the steep increase in number of components leading to an extremely high accumulated loss and, eventually, to unaffordable laser power requirements.

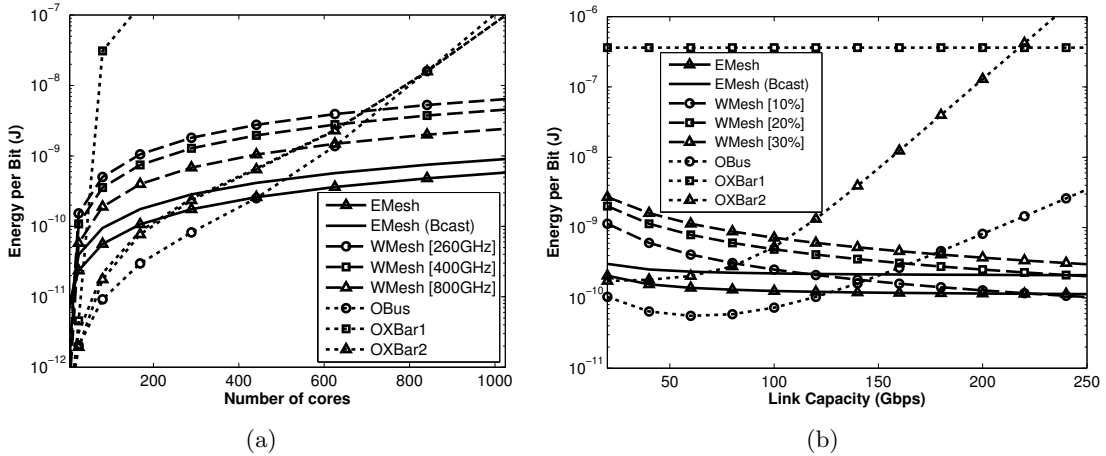


Figure 4.8: Energy per bit scaling for different interconnect technologies and architectures as functions of (a) the number of cores with  $C = 80\text{Gbps}$ , and (b) the link capacity with  $N = 256$ .

#### 4.2.2.3 Area-Energy Figure of Merit

As seen in the previous sections, a given on-chip network may scale remarkably well in terms of area and perform poorly in terms of energy, or vice versa. In order to evaluate both the area and energy scalability of each solution, we propose the following figure of merit:

$$FoM = \frac{1}{A \cdot E_{bit}^T} \text{ [bits/J/mm}^2\text{]}. \quad (4.2.18)$$

Such performance metric can be understood as the average number of bits that can be effectively transmitted for (a) each consumed joule of energy and (b) square millimeter of chip real estate. It is therefore an indicator of the joint energy and area efficiency of a given on-chip network. A large value of this figure of merit is desired.

On the one hand, Figure 4.9(a) shows how the figure of merit scales as a function of the network size in number of cores. Again, electrical and wireless NoCs show a similar trend, while a rapid decrease of the figure of merit is observed in photonic NoCs. Overall, conventional NoC yields the best performance. On the other hand, Figure 4.9(b) shows how the figure of merit scales as a function of the link capacity, in a network consisting of 256 cores. In this case, the analysis is slightly more complex. While it is clear that the optical crossbars scale poorly with the link capacity, the rest of options yield similar performance. According to our analysis, the optical bus shows the best performance for low link capacities, whereas wireless NoCs could yield an improved efficiency for high link capacities if the scaling trends observed in the state of the art continue.

#### 4.2.3 Discussion

Results here presented indicate that, in absolute terms, the baseline NoC performs remarkably better than its potential alternatives in terms of area and power. However, it is important to note that the technologies employed for electrical on-chip wires and routers is thus far much more optimized than nanophotonic or wireless chip-area technologies, which are still in their infancy and may substantially improve in the following years.

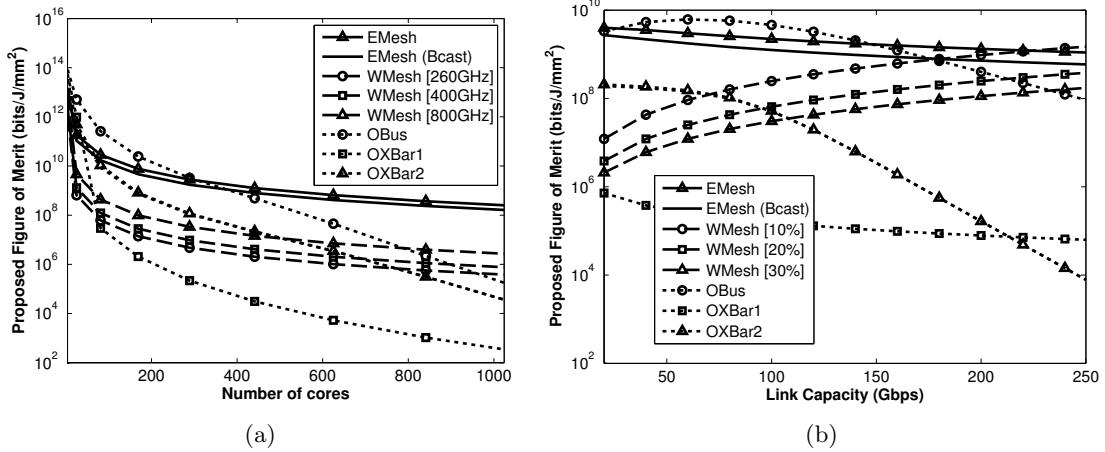


Figure 4.9: Scaling of the proposed figure of merit (higher is better) for different interconnect technologies and architectures as functions of (a) the number of cores with  $C = 80$  Gbps, and (b) the link capacity with  $N = 256$ .

For the sake of fairness, the comparison must account for the structural tendencies rather than for the absolute area and energy values. Table 4.3 summarizes the trends obtained through the application of fitting methods to the area and energy plots. In the table,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ ,  $\epsilon$  are constants. The poor scalability of nanophotonic options  $-O(N^2)$  and  $O(k^N)$ – is mainly due to the need of  $N$  rings for each of the cores of the system. Contrarily to the nanophotonic options, **WMesh** offers a good area and energy scalability with respect the number of nodes and an excellent scalability with respect to the link capacity. From this, we can infer that the concept of **BoWNoC** is better suited to the case of high data rate requirements leading to a very high radiation frequency. Conversely, in small networks working at lower speeds, electrical and photonic interconnects are expected to offer improved area and energy efficiencies.

#### 4.2.3.1 A comparison between state-of-the-art designs

To provide an approximation of the area and power of transceivers that may be available in the short term, we apply the scaling trends obtained in previous sections to a state-of-the-art design for on-chip communication [102]. Then, we compare these figures with those of recent NoC implementations.

We part from the 65-nm CMOS designs by Yu *et al.* at 60 GHz [102]. A first implementation uses OOK and achieves 16 Gbps with a bit error rate of  $10^{-15}$  while taking 31.2 mW of power and  $0.25 \text{ mm}^2$  of silicon area including an antenna of  $0.02 \text{ mm}^2$  [90]. The same authors increase the data rate by tripling the bandwidth for a total of 48 Gbps, while consuming 97.5 mW and  $0.73 \text{ mm}^2$ . They also explore the possibility of using the Quadrature Phase-Shift Keying (QPSK), which doubles the spectral efficiency with respect to OOK for a total of 32 Gbps assuming the original bandwidth. Due to the need of complex components, the power and area escalate up to 96 mW and  $\sim 0.4 \text{ mm}^2$ , respectively.

We scaled these three transceivers from 65-nm to 22-nm to obtain reasonable design points in the short term. To this end, we chose to be conservative and used scaling trends less favorable than the  $O(1/f)$  tendency pointed out in Sections 4.2.1.1 and 4.2.1.2. Specifically, we considered a sublinear scaling in terms of area given that active components do not

Table 4.3: Dominant area and energy scalability trends

Architecture	Area	Energy
<b>EMesh</b>	$O(NC)$	$O(N)$
<b>WMesh</b>	$O(N/C)$	$O(N/C)$
<b>OBus</b>	$O(N^2)$	$O(\alpha^N \beta^C)$
<b>OXBar1</b>	$O(N^2C)$	$O(\gamma^N)$
<b>OXBar2</b>	$O(N^2C)$	$O(\delta^N \epsilon^C)$

necessarily scale with technology. As for the energy, we consider that the bandwidth-to-power ratio increases around a 25% per generation, which is very well below the 40% increase predicted in [61] and the linear scaling obtained in this work.

Table 4.4 compares the area and power of the original transceivers and their scaled versions with those of different NoC implementations reported in the literature. In [6], the authors describe the mesh NoC of the Intel’s 80-core Polaris processor. This design can be considered similar to our **EMesh** baseline or the **MESH-BASE** design considered in the next chapter, as it does not implement any specific multicast support. To improve multicast performance, subsequent designs have explored the use of shorter router pipelines and multiport arbitration [21], as well as broadcast ordering capabilities similar to those attainable with BoWNoC [25]. For completeness, we also considered lower diameter topologies like the flattened butterfly (see Section 2.1). In this case, reasonable power and area are obtained for different system sizes and link widths, although recognizing serious scalability issues due to the increase in router radix.

We also include data from two recent designs that would be similar to the **OBus** considered here. On the one hand, Oh et al [62] propose to augment a conventional mesh with a global RF transmission line. Laying down the transmission line and the required transceivers takes a significant portion of area, but less power than the wireless option given that, in transmission lines, signals are guided instead of radiated. On the other hand, an all-optical 64-core NoC based on a global broadcast tree is proposed in [87]. The estimated power is high compared to the rest of alternatives, but provides a huge broadcast bandwidth of 320 bits per clock cycle. Area measurements are not provided.

Finally, to put these numbers in context, we complete Table 4.4 with the area and power consumption of two popular 22-nm cores, namely the high-performance Xeon Haswell and the energy-efficient Atom Silvermont. The thermal design power of an 18-core Haswell chip at 2.1GHz is 135W [187]. Correcting for frequency, we roughly estimate a per-core power of 5W. A similar reasoning is performed for the 8-core Silvermont chip, which works at 1.7GHz with a thermal design power of 12W [187]. Area numbers are supplied by the literature.

In overall, it is shown that a 22-nm transceiver would have an area and power consumption commensurate to that of current and future NoC designs, while representing between 1% and 10% of the area and power consumed by current core designs. Although these results do not include the area and energy required for the MAC protocol, related works [25, 150] confirm that its circuitry would take a minor part of the chip resources. Note that these figures could be reduced by means of the techniques as described above and that, in any case, the cost of the wireless channel could be in part compensated by the fact that the wired plane can be simplified.



Table 4.4: Per-Tile Area and Power Comparison

Ref.	Cores	Topology	Technology	Voltage	Frequency	Width	Area	Power
[102]	$N$	Wireless	65 nm (22 nm)	1 V	1 GHz	16 b (32 b)	0.25 (0.1) mm <sup>2</sup>	31.2 mW
						32 b (64 b)	0.4 (0.16) mm <sup>2</sup>	96 mW
						48 b (96 b)	0.73 (0.3) mm <sup>2</sup>	97.5 mW
[6]	80	Mesh	65 nm	0.7 V	1.7 GHz	39 b	0.34 mm <sup>2</sup>	98 mW
[25]	36	Mesh	45 nm	1.1 V	1 GHz	137 b	0.36 mm <sup>2</sup>	139 mW
[21]	16	Mesh	45 nm	1.1 V	1 GHz	64 b	0.32 mm <sup>2</sup>	27 mW
[37]	128	FBFly	32 nm	0.9 V	2 GHz	144 b	0.18 mm <sup>2</sup>	78 mW
[87]	64	Optics	22 nm	1 V	2.5 GHz	320 b	-	187.5 mW
[62]	64	TLs	22 nm	1 V	1 GHz	16 b	0.48 mm <sup>2</sup>	7.8 mW
Atom Silvermont (22 nm)							2.5 mm <sup>2</sup>	~1 W
Xeon Haswell (22 nm)							21.1 mm <sup>2</sup>	~5 W

#### 4.2.3.2 Towards affordable BoWNoC designs

We have seen that the area and power scalability of BoWNoC are similar than those of existing NoCs, but that antennas and transceivers are still bulky or inefficient. In light of this, the challenge is to reduce the absolute figures to levels that can be considered competitive from an implementation cost perspective. To this end, different approaches can be taken at diverse levels of design:

**System level:** Given that the BoWNoC scenario allows for an unprecedented control over propagation, the link budget process becomes quasi-deterministic. With this information, power can be allocated with extreme accuracy and thus, energy consumption may be minimized. Allocation can be even performed in an adaptive fashion thanks to reconfigurable techniques like the proposed in [167]. Other system-level techniques include power gating, which allows to dynamically power off cores that will not be used in a substantial period of time [188]. The flexibility of the wireless approach allows transceivers to be power gated with their associated core without affecting network performance, fact that is generally not possible with NoC routers. Finally, one can apply concentration by connecting  $k$  cores to a single transceiver. This approach slightly reduces the performance of the wireless network but would reduce the area and power by a factor of  $k$ .

**Transceiver level:** Unlike in traditional wireless systems, all the on-chip wireless transceivers share the same power supply and, therefore, the energy per bit metric encompasses the energy consumed by transmitter and all the receivers within the transmission range -see Equation (4.2.11)-. Thus far, we assumed  $E_{bit}^{tx} = E_{bit}^{rx}$  in order to simplify the analysis. However, the ratio between such figures could be chosen in the transceiver design process. To this end, a model accounting for the trade-offs between transceiver energy consumption, radiated power and received power, would enable the optimization of the energy efficiency. The only downturn here is the need for schemes capable of rapidly powering off the transmitter or receiver circuits while not used.

**Circuit level:** In this work, we considered a heterogeneous set of transceivers implementing different modulations and aiming at different communication scenarios, which are not necessarily oriented to low area and low power. Novel and optimized circuit topologies could allow for a substantial improvement of the area and energy efficiencies in wireless

chip communication. As we will see next, works like [102, 189] may serve as reference for future designs for efficient chip-scale transceivers.

**At the technology level:** The performance of a given wireless transceiver is undeniably limited by the underlying technology. Generally, technological advancements lead to higher operation frequency, lower area and potential for lower energy consumption. The trend set by current state-of-the-art transceivers will continue provided that the employed technologies evolve accordingly. However, the advent of a new technology or material bringing disruptive improvements, such as graphene [121, 132, 137, 190], may allow to go beyond the predicted performance. In Section 4.3 we explore the capabilities of this novel technology.

### 4.3 On the Suitability of Graphene-enabled THz Wireless Communication

In Section 3.1.2, we pointed out graphene as a technology that might be key for the implementation of WNoCs with core-level broadcast capabilities even in massive multicore settings, where processor may be several hundred microns long and wide. Scaling metallic antennas down to such dimensions is not a practical approach, since the low conductivity of nanoscale metallic structures [191] leads to a poor antenna performance. Moreover, metallic antennas of a few micrometers have a resonant frequency of several hundreds of terahertz. Such frequency band is not suitable for RF wireless communications due both to its huge channel attenuation leading to an extremely limited communication range and to the difficulty of implementing transceivers operating at such high frequency band.

Alternatively, graphene-based antennas, or shortly named *graphennas*, are uniquely suited for wireless communication within this context [192]. By virtue of its plasmonic properties, a graphenna several micrometers long is able to radiate within the terahertz band (0.1 - 10 THz) [124], this is, two orders of magnitude lower than that of metallic antennas. Low-complexity and low-power solutions for the transceivers operating at such frequencies could be achieved by adopting impulse-based modulations as discussed in Section 4.1.2. Furthermore, graphennas are tunable and show a higher radiation efficiency than typical THz metallic antennas despite their size difference as we will show throughout the rest of this chapter.

With the aim of going beyond the CMOS scaling trends identified in the previous sections, here we explore the performance of graphene-enabled wireless communications in the terahertz band and within the chip context. To this end, and since scaling down the size of the communication units intuitively suggests that nanoscale principles will largely determine the performance of communication, we define and employ a cross-cutting methodology that relates both extremes including the response of the graphennas and the effects terahertz wave propagation.

By expressing communication performance as a function of variables that define nanoscale physics through models, design space explorations can be performed during early development stages. As models evolve and improve, such methodologies would allow to obtain accurate evaluations of communication performance before graphene antennas are implemented or experimental testbeds are available. We expect that the proposed framework will allow designers not only to set minimum graphene quality requirements based on application-dependent propagation medium characteristics and communication performance guidelines, but also to explore the interaction between the frequency-selective re-

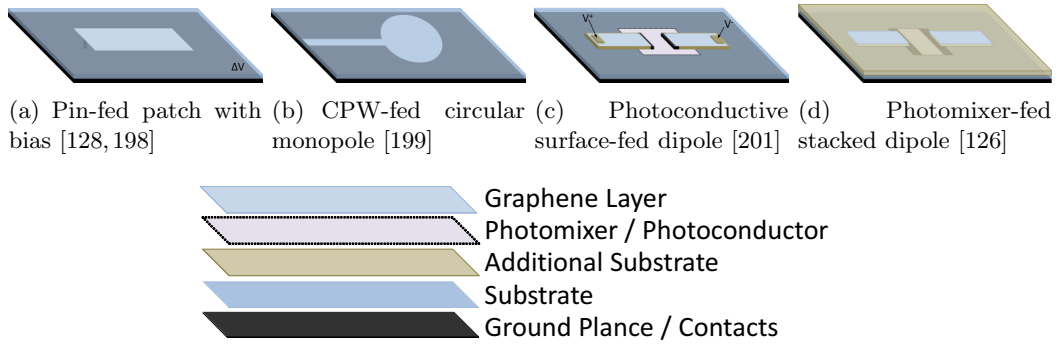


Figure 4.10: Schematic representation of a set of graphene-based antennas proposed in the literature. Silicon lenses are used in (c) and (d) but not shown for simplicity.

sponses of the antenna and the channel. Related works have inspected the impact of key technological parameters upon the antenna performance [125, 126, 128], but not from a communications standpoint. In parallel to this study, Zhang *et al* performed an end-to-end channel modeling and analysis of graphene-enabled wireless communications targeting indoor applications [193].

The remainder of the section is as follows. Sections 4.3.1 and 4.3.2 explain the theoretical background of graphene miniaturized antennas and terahertz propagation, necessary to follow the methodology proposed in Section 4.3.3. Such methodology is employed to evaluate the suitability of graphennas for ultra-short-range impulse radio communications, the results of which are shown in Section 4.3.4.

### 4.3.1 Background on Graphene-based Miniaturized Antennas

The application of carbon materials in the realm of antennas was first discussed in works that proposed the use of carbon nanotubes as potential dipole antennas and analyzed their transmission line properties and radiation pattern [91, 194, 195]. Following these discoveries and in light of the issues of this technology in terms of manufacturing, tuning, and placement on planar implementation processes, Jornet *et al* investigated the possibility of employing micrometric graphene patches for wireless communication [196]. Taking base on the work by Hanson on the propagation of electromagnetic waves on laterally-infinite graphene layers [197], it was demonstrated that a graphene patch a few micrometers long and wide would resonate in the terahertz band.

Such discovery has led to the surge of graphene antenna proposals that, in essence, consist of a number of finite-size graphene layers (the radiating elements) mounted over a metallic flat surface (the ground plane), with a dielectric material in between and a feed to drive the signals to the antenna. Patch antenna configurations have been analyzed with a pin feed [128], a punctual excitation [198] or a graphene microstrip line at one edge of the graphene layer [199]. Dipole-like designs, where the source is placed in the middle of two identical graphene layers, has been also proposed in [126, 200, 201]. Different biasing schemes have been also included in most of these works to take advantage of the unique tuning capabilities of graphennas [126, 198, 201]. Some of these proposals are conceptually represented in Fig. 4.10.

The reason behind the subwavelength behavior of graphennas is the presence of *surface plasmon polariton* (SPP) waves on the surface of graphene. Such phenomenon occurs at

the interface between any metallic and dielectric material pair when an electromagnetic wave impacts upon the metal. The properties of the SPP waves are determined by the frequency characteristics of the electrical conductivity of the metallic material (graphene in this case). For instance, while graphene shows strong plasmonic effects leading to resonance for frequencies in the terahertz band, other materials such as gold present this phenomenon in the optical range.

#### 4.3.1.1 Conductivity Models

Recent studies related to the conductivity of graphene sheets are enabling a precise modeling of the plasmonic phenomena occurring at the surface of graphennas [124, 200]. The main approach considers that a graphenna of a few micrometers in size is large enough to disregard the effects of the graphene edges. Therefore, the model for infinite graphene sheets can be employed, in which case the conductivity is calculated by means of the Kubo formula [197]. Moreover, experimental results show that the Drude-like intraband contribution dominates in the band of interest (0.1 - 10 THz), so that the conductivity can be expressed as:

$$\sigma(\omega) = \frac{2e^2 k_B T}{\pi \hbar} \ln \left[ 2 \cosh \left[ \frac{E_F}{2k_B T} \right] \right] \frac{i}{\omega + i\tau^{-1}}, \quad (4.3.1)$$

where  $e$ ,  $\hbar$ , and  $k_B$  are constants corresponding to the charge of an electron, the reduced Planck constant, and the Boltzmann constant, respectively [125, 197]. Variables  $T$ ,  $\tau$ , and  $E_F$  correspond to the temperature, the relaxation time, and the chemical potential of the graphene layer. At mid-infrared and optical frequencies, the intraband contribution fades and the conductivity approaches a much lower universal value of  $\sigma_0 = \pi e^2 / (2h)$  [202].

The conductivity plays a major role in determining the resonance of the graphenna, since the wavelength of SPPs within the graphenna is  $\lambda/n_{eff}$ , where the effective mode index  $n_{eff}$  is dependent upon the conductivity as:

$$n_{eff}(\omega) = \sqrt{1 - 4 \frac{\mu_0}{\epsilon_0} \frac{1}{\sigma(\omega)^2}}. \quad (4.3.2)$$

Knowledge on the conductivity of graphene has led to a further investigation of the characteristics of graphennas. The surface impedance of graphennas has been investigated in [200], allowing the extraction of preliminary results regarding the total efficiency of the graphenna. The work in [203] proposes a method to predict the input impedance of graphene reconfigurable dipoles. The impact of different substrates and their thickness upon the radiation characteristics of graphennas has been also studied in [125]. More importantly, the implications of varying the chemical potential and relaxation time of a graphenna have been explored in [128] and are shown below.

#### 4.3.1.2 Technological Design Parameters of Graphennas

Together with the antenna shape and dimensions, the conductivity plays a fundamental role in determining the radiation characteristics of the graphenna. As it is clearly observed in (4.3.1) and discussed next, the graphene conductivity strongly depends on the chemical potential and the relaxation time, which in turn depends on the carrier mobility. These are the parameters used in the design space exploration of Section 4.3.4.

**Chemical Potential:** Also referred to as Fermi energy, the chemical potential  $E_F$  refers to the level in the distribution of electron energies at which a quantum state is equally likely to be occupied or empty. Since it is possible to control its value by applying an electrostatic bias or by means of chemical doping, the chemical potential can be considered as a design parameter for graphennas. The impact of the chemical potential upon the frequency response of a graphenna is as shown in Fig. 4.11(a), in accordance with the tendencies revealed in [126,128]. It is observed that the radiation efficiency substantially increases with the chemical potential, whereas the resonant frequency is shifted upwards yet without an apparent effect on the resonance bandwidth. This confers graphennas unprecedented tuning possibilities that have been recently analyzed in different graphenna structures mostly at the terahertz band [126,198], but also at microwave frequencies [204].

**Relaxation Time:** The relaxation time  $\tau$  is the interval required for a material to restore a uniform charge density after a charge distortion is introduced and, in some works, it is expressed in terms of scattering rate  $\Gamma$  as  $\Gamma = (2\tau)^{-1}$ . At the band of interest ( $<10$  THz), phenomena such as interband damping or electron-phonon interaction, which appear over the interband and optical phonon threshold frequencies ( $\sim 50$  THz), can be neglected and the relaxation time can be calculated as

$$\tau \approx \tau_{DC} = \mu \hbar \sqrt{n\pi} / (ev_F), \quad (4.3.3)$$

where  $\mu$  is the carrier mobility,  $n$  is the carrier density, and  $v_F$  is the Fermi velocity [202]. The carrier density depends on the chemical potential, whereas the Fermi velocity is independent of the Fermi energy, and can be evaluated with the following expressions [205]:

$$\begin{aligned} E_F &= \sqrt{(\hbar v_F)^2 n\pi - (\pi K_B T)^2} / 3, \\ v_F &= 3ta / (2\hbar) \approx 10^6, \end{aligned} \quad (4.3.4)$$

where  $t \approx 2.8eV$  and  $a = 1.42\text{\AA}$  are tight-binding parameters for graphene [206]. Finally, considering that  $E_F \gg K_B T = 26 meV$ , the relaxation time expression in (4.3.3) becomes

$$\tau \approx \mu \frac{E_F}{v_p^2}, \quad (4.3.5)$$

which implies that the relaxation time and chemical potential can be used interchangeably under the conditions considered in this work.

Fig. 4.11(b) shows the impact of the relaxation time upon the frequency response of a graphenna. A stronger resonant behavior is observed as the relaxation time is increased, which matches the results in [128].

**Carrier Mobility:** The carrier mobility defines the average speed at which electrons can move within the material. Since diverse carrier mobility values can be achieved by means of different graphene manufacturing processes or by using different substrates [207,208], we will consider it as a design parameter for graphennas. For a suspended layer of graphene, the carrier mobility is obtained as:

$$\mu = \frac{1}{ne\rho_{xx}}, \quad (4.3.6)$$

where  $\rho_{xx}$  is the sheet resistivity.

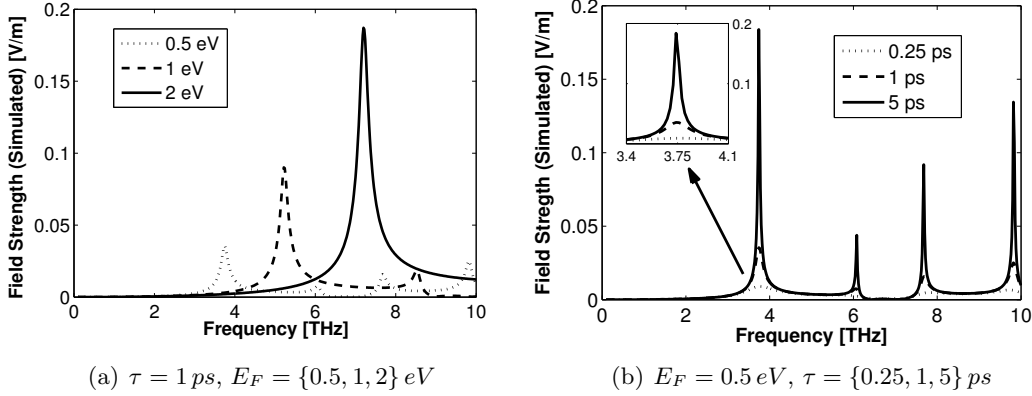


Figure 4.11: Frequency response at  $r = 1\text{m}$  of a  $5\mu\text{m} \times 1\mu\text{m}$  freestanding graphene patch fed with a  $10\text{-k}\Omega$  microstrip for different chemical potential and relaxation time values. The voltage inside the antenna is  $1\text{V}$  for all frequencies. We refer the reader to Section 4.3.3 for further methodological details.

It is worth noting that Eq. (4.3.5) gives a direct relation between the carrier mobility and the relaxation time at the frequency band of interest. This implies that both parameters can be used interchangeably under the conditions assumed in this work and that, according to the results in Fig. 4.11(b), a higher carrier mobility leads to a more resonant behavior.

### 4.3.2 Background on Terahertz Propagation

The shift of the frequency band of operation from the mmWave range up to the terahertz band has deep repercussions on the propagation of electromagnetic waves. More precisely, phenomena that are generally neglected can become significant as the wavelength reaches dimensions commensurate to the molecules found in the medium or the tiny irregularities of the surfaces upon which the waves may reflect. Their effects were first measured in indoor scenarios with the objective of their standardization within the IEEE 802.15 Terahertz Interest Group [209] and later modeled both in the frequency and time domains considering distances compatible with the chip scenario [210, 211]. Due to their frequency-selective nature and potentially detrimental impact, terahertz phenomena need to be considered on top of the channel modeling fundamentals shown in Section 3.3.

#### 4.3.2.1 Molecular Absorption and Particle Scattering

Molecular absorption is the process by which part of the wave energy is converted into internal kinetic energy of the excited molecules in the medium. Since several molecules present in the standard atmosphere have thousands of resonances in the terahertz band, they are excited by the terahertz electromagnetic waves radiated by antennas. This process converts part of the radiation into internal vibrations [209], and adds up to other factors that attenuate the propagated signals in wireless communications in the terahertz band. Molecular absorption can be modeled by the following analytical expression [210]:

$$\alpha_M(f, d) = \frac{1}{\tau} = e^{k_A(f)d}, \quad (4.3.7)$$

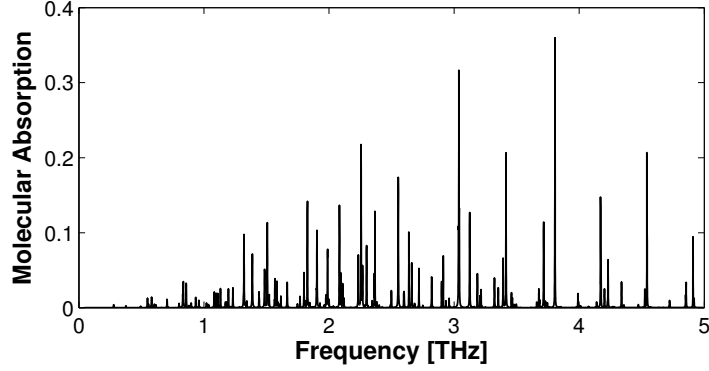


Figure 4.12: Molecular absorption of the terahertz channel at a distance of 1 cm.

where  $f$  is frequency,  $d$  is distance,  $\tau$  is defined as the transmittance of the medium and  $k_A(f)$  is the medium absorption coefficient. This last parameter depends on the medium composition, i.e., the particular mixture of molecules that the propagating wave finds along the channel, and is highly-frequency dependent as shown in Figure 4.12. We refer the reader to [210] for more details on how to calculate the medium absorption coefficient, and to Section 4.3.4 for an analysis of the impact of molecular absorption upon the performance of wireless communication in the on-chip scenario.

Another effect found at terahertz frequencies is particle scattering. As wavelengths become commensurate in size with certain particles that may be present in the environment, waves may be reflected and scattered by those particles. These effects are also frequency-selective and, therefore, have an impact upon the response of the channel in both domains [211]. The attenuation caused by particle scattering also takes an exponential form:

$$\alpha_S(f, d) = e^{k_S(f)d}, \quad (4.3.8)$$

where  $k_S(f) = \sum_j N_s^j \sigma_s^j$  is the particle scattering coefficient [211]. The calculation of this coefficient requires knowledge on the density of particles in the medium  $N$  and of the scattering cross section  $\sigma$  of each type of particle. The scattering cross section depends on the frequency band of interest, and in the terahertz band becomes:

$$\sigma_s^j = \frac{2\pi^5 x_d^6}{3\lambda^4} \left( \frac{n(f)^2 - 1}{n(f)^2 + 2} \right)^2, \quad (4.3.9)$$

where  $x_d$  is the diameter of the scattering particle,  $\lambda$  is the wavelength, and  $n(f)$  is the refractive index of the medium. Although the presence of particles large enough to produce scattering in terahertz band is highly improbable in a controlled environment like a chip package, it is important to be aware of such phenomenon.

#### 4.3.2.2 Reflections and Rough Surface Scattering

The behavior of reflections depends not only on the material, but also on the roughness of the surface. As wavelengths reach the micrometer scale, the roughness of certain material becomes significant and may produce scattering [212]. To calculate the effect of surface roughness, it is necessary to obtain a Rayleigh roughness factor that can be calculated using a measure of the surface height distribution. Although in our scenario we can consider only

specular reflections since the roughness of the materials used in electronic devices in the frequency range of interest is negligible, it is worth being aware of this phenomenon.

In any case, the reflected electric field can be obtained by multiplying the incident electric field by the Fresnel coefficient [213], which depends on the polarization of the incident wave. By assuming that the directions of the polarization vectors and the incident vectors match, we can write the Fourier transform of the channel impulse response matrix, particularized for one reflection, as:

$$H_p(f, \theta_i) = \begin{pmatrix} \Gamma_{\perp}(f, \theta_i) & 0 \\ 0 & \Gamma_{\parallel}(f, \theta_i) \end{pmatrix}, \quad (4.3.10)$$

where  $\Gamma_{\perp}(f, \theta_i)$  and  $\Gamma_{\parallel}(f, \theta_i)$  are the reflection coefficient for a perpendicular incident wave and for a parallel incident wave, respectively; and can be expressed as follows:

$$\begin{aligned} \Gamma_{\perp}(f, \theta_i) &= \frac{n_1(f) \cos \theta_i - n_2(f) \cos \theta_t}{n_1(f) \cos \theta_i + n_2(f) \cos \theta_t}, \\ \Gamma_{\parallel}(f, \theta_i) &= \frac{n_1(f) \cos \theta_t - n_2(f) \cos \theta_i}{n_1(f) \cos \theta_t + n_2(f) \cos \theta_i}, \end{aligned} \quad (4.3.11)$$

where  $n_1(f)$  and  $n_2(f)$  are the refractive index of the medium and of the material of the reflecting surface, respectively, while  $\theta_i$  and  $\theta_t$  are the incident and transmission angles. The relation between them is  $\theta_t = \arcsin\left(\frac{n_1(f)}{n_2(f)} \sin \theta_i\right)$ .

The refractive indexes will depend on the material where the waves propagate and are reflected. However, only a few materials have been characterized in the THz band. Some interesting data can be found for the time-domain spectroscopy applications [213]. In the literature, results are generally expressed by means of the real part of the reflective index  $n_i(f)$  and the absorption coefficient  $\alpha_i(f)$ . To obtain the complex refractive index, we use the following relation:  $n(f) = n_i(f) + j \frac{\alpha_i(f)c}{4\pi(2\pi f)}$  [213].

### 4.3.3 Evaluation Framework

Progressively scaling the size of the communication units (and the wavelengths) down to the microscale intuitively implies that nanoscale principles will largely determine the communication performance. As a result, it is important to carefully consider the impact of nanoscale phenomena upon performance even if communication occurs at much larger scales. Here, we explain the vertical methodology employed in this work to explicitly bridge the conceptual gap between both ends.

The methodology basically consists in the evaluation of the impulse response of the graphennas and of the terahertz channel towards a complete channel model. We will mainly follow the notation and considerations employed in [214], which stem from the work in [215] and are summarized in Figure 4.13. The model relates the voltage  $u_{RX}(t)$  at the output terminals of the receiving antenna with the input voltage  $u_{TX}(t)$  delivered at the terminals of the transmitting antenna. Although we will provide the formulations in both domains, the methodology and consequent analysis are oriented to the time domain given the very high radiation frequency and potentially wideband nature of graphennas. Also, without loss of generality, we introduce the notation for a given polarization.



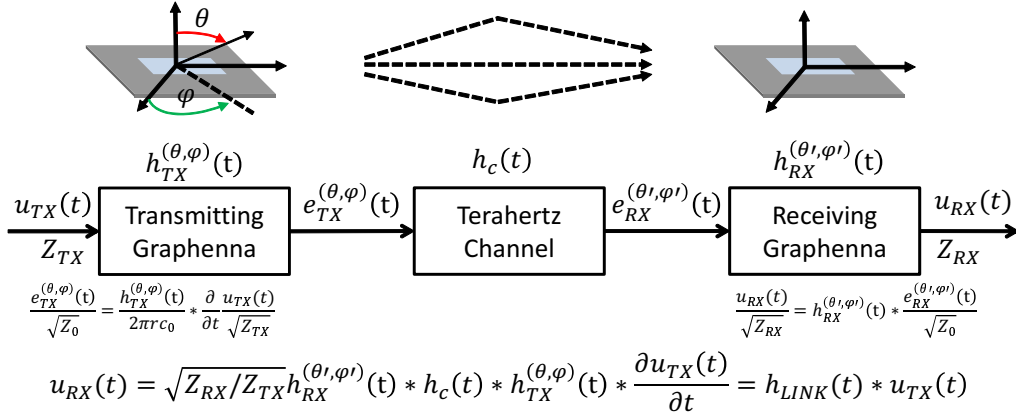


Figure 4.13: End-to-end channel model of a graphene-enabled wireless link in the time domain.

#### 4.3.3.1 Impulse Response Formulation

Consider a time-dependent voltage  $u_{TX}(t)$  delivered at the terminals of an antenna through a feed of characteristic impedance  $Z_{TX}$  and assume freespace propagation. The radiated signal  $e_{TX}^{(\theta, \phi)}(t)$  at the direction determined by the pair of angles  $\{\theta, \phi\}$  is given by:

$$\frac{e_{TX}^{(\theta, \phi)}(t)}{\sqrt{Z_0}} = \frac{\delta(t - r/c_0)}{2\pi r c_0} * h_{TX}^{(\theta, \phi)}(t) * \frac{\partial u_{TX}(t)}{\partial t \sqrt{Z_{TX}}}, \quad (4.3.12)$$

where the operator  $*$  represents convolution,  $h_{TX}^{(\theta, \phi)}(t)$  is the impulse response of the transmitting antenna at the propagation direction,  $\partial u_{TX}(t)/\partial t$  is the derivative of the input signal,  $r$  is the distance to the antenna,  $c_0$  is the speed of light in vacuum, and  $Z_0$  is the free space impedance [214]. From this equation, we infer that *the impulse response of an antenna is the relation between the excitation voltage of the antenna (input) and the radiated field strength (output)*. Therefore, the impulse response of any antenna can be obtained by applying a voltage to it and measuring, either physically or by means of simulation, the strength of the radiated fields. Note that this definition leaves the time derivative out of the impulse response of the antenna, as opposed to in [216], thus decoupling the differentiation effects inherent to all antennas from the antenna-dependent dispersion, impedance mismatch, and other losses. In the frequency domain, (4.3.12) becomes:

$$\frac{E_{TX}^{(\theta, \phi)}(f)}{\sqrt{Z_0}} = \frac{e^{j\omega r/c_0}}{2\pi r c_0} H_{TX}^{(\theta, \phi)}(f) j\omega \frac{U_{TX}(f)}{\sqrt{Z_{TX}}}, \quad (4.3.13)$$

where  $E_{TX}^{(\theta, \phi)}(f)$ ,  $H_{TX}^{(\theta, \phi)}(f)$ , and  $U_{TX}$  are the radiated field, the response of the antenna, and the input voltage in the frequency domain. The delay is modeled with the exponential factor, whereas  $j\omega$  is the equivalent to a time derivative.

A similar expression is used to calculate the voltage  $u_{RX}(t)$  at the terminals of an antenna when receiving an electromagnetic  $e_{RX}(t, \theta', \phi')$  wave arriving from the direction determined by the pair of angles  $\{\theta', \phi'\}$ :

$$\frac{u_{RX}(t)}{\sqrt{Z_{RX}}} = h_{RX}^{(\theta', \phi')}(t) * \frac{e_{RX}^{(\theta', \phi')}(t)}{\sqrt{Z_0}}, \quad (4.3.14)$$

where  $h_{RX}^{(\theta',\phi')}(t)$  is the impulse response of the receiving antenna at the incident direction and  $Z_{RX}$  is the characteristic impedance at the receiving end. With the definition of impulse response given in (4.3.12) and (4.3.14), the reciprocity theorem yields  $h_{TX} = h_{RX} = h_A$  for the same antenna [214]. Note that this also applies in the case of graphene-based antennas, since the same plasmonic principles explain their operation both in transmission and in reception [124]. In the frequency domain, we have:

$$\frac{U_{RX}(f)}{\sqrt{Z_{RX}}} = H_{RX}^{(\theta',\phi')}(f) \frac{E_{RX}^{(\theta',\phi')}(f)}{\sqrt{Z_0}}, \quad (4.3.15)$$

with  $U_{RX}(f)$ ,  $H_{RX}^{(\theta',\phi')}(f)$ , and  $E_{RX}^{(\theta',\phi')}(f)$  as the frequency-domain versions of the voltage at the output, the antenna response, and the incident field. The reciprocity condition is still  $H_{TX} = H_{RX} = H_A$  [214].

Now let us consider the radiated signal  $e_{TX}(t)$  to be incident at the receiving antenna. By combining (4.3.12) and (4.3.14), we can obtain a single expression that describes the dependence between the voltage at the receiver and at the transmitter:

$$\frac{u_{RX}(t)}{\sqrt{Z_{RX}}} = h_{RX}^{(\theta',\phi')}(t) * \frac{\delta(t - r_{TR}/c_0)}{2\pi r_{TR} c_0} * h_{TX}^{(\theta,\phi)}(t) * \frac{\partial u_{TX}(t)}{\partial t \sqrt{Z_{TX}}}, \quad (4.3.16)$$

where  $r_{TR}$  is the distance between the two antennas. The expression clearly separates the contributions of the antennas and the channel: the term that is function of  $r_{TR}$  corresponds to the freespace propagation and can be replaced by a more complex propagation model  $h_C(t)$  that could include multipath and terahertz effects. Thus, (4.3.16) becomes:

$$\frac{u_{RX}(t)}{\sqrt{Z_{RX}}} = h_{RX}^{(\theta',\phi')}(t) * h_C(t) * h_{TX}^{(\theta,\phi)}(t) * \frac{\partial u_{TX}(t)}{\partial t \sqrt{Z_{TX}}}. \quad (4.3.17)$$

Finally, the impulse response of the whole link  $h_L(t)$ , which includes the channel and the two antennas, can be calculated by assuming that  $u_{TX}(t) = \delta(t)$  and applying the properties of the convolution operation:

$$h_L(t) = \sqrt{\frac{Z_{RX}}{Z_{TX}}} \frac{\partial}{\partial t} (h_{RX}^{(\theta',\phi')}(t) * h_C(t) * h_{TX}^{(\theta,\phi)}(t)), \quad (4.3.18)$$

which becomes, in the frequency domain:

$$H_L(f) = j\omega \sqrt{\frac{Z_{RX}}{Z_{TX}}} H_{RX}^{(\theta',\phi')}(f) H_C(f) H_{TX}^{(\theta,\phi)}(f). \quad (4.3.19)$$

### 4.3.3.2 Outline of the Methodology

As summarized in Figure 4.14, the proposed methodology provides a means to *A*) characterize the antennas and the channel separately or *B*) to perform a joint evaluation of a complete graphene-enabled wireless link in the time domain. All cases basically involve the evaluation of the impulse response of the antenna  $h_A(t)$ , the channel  $h_C(t)$ , or the whole link  $h_L(t)$ , and the use of time-domain metrics to characterize their respective performances. Three steps are required to characterize the antennas, namely:

1. The conductivity  $\sigma(\omega)$  of a graphene layer with chemical potential  $E_F$ , relaxation time  $\tau$ , and carrier mobility  $\mu$  is modeled as explained in Section 4.3.3.3.

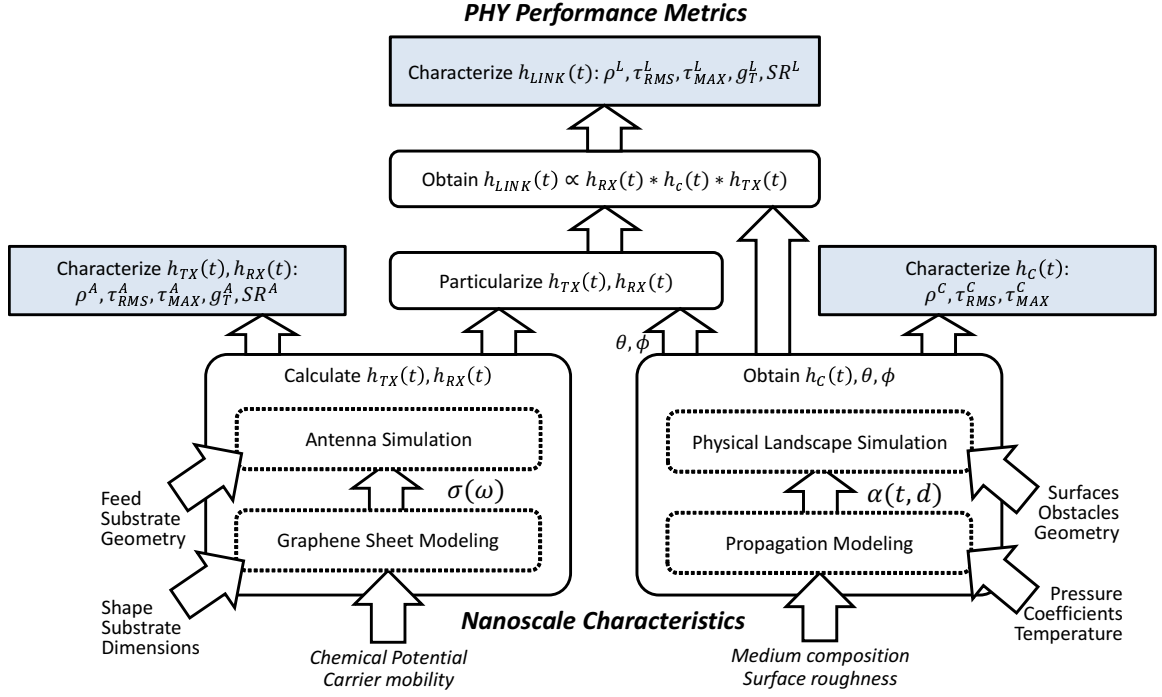


Figure 4.14: Vertical methodology for the time-domain characterization of graphene-enabled wireless links as functions of nanoscale phenomena.

2. The model of a graphene layer is integrated with the rest of antenna elements within an electromagnetic field solver, which allows to obtain the impulse response  $h_A(t)$  by using (4.3.12)-(4.3.15) as indicated in Section 4.3.3.3.
3. Finally, the communication performance of the whole graphenna is evaluated through a set of pre-defined metrics that require the impulse response  $h_A(t)$  and a voltage waveform  $u_{TX}(t)$  as inputs. These are detailed in Section 4.3.3.5.

The evaluation of the channel is performed by means of the following steps:

4. The propagation media are modeled by evaluating the extinction loss coefficient  $k(f)$  as explained in Section 4.3.3.4, which requires calculating the molecular absorption and particle scattering coefficients using Eqs. (4.3.7) and (4.3.8), respectively.
5. The reflection coefficient  $\Gamma$  of the different material interfaces found within the chip package are calculated using Eq. (4.3.11).
6. The channel impulse response  $h_C(t)$  is obtained by applying the results obtained in previous steps to the chip landscape. This defines which paths will reach the receiver, information used to solve Eq. (4.3.21) as explained in Section 4.3.3.4.
7. Finally, the channel can be characterized using the metrics enumerated in Section 4.3.3.5.

Finally, five steps are required to characterize the whole link:

8. The transient response of the transmitting and receiving antennas  $h_{TX}^{(\theta, \phi)}(t)$  and  $h_{RX}^{(\theta', \phi')}(t)$  is obtain through steps 1 and 2.

9. The impulse response of the channel  $h_C(t)$  (including the angles of radiation and arrival for the evaluated setting) is calculated using steps 4 and 5.
10. The response of the antennas obtained in step 8 is particularized for the angles of radiation and arrival obtained in step 9.
11. The impulse response of the antennas and the channel are used to evaluate  $h_L(t)$  via Eqs. (4.3.18) or (4.3.19).
12. The communication performance of the link is characterized through a set of time-domain metrics summarized in Sec. 4.3.3.5.

It is important to emphasize that, following these steps, we are able to express the different performance metrics as a function of the target parameters representing nanoscale characteristics of the antennas, the channel, or the whole link. This is the main aim and contribution of the proposed methodology. Also, there are two aspects that are worth noting in order to justify the use of time-domain metrics and to better understand their meaning. On the one hand, pulse-based modulations have been proposed as the fundamental mechanism for communication among nanosystems [176]. On the other hand, area and complexity requirements suggest the use of non-coherent IR techniques as explained in Section 4.1.2. These systems are pulse-based and better described in the time domain [175].

Next, we extend the explanations regarding the steps required to obtain the different impulse responses using the background of Sections 4.3.1 and 4.3.2.

### 4.3.3.3 Antenna Impulse Response

**Modeling the Graphene Structure:** as detailed above, the first step towards the evaluation of the impulse response of the antenna consists in calculating the conductivity of the graphene sheets that act as radiating elements. The complexity of the models used to this end will depend on the frequency band of interest and the characteristics of the graphene sheet, since they determine whether phenomena such as damping or the Hall effect should be taken into consideration [206]. The operating temperature, the chemical potential, the carrier mobility and the relaxation time of the graphene sample also need to be provided. Note that, as mentioned in Section 4.3.1.2, the substrate on which the graphene layer will be placed may affect the value of the carrier mobility.

In this work, we model and evaluate graphennas of micrometric dimensions that are expected to radiate within the frequency range where the intraband conductivity dominates, validating the use of Equation (4.3.1) to obtain the conductivity  $\sigma(\omega)$ .

**Obtaining the Impulse Response Through Simulation:** once the frequency-dependent conductivity of graphene is calculated, the radiating element of the antenna can be rigorously modeled as an infinitesimally thin surface with an equivalent impedance of  $Z(\omega) = \frac{1}{\sigma(\omega)}$ . The graphene layer needs to be shaped according to antenna geometry and then integrated with the substrate, the feed, the ground plane, or any other component that may be present in the target antenna configuration. Among other parameters, the dimensions and permittivity of the substrate, as well as the type of source and its impedance need to be defined since they determine the performance of the resulting graphenna and, by extension, its impulse response. The complexity and accuracy of the model used to describe the antenna is a design decision and will depend on the focus of the study.

Two different methods can be followed to obtain the response of the antenna by means of an electromagnetic simulator. Provided that a feeding mechanism is defined, the simu-

lator will calculate the fields radiated by the antenna as a function of the input voltage in transmission. Reciprocally, simulators generally allow to consider a wave incident to the antenna to then calculate the voltage at the antenna terminals in reception. In both cases, the response of the antenna can be derived in the time domain by relating the voltage and electromagnetic field using Equations (4.3.12) or (4.3.14), or in the frequency domain using Equations (4.3.13) or (4.3.15). The domain depends on the numerical method used to simulate the performance of the antenna; simulators commonly offer methods in both domains [217].

In case the response is calculated in the frequency domain, it is necessary to apply the inverse Fourier transform to obtain the impulse response:

$$h(t) = \mathcal{F}^{-1}(H(\omega)) = \int_{-\infty}^{\infty} H(\varepsilon) e^{2\pi i \varepsilon t} d\varepsilon. \quad (4.3.20)$$

Note that the frequency response needs to be defined over all frequencies, so that limiting the band will introduce errors in the inverse transform if the response is non-zero outside the band of interest. These errors will have an impact upon the performance metrics that only depend on the impulse response of the antenna. However, this fact does not truly affect the study of the communication performance of any antenna provided that the transmitted signals will be band-pass and within the band of interest.

#### 4.3.3.4 Channel Impulse Response

The impulse response of the channel  $h_C(t)$  accounts for the different phenomena that attenuate and disperse the signal during propagation. These can be inherent to the propagation medium, or result from the interaction of signals with the elements located between transmitter and receiver. In general, they can be mathematically represented as:

$$h_C(t) = \sum_{i=1}^L \Gamma_i \alpha_i^{-1}(t, d) e^{j\varphi_i(t, d)} \delta(t - \tau_i), \quad (4.3.21)$$

where  $d$  denotes separation between transmitter and receiver and  $L$  is the number of components that reach the receiving antenna. This equation basically expresses that signals will suffer, for each path to the receiver, a given attenuation  $\alpha$ , a phase shift  $\varphi$  and a delay  $\tau$ . The term  $\Gamma$  adds the effects of the different reflections.

As summarized in Fig. 4.14 and according to Eq. (4.3.21), calculating the impulse response of the channel implies modeling the propagation medium to assess the attenuation and phase shift per unit of distance, and then simulating the physical landscape in order to obtain the number, distribution, and characteristics of multipath components.

**Modeling the propagation medium:** When considering free space propagation through air, we normally have that  $\alpha(t, d) = (2\pi d c_0)^{-1}$ ,  $\phi(t, d) = 0$  and  $\tau(t, d) = d/c_0$  with  $c_0$  representing the speed of light [214]. However, terahertz waves are susceptible to molecular effects negligible in lower frequencies [210, 211]. Basically, electromagnetic waves *A*) excite gas molecules whose resonant frequency is in the terahertz band, losing energy in the process (molecular absorption), and *B*) are scattered by particles whose size is comparable to the wavelength of the signal. These effects are frequency-selective and, therefore, have an impact upon the response of the channel in both domains. Specifically, we have:

$$\alpha(f, d) = \frac{e^{k(f)d}}{2\pi dc_0}, \quad \alpha(t, d) = \mathcal{F}^{-1}(\alpha(f, d)), \quad (4.3.22)$$

where  $k(f) = \sum_j k_A^j + k_S^j$  is the extinction loss coefficient, which accounts for the molecular absorption and scattering contributions of all the elements present in the medium through the coefficients  $k_A$  and  $k_S$ . These coefficients model the quantity of molecular absorption and scattering per type of molecule and can be evaluated using Eqs. (4.3.7) and (4.3.8). Therefore, the particular mixture of molecules that waves will find along its path, as well as the transmission distance, will be the two main determinants of the propagation loss. We refer the reader to [211] for more details and refined models.

**Modeling the surfaces:** The channel characterization for wireless communications in the terahertz band has a particularity besides the presence of molecular effects inherent to the propagation. In case surfaces present a roughness comparable to the wavelength of the signal, diffuse scattering appears upon reflection. Given the small wavelength of terahertz waves, such diffuse scattering effects cannot be neglected anymore and need to be considered [212]. In either case, the presence of surfaces will determine the direction and intensity of reflections, aspects that need to be taken into account when simulating the channel.

**Simulating the physical landscape:** finally, the scenario must be modeled and simulated to obtain the multipath behavior of the channel. Full-EM simulation becomes too costly due to the density of elements to consider and, instead, one can consider the use of techniques such as ray tracing. The outcome of these simulations will depend on the application scenario: most applications generally require of an statistical model comprehending the most representative settings in terms of obstacle shape, composition, or motion [218]. In our case, the scenario is static and can be not only known *a priori*, but even modified at design time to optimize certain metrics. For instance, the characteristics of the dielectric used throughout the chip can be tuned as they have an impact upon performance [172].

#### 4.3.3.5 Time-domain Performance Metrics

The final step in each branch of the methodology is to employ the impulse response  $h_A(t)$ ,  $h_C(t)$ , or  $h_L(t)$  to fully characterize the antenna, the channel, or the whole link, respectively. This process is independent of the antenna type or the channel specificities, and requires a set of performance metrics different from the ones used in narrowband systems. In works that characterize Ultra-wideband (UWB) antennas [214, 219], metrics are evaluated using the analytic response of the antenna, which is given by:

$$h^+(t) = h(t) + j\mathcal{H}, \quad (4.3.23)$$

where  $\mathcal{H}$  is the Hilbert transform of the impulse response. The reason behind the choice of the analytic response is that its envelope  $|h^+(t)|$  is a faithful representation of the dispersion of the antenna since it gives insight into how the energy input to the antenna is distributed over time. Alternatively, channel modeling in general and terahertz band research in particular employs the Power Delay Profile (PDP)  $p(t)$  of the channel or the entire link, which is obtained as:

$$p(t) = |h(t)|^2. \quad (4.3.24)$$

Besides this formulation difference, both research lines refer to the same phenomena and effects: attenuation and dispersion are analyzed by evaluating, respectively, the characteristics in terms of amplitude and temporal length of the response. In light of this, we propose a unified set of metrics that operate upon  $h^+(t)$  if we only inspect the antenna or  $p(t)$  if we inspect the channel or the whole link, and that characterize the amplitude and length of the response. The novelty resides in that the methodology is capable of expressing these metrics as functions of nanoscale phenomena.

**Response Peak:** the peak  $\rho$  of a response is defined as the maximum value of its envelope:

$$\begin{aligned}\rho_A &= \max_t |h^+(t)| [m/ns], \\ \rho_C &= \max_t p_C(t) [dB], \\ \rho_L &= \max_t p_L(t) [dB].\end{aligned}\tag{4.3.25}$$

A high peak value could mean that the energy is highly concentrated around a given time instant. Receiving a strong peak allows for a precise detection of the pulse position, which is desirable in location and ranging applications, as well as in coherent communication systems. In the case of non-coherent communication, a high peak value is not necessarily a decisive factor since the receiver accounts for the energy in a time interval that may span the whole received pulse.

**Response Energy:** the response energy  $E$  of the impulse response is defined as the integral of its instantaneous power over its duration:

$$\begin{aligned}E_A &= \|h_A(t)\|^2, \\ E_C &= \|h_C(t)\|^2 = \int p_C(t), \\ E_L &= \|h_L(t)\|^2 = \int p_L(t),\end{aligned}\tag{4.3.26}$$

where the norm is  $\|f(x)\|^k = \int_{-\infty}^{\infty} |f(x)|^k dx$ . The value of  $E$  indicates how the antenna and channel effects impact on the radiated and propagated energy.

**Transient Gain:** the transient gain  $g_T$  is the time domain version of the antenna gain or the channel path loss, and is an indicator of how efficiently an antenna or a link is able to transmit a given input signal  $u_{TX}$ . This is a specially relevant metric since since a design objective is to maximize the power that reaches the receiver towards reducing the error rate. We derive expressions for the antenna and the channel using the original definition, which expresses the transient gain as the the ratio of the radiation intensity of the antenna,

$$U_{rad} = r^2 \frac{\|e_{TX}(t)\|^2}{Z_0},\tag{4.3.27}$$

to the radiation intensity of an isotropic radiator,

$$U_{rad}^{iso} = \frac{P_{in}}{4\pi} = \frac{\|u_{TX}(t)\|^2}{4\pi Z_{TX}},\tag{4.3.28}$$

with  $u_{TX}$  as the input voltage. A similar definition can be extended to the channel or the whole link, obtaining:

$$\begin{aligned}
g_T^A(u_{TX}) &= \frac{\|h_A(t) * \frac{\partial u_{TX}(t)}{\partial t}\|^2}{\|\sqrt{\pi}c_0 u_{TX}(t)\|^2}, \\
g_T^C(e_{TX}) &= \frac{\|h_C * e_{TX}(t)\|^2}{\|e_{TX}\|^2}, \\
g_T^L(u_{TX}) &= \frac{Z_{TX} \|h_L(t) * u_{TX}(t)\|^2}{Z_{RX} \|u_{TX}^2\|}.
\end{aligned} \tag{4.3.29}$$

Note that the channel transient gain considers an electrical field as an input, but that can be expressed in terms of an input waveform as long as the impulse response of the antenna is available. This dependence will allow us to evaluate the frequency-dependent effects of the antenna and the channel in the time domain.

**Response Width:** one way to evaluate the width  $\tau_W$  of the response is by calculating the Full Width at Half Maximum (FWHM) of the response; this is, the difference between the time instants wherein such magnitude is half of the maximum:

$$\begin{aligned}
\tau_{FWHM}^A &= t_{h2}^A - t_{h1}^A, \\
\tau_{FWHM}^C &= t_{h2}^C - t_{h1}^C, \\
\tau_{FWHM}^L &= t_{h2}^L - t_{h1}^L,
\end{aligned} \tag{4.3.30}$$

where  $t_{h1}^A = t'$  so that  $|h_A^+(t')| = \rho^A/2$  and  $t_{h2}^A = t''$  so that  $t'' > t_{h1} \wedge |h_A^+(t'')| = \rho^A/2$ . Similar expressions are used for  $t_{h2}^C, t_{h1}^C, t_{h2}^L$ , and  $t_{h1}^L$  using the PDP. The envelope width is a clear indicator of the dispersion introduced around the peak by the antenna, the channel, or the link. The lower is this value, the lower is the broadening that pulses will suffer. In this case, the inter-pulse interval could be reduced leading to higher data rates.

Another way to calculate the width of the response is by means of the RMS delay spread, which is the square root of the second central moment of the response:

$$\begin{aligned}
\tau_{RMS}^A &= \sqrt{\int (t - \tau_m^A)^2 \cdot |h^+(t)|^2}, \\
\tau_{RMS}^C &= \sqrt{\int (t - \tau_m^C)^2 \cdot p_C(t)}, \\
\tau_{RMS}^L &= \sqrt{\int (t - \tau_m^L)^2 \cdot p_L(t)},
\end{aligned} \tag{4.3.31}$$

where  $\tau_m^A = \int t \cdot |h_A^+(t)|^2$ ,  $\tau_m^C = \int t \cdot p_C(t)$  and  $\tau_m^L = \int t \cdot p_L(t)$  are the mean excess delay of the antenna, the channel, and the link. The RMS delay spread is a measure of the time dispersion introduced by the frequency selectivity of the antenna or the channel. Generally, the coherence bandwidth  $B_C = \tau_{RMS}^{-1}$  of a given channel is used to express the frequency interval over which the response does not change significantly, giving a hint of the available bandwidth. Low values of the delay spread are desired, as this implies that the link admits the transmission of short pulses towards obtaining high data rates.

**Response Duration:** the duration of a response  $\tau_R$  is generally defined with respect to a parameter  $\alpha$  that represents a portion of energy that can be considered negligible. We express the duration as the time interval between the response peak and the response reaching the lower bound determined by  $\alpha$ :



$$\begin{aligned}
\tau_R^A(\alpha) &= t_\alpha^A - t_{\rho_A}^A, \\
\tau_R^C(\alpha) &= t_\alpha^C - t_{\rho_C}^C, \\
\tau_R^L(\alpha) &= t_\alpha^L - t_{\rho_L}^L,
\end{aligned} \tag{4.3.32}$$

where  $t_{\rho_A}^A = t'$  so that  $|h_A^+(t')| = \rho_A$  and  $t_\alpha^A = t''$  so that  $t'' > t_{\rho_A}^A \wedge |h_A^+(t'')| = \alpha \cdot \rho_A$ . Parameters  $t_\alpha^C$ ,  $t_{\rho_C}^C$ ,  $t_\alpha^L$ , and  $t_{\rho_L}^L$  are defined analogously. In an antenna, the response duration is often referred to as *ringing duration* due to the ringing tail of the response, which is caused by the resonant behavior of energy storage or multiple reflections within the radiating structure [214]. In a channel, the response is referred to as *maximum excess delay* can be lengthened by isolated multipath components. For high data rates, these ringing effects or multipath reflections may overlap with the following symbol, causing a raise of the inter-symbol interference and therefore limiting the maximum achievable data rate of the transmission. Hence, a low response duration is desirable.

**Pulse Width Stretch Ratio:** similarly to the transient gain, the stretch ratio  $SR$  can be defined with respect an input waveform. Let the normalized cumulative energy function of a given signal  $s(t)$  be defined as:

$$E_s(t) = \frac{\int_{-\infty}^t |s(t)|^2}{\|s(t)\|^2}. \tag{4.3.33}$$

Assuming that a certain fraction  $\alpha$  of ringing energy can be neglected, the width of the signal  $W(s)$  can be then obtained with the following equation:

$$W(s) = E_s^{-1}(1 - \alpha/2) - E_s^{-1}(\alpha/2). \tag{4.3.34}$$

The stretch ratio is obtained by dividing the width of the output signal by the width of the input signal [219]. Depending on whether we consider the antenna, the channel or the whole link, the stretch ratio becomes:

$$\begin{aligned}
SR_A(u_{TX}) &= \frac{W(h_A * \frac{\partial u_{TX}}{\partial t})}{W(u_{TX})}, \\
SR_C(u_{TX}) &= \frac{W(h_C * e_{TX})}{W(e_{TX})}, \\
SR_L(u_{TX}) &= \frac{W(h_L * u_{TX})}{W(u_{TX})},
\end{aligned} \tag{4.3.35}$$

Therefore, the pulse width stretch ratio quantifies the broadening of a pulse caused by the antenna. Note that values lower than 1 are possible, which do not imply that the output pulse is shorter than the input signal, but that antenna concentrates a significant fraction of the output pulse energy around the peak. A value close to 1 or below is desired, which means that the antenna has a nearly flat response in the frequency band of the input signal. Otherwise, the pulse width would increase leading to reduced transmission data rates.

#### 4.3.4 Design Space Exploration

This section is devoted to analyzing the results of the design space exploration. Here, we evaluate the graphennas and the channel separately, to then extract the impulse response of the whole link and provide a joint assessment. We choose a rather simple scenario to focus

Table 4.5: Parameters for graphene-enabled wireless links design space exploration.

Parameter	Value
Dimensions ( $W \times L$ )	$5 \times 1 \mu\text{m}^2$
Chemical Potential ( $E_F$ )	0.1 — 2 eV
Carrier Mobility ( $\mu$ )	0.5 — $6 \text{ m}^2\text{V}^{-1}\text{s}^{-1}$
Feed	10-k $\Omega$ Microstrip
Substrate ( $\epsilon_r$ )	Air (freestanding, $\epsilon_r = 1$ )
Frequency Range	0.1 — 10 THz
Impulse Pulse Types	Sinc, Gaussian
Impulse Bandwidth	1 THz
Propagation	Line-of-sight
Vapor Concentration ( $VC$ )	0.5% — 50%
Distance ( $d$ )	1 mm — 10 m

on the methodological details rather than on the specific antenna design or channel. Unless noted, the link to be explored has the following characteristics and parameters, summarized in Table 4.5:

- *Antennas*: we assume a family of freestanding 5- $\mu\text{m}$  by 1- $\mu\text{m}$  graphennas with different carrier mobility and chemical potential values. In transmission, the antenna is fed with a 1-THz wide sinc pulse centered at the resonant frequency of the antenna. The feed is a microstrip line with an impedance of 10  $k\Omega$ . This is due to the high impedance of graphennas, which is typically of up to a few  $k\Omega$  and suggests the use of high-impedance sources such as photomixers to reduce the impedance mismatch [201, 203].
- *Medium*: the medium is air, modeled as a standard medium with a variable percentage of water vapor molecules. Small-particle scattering is negligible.
- *Landscape*: we assume that antennas are face to face and, therefore, line-of-sight propagation will take place through the boresight direction of the antennas. Transmission distance is variable.

We use FEKO [217] to evaluate the family of graphennas, and MATLAB scripts to implement the channel and link models and extract the different performance metrics. In our case, we evaluate the frequency response of the antenna and/or the channel in the band of interest (0.1 - 10 THz) and we then apply the inverse Fourier transform to it. When inspecting the graphennas, we will focus on the impact of the carrier mobility and chemical potential on their performance. As for the channel, we will analyze the key impact of the highly frequency-dependent molecular absorption on the communication performance of chip-scale links.

#### 4.3.4.1 Antenna

Fig. 4.15 compares the envelope of the analytic response of the graphenna under investigation considering two different pairs of carrier mobility and chemical potential. The first graphenna has a carrier mobility of  $\mu = 6 \text{ m}^2\text{V}^{-1}\text{s}^{-1}$  and a chemical potential of  $E_F = 0.1 \text{ eV}$  for a relaxation time of  $\tau = 0.6 \text{ ps}$ , whereas the parameters of the second graphenna are  $\mu = 1 \text{ m}^2\text{V}^{-1}\text{s}^{-1}$  and  $E_F = 2 \text{ eV}$  for a relaxation time of  $\tau = 2 \text{ ps}$ . Both plots are delayed 1 ps for the sake of clarity. The envelope of the analytic response increases at that point, rapidly

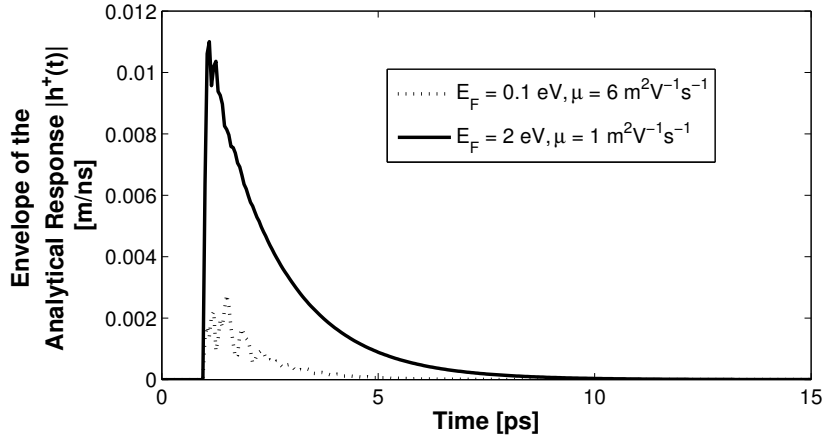


Figure 4.15: Envelope of the analytic response of two graphennas with different chemical potential and carrier mobility.

reaching the response peak, and then an apparently exponential decay follows. Ringing effects cause oscillations to appear in the analytic response after the main peak and broaden the impulse response.

Table 4.6 summarizes the performance of the two graphennas mentioned above. In order to complete the analysis, the table includes two metallic antennas: a micrometric patch, which has a resonant frequency of several tens of terahertz; as well as a metallic patch resonating in a similar frequency band than the graphennas. The metrics  $g_{T,G}$ ,  $SR_G$  and  $g_{T,S}$ ,  $SR_S$  correspond to the transient gain and stretch ratio for a Gaussian pulse and a sinc-shaped pulse centered at the resonant frequency of each considered antenna. Note that since the impedance of gold antennas is in the order of  $100\Omega$ , we will use a  $75\text{-}\Omega$  source impedance to drive the antenna in these cases for the sake of fairness.

In order to reach the same resonant frequency than a graphenna, a gold patch is approximately two orders of magnitude larger in terms of area. Moreover, the patch is  $0.5\ \mu\text{m}$  thick (gold cannot be infinitesimally thin). This size difference allows this antenna to radiate more energy, resulting in a higher envelope peak. However, it is observed that this fact does not ensure the best performance in terms of relative radiation efficiency (transient gain). The response of the  $75\ \mu\text{m} \times 15\ \mu\text{m}$  gold patch is the narrowest among the compared antennas, which allows it to perform remarkably well in terms of stretch ratio. The second gold antenna, the dimensions of which are comparable to that of the graphennas (but  $0.033\text{-}\mu\text{m}$  thick), resonates at a much higher frequency than the other antennas. In this case, the gold antenna shows an outstanding performance in terms of response width and ringing, leading to the lowest stretch ratio figures. Although remarkable performance is observed in terms of envelope peak, the reduced transient gain implies that graphennas of the same size will be able to radiate with higher efficiency.

In light of these results, it is reasonable to conclude that even though gold antennas show a slightly improved potential performance, the difference will be compensated by the unique size and resonant frequency characteristics of graphennas.

To further investigate the impact of the carrier mobility and chemical potential on the time-domain behavior of graphennas, we obtained the impulse response of a set of different graphennas within the parametric design space. Even though carrier mobilities of a few hundred thousands of  $\text{cm}^2\text{V}^{-1}\text{s}^{-1}$  have been measured in nearly ideal conditions [207], we will

Table 4.6: Performance comparison of different metallic and graphene-based antennas.

	Graphenna 1 $E_F=0.1, \mu=6$	Graphenna 2 $E_F=2, \mu=1$	Gold Antenna 1	Gold Antenna 2
$Area$	$5 \mu\text{m}^2$	$5 \mu\text{m}^2$	$1125 \mu\text{m}^2$	$5 \mu\text{m}^2$
$f_{res}$	1.74 THz	7.48 THz	1.52 THz	22.5 THz
$p$	0.003 m/ns	0.01 m/ns	0.12 m/ns	0.04 m/ns
$E$ (arb. units)	1.76	50.23	264.94	15.04
$\tau_W$	1 ps	1.2 ps	0.45 ps	0.06 ps
$\tau_R(10\%)$	1.95 ps	3.6 ps	1 ps	0.1 ps
$g_{T,G}$	-31.56 dBi	-4.55 dBi	-8.17 dBi	-9.45 dBi
$SR_G$	1.013	1.026	0.9957	0.9843
$g_{T,S}$	-31.6 dBi	-4.623 dBi	-8.19 dBi	-9.45 dBi
$SR_S$	0.9912	1	0.9912	0.9888

evaluate a more conservative range between 5000 and 60000  $\text{cm}^2\text{V}^{-1}\text{s}^{-1}$ , proved achievable with current graphene manufacturing techniques [202]. In the case of the chemical potential, typical values between 0.1 and 2 eV are considered, which are below the graphene electrical breakdown [220]. The resulting relaxation time ranges from 0.05 to 12 ps in the frequency band of interest.

Fig. 4.16 shows the matrix of impulse responses, wherein each row and column corresponds to a chemical potential and carrier mobility value, respectively. The time interval is fixed and ranges from 0 to 10 picoseconds in all cases, whereas the vertical axis limits are also fixed to  $[-P, P]$ , where  $P$  is the maximum envelope peak among all the temporal responses.

In very low chemical potential and carrier mobility conditions, almost null impulse responses are obtained. This implies that resonance is not achieved due to the attenuation of SPP waves as they propagate on the surface of the graphenna. In order to obtain a non-negligible resonant behavior, such effects must be reduced by means of improving either the chemical potential or the carrier mobility. As mentioned in Section 4.3.1.2, this increases the relaxation time of the material and leads to a stronger radiated field.

Although both the increase of the chemical potential and of the carrier mobility contribute to a raise of the radiated energy, the impact on the impulse response is different in each case. On the one hand, the energy variation observed as we modify the chemical potential is evenly distributed along the response, impacting both the response peak and pulse width. Such behavior is coherent with the frequency-domain behavior shown in Figure 4.11 and is further analyzed in the following paragraphs using performance metrics presented in Section 4.3.3.5.

**Response Peak:** Fig. 4.17(a) shows the results regarding the response peak, as functions of both the carrier mobility and the chemical potential. The results therein confirm that the peak value is clearly proportional to the chemical potential since contour lines are parallel to the Y-axis. The weak dependence shown with respect to the carrier mobility confirms that the raise in antenna efficiency impacts on the impulse response width and ringing length rather than on the peak value.

**Response Energy:** Fig. 4.17(c) shows the energy of the response as functions of both the carrier mobility and the chemical potential. The results therein indicate that both technological parameters have a similar impact upon the transient gain for this test signal.

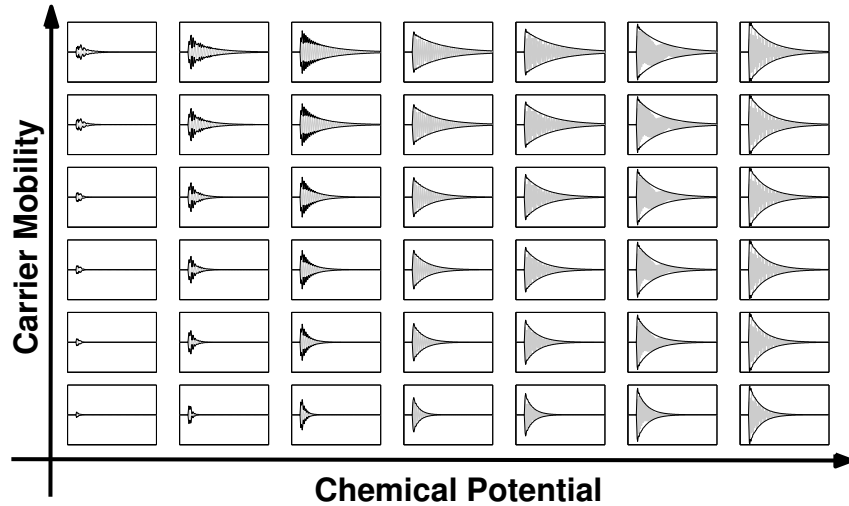


Figure 4.16: Impulse Response matrix of  $5\mu\text{m} \times 1\mu\text{m}$  graphennas for a design space exploration with different carrier mobility and chemical potential values. The time interval ranges from 0 to 10 picoseconds.

Such behavior matches the results observed in Fig. 4.16, which show an increase in terms of impulse response in the presence of high chemical potential and carrier mobilities.

**Transient Gain:** Fig. 4.17(e) shows the transient gain as functions of both the carrier mobility and the chemical potential when a gaussian pulse is radiated. The behavior is similar than for the response peak and response energy metrics. Apparently, whether this such energy surge revolves around the response peak or not does not make a difference when evaluating the transient gain for this test signal. However, this tendency may vary when changing the bandwidth of the input signal. The same analysis has been also performed using a sinc function with the same bandwidth and energy, showing identical tendencies with very similar absolute values.

**Peak Width:** Fig. 4.17(b) plots the peak width as a function of both the carrier mobility and the chemical potential. Narrow responses are obtained for low chemical potentials and carrier mobilities. The peak width then sharply increases with the chemical potential for values up to approximately 1 eV and with the carrier mobility. When the chemical potential surpasses 1 eV, the peak width plateaus or slightly decreases depending on the carrier mobility value. This behavior can be explained as follows: the response becomes wider with the carrier mobility due to the increase of ringing effects clearly observed in Fig. 4.16. However, the specific impact upon the response width depends on the value of the half-maximum: at low chemical potentials, the peak value is low and therefore ringing effects dominate; whereas at high chemical potentials, the peak value is high and the impact of incremental ringing diminishes.

**Ringling Duration:** Fig. 4.17(d) shows the ringling duration as a function of both the carrier mobility and the chemical potential, assuming  $\alpha = 10\%$ . Two main tendencies are clearly observed regarding the ringling duration: first, that it increases with the carrier mobility and, second, that its dependence on the chemical potential is rather parabolic as it increases until approximately  $E_F = 1$  eV and then moderately decreases. The highest ringling values are therefore obtained with the combination of high carrier mobilities and  $E_F = 1$  eV. Since the ringling duration also depends on the peak value, a similar reasoning to that of the peak width can be made to explain this behavior.

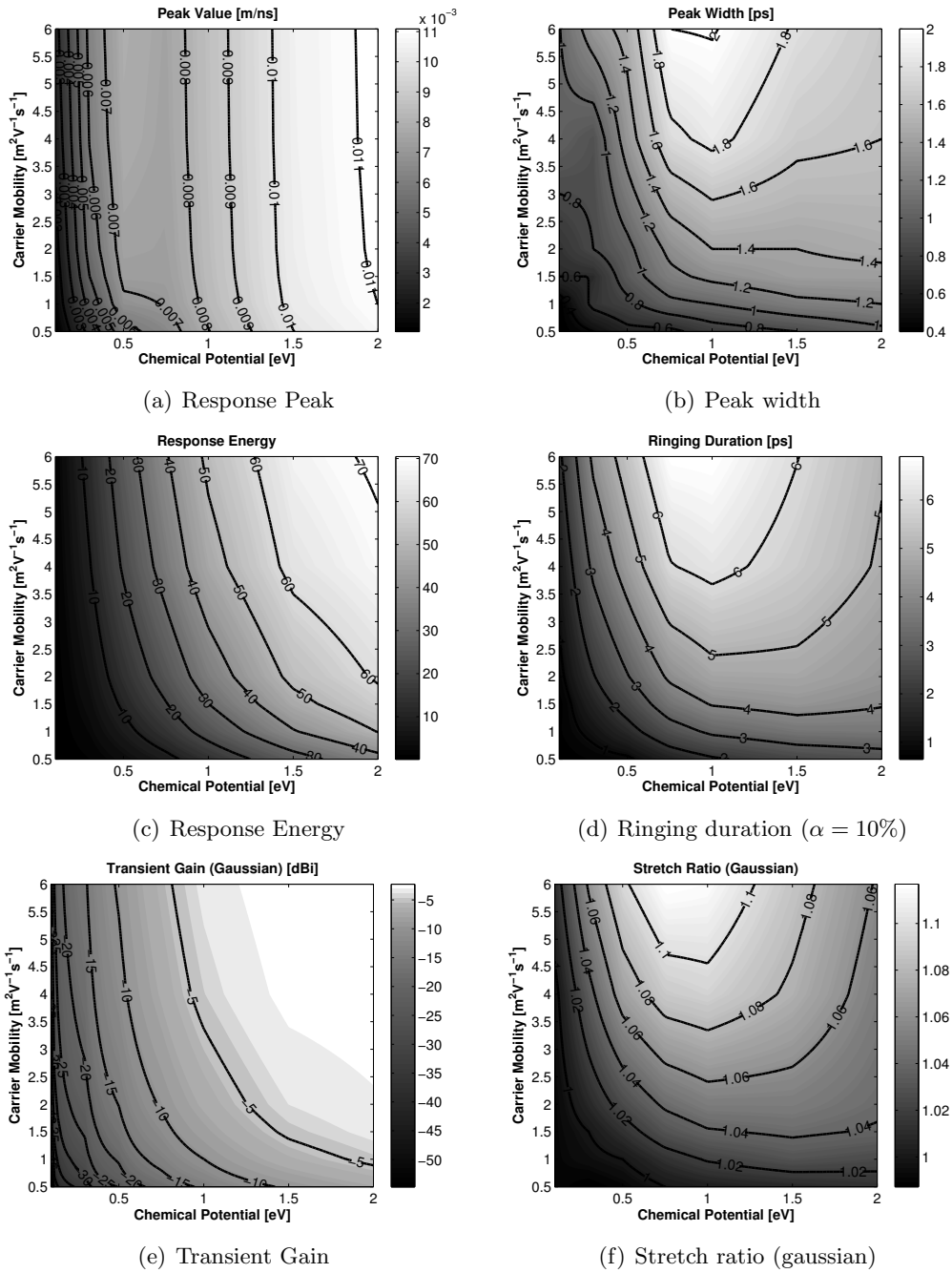


Figure 4.17: Response amplitude metrics (left) and response length characteristics (right) as functions of the carrier mobility and chemical potential.

**Pulse Width Stretch Ratio:** Fig. 4.17(f) shows the pulse width stretch ratio as a function of both the carrier mobility and the chemical potential, assuming a gaussian pulse as input. The trend shown therein is similar to that of the ringing. A clear and undesired rise of the stretch ratio is observed when the chemical potential is increased below 0.9 eV, with a particularly strong transition around 0.5 eV and high carrier mobilities. After reaching its highest value, the stretch ratio moderately decreases for both high carrier mobilities and

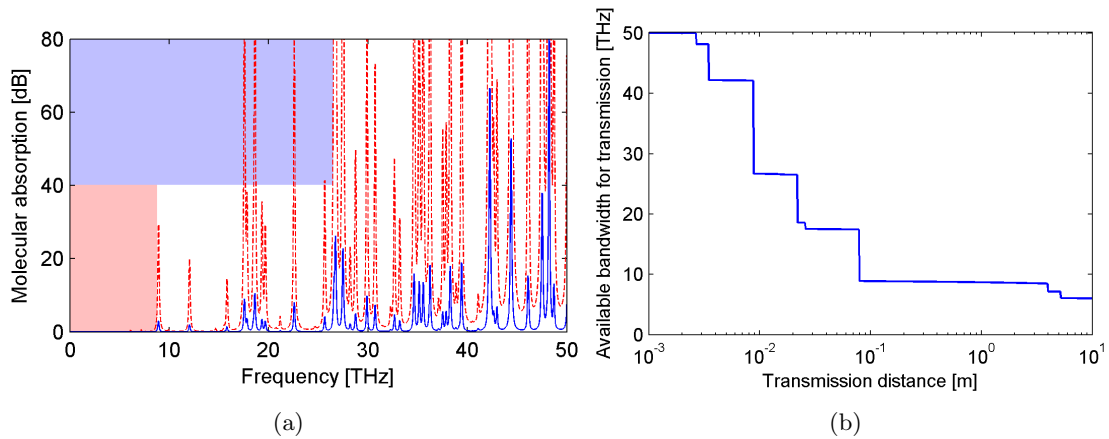


Figure 4.18: (a) Molecular absorption in dB for transmission distances of 1 cm (blue solid line) and 10 cm (red dashed line). The top blue (bottom red) background shows the frequency region which determines the available bandwidth for a transmission distance of 1 cm (10 cm). (b) Available bandwidth in the frequency band up to 50 THz.

high chemical potentials. The same relative behavior is observed when a sinc-shaped pulse is used as input, but with a general improvement in absolute terms.

#### 4.3.4.2 Channel

Fig. 4.18 shows the molecular absorption of the terahertz channel as a function of frequency, for transmission distances of 1 cm and 10 cm, well within the range of current multiprocessor die sizes. We observe that molecular absorption is highly dependent on the transmission distance, derived from its exponential dependence with the distance, as shown in Eq. (4.3.7). In this particular case, both the number of absorption peaks and their amplitude notably increase when the transmission distance changes from 1 to 10 cm.

Fig. 4.18(a) shows that molecular absorption creates several peaks of very high attenuation, which will create a limitation in the available bandwidth. In order to quantify the available bandwidth as a function of the transmission distance, we consider the frequency band at which the value of molecular absorption is below a given threshold. Since this threshold will depend on the final implementation and the tolerable distortion, we choose 10 dB as an example. A different threshold would result in different quantitative values for the effective bandwidth, but with the same dependence on the transmission distance. In line with the pulse-based modulations which have been proposed in short-range terahertz communications [176], the considered frequency band ranges from baseband to 50 THz.

According with Fig. 4.18(a), the available bandwidth for a transmission distance of 1 cm would be of approximately 27 THz as shown in blue, whereas it would be around 9 THz for a transmission distance of 10 cm, as shown in red. Fig. 4.18(b) shows a semi-log plot of the scalability of the available bandwidth with respect to the transmission distance. We observe a rapid decrease of the bandwidth as the distance increases, with several steps of different sizes corresponding to the molecular absorption peaks. For instance, molecular absorption has a negligible effect at short transmission distances and the whole 50 THz band is usable for distances up to 2.5 mm. This basically means that molecular absorption does not become an impairment at chip-scale distances, confirming the suitability of the

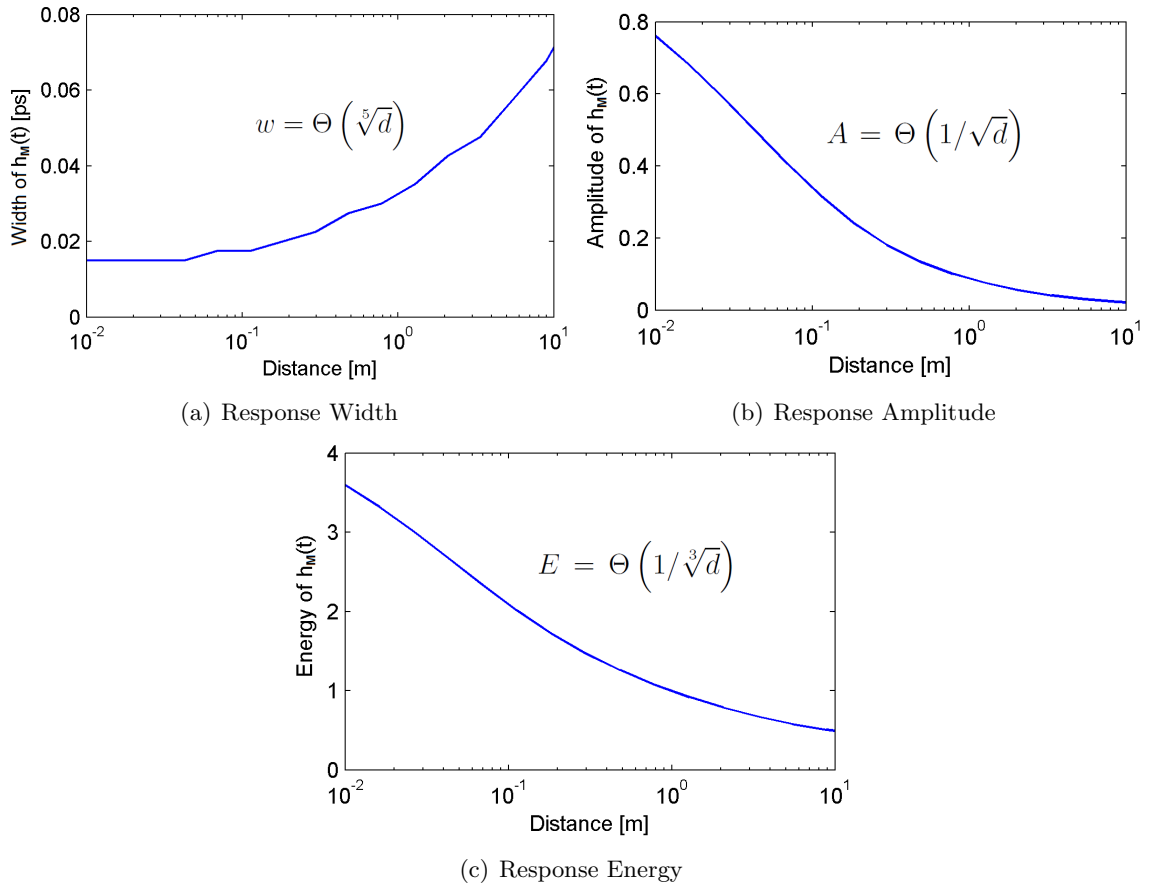


Figure 4.19: Semi-log plots of the different characteristics of the channel impulse response due to molecular absorption.

terahertz channel for on-chip communication. On the other hand, for distances greater than 5 m, less than 6 THz are available for a molecular absorption-free transmission.

The increase of the importance of molecular absorption with the distance can be also evaluated in the time domain. The attenuation peaks at high-frequency components of the signal causes the channel impulse response to lower the peak amplitude, and to show a smaller width and a less smooth shape for the higher transmission distance. Assuming that the transmitted signals consist of subpicosecond pulses [176], the main implication of the dispersive behavior of the channel impulse response is that molecular absorption distorts the transmitted pulses, decreasing their amplitude and increasing their width. We will study these effects in more detail next.

Fig. 4.19 shows a semi-log plot of the width of the molecular absorption impulse response  $w$  as a function of the transmission distance  $d$ . A clear dependence of the width of the impulse response on the distance between transmitter and receiver is observed, with a scaling trend of  $O(\sqrt[5]{d})$ . In particular, in order to achieve a communication throughput in the order of 1 Tbps, the received pulses will need to have a width of less than 1 ps. This result shows that, for instance, for a transmission distance of 3 m, the distortion introduced by molecular absorption is limited to 0.05 ps, thereby reducing the maximum achievable throughput by around 5%. At chip-scale distances, however, the effect of molecular absorption in the response width is almost negligible.



The amplitude of the channel impulse response  $A$  is related to the attenuation caused by molecular absorption to a signal propagating in the terahertz channel. In Fig. 4.19 we can observe that, as expected, the amplitude of the impulse response decreases as the transmission distance increases, with a scaling trend of  $O(1/\sqrt{d})$ . For instance, the amplitude decreases by a factor of 10 when the transmission distance changes from 1 cm to 1 m. At chip-scale distances of a few centimeters, though, the attenuation caused by molecular absorption is mild. This demonstrates that molecular absorption will not be an impairment in the BoWNoC scenario.

Another important metric to consider is how molecular absorption affects the energy of the transmitted signals. Since we have found that molecular absorption attenuates the transmitted signal, but it also increases their width, it is not clear how the signal energy measured by the receiver will scale with the transmission distance. Fig. 4.19 shows that the energy of the channel impulse response due to molecular absorption decreases with respect to the distance, with a scaling trend of  $O(1/\sqrt[3]{d})$ . This result shows that the reduction of the signal energy occurs at a lower pace than that of its amplitude; by comparison, the energy decreases only by a factor of 4 when increasing the transmission distance by two orders of magnitude, from 1 cm to 1 m. As a consequence, the use of non-coherent detection schemes like the introduced in Section 4.1.2 may be better suited than a coherent detector in environments affected by molecular absorption.

For the sake of comparison, in a scenario of an ideal free-space wireless communication channel with no molecular absorption, the channel impulse response would be a delta function independently of the transmission distance (without considering other factors affecting the loss). Thus, the width, amplitude, and energy of the temporal response of the channel would scale as  $O(1)$  instead of as  $O(\sqrt[5]{d})$ ,  $O(1/\sqrt{d})$ , and  $O(1/\sqrt[3]{d})$ , respectively.

#### 4.3.4.3 Joint Exploration

One of the virtues of the methodology presented in Section 4.3.3 is that it allows the end-to-end evaluation of graphene-enabled wireless communication links. Thus, interactions between nanoscale effects at the graphene and nanoscale effects at the channel can be investigated towards a complete design space exploration.

We have seen that the chemical potential affects the electrical size of the antenna as the resonance frequency increases without changing the physical size. Thus, a higher chemical potential implies a higher efficiency. On the other hand, we have seen that the effects of molecular absorption appear at around 1 THz and that they increase with the frequency of the EM waves. Thus, there is a need to inspect whether the path loss should be minimized either by increasing the chemical potential or by reducing the detrimental effects of molecular absorption.

Figure 4.20 shows the peak value and the RMS delay spread as functions of both the chemical potential of the antenna and the amount of water vapor in the channel. Plots 4.20(a) and 4.20(b) represent these metrics for a transmitter-receiver separation of 3 cm, commensurate with the size of standard chips. At such short distance, molecular absorption is far from being an impairment as suggested above. However, a dependence between both the chemical potential and the molecular absorption can be noticed. Specifically, the absorption can become significant in terms of attenuation at high chemical potential values, whereas the RMS shows a more stable value when the molecular absorption is low.

To extend this joint evaluation to the off-chip scenario, we plotted the peak value and RMS delay spread for a distance of 1m in Figures 4.20(c) and 4.20(d). Here, the difference

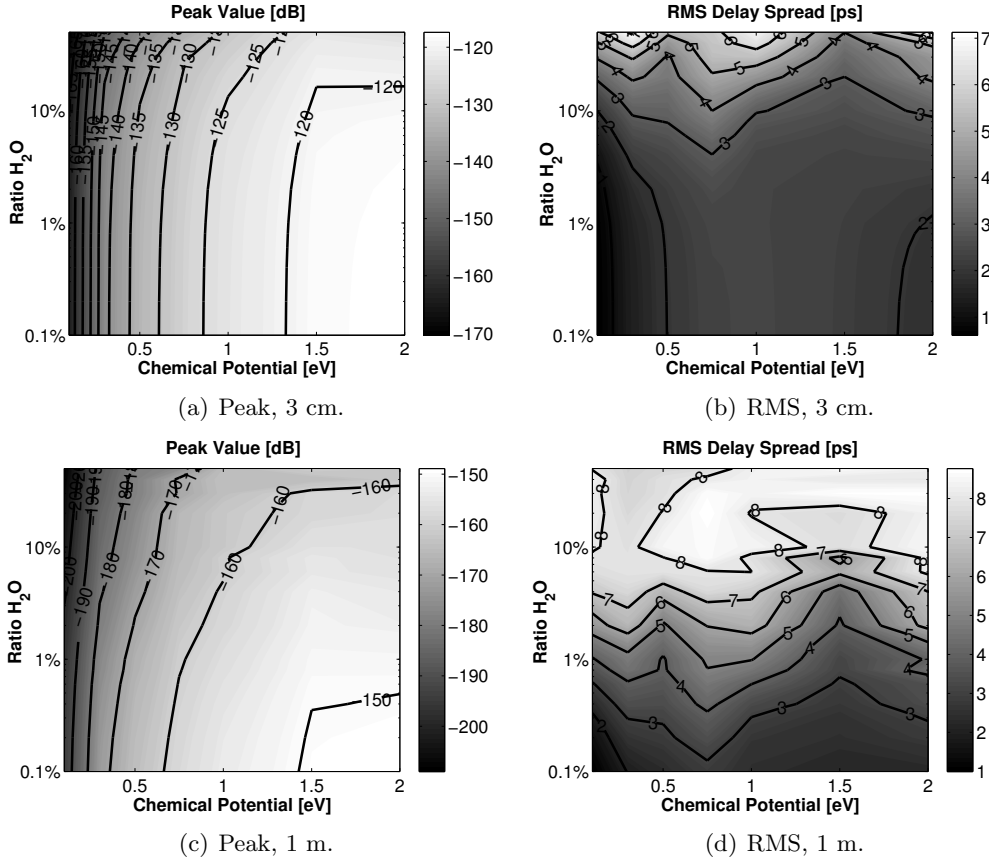


Figure 4.20: Response peak and RMS delay spread as functions of the chemical potential of the antenna and the amount of water vapor in the channel for a distance of 3 cm (top) and 1 m (bottom).  $\mu=4\text{m}^2\text{V}^{-1}\text{s}^{-1}$ .

in terms of molecular absorption is clear: increasing the water vapor concentration does not yield significant changes in terms of peak value for  $E_F = 0.5$  eV, whereas for  $E_F = 2$  eV there is a gap of almost 10 dB when considering low and high water vapor concentrations. The RMS delay spread shows a rather irregular pattern due to the highly frequency-dependent distribution of the molecular absorption.

Finally, note that the use of this methodology could be applied to other investigations. For instance, since multipath is a frequency-dependent phenomenon, one could explore the multipath effects with multiple chemical potentials and, thus, multiple frequencies of resonance. In simple terms, reflections that might be harmless for an antenna resonating at a certain frequency may destroy signals when the antenna is tuned at another frequency. In an scenario like that of WNoC where we seek maximum performance, we may benefit from the inherent tunability of graphene antennas, and multipath components are fixed, this vertical methodology can be extremely useful to avoid certain performance-degrading situations.

## Chapter 5

# MAC: Seeking Reliable and Scalable Broadcast Communication

The Medium Access Control (MAC) layer design has been a key research issue since the creation of the first computer networks. Basically, the MAC protocol defines mechanisms to ensure that all nodes can access to a shared medium in a reliable manner. This is mandatory because two or more simultaneous accesses to the same channel generally *collide* and cannot be received correctly, resulting into a waste of resources. Thus, the protocol needs to determine how to avoid them and/or how to recover from them in order to guarantee a successful transmission. These decisions play a decisive role in determining the performance of the network.

The ALOHA system [221] is considered one of the first wireless packet data networks with a MAC implementation. In this case, nodes simply attempt to transmit whenever data is ready and wait for acknowledgment. A collision is assumed at the receiver if the CRC fails, in which case the transmitter is not acknowledged and will have to retry after waiting a random amount of time. This primitive scheme works, but the cost of collisions is high. As a result, MAC protocol design has evolved to reduce their cost and to address multiple challenges associated with node mobility, disjoint transmission ranges, intermittent failures, so as to be applicable at a wide range of wireless network scenarios such as local area networks, mobile networks, or wireless sensor networks [162, 163, 222].

At a first glance, the on-chip scenario does not show strong resemblances with traditional wireless networks. Most of the constraints, performance objectives, and input traffic characteristics are unique to the on-chip communication paradigm and, as such, there is a need for a complete rethinking of existing MAC protocols. As analyzed in Section 3.4, first WNoC proposals have not provided a deep analysis of all these factors and, instead, resorted to basic MAC schemes mainly based on several types of multiplexing. Be it time, frequency, code multiplexing or any combination thereof, the architectures based on this technique serve as a proof-of-concept of the WNoC paradigm in specific applications [141], but often do not represent a realistic design point for BoWNoC due to their fundamental rigidity and scalability issues. Thus, coordinating the access to the shared medium remains as a grand challenge in BoWNoC.

This chapter aims provide a comprehensive context analysis that details the main particularities of the on-chip scenario. This study aims to deliver a taxonomy of the uniquenesses of this field, and to inspect how these uniqueness will impact the design of MAC protocols. We will see that although some traditional MAC techniques may be worth

revisiting, co-design and cross-layer design are feasible and should be tackled to obtain near-optimal results. As a second contribution, we present the design of a MAC protocol that takes advantage of the knowledge gained in the context analysis to provide improved flexibility and performance with realistic cost.

The remainder of the chapter is as follows. In Section 5.1, we first detail the general principles that have been traditionally applied in the MAC layer of conventional networks, to then review the existing proposals for the on-chip communication scenario. In Section 5.2, we present the context analysis. Section 5.3 is devoted to the description and performance evaluation of BRS-MAC, a realistic protocol for scalable BoWNoC designs. The performance is assessed by means of numerical analysis and cycle-accurate simulation in a wide set of configurations, and benchmarked against a selection of wired and wireless architectures. Finally, in Section 5.4 we give a brief set of guidelines for future research on MAC protocol optimization for BoWNoC.

## 5.1 Design Principles, Objectives and Challenges

The MAC layer manages access to the medium by telling nodes when to transmit and how to deal with collisions. As outlined in Section 3.3, MAC protocols enforce these rules seeking to maximize performance and fairness, as well as to reduce cost. If the MAC protocol needs to be applied in scenarios with significant node densities, it must also be scalable. Section 3.3 also explained that MAC mechanisms can be broadly divided into three groups: *channelization*, *coordinated access* or *random access*.

For simplicity reasons, primitive MAC protocols mostly implemented random access strategies. Several proposals followed the seminal ALOHA protocol [221], increasing in sophistication to achieve the performance, fairness, and scalability objectives outlined above. Slotted ALOHA [223] reduced the collision probability by only allowing the user to transmit in pre-defined time instants, whereas Carrier Sense Multiple Access (CSMA, [224]) did so by checking whether the channel is free before transmitting. Ethernet [225] uses CSMA/CD, which detects collisions and prematurely aborts transmissions to minimize the performance penalty. A classic alternative implementing distributed coordination is token passing [226], where the node that possesses the token transmits and then hands it off to the next node of an ordered list, totally avoiding collisions yet at the cost of extra latency.

These simple protocols work moderately well as long as certain assumptions are met. However, real scenarios present a set of issues that complicate the protocol and reduce performance. For instance, the IEEE 802.11 standard for wireless local area networks [222] uses a CSMA with collision avoidance instead of CSMA/CD since collisions cannot be detected during transmission. It also provides a contention-free mode of operation and addresses issues related to mobility, asynchronous operation, or the *hidden terminal* problem, wherein nodes located in different transmission ranges cannot correctly assess if the medium is free.

As we will see in Section 5.2, the chip scenario does not present as many issues as traditional wireless networks, but is in overall more demanding in terms of performance and reliability. Moreover, our vision has significant scalability requirements as it considers a potentially large number of nodes within the same chip sharing the same set of channels. Finally, on-chip traffic is highly variable and requires certain architectural flexibility.

Existing WNoC proposals employ a relatively low number of antennas even in large-scale architectures and, as mentioned in Section 3.4, mostly rely on channelization schemes. These techniques are a valid option for rather uniformly distributed traffic and only up

Table 5.1: Wireless Manycore Scenario Requirements

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

to a moderate number of nodes, as they are not scalable in terms of performance or implementation complexity. Increasing the number of time, frequency, or code-multiplexed channels implies, respectively, a non-scalable increase in latency, the use a large set of fine-tuned filters, and unrealistic synchronization requirements [141]. Also, since bandwidth is typically allocated in a static manner, performance drops dramatically if traffic bursts or hotspots appear. For greater flexibility and simplicity, other proposals employ a version of the token passing protocol [90], but the token round-trip time remains as a scalability concern. ALOHA and CSMA-based protocols have been rarely evaluated [151, 152] despite of their low latency and simplicity, arguably due to the relatively low performance in terms of throughput.

The brief analysis of related work reveals that current MAC protocols for wireless on-chip communication are not suitable for BoWNoC since the balance between the performance, cost, fairness, and scalability objectives is not accomplished. As we will see subsequently, the scenario presents certain peculiarities that will drive the design process and offers a set of enhancement opportunities mainly arising from the knowledge and certain degree of control that the architect has on the system.

## 5.2 Context Analysis

Table 5.1 provides a rough quantification of the main requirements of the wireless manycore scenario. The number of nodes per transmission range reaches levels commensurate to those of massive WSN or Machine-to-Machine (M2M) networks [227]. The throughput demands also lead to strong resemblances with M2M networks, including the use of mmWave technologies. Finally, the on-chip networking scenario shares with mission-critical WSNs the need for latency and reliability guarantees, although with much more restrictive deadlines and power budget [228].

Such a distinctive combination of requirements would be unsolvable in the aforementioned scenarios due to a long list of design issues: unknown topology, intermittent nodes, blockage, multipath, energy preservation constraints, or deafness problems due to the use of directive antennas, to name a few. However, we will see that the uniquenesses of the on-chip scenario virtually eliminate most of these problems, thus pointing towards simple and streamlined solutions through informed design decisions.

Figure 5.1 shows a summary of the traits that will define the MAC design process in the BoWNoC paradigm. Also, Table 5.2 lists the differences between on-chip networks and traditional wireless networks in aspects that affect the design of the protocol. The rest of the section delves into all these aspects, with the aim to provide specific guidelines of

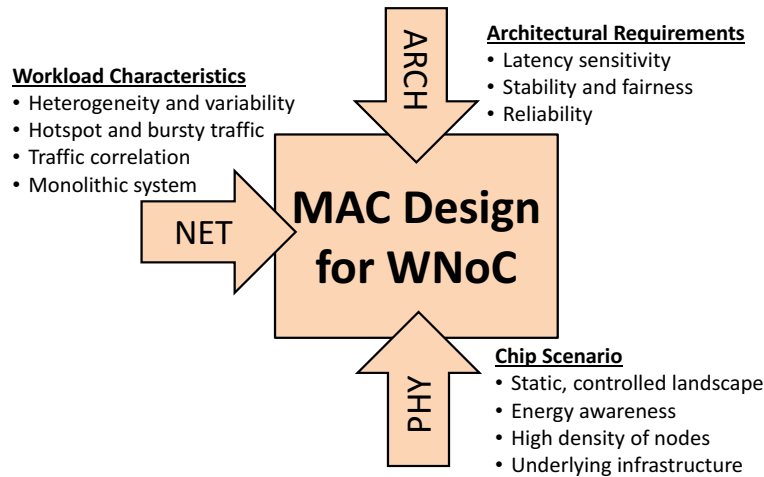


Figure 5.1: Main facets that define the MAC design process in the WNoC scenario.

design that will be used later to propose a custom MAC protocol and specific optimization opportunities for future research.

### 5.2.1 The Chip Scenario

We first focus our analysis on the physical characteristics of the on-chip scenario.

#### Static and controlled landscape

The propagation of the wireless information takes place in a confined space. This physical landscape, including the network topology, the chip layout, and the characteristics of the employed materials, is fixed and known beforehand. This represents one of the main uniquenesses of the WNoC scenario, since nodes in wireless networks generally move within a propagation environment that can be dynamic. The *a priori* knowledge of the physical landscape enables, through an accurate channel characterization, an unprecedented reduction of the randomness of the propagation process. In fact, the channel becomes quasi-deterministic at the data-link layer.

These considerations have profound implications on the MAC design. Designers have a unique control over the transmission range, enabling one-hop communication and virtually eliminating problems such as *hidden* and *exposed* terminals. Therefore, techniques generally used to address these issues and mobility at the cost of complexity and performance, e.g. RTS/CTS, are not necessary. This also opens the door to adaptive methods that may require consensus to work, simplifying distributed protocols. It is finally worth noting that such control over propagation may also enable the detection of collisions, a functionality normally restricted to wired networks that could greatly enhance the performance of random access protocols.

#### High density of nodes

CMPs are expected to reach stunning levels of integration, enabling the development of thousand-core processors [9]. This results in a wireless network where tens to hundreds of wireless nodes communicate simultaneously, context that sets *scalability* as a primary design objective. Although high-density wireless networks are not new, e.g. WSNs [162], nodes in such networks typically decrease their transmission range to reduce contention and resort to

Table 5.2: Differences between traditional wireless networks and the WNoC scenario in terms of MAC protocol design.

	Conventional Scenarios	WNoC Scenario	Implications
<b>Physical Constraints</b>			
Frequency	Up to 60 GHz	60 GHz and beyond	Potentially high impact of propagation delay
Distance	Meters to kilometers	A few centimeters	
Landscape	Dynamic environment	Confined and static	Known range, consensus is easy to achieve
Node Density	Moderate	High	Emphasis on scalability, channelization is discouraged
Energy Policy	Case-dependent	Limited by dissipation: energy-aware	Reduce overhead or penalty of collisions
Auxiliary Network	Expensive, unreliable	Wired, cheap and reliable	Can be used for control, or prevent saturation
<b>Workload Characteristics</b>			
System Design	Multi-vendor standardized system	Monolithic system	Knowledge on traffic, co-design with upper layers, prediction
Packet Length	Variable	Fixed, typically short	Nodes know when transmissions finish, can help fairness
Broadcast	Few sources, generally unreliable	Any node can broadcast, needs to be reliable	Unified broadcast domain is preferable, scalable ACK
Variability	Due to traffic aggregation	Phase behavior; difference between applications	Reconfigurability is desirable
Spatiotemporal Characteristics	Depends on context, correlations may exist	Often bursty and hotspot	Adaptivity to fast changes is desirable
<b>Architectural Requirements</b>			
Latency	Variable, generally not a strong requirement	May be critical	QoS-aware design principles are desired
Throughput	Generally important	Important but not critical	Give priority to latency over throughput
Reliability	Errors can be tolerated	Errors are mostly not tolerated	Strong emphasis on reliability

multiple hops to reach the intended destinations. However, this approach is not advisable here due to the stringent latency requirements of the application.

Scalability requirements also limit the practicality of channelization mechanisms. Creating tens to hundreds of orthogonal channels with the physical constraints of manycore chips is unfeasible due to the increase of complexity of key hardware, e.g., passband filters in frequency multiplexing or synchronization components in code multiplexing. Some works have tried to alleviate this issue by combining different multiplexing mechanisms, e.g. frequency and time in [153]. However, problems associated with the its intrinsically rigid nature of the approach still arise whenever traffic conditions vary. Consider, as a specific yet relevant example, the poor performance of multiplexing with a fair distribution of bandwidth in the presence of hotspot traffic.

Scalability is also an issue in scheduling or random access protocols. In the former case, scheduling nodes need to manage requests from an increasing number of nodes and can easily become a performance bottleneck. In the latter case, designers can expect an increase of the collision probability as more nodes contend for the channel. Also, acknowledging becomes challenging in WNoCs devised to transport multicast traffic due to the expected burst of responses known as the *ACK implosion* [229].

### Energy awareness

Typically nodes in wireless networks are mobile and hence have a limited battery. As a result, a large amount of research have been devoted to the development of protocols that are energy-efficient or even energy-constrained in extreme cases of WSNs. On the contrary, the energy availability is guaranteed in chip environments, yet it cannot be considered unlimited since heat dissipation is expensive. Actually, power has recently become a driver of multiprocessor design, suggesting the use of DVFS techniques or the power-gating of processor cores that are not indispensable during a certain period.

This basically implies that MAC protocols for WNoC need to be *energy-aware* but that, as we will see, should not substantially sacrifice performance to increase the energy efficiency. Latency constraints discourage the use of techniques with low duty cycles as proposed in numerous MAC protocols for WSN [162]. Instead, the protocol should reduce energy wastage by reducing the overhead of the protocol and/or minimizing the penalty of collisions. Additionally, energy could be opportunistically saved by tuning –not turning off– certain parts of the transceiver if a node is not expected to transmit in a while, e.g. during a backoff. Consistently with the DVFS paradigm, protocols should also to adapt to changes in the frequency of cores as these may introduce variations not only in the traffic characteristics, but also on the performance of the wireless link. Last but not least, the correct operation of the protocol must not be disrupted if a given core, including its antenna and transceiver, is powered off. A simple example would be token passing, where the token could be lost when reaching the core wireless unit that has been shut down.

### Underlying wired network

The WNoC paradigm does not aim to replace wired on-chip networks, but rather to complement them. As a consequence, it is reasonable to assume the existence of underlying wired networks that provide a synchronized clock and can efficiently transport unicast flows. This wired backbone is unique to this scenario and can assist the MAC protocol by either transporting control information for scheduling or handshaking purposes, or absorbing traffic originally intended for the wireless network in congestion situations. For instance, we discuss the use of a lightweight wired plane in token passing schemes to increase their scalability in Section 5.4. Synchronization at the processor clock granularity can also be appealing when implementing slotted protocols.

## 5.2.2 Workload Characteristics

Now, we focus our analysis on the characteristics of on-chip traffic. To support the observations here presented, we include a few results of a traffic analysis performed over a tiled architecture with private 32-kB L1-D/L1-I caches, 512-kB of shared L2 per core, and three different flavors of cache coherence: full-map directory with MESI [1], HyperTransport [15], and TokenB [16]. We use GEM5 [230] to extract the communication traces generated by the SPLASH-2 [10] and PARSEC [11] applications running over the architecture under test.

### Heterogeneity and Variability

Although architectures are generally designed trying to avoid expensive communication transactions, manycore processors face the challenge of having to support heterogeneous traffic profiles. Local and unicast communications dominate, but the work by Heirman first demonstrated that the presence of global flows can become significant in cache-coherent processors as the number of cores grows [231]. We performed a similar analysis for multicast traffic: Figure 5.2 shows the number of multicast messages per instruction, as well as the



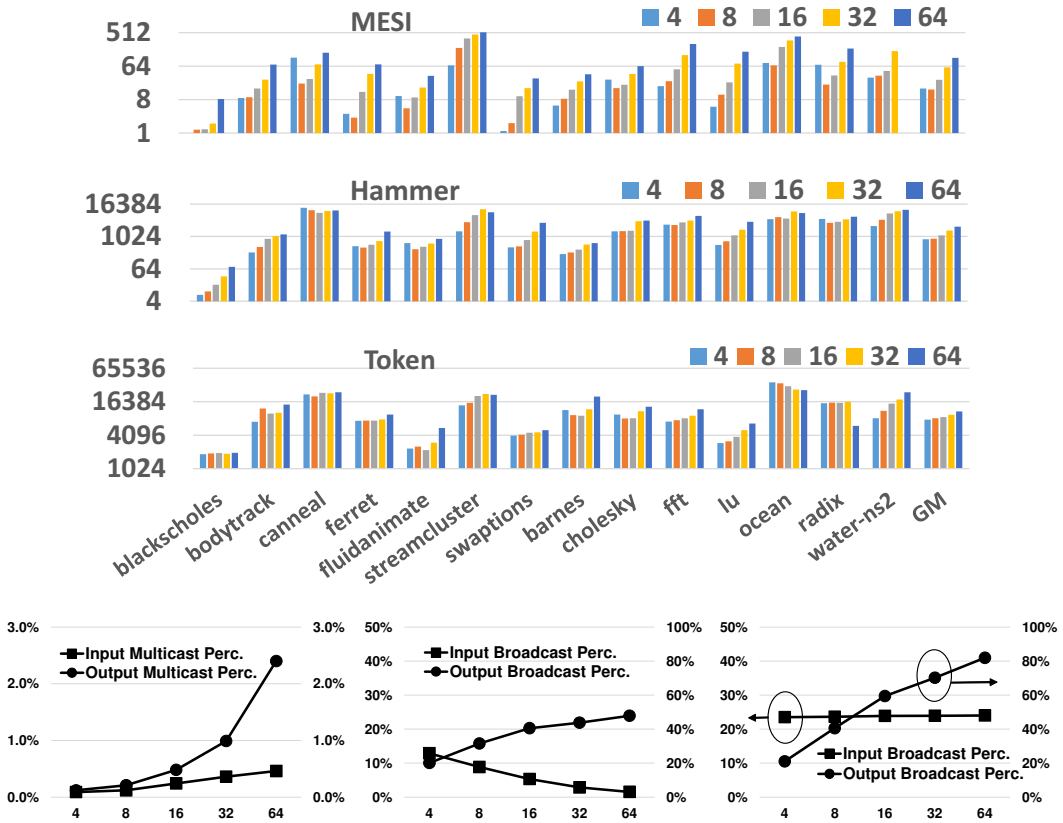


Figure 5.2: Number of multicast flits per  $10^6$  instructions for MESI, HT and TokenB protocols as a function of the processor size (logarithmic scale). Lower charts represent the percentage of multicast flits before and after replication for each respective case.

percentage of flits that are multicast, for a wide set of applications running over different architectures. It is observed that the multicast traffic intensity grows substantially when increasing the number of cores, becoming accountable for more than half of the traffic within the NoC in certain coherence protocols.

On-chip traffic also possesses a large degree of variability. The existence of a wide range of programming models or application domains causes large changes in terms of communication demands from one application to another. As shown in Figure 5.2, there is over one order of magnitude difference between applications that are communication-intensive and those that are not. Note that, within each particular application, phase behavior also leads to wild variations on the traffic characteristics over time [232]. Figure 5.3 exemplifies this by showing how the traffic generated by a particular application varies in an iterative way over its execution. Finally, it is important to note that manycore processors will likely execute different multithreaded applications at the same time, creating a virtually infinite number of workload possibilities. All of these intra- and inter-application changes constitute sources of *coarse-grained variability*.

The coarse-grained variability of traffic suggests that the MAC protocol should be reconfigurable to adapt to large-scale changes with a reasonable cost. This encourages the use of schemes that can be reconfigured periodically [153] or co-design techniques capable of reducing the uncertainty of traffic variations. For instance, application phase prediction

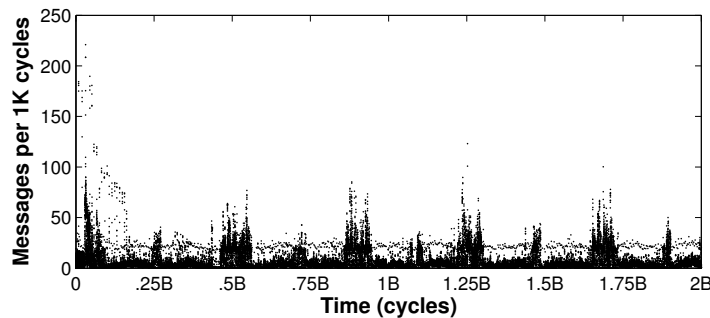


Figure 5.3: Phase behavior exhibited by the traffic generated by the *fluidanimate* application during 2B cycles. Communication-intensive and computation-intensive phase are interspersed.

[232] could be employed to foresee future traffic requirements and reconfigure the MAC protocol accordingly.

### Hotspot and Bursty Traffic

As the predominant programming model in CMPs, shared memory has been assumed in the vast majority of NoC works. Thus, coherence traffic has been characterized for a set of popular architectures and benchmarks with the aim to drive subsequent NoC optimizations and provide tools for their accurate evaluation. The pioneering work by Soteriou *et al* revealed that, in most applications, a large fraction of the traffic is not only generated by a rather small subset of cores, but also injected in bursts [233]. These factors constitute different agents of *fine-grained variability*.

We performed a similar characterization focusing on the multicast traffic targeted by the BoWNoC paradigm. To evaluate the spatial distribution, we calculated the coefficient of variation (CoV) as  $c_v = \sigma/\mu$ , where  $\sigma$  and  $\mu$  are the standard deviation and mean of the multicasts injected by each node. 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 multicast injection over given cores. Figure 5.4(a) plots average CoV calculated over all the applications for the different coherence protocols, confirming an increase of the spatial imbalance of traffic injection as the number of cores grows.

To evaluate the burstiness of traffic, we calculated the Hurst exponent  $H$  ( $0.5 < H \leq 1$ ) applying the RS plot method [233] on the temporal information of the application traces. In light of the results of Figure 5.4(b) and given that an  $H$  value close to 1 denotes strong self-similarity, it can be concluded that multicast traffic is self-similar and that burstiness generally increases with the core count.

Hotspot and bursty traffic profiles are generally detrimental to performance. Token passing protocols struggle when traffic injection is concentrated around a set of cores, whereas many protocols suffer severe performance losses when traffic is bursty [162]. These spatiotemporal characteristics call for flexible solutions that can provide fast and fine-grained adaptivity and would, ideally, devote all resources to the service of nodes in the hotspots or to the absorption of traffic bursts. Multiplexing schemes lack such capability and thus are highly suboptimal within this context. Random access and scheduling protocols are more malleable and can manage hotspot traffic fairly well, but still suffer of high collision rates in the presence of bursty communication. In the latter case, the performance

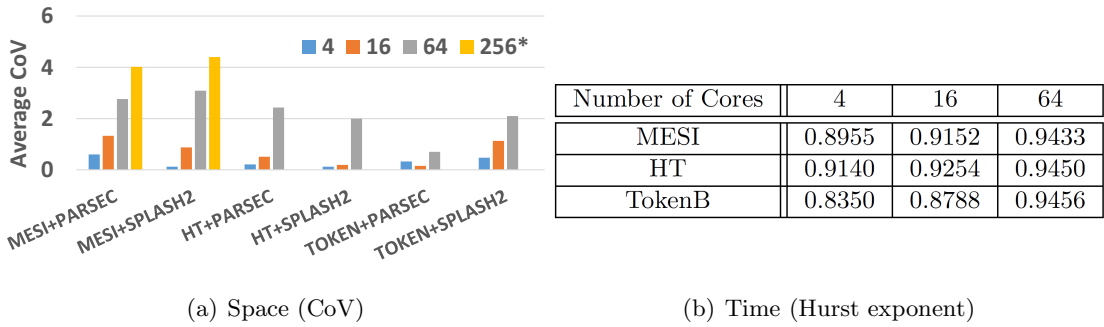


Figure 5.4: Spatiotemporal characterization of the injection distribution of multicasts (geometric mean).

drop could be alleviated by anticipating upcoming bursts and adapting the protocol to them. This could be achieved by identifying recurrent correlation patterns or, as outlined next, exploiting the monolithic nature of the system.

### Traffic Correlation

The iterative nature of computer applications inevitably leads to a certain degree of correlation between certain events [1]. This is the main reason behind the existence of the phase behavior and burstiness effects mentioned above. As we will see in Section 5.4, predictive strategies could be applied to MAC protocols to exploit such correlation characteristics and, thus, to enhance their performance and efficiency. Here, we evaluate the degree of spatiotemporal correlation and the predictability of the multicast flows, which are the main target of the BoWNoC paradigm.

To evaluate spatiotemporal correlation, we consider transmissions from different nodes separated less than a given time period  $\tau$ . The choice of  $\tau$  may depend on several factors and must capture meaningful correlations. A value too low will not yield any correlation, while a value too high will dilute meaningful cases within high correlation probabilities. Besides investigating the amount of correlation between multicasts, we also study how *strong* is this correlation in an attempt to quantify how predictable are these transmissions. To this end, we define the predictability of node X as:

$$pred_X = \frac{\max_i N_{Xi}}{\sum_i N_{Xi}} \quad i \neq X, \quad (5.2.1)$$

where  $N_{XY}$  is the number of transmissions of Y correlated to X. A low value means that crosscorrelation is spread over a set of cores, therefore complicating the prediction (0 if transmissions are not correlated); whereas a high value indicates a strong correlation with few cores (1 if correlation is deterministic). The factor of predictability between any two pairs is calculated as the weighted average of the predictability of each core as

$$pred = \frac{\sum_{i \neq X} N_{Xi}}{\sum_i \sum_{j \neq i} N_{ij}} pred_i = \frac{\sum_i \max_{j \neq i} N_{ij}}{\sum_i \sum_{j \neq i} N_{ij}}. \quad (5.2.2)$$

Figure 5.5 shows the average degree of correlation and the factor of predictability for  $\tau = 50 \cdot T_{CLK}$  of the SPLASH2 and PARSEC benchmark suites. It is observed that both the correlation and predictability levels increase with the number of cores in MESI and HT, supporting the hypothesis that predictive strategies have more potential in larger multi-processors. In TokenB, the predictability is inversely proportional to the number of cores,

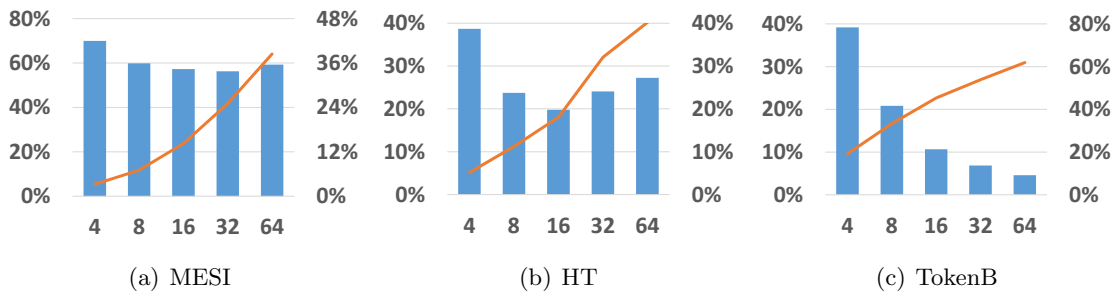


Figure 5.5: Factor of predictability (blue bars, left axis) and degree of correlation (red lines, right axis) with  $\tau = 50 \cdot T_{CLK}$ .

which suggests that the injection of multicast traffic behaves as a uniformly distributed random variable and that predictive techniques would be of little use.

### Monolithic system

A multicore processor is basically a monolithic system from the designers point of view and often a proprietary solution. The team responsible of designing the system therefore has –a rough– control over the entire architecture, from the physical implementation up to the compiler that outputs the code. This represents a big departure from traditional wireless systems where the nodes, the network stack, and the applications are designed and developed by different teams and often rely on open standards.

*The monolithic nature of the system basically implies that protocols can be streamlined by entering into the design loop of the whole architecture, enabling an unprecedented level of optimization of MAC protocols.* The techniques that could stem from this observation differ from those based on the identification of correlation patterns in that, here, we take advantage of the control that we have over the system. We subsequently give a brief description of possible examples where the monolithic nature of CMPs can be exploited.

For instance, the compiler determines –to a large extent– the distribution of the operations that generate the on-chip traffic within a given application, somehow fixing what and when the nodes will transmit. This knowledge could be leveraged to anticipate potentially harming situations or even to avoid these situations using a set of new compiling rules.

A similar approach can be thought from the privileged perspective of the programmer. Consider a message passing system, where the programmer explicitly defines communication using a library of primitives. The MAC layer can be highly optimized by finely tuning the protocol to each primitive, especially to those related to collective communication. Similarly, the programmer could be provided with a set of special instructions that can make explicit the behavior of the protocol for program sections that may exhibit hotspot or bursty behavior.

Finally, one can think on optimizations that take advantage of the knowledge of the architecture. A particular example refers to the architecture developed in Chapter 7 to better support lock and barrier synchronization in shared-memory applications. Both locks and barriers generate global communication, but barriers often do it in bursts caused by the arrival of several threads to the same barrier in a short period of time. In this scenario, the MAC protocol could take the type of synchronization operation into account when determining its assertiveness.

### 5.2.3 Architectural Requirements

A wireless network represents the communication backbone for a certain architecture. This architecture is created for a given purpose or application, which imposes a set of performance objectives that drive the design process. Next, we detail the main objectives of the multiprocessor architecture scenario.

#### Latency Sensitivity

The latency introduced by on-chip communication becomes critical as it essentially delays the progress of operations lying in the critical path of the processor. Supporting this, Sanchez *et al* identified latency as a more limiting factor than throughput in NoCs for cache-coherent manycores [37]. This highlights the appeal of MAC protocols that prioritize latency over throughput while taking into account the energy considerations discussed above.

In wireless networks, latency has traditionally been reduced by increasing the data rate, coordinating the multiple hops towards the receiver, or optimizing the periods of contention. The application of the first design rule to the multiprocessor context discourages the use of techniques that avoid contention by performing a fixed division of bandwidth, and favors random access and scheduling protocols instead. The second rule does not apply here as one-hop wireless transmissions are expected. With respect to the third rule, it is worth noting that the multiprocessor scenario offers unique optimization opportunities arising from the monolithic nature of the system as discussed previously. This information can be used to minimize the collision probability by adopting optimal persistence, backoff, or reservation policies, or by prioritizing latency-critical messages as per architecture, compiler or programmer recommendations.

#### Stability and Fairness

A link layer providing an homogeneous and bounded latency is very appealing feature from an architect perspective, as it potentially reduces the complexity of the upper layers by rendering design decisions easier to reason and verify. This first implies that the MAC strategy must be stable, meaning that its performance should not decrease sharply beyond the saturation load to avoid latency peaks. Random access and scheduling protocols need to be carefully reviewed due to this, whereas channelization techniques are generally stable. In either case, protocols can always resort to the underlying wired NoC to maintain the load below the saturation point.

In practice, latency guarantees are generally addressed by means of protocols with QoS [228], which can apply priority policies to minimize or bound the latency of certain transactions. This will affect fairness, another important aspect that impacts on the latency statistics and that has to be maintained at least among packets belonging to a specific class. In the on-chip scenario, knowledge on the application can guide the application of QoS techniques to reduce the latency of certain critical flows as mentioned above.

#### Reliability

Multicore processors generally require reliable communications, although this depends on the specific architecture. On the one hand, most processors implement error control and correction schemes to provide high-level reliability, as seemingly minor errors may corrupt an entire computation. Within this context, a reliable MAC solution is desirable to minimize the performance reduction caused by errors in upper layers. On the other hand, recent times have seen the emergence of approximate computing [166], which relaxes the need for fully precise computations to provide much higher energy efficiency.

Unlike in traditional wireless networks and as analyzed in Chapter 4, WNoC proposals generally consider a very high BER commensurate with that of RC wires, i.e.  $\sim 10^{-15}$ . As a result, the MAC layer can safely assume that errors in the reception of a message will be caused by collisions in virtually all cases. Assuming a contention-based protocol, the objective will be to adjust the effort spent in resolving collisions according to the specific combination of latency and reliability requirements of the architecture.

An important aspect to consider here is whether the static and controlled environment allows for the proactive detection of collisions. This would bring the contention-based MAC solution closer to that of wired networks, e.g. Ethernet, opening the door to strategies such as negative acknowledging. The implications are twofold. First, the latency of the protocol is highly reduced as there is no need for burdensome timeout approaches. Second, reliable broadcast support can be achieved with a reasonable cost. Note that, in wireless networks, broadcasts are generally *best effort* due to the severe congestion that would be caused by the subsequent acknowledgments. As we will see in next sections, a simple solution here would consist in the transmission of a *tone* signal upon detecting a collision. Tones would be detected by idle nodes, including the colliding transmitters, and interpreted as a negative acknowledgment. If this approach is not technically viable, one can also relay any acknowledgment to the wired network, for which many-to-one traffic optimizations have been developed [20]. In any case, it would be advisable to implement a method to distinguish between errors due to propagation impairments and errors due to collisions, so that protocols can act accordingly without wasting resources.

### 5.3 BRS: A MAC Protocol for Reliable Broadcast in WNoC

The context analysis performed in the previous section provides an excellent starting point for the development of MAC protocols for BoWNoCs. Here, we propose a first design that aims to deliver the reliability, scalability, and flexibility demanded by the application. The protocol is based on well-known carrier sensing techniques and places emphasis on the reliable reception of broadcast messages. Due to the convergence these three facets, we call the protocol BRS-MAC (Broadcast, Reliability, Sensing). Our design considers a single broadband channel, although it could be extended to support multiple channels.

The design decisions of the BRS-MAC protocol are strongly influenced by the different particularities pointed out in the context analysis. The protocol needs to be flexible to adapt to the potentially bursty and hotspot nature of the traffic, and be able to support the transmission of packets of different lengths. It also needs to address the *ACK implosion* issue arising from the reliable broadcast requirement [229]. While scalable, the solution needs to be lightweight given the strong area and power limitations of the scenario. Finally, it is desirable to provide a protocol that can be extended or improved with architecture co-design processes since these are expected to provide unprecedented levels of performance.

In next sections, we detail the BRS-MAC protocol and provide a thorough performance evaluation. The protocol is described in Section 5.3.1. Delay and throughput models for BRS-MAC are built considering both generic environments and the on-chip scenario in Section 5.3.2. In the latter case, the novelty resides in the fact that the location of nodes is fixed and known a priori, which enables the modeling of the performance with much higher accuracy than using conventional assumptions and methodologies. Then, in Section 5.3.3, we integrate the protocol within an on-chip network simulator to experimentally assess its performance. We explore how the latency and throughput vary with (*i*) the number

of nodes; *(ii)* the link capacity; and *(iii)* the percentage of broadcast traffic. The results are finally benchmarked against those of a set of representative MAC protocols based on scheduling and different wireline NoCs. With this, we expect to identify the suitability of BRS-MAC in different scenarios by capturing performance break-even points, as well as justify the possible optimizations proposed in Section 5.4.

### 5.3.1 Protocol Description

BRS-MAC is a random access protocol. This provides greater flexibility and potential for low latency, which contrasts with the rigid channelization proposals of related work [141]. We base our design on the non-persistent version of the Carrier Sensing Multiple Access (CSMA) protocol [224] aiming to provide a reasonable performance in the presence of bursty traffic [162], and to respond to the importance of latency in manycore settings [37]. We adapt the protocol to the particularities of the scenario, aiming to reduce the penalty of collisions and thereby to maximize the throughput. As we will see next, the design objectives are accomplished by implementing a collision detection scheme and acknowledging broadcasts in a fast and scalable way.

Typically, wireless systems with carrier sensing use collision avoidance techniques as they cannot detect collisions, and resort to positive acknowledgment schemes to solve the exposed or hidden node problems. This means that the whole message is transmitted even if a collision occurs, and that the transmitter needs to wait during a timeout period before retransmitting, thereby reducing performance. In wired systems, where collisions can be detected by the transmitter, carrier sensing schemes reduce waste of resources by canceling the transmission as soon as the collision is detected and notified.

The on-chip scenario suggests a solution closer to the collision detection approach. Since nodes are static and the environment is known beforehand, collisions may be detected with high probability. Also, since hidden or exposed terminal problems are eliminated, consensus on the necessity of retransmitting can be reached. Cores may not be able to use the jamming strategies used in collision detection schemes [225], but could resort to negative acknowledgments with the same result instead. Relying on this fundamental aspect, we next detail the principles of operation of the BRS-MAC protocol.

#### Transmission policy

When a node is ready to send data, it will only transmit if the channel is sensed idle, thereby preventing the interruption of on-going transmissions; otherwise, the node backs off. In the former case, the transmission starts and the protocol performs the steps summarized in Figure 5.6:

- **Step 1 – Preamble Transmission:** the sender will transmit a preamble and then wait. If there is only one transmitter, the rest of nodes will correctly receive these initial bits and remain silent; otherwise, there is a collision and nodes will start transmitting a Negative ACKnowledgment (NACK) signal by the end of this step.
- **Step 2 – Collision Handling:** During the second step, the sender listens again to the medium. If the medium remains idle, it means that the first phase of the transmission was successful and the sender will resume it. Otherwise, the presence of a NACK indicates that there was a collision, in which case the original senders will cancel the transmission and back off. Receivers know that there was a collision because they

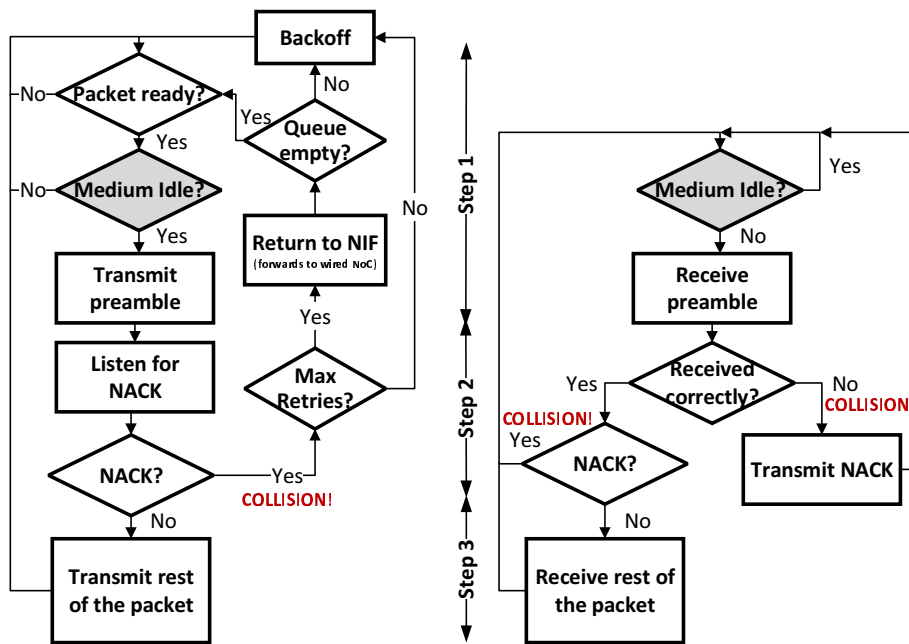


Figure 5.6: Flowchart of the BRS-MAC protocol for the transmitter (left) and the receiver (right). Transitions due to collisions are made explicit with red labels.

either detected it, or hear the NACK signal as well. Thus, a node cannot initiate a new transmission during this phase. By detecting and notifying collisions here, BRS-MAC reduces their penalty as it avoids the unnecessary transmission of the whole message.

- **Step 3 – Data Transmission:** the source transmits the rest of the packet only in the absence of a collision. Otherwise, the channel becomes free for new transmissions. The length of this step will depend on the packet size and transmission speed. Both are known by all since the transmission speed is fixed and the packet size can be indicated in the preamble.

## Unslotted and Slotted Configurations

Note that the algorithm here described can be implemented in both slotted and unslotted configurations. In the former case, the channel is slotted at the processor clock granularity and the duration of the different steps needs to be of an integer number of cycles. This facilitates the integration of the wireless network into the multiprocessor environment and relaxes the synchronization requirements and, due to this, we will consider this design in the simulations of Section 5.3.3. In the unslotted protocol, transmissions can begin at any point in time and not only at the beginning of the pre-defined slots. This increases performance by reducing the length of the vulnerable interval from half of the clock period to only the propagation time. For the sake of mathematical tractability, we will consider this option in the models of Section 5.3.2. In future work we will model the slotted protocol as well.

## Transmission Preamble – Support for Variable Transmission Lengths

The preamble carries packet headers that allow to detect collisions in the second step. The size of the preamble is fixed and should be large enough to ensure the reliability of the



collision detection, but still short to reduce the penalty of collisions. Section 5.3.2 analyzes the impact of the preamble size on the performance of BRS-MAC.

The preamble also allows nodes to determine the length of the third step, which depends on the size of the packet being transmitted. Encoding the transfer length will probably require only a few bits as packet lengths are fixed, known, and take few different values.

## Channel Sensing and Collision Detection

In this part of the work, we do not make any assumptions on the modulation used by the nodes, which can be CW or IR as explained in Section 3.3. In either case, we consider that nodes can sense the presence of signals in the medium and detect collisions.

In wired networks, collision detection is essentially performed by the transmitter, which can sense the medium through a separate channel and compare the input voltage and the transmitted signal. A collision occurred if both signals differ by a large margin. In wireless systems, this scheme cannot be reproduced since nodes generally cannot transmit and sense the medium at the same time. Also, transmitted signals are typically tens of dBs stronger than any received signal, causing the received signals to be *invisible* to the colliding sources and thus invalidating the comparative approach.

For all this, BRS-MAC leaves the collision detection responsibility to the receivers. This is possible thanks to the uniquenesses of the scenario with respect to the static and known propagation medium, which allow to model the channel in a quasi-deterministic way. At the time of this writing, there is not a closed methodology to detect collisions, and we rather propose different approaches to be explored in future research, namely:

- **Redundancy check:** error detection has typically been the only way to know whether a transmission collided or not. However, this is performed at the end of the transmission, wasting precious resources. Here, we propose to perform a special redundancy check over the preamble at the second step of the transmission. In this case, it would be desirable to encode the preamble in order to maximize the error detection probability.
- **Non-coherent detection:** this approach takes advantage of the static and known environment, leading to a quasi-deterministic power budget. By attaching the source address in the preamble, receiving nodes should be able to know what energy level to expect with a high accuracy. A collision is assumed in case there is a significant mismatch between the theoretical and actual received energy, which will be likely caused by the overlapping of two colliding transmissions.
- **Coherent detection:** a strategy similar than the proposed above can be used in coherent systems. This would consist in identifying the phase at which signals should be received depending on the source of the message, and comparing it with that of the actual received signals. Note, however, that this is a more challenging approach from a technical perspective since it requires fine synchronization at the bit level.
- **Correlation:** a recent work [234] proposes the placement of unique signatures at the preamble to discern between correct and incorrect transmissions. A similar approach could be used here: all nodes could use the same (delayed) preamble, which would be received correctly in all cores only in the absence of collisions. With two or more cores transmitting at the same time, the preamble would be corrupted and correlation spike would not appear in at least one of the receivers.

As we will see, the protocol will work as long as one or more receivers detect and notify the collision during the second step of the transmission policy. In the unlikely case of a false

negative, i.e., a collision gone unnoticed by each and every core of the chip, we have to rely on existing error control strategies at the packet level or at higher layers of the architecture. The latter option is costly and should be avoided since it may involve the use of rollback mechanisms. In case of a false positive, i.e., a core detects an nonexistent collision, there will be a certain performance reduction due to the subsequent backoff and retransmission; correctness will not be affected.

### Acknowledgment policy

Due to the different distances among cores and depending on the detection scheme, collisions may go unnoticed at certain nodes. This cannot be tolerated as all receivers need to be aware of all collisions. This suggests the use of a global NACK strategy that warns all nodes that a collision took place. Sending explicit NACK packets would be impractical as it would require the serialization of many messages. To avoid this *ACK implosion* [229], we resort to the unconventional many-to-one communication scheme explained in Section 3.2: we model the burst of NACKs as an aggregated binary signal, the presence of which indicates that one or more cores detected the collision. From the original transmitter perspective, the presence of this signal acts as an advanced NACK that prompts the transmitter to back off and retry later; whereas *silence means consent*, acting as a Clear to Send (CTS) that prompts the source to go on and transmit the rest of the packet. From the receiver perspective, the presence of this signal acts as a *jamming signal* that prompts them to cancel the reception and start over. As we will see, performing this step at an early stage of the transmission reduces the penalty of collisions.

### Retransmission policy

Unless noted, we use the Binary Exponential Backoff (BEB) strategy [225], which calculates the backoff interval  $BO$  as a function of the number of collisions,

$$BO \in [0, BO_0(2^c - 1)], \quad (5.3.1)$$

where  $BO_0$  is the nominal backoff length and  $c \in [0, c_{max})$  is an integer that is incremented on every collision and decremented on every successful transmission. Therefore, this approach reduces the assertiveness of the protocol as the collision rate increases. The nominal backoff length is chosen *a priori* as a function of the number of nodes and channel capacity. In future iterations, though, we envisage the use of prediction techniques to adapt the persistence and backoff intervals to the input load. Such strategy is not commonly used in wireless networks due to the lack of consensus, but could work in this scenario as there are no hidden or exposed terminals.

In the unlikely case that a packet in any MAC queue exceeds a given number of retries, it will not be retransmitted through the wireless network again and, instead, will be forwarded to an alternative network plane. To this end, the head of the MAC queue is popped and sent back to the network interface, which will forward it to the alternative network. At that point, the retry counter is reset and the transmitter backs off if the queue is not empty. We refer the reader to Section 6.2.1 for more details on this.

Table 5.3: Notation for BRS-MAC performance analysis.

Symbol	Concept
$S$	Link throughput
$G$	Offered Load
$D$	Average transmission delay
$U$	Duration of successful transmissions
$B$	Duration of busy periods
$I$	Duration of idle periods
$d_{ij}$	Distance between nodes $i$ and $j$
$a_{ij}$	Normalized propagation time between nodes $i$ and $j$
$a$	Normalized propagation time (worst case)
$\alpha$	Normalization factor for exact propagation time evaluation
$T$	Average transmission time of a complete packet
$T_{OK}$	Time spent in a successful transmission
$T_{KO}$	Time spent in a collision
$b$	Normalized preamble transmission time
$P_b$	Blocking probability
$N_{re}$	Average number of retransmissions
$N_c$	Average number of collisions
$BO$	Duration of the backoff period
$N$	Number of nodes
$E\{\cdot\}$	Expected value

### 5.3.2 Performance Analysis

This section presents the analytical models for the performance of the unslotted version of BRS-MAC. Since BRS-MAC is based on the non-persistent CSMA protocol, we use the notation and models by Kleinrock and Tobagi [224] as starting point. We maintain the definitions of throughput  $S$ , offered traffic  $G$ , and delay  $D$ . We keep the propagation time  $a$  and add the preamble time  $b$  as main parameters. All variables are normalized by the average transmission time  $T$ , which takes into consideration the packet size distribution and transmission probabilities. The reader will find a summary of the notation in Table 5.3.

The models also maintain most of the system assumptions by Kleinrock [224], which let us use renewal theory and obtain closed-form expressions. We basically consider that the time required to switch between TX and RX modes and to detect the channel busy are negligible. All arrivals (including retries) follow a Poisson process and are uniformly distributed among an infinite population, which is reasonable considering the high number of nodes in a manycore processor. We also consider a deterministic backoff instead of BEB to keep the model tractable. Finally, unlike in [224], our work does not make assumptions about the NACK channel as BRS-MAC uses the same channel for data and NACKs.

Next, we present the analytical model for the unslotted version of BRS-MAC and study its throughput and latency for different propagation time and preamble length conditions. In the interest of comparison, we take a non-persistent CSMA protocol as the baseline.

#### 5.3.2.1 Walkthrough Example

In order to better understand the timing of the protocol and the models presented below, Figure 5.7 shows how BRS-MAC operates in the presence of a collision and for a successful transmission. The diagram includes the duration of the different steps and defines busy and idle periods, which are used later in the model. The specific example shows three cores out of a CMP of more cores. The propagation time between these three cores takes different

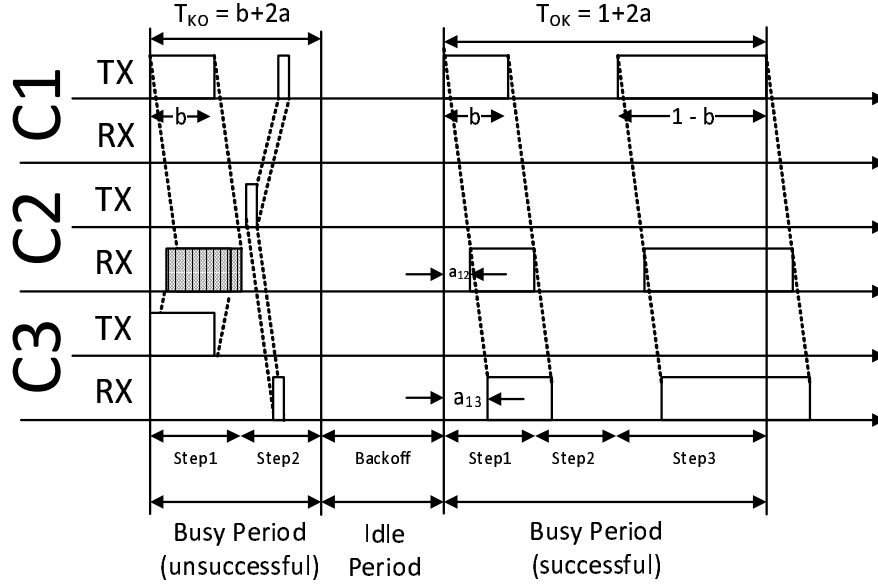


Figure 5.7: Operation of BRS-MAC both for a collision and a successful transmission.

values, e.g.,  $a_{12}$  and  $a_{13}$ , whereas we denote the longest propagation time between any pair of cores of the CMP as  $a$ .

At an initial time, cores 1 and 3 attempt to transmit a message by sending the preamble during  $b$  normalized units. On the one hand, the preamble is not heard by cores 1 and 3 since they cannot receive while transmitting; on the other hand, core 2 receives a combination of both preambles, which is seen as a collision. Then, core 2 answers with a very short NACK signal, which is received by cores 1 and 3 that are now listening to the shared medium. The preamble and NACK steps take  $b + 2a$  units in total.

After sensing the collision, cores 1 and 3 back off. Suppose that core 1 chooses a shorter backoff time and retries sooner. In this case, the preamble is correctly received, the channel is sensed idle during the NACK period, and finally core 1 can transmit the rest of the message. The correct transmission takes  $1 + 2a$  normalized units in total.

### 5.3.2.2 Propagation Time in Manycore Processors

As in [224], we will initially consider an equal propagation time for all nodes, which is an assumption generally taken due to the unknown and dynamic position of nodes. However, knowing the propagation environment *a priori* allows to determine the exact propagation time among nodes (e.g.  $a_{12}$  and  $a_{13}$  in Fig. 5.7). This is important since unslotted CSMA protocols (including BRS-MAC) define a *vulnerability interval* that depends on the propagation time and during which transmissions can collide. Therefore, we can evaluate these protocols with higher accuracy as we can consider the exact propagation times.

To relate classical models with our models with exact propagation times, we relate the average distance between cores with the  $a$  parameter, which represents the normalized propagation time in the worst case scenario and is given by

$$a \simeq \frac{WC\sqrt{2}}{v_p L}, \quad (5.3.2)$$

where  $W \sim 1$  cm is the side of a square chip,  $C$  is the channel capacity,  $v_p$  is the propagation speed, and  $L \sim 100$  bits is the average packet length.

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

$$\alpha = \frac{1}{N(N-1)} \sum_{\forall i} \sum_{\forall j \neq i} d(i, j) \simeq \frac{\Delta(2)}{\sqrt{2}} \simeq 0.3687, \quad (5.3.3)$$

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

### 5.3.2.3 Throughput Model

The throughput of the network is calculated as:

$$S = \frac{E\{U\}}{E\{B\} + E\{I\}}, \quad (5.3.4)$$

where  $E\{U\}$  is the expected useful occupancy of the channel of successfully transmitted messages. The term  $E\{B\} + E\{I\}$  represents a cycle, i.e. the time between two transmissions taking into consideration the average duration of the busy and idle periods, respectively. The throughput takes a value between 0 and 1, and *can be seen as a metric of the effective use of the wireless medium* [224].

The expected duration of successfully used slots  $E\{U\}$  can be obtained by multiplying the useful transmission time (not including the collision handling overhead) and the probability of success, which corresponds to the probability that no terminal starts a transmission during the propagation time  $a$ . By assuming Poisson traffic and an identical propagation time among all nodes, we have

$$E\{U\} = e^{-aG}. \quad (5.3.5)$$

The expected duration of the idle period  $E\{I\}$  is given by the source inter-arrival time, which is given by:

$$E\{I\} = \frac{1}{G}. \quad (5.3.6)$$

The expected duration of the busy period  $E\{B\}$  takes into account the propagation times on the duration of successful and colliding transmissions. Since the duration of the NACK message and the switching delay from transmitting and receiving and *vice versa* are neglected, the length of the collision handling step is equivalent to a propagation round trip time  $2a$ . Thus, as shown in Figure 5.7, successful transmissions occupy the channel during  $1 + 2a$ ; whereas in colliding transmissions the *jamming* nature of the NACK burst ensures that the cancellation takes effect after  $b + 2a$ . Thus, we have:

$$E\{B\} = e^{-aG}(1 + 2a) + (1 - e^{-aG})(b + 2a). \quad (5.3.7)$$

Finally, we obtain the throughput with Equation (5.3.4) as

$$S = \frac{e^{-aG}}{e^{-aG}(1 - b) + b + 2a + 1/G}. \quad (5.3.8)$$

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

$$E\{U^e\} = \frac{1}{N} \frac{1}{N-1} \sum_{\forall i} \sum_{\forall j \neq i} e^{a_{i,j}G}. \quad (5.3.9)$$

It is worth noting that this expression cannot be simplified. However, when the probability of collision is low, it is possible to use the Taylor approximation for the exponential term ( $e^x \simeq 1 - x$ ) and apply Eq. (5.3.3) to obtain

$$E\{U^e\} \simeq 1 - G\alpha a. \quad (5.3.10)$$

In the case of  $E\{B^e\}$ , we need to evaluate the average duration of the busy period for each pair of nodes  $i, j$ . This requires re-evaluating Equation (5.3.7) which, using the Taylor expansion approximations outlined above, becomes

$$E\{B^e\} \simeq 1 + (2 + \alpha)a - (1 - b)G\alpha a. \quad (5.3.11)$$

Thus, the throughput  $S^e$  can be re-calculated using Equation (5.3.4), which yields

$$S^e \simeq \frac{(1 - G\alpha a)}{1 + (2 + \alpha)a - (1 - b)G\alpha a + 1/G}. \quad (5.3.12)$$

Using the same methods and approximations, it is easy to prove that the baseline CSMA becomes

$$S_{CSMA}^e \simeq \frac{(1 - G\alpha a)}{1 + a + 1/G}. \quad (5.3.13)$$

#### 5.3.2.4 Delay Model

Here, we calculate the time required in average to successfully transmit a packet. We define the transmission delay  $D$  as the time between the instant the packet is generated until the packet is successfully delivered to *all* receivers. The delay can be expressed as

$$D = N_{re}E\{BO\} + N_c T_{KO} + T_{OK}, \quad (5.3.14)$$

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

To calculate the average number of retransmissions  $N_{re}$ , we simply need to resort to the definition of offered rate and throughput. The offered rate  $G$  includes the retransmissions, whereas the throughput  $S$  only includes transmission attempts that succeed. Thus,

$$N_{re} = \frac{G}{S} - 1. \quad (5.3.15)$$

To calculate the number of collisions per packet  $N_c$ , let us denote  $P_b$  as the probability of finding the channel busy within a cycle. We have that  $1 - P_b = \frac{a+1/G}{E\{B\}+E\{T\}}$  as arrivals see

the medium free in idle periods plus the propagation time [224]. From this, we can obtain the average number of attempts that find the channel free as  $(1 - P_b)\frac{G}{S}$  [224]. Since the number of collisions is the number of attempts minus one, we have

$$N_c = \frac{a + 1/G}{E\{B\} + E\{I\}} \frac{G}{S} - 1. \quad (5.3.16)$$

**Exact propagation time:** when considering exact propagation times, the delay model is slightly different. We basically need to consider  $S^e$  instead of  $S$  in Eq. (5.3.16), and reevaluate the average number of collisions as

$$N_c^e = \frac{\alpha a + 1/G}{E\{B^e\} + E\{I\}} \frac{G}{S^e} - 1 \quad (5.3.17)$$

regardless of the protocol choice.

### 5.3.2.5 Validation and Results

In this section, we use the models developed in Sections 5.3.2.3 and 5.3.2.4 to obtain the throughput and delay of the BRS-MAC protocol under different configurations. We will compare the results with those of the ideal non-persistent CSMA model developed in [224]. Also, in order to validate the applicability of the proposed models, we have implemented an ad-hoc event-driven simulator and compared its output with the analytical results. The simulator considers exponentially distributed arrivals, a random uniform injection distribution, and the exact propagation time among nodes. We have simulated a homogeneous manycore processor with 1024 cores and obtained  $G$  by averaging the number of attempts of all nodes, and  $S$  by averaging the number of successful transmissions. Unless noted, we assume  $a = 0.1$ ,  $b = 0.1$ ,  $BO = 40$  ns, as well as a transmission time of  $T = 1$  ns, achievable with speeds around 100 Gbps considering the length of packets in the CMP scenario as discussed in previous sections.

To validate the models, we compare the throughput obtained with the simulator and the models for both BRS-MAC and CSMA, considering both worst-case and exact propagation time. The plots, shown in Figure 5.8, reveal a good agreement between the model and the simulation results for  $G \leq 10$ , which is the input range of interest. A significant performance improvement is also observed when considering the exact propagation time. In light of its validity, we will henceforth use the exact propagation time model.

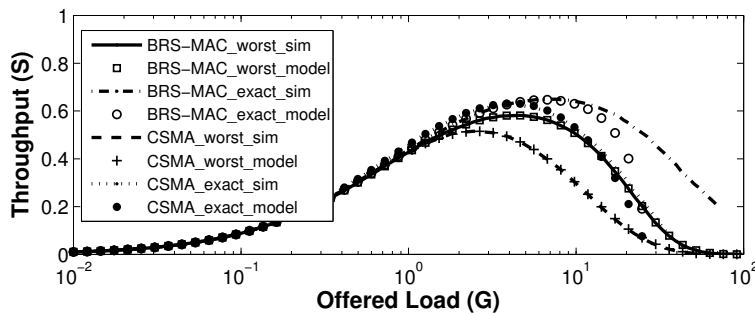


Figure 5.8: Simulated and analytical throughput for worst-case and exact propagation times.

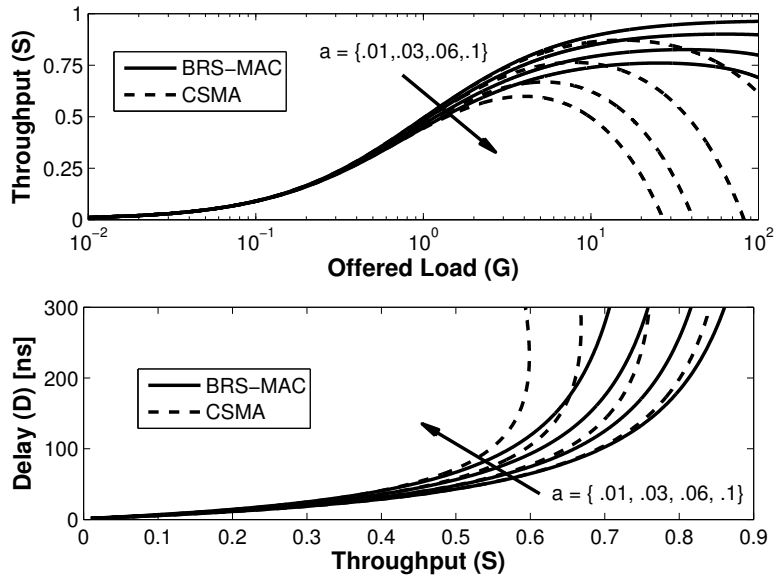


Figure 5.9: Throughput and delay characteristics as functions of the propagation time.

Figure 5.9 plots the throughput and delay characteristics of the evaluated protocols considering different propagation times. Increasing the propagation time negatively impacts on performance due to an increase of the collision rate. Thanks to its early reaction to collisions, BRS-MAC outperforms CSMA for all propagation times, with the peak being 10%–26% higher. This throughput improvement also relates to the delay: at a fixed delay of 200 ns, BRS-MAC admits 2%–11% more throughput.

Let us now evaluate the impact of the preamble time on the performance of BRS-MAC. Figure 5.9 shows the throughput and delay characteristics of BRS-MAC for different preamble lengths. The plot reveals that handling collisions using the same channel than the transmission itself incurs into an overhead with respect to CSMA. This is compensated

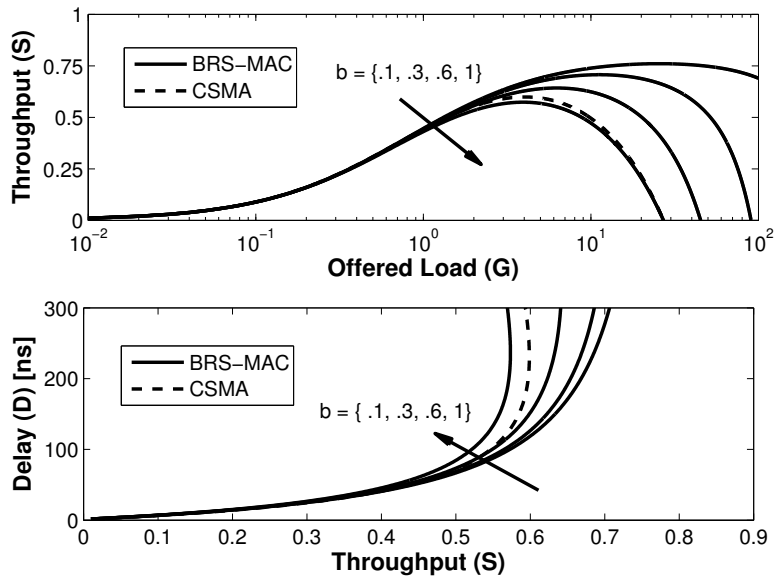


Figure 5.10: Throughput and delay as functions of the position of the preamble.



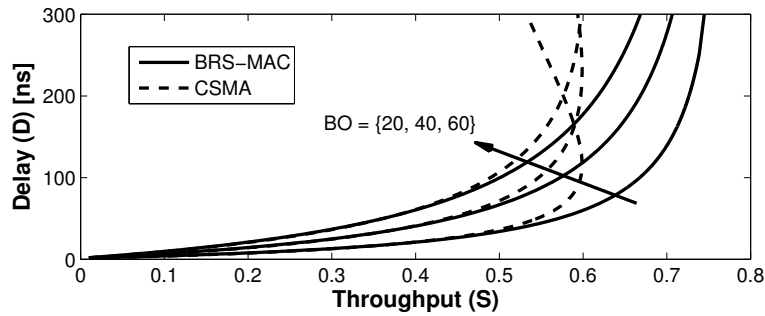


Figure 5.11: Delay as a function of the throughput for different backoff values.

when collisions are handled at the beginning of the transmission ( $b = 0.1$ ), which actually improves throughput up to a 27%. A similar behavior is observed with respect to the delay: at 200 ns, the throughput increases 9%–12% with respect to CSMA. Note that  $b = 0.1$  corresponds to a preamble of 10–30 bits in light of the size of messages in CMPs [37].

To showcase the importance of the backoff mechanism, Figure 5.11 plots the latency of BRS-MAC and CSMA for different backoff values. Both show a lower delay when the average backoff time is low, but performance can quickly degrade in high-contention phases. Alternatively, larger backoff periods imply longer delays but allow the network to better absorb large bursts of traffic. Comparatively, BRS-MAC delivers a much better performance at high throughput due to the lower impact of collisions. Finally, note that these results consider a fixed average backoff time; real scenarios will account for the exponential backoff algorithm depicted in previous sections which should, theoretically, provide a lower delay at low throughput values and a similar saturation throughput.

### 5.3.3 Comparative Scalability Exploration

The main objective of this section is to contextualize the performance of a BoWNoC with BRS-MAC by comparing it against other WNoC designs and state-of-the-art wireline NoCs. To this end, we evaluate how the latency and throughput of different architectures scale as a function of the number of nodes  $N$ , the capacity of the wireless channel  $C$  and the percentage of broadcast traffic  $\beta$ . We also provide hints on the behavior of the different alternatives in the presence of bursty and hotspot traffic. Unlike in the previous section, this evaluation considers the slotted version of BRS-MAC and tests its implementation within a fully functional and cycle-accurate simulator. Here, we first describe the evaluation framework, to then detail the comparative evaluation results that shed light on the broadcast scalability of the BRS-MAC protocol and the rest of wired and wireless alternatives.

#### 5.3.3.1 Evaluation Framework

We use PhoenixSim [235], a cycle-accurate NoC simulator, to perform the evaluation. As detailed in Appendix A, PhoenixSim includes a complete set of methods for the evaluation of switched NoCs, on top of which we implemented the modules required for the simulation of WNoCs. Table 5.4 shows a summary of the simulation parameters, whereas the input traffic profiles and investigated network architectures are subsequently detailed.

Table 5.4: Simulation Parameters

System	400 mm <sup>2</sup> die, 1 V, 1 GHz, 16 to 1024 cores.
Wireline NoC	128 bits, XY routing, 4 virtual channels, credit-based flow control, fixed tree multicast, single- and multiport allocation. MESH: 2 cycles per hop. FBFLY: 3-7 cycles per hop.
Wireless NoC	Single channel, 8 to 160 Gbps, 1-cycle token passing delay, 1-cycle from and to central buffer. CSMA: Non-persistent, NACK burst, $a \leq 0.1$ , 128-bit preamble, truncated exponential backoff, 8 maximum retries.

### Traffic Profiles

Initially, cores are modeled as generators of memoryless traffic with a constant arrival rate  $\lambda$  over time. Broadcasts represent a fraction  $\beta$  of all the packets, while the rest  $(1-\beta)$  are unicasts. Since the main objective is to inspect the broadcast performance, we consider a simple random uniform pattern for unicast traffic. Even though we do not want to bind to specific architectures, we consider two packet sizes as commonly found in cache-coherent systems [1]: short for requests and long for responses (the size of one address and one address plus one data block, respectively). Here, we assume these to be equivalent to 1 flit and 4 flits. This simple synthetic traffic is used to investigate the scalability of different networks under a broad range of conditions. To provide hints of performance in more realistic scenarios, we later perform a sensitivity analysis considering bursty and hotspot traffic. We refer the reader to Appendix A for more details on the traffic generator used in this part of the dissertation, as well as its validation.

### Investigated Architectures

We aim to cover a large fraction of the solution space by considering the following schemes. Unless noted, acknowledging is implicit since negligible bit error rates can be assumed with proper coding and power allocation. Wireless capabilities are given at the core level.

**Carrier Sensing (W-CSMA)** with which we aim to represent contention-based protocols.

We model the slotted version of the BRS-MAC protocol explained in Section 5.3, using non-persistence and adopting the NACK burst mechanism to reduce the control overhead. The preamble size is of 128 bits, which yields a normalized preamble size of  $b \simeq 0.4$  for the traffic considered here. For the wireless transmission speeds depicted in Table 5.4, the propagation time is  $a \leq 0.1$ . A packet will be forwarded to an alternative network plane 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 [226]. One full packet can be transmitted in each round. We do not split long messages into flits here as the packet latency would be unacceptable, whereas bulk transmissions are not allowed for fairness reasons. Upon completion, or in case there is nothing to transmit, the token is handed off to the next core. We assume that the token passing is performed through a lightweight wired ring and is pipelined with the wireless transmission. Since the token somehow divides time in slots, we can

consider this as feasible way to implement a multiple access scheme based on dividing the spectrum in orthogonal channels.

**Centralized Buffer Arbitration (W-CBUF)** for the sake of comparison, we also study the performance of a centralized MAC scheme. With unlimited resources, it would be possible to have an arbiter connected to every core with one-cycle bidirectional links and that works as follows. When a node is ready to transmit, it sends a request to the arbiter with its identity and the packet size. The arbiter stores this in a FIFO buffer and grants access to the node whose request is in the buffer head, waiting exactly the wireless transmission time between consecutive grants. Contention only appears when multiple nodes request access at the same clock cycle. This is resolved by the arbiter. This scheme therefore provides fair, ordered and contention-free access in a flexible way, with resources that are not available in other wireless networks. The main reason for evaluating this scheme is to motivate unconventional MAC designs and to quantify the improvement margin of the protocols mentioned above.

For benchmarking purposes we compare the performance with two wired topologies, considering two router microarchitectures in each of them:

**Routed Mesh (MESH)** as baseline, we consider a wireline 2D mesh with two cycles per hop in the absence of contention. This is achievable with bypass strategies [21]. The choice of a mesh topology is backed up by extensive use as baseline design in the literature. Flow control is credit-based and uses the data links to exchange buffer utilization information between neighboring nodes. Tree-based multicast is studied due to its better latency scalability, which becomes a critical performance factor in many-core scenarios [37]. Note, though, that path-based and tree-based methods become very similar for broadcast (e.g. column-path routing [18] is equivalent to fixed tree with XY routing). We consider two router microarchitecture flavors: single-port allocation in **MESH-BASE** routers, and multiport allocation with multicast crossbar in **MESH-FT**. The delay is proportional to and independent of the number of flit replications in the former and latter case, respectively. These configurations are equivalent to the **EMesh** profile evaluated in Chapter 4.

**Flattened Butterfly (FBFLY)** given its much lower diameter, this topology represents a more aggressive competitor. As described in [36], FBFLY generally employs 4-way concentration and then connects each router with every router in its row or in its column. Thus, less than 4 hops suffice to reach any core from any other core. This comes at the cost of increasing the radix of the routers and complicating the design of its arbiter and crossbar as the network is scaled [37]. These designs do not scale well in terms of area and power (see Section 4.2.3), but are evaluated as to include low-diameter NoCs in the design space exploration. Due to the increase in design complexity, we assume that the pipeline depth logarithmically increases with the router radix, for a total of 3 to 7 cycles per hop. As in **MESH**, we consider fixed tree routing and two types of router: single-flit allocation in **FBFLY-BASE** and multiport allocation in **FBFLY-FT**.

### 5.3.3.2 Scalability Results

Next, we present a comparative evaluation of the network architectures outlined above. First, we measure latency as the time passed between the generation of one message and its the complete reception at *all* the intended destinations. Then, we measure throughput from

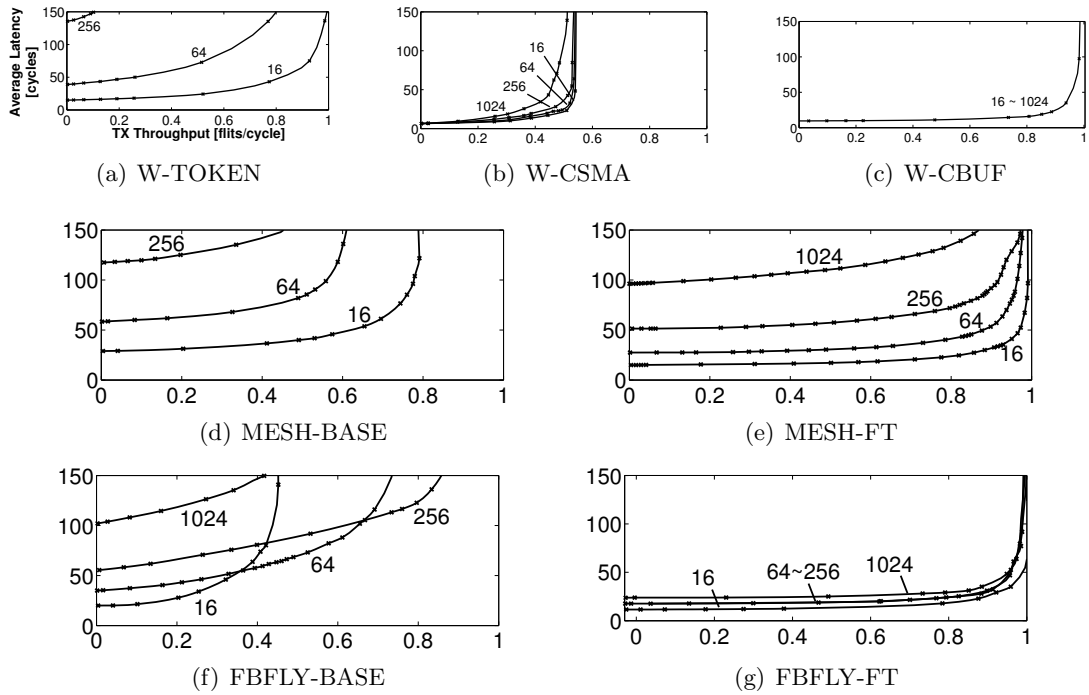


Figure 5.12: Latency-throughput curves for representative system sizes for 100% of broadcast traffic and a wireless capacity of one flit per cycle.

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

To obtain the latency-throughput characteristic of each network architecture, we gradually increase the offered load. For simplicity, though, we will generally show two critical metrics: the low-load latency and the saturation throughput. The former corresponds to the average communication latency in the presence of mild contention, and models the performance of the network for at least 50% of the maximum admitted load with reasonable accuracy. The latter evaluates the throughput of the network for very high loads. For the sake of fairness, we measure the throughput obtained when the latency reaches a given common limit instead that of simply measuring the load at which the network saturates (both metrics may differ). The limit is set to 150 cycles, a value commensurate with the latency of accessing data in the main memory. Relating the throughput to a bounded latency is important given the characteristics of the scenario as explained in Section 5.2.3.

### Scaling the Number of Nodes

Figure 5.12 shows the latency-throughput characteristics of the considered network architectures for a set of representative system sizes  $N = \{16, 64, 256, 1024\}$ , assuming a wireless capacity of one flit per cycle and only broadcast traffic. As intuition suggests, the increase in system size has a significant impact upon performance. In conventional NoCs, the latency increases due to the growth of the number of hops required to reach all destinations, while the throughput suffers a contained drop. In wireless NoCs, the performance both in terms of latency and throughput depends on whether the protocol is fixed or works on

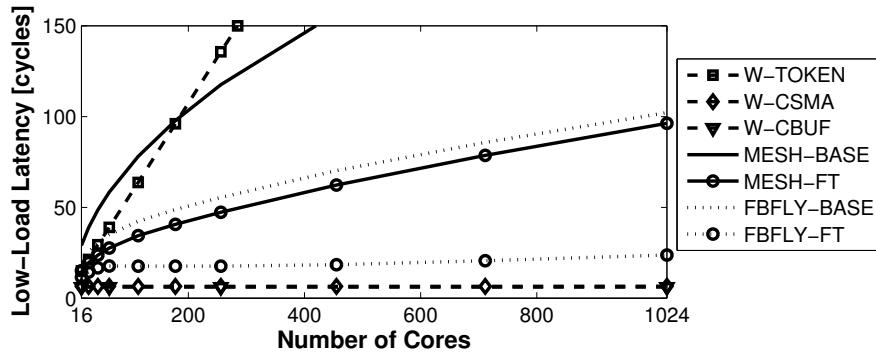


Figure 5.13: Low-load latency of broadcast transmissions as a function of the number of nodes  $N$  for  $C = 1$ .

demand. A notable case is that of **W-CSMA**, which saturates significantly earlier than the rest of alternatives. This is basically due of collisions: retries compete with newly generated packets, causing the throughput to gradually become lower than the offered load. Next, we analyze the results in more detail.

The behavior of the low-load latency as a function of the system size is shown in Figure 5.13. Three different scaling trends can be clearly identified. First, the latency of **W-TOKEN** scales linearly with the number of cores due to delay introduced by the arbitration phase. Since the token is passed through a ring, the latency scales as the average hop distance of such topology,  $O(N/2)$ . Due to their on-demand nature, the latency of the rest of wireless schemes remains flat given that, at low loads, a node will most likely be able to transmit immediately. In the wired NoCs, the latency scales proportionally to the average hop distance of the topology:  $O(\sqrt{N})$  in **MESH** and almost constant in **FBFLY**. In both topologies, the latency observed in their base configurations is considerably higher due to the additional delay incurred by the flit forking process. (**FBFLY** actually loses its scalability advantage since every flit spends  $\propto N$  cycles in each router).

For the sake of comparison, it is important to remark the results from related work. The NoC implemented in [21] is expected to show a similar scaling trend than **MESH-FT**, but with a lower absolute value since it attains one cycle per hop. The scheme in [44] is a mesh with unconventional multi-hop bypass links, and achieves a latency (not counting router-to-processor communication) as low as 5.6 clock cycles for 64 cores. Although its scalability is theoretically close to **FBFLY-FT** as it is potentially able to reach all cores in two network hops, several technological assumptions need to be made to confirm this potential [32].

Figure 5.14 illustrates how the throughput of the different schemes scales with the system size. In **MESH-BASE** and **FBFLY-BASE**, the throughput decreases with the number of cores mainly because latency scaling induces the network to reach the latency limit at lower loads. With unbounded latency, the increase in terms of bisection bandwidth would mostly compensate the increase in number of destinations per message and the throughput would remain constant. This is the case for **RMESH-BASE** and **FBFLY-BASE**, which are below the latency limit and almost achieve maximum throughput. Given by the inherent broadcast nature of wireless NoCs and in spite of having a much lower bisection bandwidth, **W-CBUF** and **W-CSMA** also achieve a rather flat scaling with a lower absolute value in the latter case. Finally, we have that **W-TOKEN** is clearly dominated by the token passing delay and that it is not able to provide any throughput with acceptable delay beyond a few hundreds of cores.

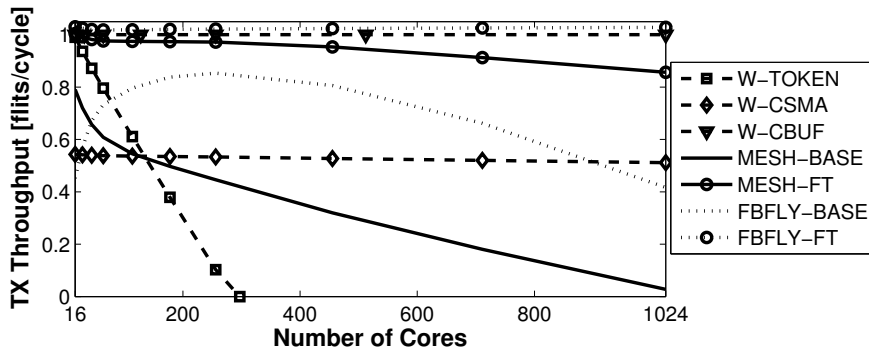


Figure 5.14: Throughput of broadcast transmissions at the maximum admissible latency (150 cycles) as a function of the number of nodes  $N$  for  $C = 1$ .

Again, we compare our results with those of related work. The 16-core NoC implemented in [21] reports a throughput similar to that our MESH-FT. Since it is able to perform single-cycle hops, it will probably provide better throughput scalability than the meshes evaluated here. The work in [44] reports a throughput of  $\sim 0.9$  flits per cycle for 64 cores. This value directly competes with MESH-FT, FBFLY-FT, or W-CBUF. The rest of wireless alternatives will need to improve to be comparable to it in large systems.

### Scaling the Channel Capacity

For the conditions evaluated above, wireless strategies are capable of consistently achieving very low latencies with moderate-to-high throughput. However, we have assumed a bandwidth of one flit per cycle thus far, which is around 160 Gbps in a system running at 1 GHz and with 128-bit links and including the propagation latency. As discussed in Chapter 4, these figures may not be available in the near future. Therefore, it is important to understand the dependence between performance and channel capacity in order to guide the design of future WNoCs.

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

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

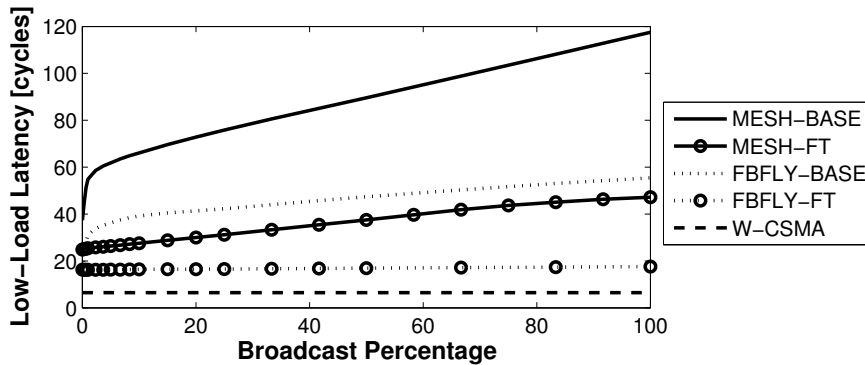


Figure 5.15: Low-load latency as a function of the percentage of broadcast traffic  $\beta$  for  $N = 256$  and  $C = 1$ .

### Scaling the Broadcast Intensity

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

In the absence of contention, the latency of a broadcast transmission is equivalent to the latency of reaching the furthest destination. As shown in Figure 5.15, this causes the latency to drop as the broadcast probability decreases. The impact is more patent in the base configurations, since it takes several cycles to complete the flit forking process in each router. Remarkably, there is no break-even point of latency even with the best wired alternative: given enough channel capacity and due to its unique one-hop communication capabilities, the latency of any wireless transmission will be always the lowest.

From a throughput perspective, the percentage of broadcast traffic has a very strong impact on performance. As illustrated in Figure 5.16, the throughput is of a several tens of flits per cycle in the wired topologies and decreases as each message has to reach more destinations. Approaching  $\beta = 100\%$ , the performance of wireline and wireless schemes become comparable despite the huge difference in terms of bisection bandwidth, to the point that break-even points with MESH-BASE are achievable given enough wireless capacity and MAC efficiency. This, together with the latency results above, suggests that the wireless plane could be used not only for broadcast transmissions, but also for selected latency-sensitive unicasts to further enhance performance.

On the one hand, we estimate that the work in [21] will provide curves very similar to that of MESH-FT. On the other hand, it remains unknown how the multihop capabilities of [44] will affect throughput in mixed traffic, given that broadcasts that occupy several router ports within the same clock cycle may greatly affect unicast transmissions.

#### 5.3.3.3 Sensitivity Analysis

Section 5.2.2 has revealed that on-chip traffic is often injected by a small subset of nodes and in a bursty manner. This fact has a notable impact upon the performance of the chosen NoC and, thus, should also influence its design. Here, we evaluate the sensitivity of the different network architectures to the traffic characteristics using, to this end, synthetic

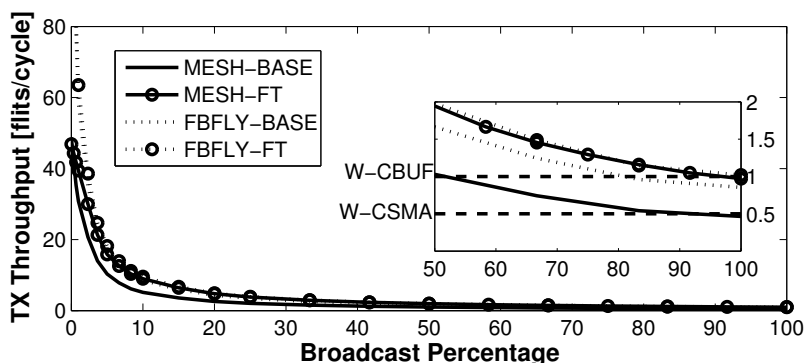


Figure 5.16: Throughput at the maximum admissible latency (150 cycles) as a function of the percentage of broadcast traffic  $\beta$  for  $N = 256$  and  $C = 1$ .

traffic generated with the method described in A. Due to the random nature of the method, we performed 15 runs for each design point and calculated the geometric mean. Results are normalized to the exponential uniform random traffic case.

**Hotspot traffic:** spatial concentration typically reduces the network throughput due to the uneven use of resources. Figure 5.17 illustrates this effect by plotting the throughput for different levels of hotspot traffic. As we reduce  $\sigma$ , the injection process becomes more concentrated, significantly impacting the performance of most networks. W-TOKEN suffers an important reduction since the token needs to travel around the ring even if the processors willing to transmit are highly clustered, whereas W-CBUF and W-CSMA perform well independently of the injection profile. Concentration is even helpful in the latter case, as it reduces the average number of contending stations. In wired schemes, concentration creates congestion around the source, particularly in those configurations with high per-hop time, e.g. FBFLY-BASE.

**Bursty traffic:** Figure 5.18 shows the performance of the different schemes for increasing levels of burstiness. All networks see their performance reduced due to the backlogging of flits in routers and interfaces during packet bursts. This increases the mean latency and reduces the achievable throughput due to momentary congestion. W-CSMA is a particularly concerning case, as the probability of collision increases with the burstiness of traffic. Congestion is also aggravated in wired schemes, which see the admissible throughput to drop substantially. On the other hand, W-TOKEN and W-CBUF perform reasonably well due to their collision-free and short pipelined nature.

#### 5.3.3.4 Implementation Cost

To complete the analysis, it is important to remark the cost of implementing the different MAC schemes and compare those overheads with the absolute cost of each topology. At the time of this writing, we cannot provide a circuitual implementation of the BRS-MAC protocol here presented. However, note that SD-MAC [150], which represents one of the few MAC protocols for WNoC implemented to date, consumes very low area and bit energy ( $\sim 0.01 \text{ mm}^2$  and  $\sim 70 \text{ pJ/packet}$  in  $0.18 \mu\text{m}$  CMOS), suggesting that the impact of including BRS-MAC within the analysis is negligible in light of the area and power consumed by the cores as shown in Table 4.4 of Section 4.2.3.



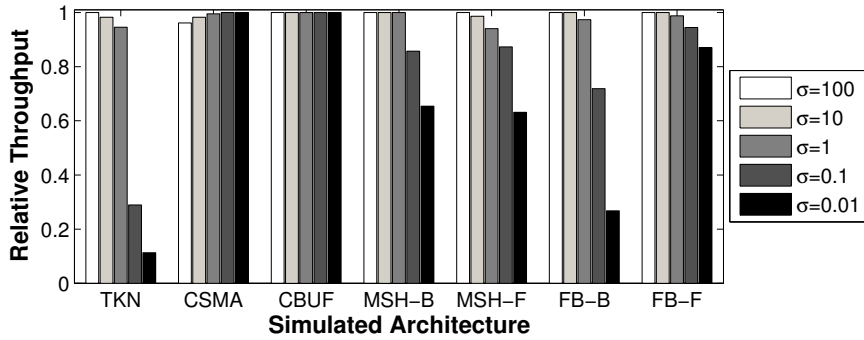


Figure 5.17: Throughput for different spatial injection distributions, from spread out ( $\sigma = 100$ ) to extremely hotspot ( $\sigma = 0.1$ ), with  $N = 64$ ,  $\beta = 100\%$  and  $C = 1$ .

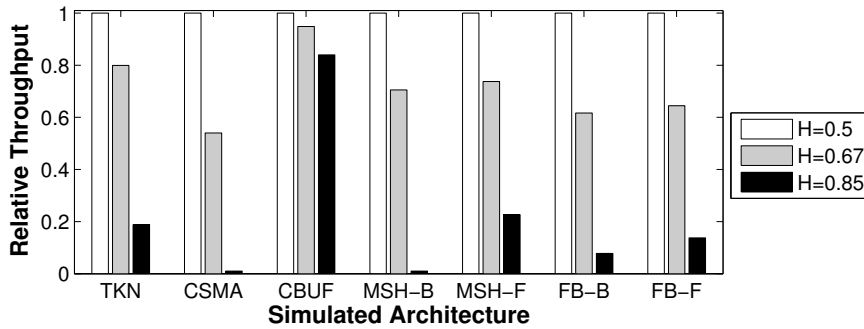


Figure 5.18: Throughput for different temporal injection distributions, from exponential ( $H = 0.5$ ) to extremely bursty ( $H = 0.85$ ), with  $N = 64$ ,  $\beta = 100\%$  and  $C = 1$ .

In the W-TOKEN and W-CBUF options, the MAC overheads are basically the area and power consumed by the wires and switches required to implement the token ring and the connections to the centralized buffer, respectively. In the former case, the token ring is composed by narrow links (potentially of one bit) between the MAC modules of each tile. From an implementation standpoint, the recent SCORPIO prototype [25] has shown that similar lightweight networks may consume less 1% of the tile area and power and, therefore, we will assume that the implementation cost of W-TOKEN is reasonable. We refer the reader to the next section for a discussion on the feasibility of W-CBUF.

## 5.4 Optimization of Existing Protocols

Previous sections have analyzed the on-chip communication context and proposed and evaluated a MAC protocol based on a better understanding of the requirements and objectives of BoWNoC. Although the performance of the analyzed alternatives may be enough to initially validate the BoWNoC, the context analysis suggests that there is still a large margin for improvement. Here, we take advantage of the knowledge gained during the processes of analysis and design to propose a set of enhancements that could be developed in future research.

## Towards Optimal and Adaptive MAC

The on-chip scenario is driven by latency and reliability, yet with a significant emphasis on energy efficiency. This combination of requirements may have been, together with simplicity, the main cause of the arguably widespread proposal of multiplexing. However, the good scalability and flexibility at both coarse and fine temporal granularities of random access techniques such as W-CSMA turns them into an interesting approach for broadcast-based wireless NoCs. However, their efficiency still needs to be improved in order to consider them as a serious contender. There are different design facets that affect performance and that must be carefully considered when optimizing a random access protocol.

The first design decision concerns the slotting of the channel. The models developed in Section 5.3.2 consider an unslotted asynchronous design, whereas the simulations in Section 5.3.3 assume a slotted protocol. Whereas the slotted option is considered easier to integrate in a multiprocessor where cores are synchronized and work using the same clock, it also suffers a performance loss due to the reduction of transmission opportunities (note that the slot length is proportional to the period of the clock, much larger than the propagation time). A preliminary comparison between both protocols reveals that the unslotted protocol achieves a 14% higher throughput at a latency of 150 cycles, suggesting that an asynchronous approach could deliver higher performance. This needs to be confirmed in further investigations.

Persistence and the backoff time are perhaps the most important aspects to consider [224]. When seeing the channel busy, the  $p$ -persistent protocol grants access with probability  $p$  immediately after the on-going transmission ends, or backs off with probability  $1 - p$ . In this work, we assumed a non-persistent protocol ( $p = 0$ ) and a backoff algorithm that incurs into recurrent collisions in high contention scenarios. However, there exist optimal persistence and backoff interval values depending on the traffic characteristics [236]. These values are hard to find and to change in a consistent way at runtime in typical CSMA environments, resulting in suboptimal performance. In a chip environment, though,  $p$  and the backoff interval could be evaluated more precisely given the relatively high knowledge on the traffic and of the application demands. High  $p$  values with shorter backoff times can be employed to reduce the latency in program phases with lower load, whereas lower values of  $p$  and longer backoff times can be used in high-contention phases requiring less assertiveness.

The chip scenario does not only allow to find near-optimal values of  $p$  and the backoff interval, but also to change them at runtime with ease. Dynamic reconfiguration could be performed using network performance metrics as inputs, or could be assisted by the programmer or the compiler, which could insert instructions that would tell the protocol how to operate. In any case, and as explained in Section 5.2.1, adaptive algorithms are not generally used in conventional non-broadcast wireless networks because it is hard to get consensus on the decisions. However, in our case, they would be easy to support because all nodes have all the information at all times. A single special message would be enough to ensure consistency.

Another possibility would be to have a hybrid MAC scheme that adapts to the load to optimize performance. We believe that the sweet spot in terms of performance and flexibility is somewhere in between the *channelization* and *random access* extremes; in schemes that combine time multiplexing, perhaps in the form of token passing, and random access. For instance, one could devise a protocol for WNoC that takes advantage of the better performance of random access at low loads and of the higher throughput of scheduling

techniques at high loads [152]. Outside the WNoC world, the literature contains different proposals that naturally implement such mixture by means of unconventional persistence mechanisms [237], random access with reservation [238], or probabilistic time division [239]; the challenge will be to find the solution that leads to the sweet spot in on-chip networks.

Finally, multiprogramming also needs to be carefully considered since it opens an unconventional design space. In these workloads, different sets of cores execute different applications, each with its own phased communication requirements. In this case, the assertiveness of the protocol could still be managed on a per-application basis, but should be coordinated among the different applications, similarly to the way prefetchers are adjusted in multicore settings [240], to maintain fairness and performance.

### Multicast Source Prediction

In computer architecture, prediction has been pervasively used as a tool to improve performance. The outcome of a conditional branch or the sharer set of certain cache lines are aspects that show correlation in different situations due to, among other factors, the iterative nature of computer programs. Predictive systems exploit such information to optimize the performance of a given multiprocessor architecture [1, 8] or, as recently proposed, its on-chip network [35, 76].

Taking into account the numerous precedents of the use of prediction to enhance the performance and efficiency of multiprocessors, it is reasonable to think that the same principle may be applied to the design of MAC protocols for BoWNoC. Specifically, we propose the concept of *multicast source prediction*, wherein we exploit the high spatiotemporal correlations exhibited by multicast traffic in a wide selection of architectures –see Section 5.2.2– to predict the source of next transmissions. Such predictability is even higher if we consider the well-studied phenomenon of application phase behavior [232].

Figure 5.19 shows an abstract representation of the basic architecture of a NoC-juxtaposed predictive system. In our case, the predictor is attached to the local MAC module and reads the packets that go through the wireless plane to extract the source of each multicast packet. From this, the predictor guesses the next multicast source and vali-

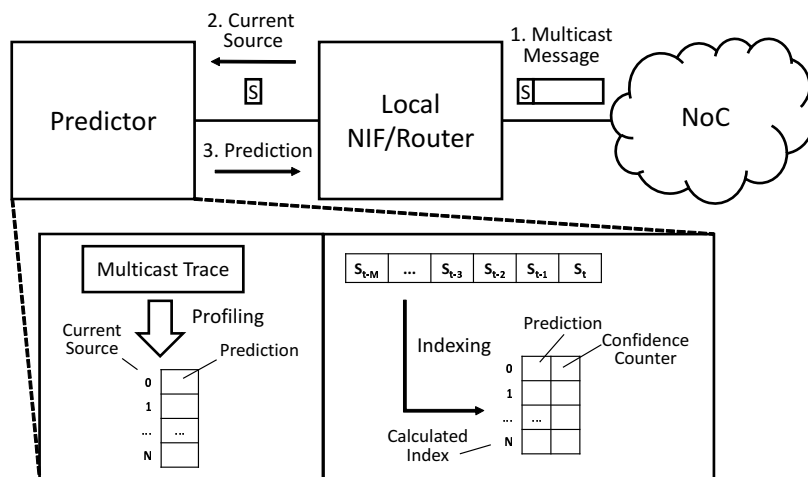


Figure 5.19: Multicast source prediction scheme, with the detail of a static predictor (left box) and a last value predictor (right box).

dates previous predictions. Since all nodes have access to all transmissions, global and local decisions are identical. The specific implementation of the predictor will depend on the nature of the multicast correlation and on the required level of accuracy to obtain acceptable performance improvements.

In order to avoid predictions over distant and potentially non-correlated events, both the header of the packet and the prediction are kept by the predictor during a pre-defined amount of time (i.e. observation window) and then discarded. If the next multicast transmission occurs before the information is discarded, the predictor may update its table with the new source. The choice of the length of the observation window will depend on how the impact of consecutive multicasts behaves over time.

In random access MAC protocols, certain collisions can be avoided if nodes know in advance who will transmit next. Thus, prediction-assisted MAC protocol could be modified by allowing only predicted sources to transmit under given circumstances. This could suppose a great performance improvement over non-optimized protocols since bursts of contention can be alleviated or, in extreme cases, even eliminated. Prediction can be also used to save energy by reasoning whether a given wireless interface will be idle for a significant amount of time and, thus, can be switched off without causing performance losses.

To further motivate the usefulness of the *multicast source prediction* approach, we evaluate the accuracy of two simple and widely-known predictors by running the different sets of traces over a simulated environment. The predictors are sketched in Figure 5.19. First, we consider a static predictor (SP), which assigns a prediction to every possible input at design time or compile time based on off-line profiling. We also consider a last value predictor (LVP), which consists of a buffer containing the last  $M$  multicast sources and which indexes a table of  $N$  entries. Here, the table is built and updated at runtime. In order to further increase the accuracy, 2-bit saturating counters are associated to each entry. These counters are incremented when predictions are correct and decremented otherwise, and are only made effective if the value of the counter is '10' or '11', thus reducing the frequency of incorrect predictions. Both predictors are implemented in MATLAB.

Figure 5.20 shows the coverage (i.e. number of predictions made effective over the number of events of interest) and accuracy (i.e. number of correct predictions over the number of predictions cast and executed) of SP and LVP averaged over all the SPLASH2 and PARSEC applications for different CMP sizes and coherence protocols. We assume that the observation window is  $\tau = 50 \cdot T_{CLK}$ . Consistently with the correlation results shown in Section 5.2.2, both SP and LVP show substantially better accuracy in MESI. In HT and TokenB, the use of the static predictor is highly discouraged due to its low reliability.

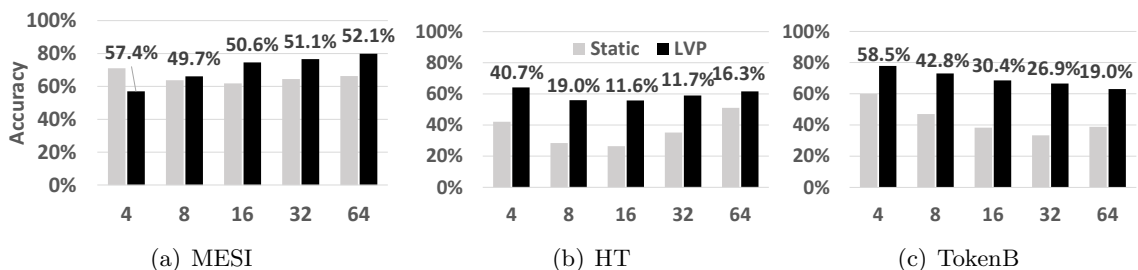


Figure 5.20: Geometric mean of the prediction accuracy in SPLASH2 and PARSEC for the SP and LVP assuming a 50-cycle observation window. Labels indicate the coverage in LVP (we assume 100% coverage in the static case). Higher is better.

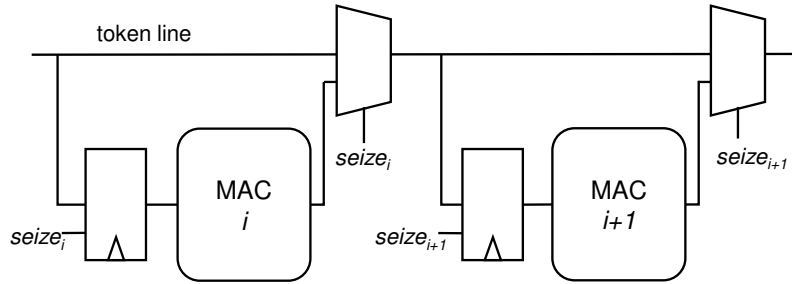


Figure 5.21: Multihop ring for token passing.

LVP, by means of the added confidence counter, achieves accuracies over the 50% yet with decreasing prediction coverages. Note, however, that the performance of the prediction scheme strongly depends on the application: the accuracy of SP using HT is below 15% for 64-core *canneal* and above 86% for 64-core *cholesky*, to cite an example.

These figures could be improved with more sophisticated predicting schemes like two-level predictors [1]. Also, phase detection can be achieved with higher accuracy by limiting predictions to phases in which the predictor has historically been more effective [232].

### Multi-hop Token Passing

The rigidity of the token passing scheme is the main barrier that prevents W-TOKEN from being a valid alternative in the manycore scenario. We assumed that passing the token takes one clock cycle per node, but this is clearly not enough in light of the results shown above, especially for hotspot traffic. By making the token ring asynchronous and allowing the token to traverse multiple nodes within the same clock cycle (assuming that these nodes do not have anything to transmit), the performance of W-TOKEN would greatly improve. This could be implemented by employing multi-hop asynchronous schemes such as the recently proposed in [32].

Figure 5.21 shows a possible way to implement such an asynchronous token ring. Each MAC module is connected to the ring through a register controlled by a *seize* bit, which is enabled when the node wants to transmit or it is “multiple of  $M$ ” hops away from the previous transmitter (whose ID is known by all). This way, the token will traverse  $M$  MAC modules within the same clock cycle unless someone along the path needs to transmit.  $M$  is limited by technology and energy consumption as indicated in [32]. As in the original scheme, the passing of the token overlaps with part of the transmission.

To demonstrate the usefulness of the approach, we reevaluated W-TOKEN by considering a multi-hop token passing with  $M = 4$  as a conservative value. Figure 5.22 shows a latency and throughput comparison between both alternatives. It is observed that the latency is cut by a factor of 4, extending the range of system sizes for which the latency is below the maximum admissible latency up to 1024 cores. As a result, the throughput obtained for such latency improves significantly, reaching 0.3 flits per cycle for a thousand cores.

### Implementing a Centralized Arbiter

Results shown in Section 5.3.3 have revealed that, as expected, the W-CBUF strategy delivers the best performance in both latency and performance. However, this is clearly an unrealistic option due to the implementation complexity and unaffordable scalability as achieving

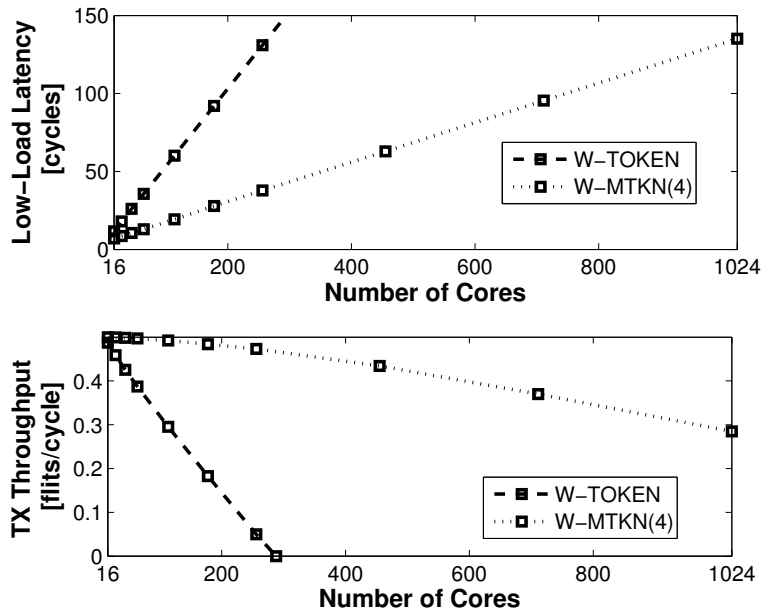


Figure 5.22: Performance of token passing with and without multi-hop capabilities.

centralized arbitration requires control signals to traverse global links and an  $N$ -to-1 multiplexer within the same clock cycle. This is a daunting task as both the system size and frequency increase.

One alternative way to perform centralized arbitration would be to use the low bandwidth channel in the wireless medium to exchange access requests and grants. However, these can still collide. This can be avoided by performing arbitration by means of a hierarchy of simple multiplexers and on-chip wires that progressively lead to the central buffer. From an implementation standpoint, recent works [25] have shown that similar lightweight networks may consume a negligible fraction of the tile area and power. Our network performance results, not shown for brevity, demonstrate that this approach would cause an increase in latency but would have a negligible impact on throughput. This way, the problem boils down to striking a balance between latency requirements and implementation complexity.

## Chapter 6

# NET: Exploring the Hybrid Wired-Wireless Design Space

Results in the previous chapter have shown that designs based on the BoWNoC paradigm greatly outperform several wired networks in terms of broadcast performance. This is an expected result in light of both the natural broadcast capabilities of wireless on-chip communication and of the strong emphasis placed over taking advantage of such potential. However, it is well known that a multiprocessor will generate a heterogeneous and highly-variable traffic profile with mixed broadcast and unicast flows. Such heterogeneity will be exacerbated in manycore environments and, within this context, a BoWNoC will need to be fit within a larger hybrid network architecture.

Here, we present OrthoNoC, a simple yet powerful hybrid design that takes advantage of the uniquenesses of the BoWNoC paradigm, while being backed up by a more conventional wired NoC. This is a first specification of the vision set by the thesis, outlined in Sec. 1.3): the BoWNoC is conceived as a latency-driven and broadcast-oriented *wireless plane* that will serve global traffic, whereas the *wired plane* NoC remains throughput-oriented, ideal for the transport the rest of the communication flows. After defining the architecture and explaining the main design decisions, the objective of this chapter is to perform an exploration of the hybrid wired-wireless design space.

OrthoNoC takes its name from *Orthos*, a two-headed dog belonging to the Greek mythology, due to its structural resemblance. OrthoNoC is composed by two independent network planes or *heads*, both driven by a unique traffic steering policy that would represent the *core* of the architecture. This is in contrast to most hybrid proposals described in the literature [90, 94, 141], where packets may go through both network planes in the pathway to their destination. *Ortho* is also a Greek prefix sometimes used to express uncorrelation or independence between two objects, thus reinforcing the idea behind our network architecture. This strategy also allows OrthoNoC to implement load balancing easily, thereby helping to mitigate the adverse effects of traffic heterogeneity and variability.

This chapter is organized as follows. In Section 6.1, we first motivate our design by inspecting the potential performance gains and drawbacks of a simple hybrid wired-wireless architecture. Then, in Section 6.2, we briefly describe how OrthoNoC attempts to achieve these potential gains while covering the main drawbacks. Section 6.2.1 details the main decision designs taken at the network and system levels, which complement the PHY and MAC decisions justified in previous chapters. Finally, we explore the hybrid design space by evaluating the performance of OrthoNoC in a wide range of configurations in Section

6.2.2. We demonstrate that OrthoNoC is capable of improving severalfold latency speedups and a throughput increase of up to 56% in the presence of broadcast traffic and of up to 26% in the presence of global traffic.

## 6.1 Motivation

Results obtained in Chapter 5 suggest that a hybrid network architecture combining a wired mesh with a BoWNoC would not only greatly reduce the latency of broadcast messages, but also achieve a significant throughput boost by relieving the wired plane of having to serve such type of traffic. To quantify such potential gains, we compare the performance of the MESH-FT wired network with and without an overlaid W-CARB wireless plane (see Section 5.3.3.1 for more details on the networks). In this scheme, unicast and broadcast messages are forwarded to the wired and wireless planes, respectively.

The evaluation is performed using the simulation framework defined in Section 5.3.3.1, considering different system sizes  $N$ , wireless channel capacities  $C$ , and broadcast traffic percentages  $\beta$ . We put emphasis on the results for  $\beta \leq 25\%$ , as these correspond to the range of broadcast percentages found in cache-coherent processors as detailed in Section 5.2.2. Results for  $\beta > 25\%$  could be also of interest given that new multicast-intensive architectures may arise if the cost of multicast is reduced, as we will see in Chapter 7.

### Potential Gains

Figure 6.1 illustrates the improvement in terms of low-load latency as a function of all the variables considered. Here, a value of 4 implies that the hybrid architecture goes four times faster. The improvement is quite consistent and, as the intuition suggests, increases with the broadcast percentage, the channel capacity and the system size. Assuming  $N = 256$ , the latency is reduced by a 20% for  $\beta \approx 10\%$ , cut in half by  $\beta \approx 50\%$  and to one third for  $\beta \approx 70\%$ . A similar tendency, yet with lower absolute gains, would be obtained when comparing the hybrid architecture to a network topology with higher radix.

Figure 6.2 shows the throughput improvement (or deterioration) as a function of all the design variables. Up to a given broadcast percentage  $\beta_1$ , the addition of the wireless plane increases performance (over 2X speedup in some cases) as it significantly offloads the wired plane. This percentage range is reduced as the network scales and the bisection bandwidth of the wired plane increases. Beyond that point  $\beta_1$ , the percentage of broadcast traffic

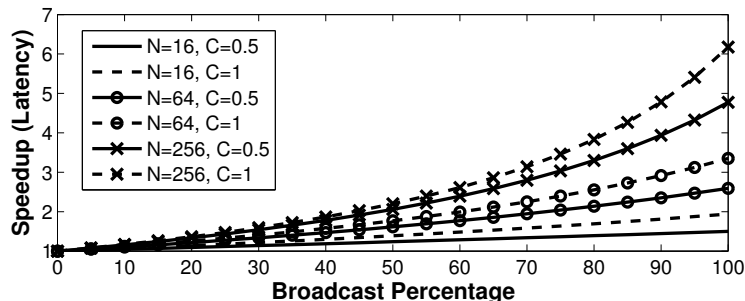


Figure 6.1: Latency speedup of a hybrid NoC with respect to MESH-FT as a function of the broadcast percentage for different system sizes and channel capacities.



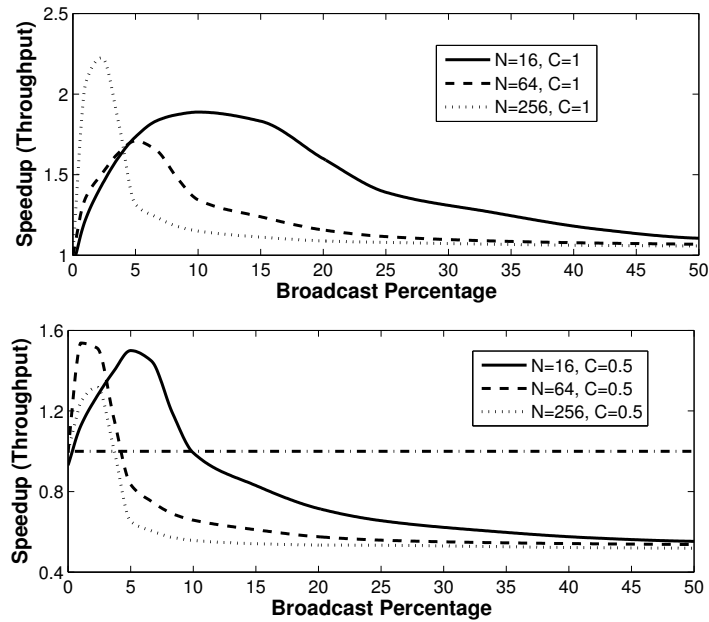


Figure 6.2: Throughput speedup of a hybrid NoC with respect to MESH-FT as a function of the broadcast percentage for different system sizes and channel capacities.

saturates the wireless plane, whereas the unicast traffic does not take the wired plane to the throughput limit. Therefore, the inclusion of the wireless plane at that point only helps to reduce the overall latency, whereas the throughput is progressively diminished. At very large broadcast percentages, the throughput speedup is the difference in terms of saturation throughput between W-CBUF and a RMESH-FT. Although the throughput improvement is sensitive to the capacity of the wireless channel, improvements can be achieved over a significant range of broadcast percentages even with a modest capacity. These results are similar when compared to a network topology with higher radix, but not shown for brevity.

### Potential Drawbacks

Results shown above demonstrate that a simple approach can be useful by providing impressive performance gains in a wide variety of scenarios. However, we must take into consideration that the evaluation has been made assuming a rather ideal MAC strategy that provides low latency and ideal throughput. This may not be achievable at least in the short term, forcing architects to resort to schemes with higher latency, e.g. token passing, or lower throughput, e.g. CSMA. We also need to carefully consider systems with lower channel capacities than half a flit per cycle, fixed by the realistic estimations given in Chapter 4 regarding current and short-term transceiver implementations.

The channel capacity issue connects with the problem of load balancing. For instance, consider the performance shown in Figures 6.1 and 6.2 for  $\beta \geq 10\%$  and  $C = 0.5$ . Although the wireless plane offers much lower latency in these cases, the wired plane could be helpful when the load is high and the wireless network saturates. Therefore, it is necessary to have an intelligent hybrid controller that steers traffic as a function of the load and performance objectives. Reconfigurability is indeed a desirable option given, besides the aforementioned considerations, the high variability of traffic depending on the application being run in the underlying processor.

The lack of versatility of the solution is another issue to consider. While BoWNoC is clearly better than other NoCs in the presence of broadcast traffic, it may be also advisable to use BoWNoC in other scenarios. A relevant example is that of long-range traffic, which has been the target of most of the hybrid wired-wireless proposals [90, 94, 96] due to the latency advantage of the wireless plane. Although this network architecture has shown substantial performance and energy efficiency gains, its scaling becomes difficult due to the fixed position of antennas, the rigidity of the MAC mechanism, or unrealistic bandwidth requirements. For a reasonable cost, the knowledge gained in such related work could be very well applied to a hybrid proposal that would also serve broadcast traffic.

## 6.2 OrthoNoC: A Dual-Plane Wired-Wireless Network Architecture

Recent years have seen the emergence of several wired-wireless NoC architectures [90, 94, 96, 153]. Most of these hybrid architectures use the wireless technology to implement long-range links between selected nodes, and are laid over a conventional NoC that takes care of local transmissions. To implement this strategy, a subset of routers is equipped with a wireless interface. This way, transmissions from far-apart cores will go from the home node to the local router with wireless capabilities, then perform one or a few hops in the wireless network, and then go through the wired network towards the final destination. To ensure the correct operation of the whole architecture, the routing protocol needs to be revised and deadlock conditions need to be proved again.

Our architecture, OrthoNoC, attempts to break away from these proposals by implementing two independent network planes. This mainly implies that a packet will most likely not switch planes during its transmission. The choice of network plane is performed at the source by the hybrid controller, which interconnects the network interfaces of both planes with the processor. Since the traffic steering policy enforced by the controller is not fixed, can adapt to network-side or processor-side inputs, and can balance the load between network planes, we claim that OrthoNoC is a powerful solution to solve the throughput, reconfigurability and versatility issues of existing hybrid proposals.

Figure 6.3 pictorially represents the main idea of OrthoNoC. The processor is organized in computing tiles. Each tile contains a number of processing cores with their respective instruction and data L1 caches, a slice of the shared L2 cache, and the network interface. The network interface directly connects the tile with a *wireless plane* by means of a transceiver and an antenna, and with a *wired plane* by means of a local router.

The OrthoNoC architecture embodies the hybrid vision described in Chapter 1. Within this vision, the wireless plane leverages the unique properties of the technology to transmit latency-sensitive, global and broadcast traffic. In the interest of simplicity, flexibility, and to provide full broadcast support, this plane accounts for a very small set of broadband channels shared by potentially every computing core. On the other hand, the wired plane is takes care of unicast and local traffic, for which it can achieve high throughput and contained latency. Remind that we initially consider conventional wires in a mesh topology given its scalability, regularity and simplicity, but that we also anticipate the use of nanophotonic technologies in this plane seeking unbeatable bandwidth and energy efficiency.

Most of the design decisions of OrthoNoC, including the steering policy, seek to emphasize and take advantage of the strengths of wireless communication: simplicity, flexibility, and broadcast. By separating both network planes, OrthoNoC keeps the architecture sim-

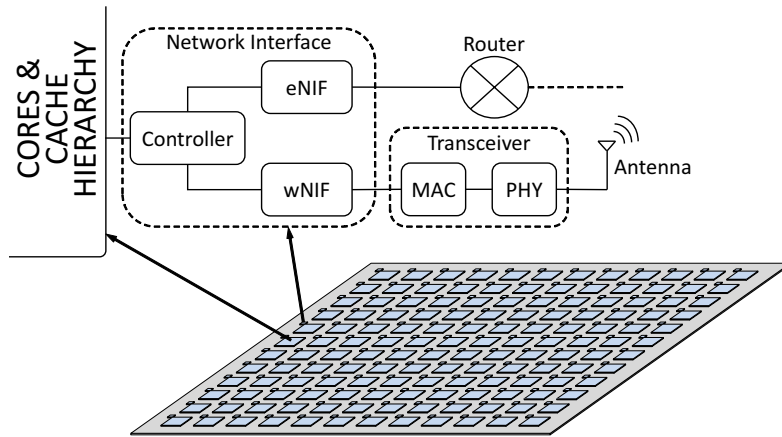


Figure 6.3: Schematic diagram of the OrthoNoC architecture.

ple as there is no need for a redefinition of the routing protocol or a reevaluation of the deadlock freedom. This also allows to increase the bandwidth at the ejection points of the network, thereby improving the network throughput for moderate to high levels of broadcast. By integrating one transceiver per tile, OrthoNoC provides flexibility as the antenna and transceiver can be power gated with its associated tile without affecting the rest of the network. Finally, by allowing all tiles to share the same channels, OrthoNoC delivers low-latency and ordered broadcast capabilities to the network as wireless transmissions reach all cores regardless of their location.

Flexibility is arguably the most powerful feature of OrthoNoC. Indeed, adaptivity and reconfigurability capabilities are of critical importance in the manycore era, where dark silicon constraints force certain parts of the chip to be powered off and where multiprogramming workloads introduce high spatiotemporal traffic variability. As pointed out above, OrthoNoC allows to power gate transceivers that are not being used to reduce energy without affecting the rest of the network [188]. This could be also used to, for instance, implement an adaptable hybrid small-world network based on the hybrid proposal in [90]. Instead of deploying wireless hubs in fixed locations for general purpose, OrthoNoC could periodically re-evaluate the target cost function considering the characteristics of the input traffic. The introduction of the hybrid network interface also allows to reconfigure the fraction of traffic that is injected into each network, as well as the minimum distance to consider in cases where the wireless plane is used for long-range transmissions. Finally, one can also adapt the topology depending on possible hotspots introduced by the application. All these aspects provide the application mapping algorithms with higher degree of freedom [241], as well as the network with a much larger reconfigurability.

### 6.2.1 Design Decisions

The design process of OrthoNoC requires addressing several implementation issues present at different abstraction layers. Here, we detail a selection of them mainly from the system-level perspective.

## Clusterization

The tiled scheme summarized in Figure 6.3 can be easily modified to perform core concentration at the processor side, helping to keep the network balanced and tractable and reducing the number of on-chip antennas as per area and power reasons. Specifically, concentration will be accomplished by integrating several processing cores within the same tile. Since the L2 is shared among the cores of a tile, there is no need to modify the controller or the network interface.

Concentration at the network side is also possible by sharing a single controller, network interface, and wireless communication unit among a set of tiles. To implement this scheme, it is necessary to employ a local switch that interconnects the tiles and the controller. Moreover, concentration at the network side can be performed asymmetrically by using different concentration factors at both network planes. For instance, the wired plane may still connect one tile to each router, whereas the wireless plane may interconnect four tiles to each antenna and transceiver. In this latter case, a multiplexer and an amplifier would be required between the NIF of each tile and the MAC module of the wireless plane. For small values of concentration, the latency overhead of these components would be of around one clock cycle per direction at most.

Note that, in both approaches, concentration affects system-level decisions such as load balancing or the power gating of transceivers.

## Heterogeneous Systems

Due to the flexibility of OrthoNoC and of the tiled organization, this scheme can be applied in heterogeneous systems. In CPU+GPU architectures, or in processors with *small* and *large* cores like ARM's big.LITTLE, heterogeneity can be hidden behind the concept of tile. This basically means that the tile composition can change from tile to tile. In the end, OrthoNoC will interconnect tiles, not cores, regardless of its type. As we will see, heterogeneous systems may benefit from the use of different broadcast domains enabled by the implementation of different frequency channels.

## Hybrid Controller Design

A traffic steering policy, enforced by the hybrid controller within the network interface, determines the plane through which a message will be transmitted. As we will see, this policy can be simple or complex, fixed or determined at runtime, and agnostic or aware of the underlying multiprocessor architecture. In any case, the policy needs to be aware of the strengths and limitations of the wireless plane, caring to not overload it as it can easily become a bottleneck due to its moderate aggregate bandwidth. At the receiving end, the controller performs admission control as part of the functionality of the network interface.

The traffic steering policy is a key determinant of the performance of OrthoNoC and, as such, it represents the main object of study of this section. Ideally, the hybrid controller would always choose the optimal path while balancing the load. To this end, the controller considered here implements the traffic steering methods shown in Fig. 6.4. First and foremost, the controller performs *plane selection* following a static or dynamic policy. This policy can balance the load in a preventive manner if the traffic characteristics are known and stable. Otherwise, the controller may need to resort to additional methods. *Plane switching* can be used when packets suffer unacceptable delays in the wireless plane, in which case the MAC protocol re-sends packets through the wired plane. *Plane blocking*

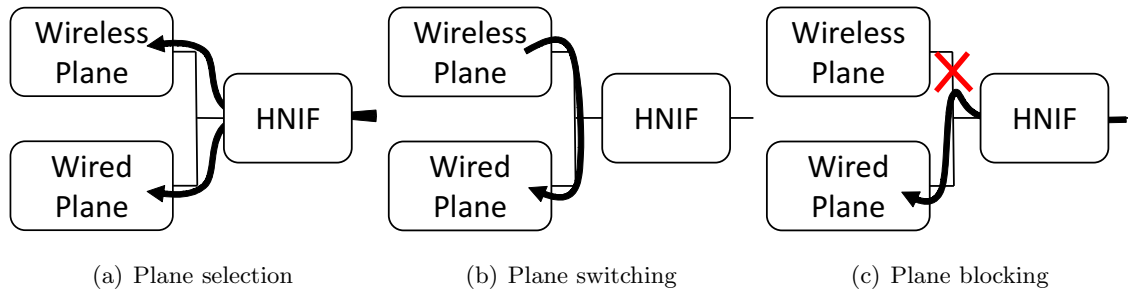


Figure 6.4: Controller methods for load balancing.

can be used when one of the planes suffers severe congestion, in which case the controller temporarily deflects all packets towards the uncongested plane.

The plane selection policy can take two main approaches. On the one hand, *network-oriented* or *architecture-agnostic* controllers base their decisions solely on the characteristics of the message, the network plane, or the load. Simple instances of this type of controller only need to inspect the packet header, whereas more complex policies may take decisions based on load-dependent latency estimations [62]. This is the type of controller we will evaluate in this chapter.

On the other hand, *architecture-aware* controllers take advantage of architectural information to optimize traffic steering. Knowledge on the criticality of a given message, given either by the programmer, the compiler, or the architecture, is of great value when determining the plane to use. Invalidation in conventional coherence protocols or collective communication primitives in message passing could be targeted using this approach. In Chapter 7, we explore this path by using the hybrid controller, in conjunction with a small piece of special memory, to implement fast synchronization. Synchronization messages (generated by locks and barriers, which tend to be latency-critical) are sent through the wireless plane and the rest of messages through the wired plane.

### Channelization and RF Planning

OrthoNoC will initially consider one broadband channel shared by all cores. The use of a single channel forces OrthoNoC to manage access to the shared medium through arbitration, but also guarantees one-hop delivery with all nodes having a consistent view of the order of delivery. Thus, a single broadcast domain is created. Notwithstanding this, we envisage that future iterations of the architecture will account for a small set of channels and that, ideally, each tile will be able to access those channels and dynamically switch between them. This could be used to, for instance, increase the throughput of the wireless plane as implied in related work [96,153]. Having multiple channels could also be interesting from an architectural perspective, as multiple broadcast domains may be required to accommodate either multiple applications mapped within the same processor, or different components within an heterogeneous system as described above.

Determining the number and position of the broadcast domains concerns the design of the antenna and the power allocation. Basically, antennas need to transmit with enough power to reliably transmit to the furthest nodes of its domain, which can be in any direction. This suggests the use of omnidirectional antenna and affects the overall energy consumption. In all cases, OrthoNoC considers variable-gain amplifier techniques [167] to perform a fine tuning of the allocated power and save precious energy.

Table 6.1: Simulation Parameters

System	20×20 mm <sup>2</sup> die, 1 V, 1 GHz, <b>64/256 tiles</b> (def: 64)
Hybrid Controller	<b>Broadcast/Global</b> , 1-cycle delay
Wired Plane	MESH-FT (128b links, 2 cycles/hop, wormhole XY routing, 4 VCs, fixed tree multicast, multiport allocation)
Wireless Plane	Single channel, <b>1–10 cycles/flit</b> (def: 2), BRS-MAC protocol (non-persistent, NACK burst, exponential backoff), 3 retries max, 4 backlogged packets max

(Explored variables are shown in bold)

## PHY and MAC

The PHY and MAC layers follow the design principles shown in Chapters 4 and 5, respectively. In the former, we consider a simple modulation seeking high transmission speeds with moderate area and power overheads, with a coding scheme capable of ensure a BER commensurate to that of a RC wire ( $\sim 10^{-15}$ ). In the latter, we use the BRS-MAC protocol developed in Section 5.3, putting emphasis on its flexibility, potential for low-latency, and reduction of the penalty of recurrent collisions.

### 6.2.2 Evaluation Framework

We use the same simulation framework than in Chapter 5 to evaluate how the inclusion of a wireless plane in OrthoNoC improves performance as compared to a baseline NoC consisting of only the wired plane. The baseline corresponds to the MESH-FT configuration used in Chapter 5. Note that this wired NoC is a very aggressive baseline, which implies that any improvement obtained over this baseline setting is highly valuable.

Table 6.1 details the general parameters of the simulated architecture, showing in bold the design variables with which we explore the hybrid design space of OrthoNoC. We basically change the system size and channel capacity to inspect the scalability of OrthoNoC but, most importantly, we modify the hybrid controller policy to evaluate the applicability of the architecture in different scenarios. Our goal is to validate the versatility and reconfigurability potential of the architecture. Note that the wireless plane will attempt to transmit the same message three times before redirecting it through the wired plane. Blocking of the wireless plane is performed on a per-core basis and occurs when the MAC queue reaches a backlog of four packets. The blocking is lifted as soon as the backlog is reduced to two packets.

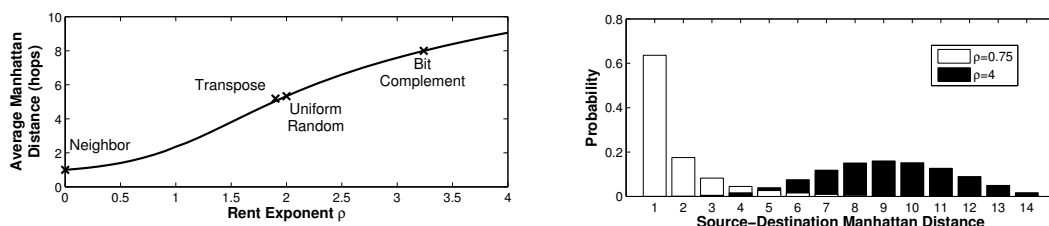
### Traffic Profiles

The objective of chapter is to evaluate the performance and flexibility of the OrthoNoC architecture. To this end, we need to consider a wide range of scenarios involving variable amounts of broadcast traffic and different hop distance distributions. We use synthetic traffic as it can effectively model both characteristics.

Table 6.2 lists the characteristics of the traffic profiles considered in this section. We use the traffic generator detailed in appendix A to model different broadcast intensities, as well as different hop distance distributions. Regarding the latter aspect, besides using the typical neighbour, uniform random, transpose and bit complement profiles [242], we

Table 6.2: Synthetic traffic profiles

Injection Distribution	Temporal: Poisson arrivals, Spatial: Uniform Distribution
Message Size	1 and 4 flits (same probability)
Hop Distance	<b>Neighbour (NE), Uniform Random (UR), Transpose (TR), Bit Complement (BC), Rent (RE-<math>\rho</math>)</b> (def: UR)
Broadcast Percentage	<b>0–100%</b> (def: 0%)

(a) Average distance as a function of  $\rho$  (b) Probability density function for two  $\rho$  valuesFigure 6.5: Manhattan distance statistics for Rent traffic in a  $8 \times 8$  mesh.

consider variable locality though the use of the Rent traffic model [231]. In this power-law model, the Rent exponent  $\rho$  determines the probability density function of the distance between source and destination. Figure 6.5 shows the Manhattan distance statistics of traffic generated with the Rent methodology [243] as a function of the  $\rho$  exponent.

## Investigated Architectures

To explore the effectiveness of the hybrid controller with architecture-agnostic configurations, we evaluate the performance of two simple policies:

**Global** source and destination are compared. If the Manhattan distance is above a certain threshold  $H$ , the message is considered global and is sent through the wireless plane; otherwise, it is sent through the wired plane. The threshold can be decided *a priori* or at runtime. For load balancing purposes, we will consider different fixed thresholds calculated as a fraction of the network diameter of the wired plane topology.

**Broadcast** the number of destinations is inspected. If the message is broadcast or a multicast with a large destination set, the message is sent through the wireless plane with probability  $P$ , whereas the rest of messages are sent through the wired plane. In this work, we will test different probabilities in the interest of load balancing.

As summarized in Table 6.3, the evaluation basically considers synthetic traffic with variable hop distribution to test the *global* policy, and with variable broadcast percentage to test the *broadcast* policy. The minimum distance and the probability of using the wireless plane remain as main design parameters of each policy. Remind that, for congestion reasons, packets intended for the wireless plane may be redirected through the wired plane.

### 6.2.3 Performance Evaluation

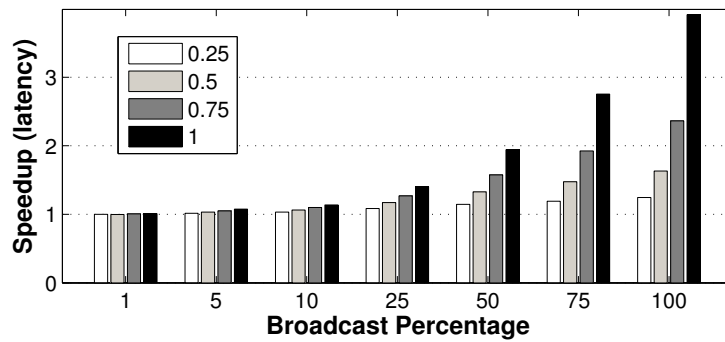
We evaluate the performance of OrthoNoC by obtain the speedup in terms of latency and throughput that it introduces with respect to the baseline. We measure the low-load latency

Table 6.3: Explored Controller Policies

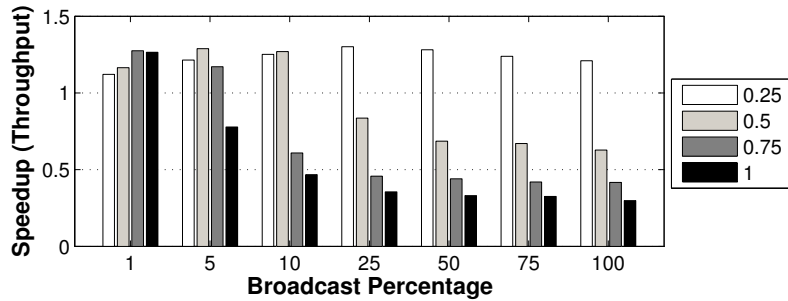
Application			Controller		
Profile	Variable	Range	Policy	Variable	Range
Synthetic #1	Rent Exponent	0.75–4	Global	Threshold $H$	$.5D-.75D$
Synthetic #2	Broadcast %	0–100	Broadcast	Admission Prob. $P$	0.25–1

( $D$  is the network diameter of the wired plane)

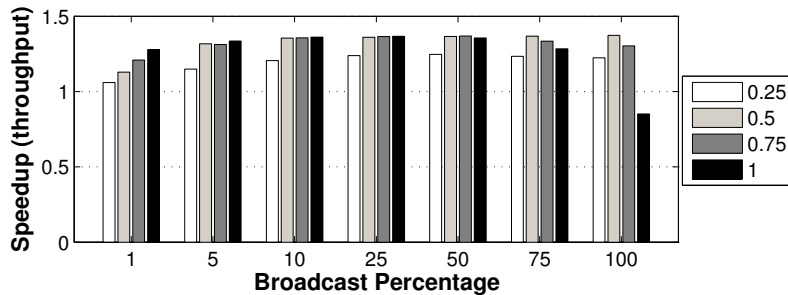
and establish a common limit of 200 cycles to measure the throughput of the network. This value is commensurate to the access latency to main memory in modern multiprocessors. Refer to Tables 6.1 and 6.2 for default system and traffic parameter values.



(a) Latency



(b) Throughput (without plane blocking)



(c) Throughput (with plane blocking)

Figure 6.6: Network performance speedups of the broadcast policy of OrthoNoC for different admission probabilities and percentages of broadcast traffic.



## Broadcast Policy

Remind that, here, the hybrid controller forwards broadcast packets to the wireless plane with probability  $P$ . Figure 6.6(a) shows the network latency speedup as a function of the broadcast intensity for different values of  $P$ . Clearly, the latency is improved as much as the wireless plane is used to transport the broadcast flows (higher  $P$ ). The improvement is maximized as the broadcast traffic becomes dominant.

Figures 6.6(b) and 6.6(c) show the network throughput speedup for different values of  $P$  in two cases. Figure 6.6(b) considers that the network interface does not support plane blocking. In this case, OrthoNoC performs remarkably better than the baseline starting at very low broadcast percentages, yet as long as the pressure to the wireless plane is moderate; this is, for low admission probabilities or broadcast percentages. Here,

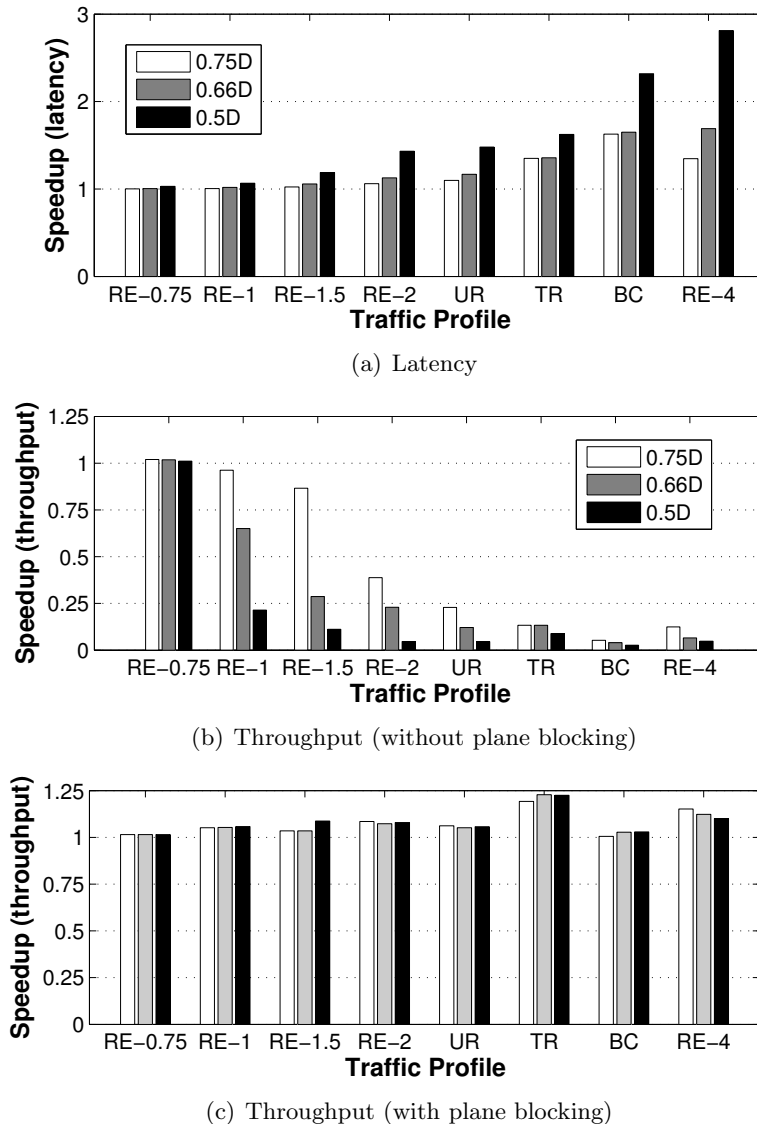


Figure 6.7: Network performance speedups of the global policy of OrthoNoC for different distance thresholds and traffic profiles.

the admission probability acts as a preemptive load balancing mechanism. Now consider Figure 6.6(c), which assumes that plane blocking is supported and packets can be deflected through the wired plane. We observe that the throughput is enhanced up to a 40% in a wide range of scenarios with respect to the baseline, and that the optimal  $P$  value depends on the broadcast percentage. Also, note that plane blocking does not solve congestion in all cases, suggesting that admission control may still be useful.

At this point, there is one aspect worth highlighting. The dual-plane structure of OrthoNoC increases the bandwidth at the ejection points of the network by providing direct connections to both network planes. This allows the network throughput to improve consistently for  $\beta > 10\%$ , approximate point where the baseline network becomes limited by the ejection links. In hybrid architectures that integrate a few wireless links within the wired topology, this improvement would be lost since the throughput is still limited by the bandwidth offered by the wired edges.

### Global Policy

In this case, the hybrid controller forwards to the wireless plane messages the destination of which is more than  $H$  hops away. Figure 6.7(a) represents the network latency speedup as a function of the traffic profile for different thresholds. Traffic profiles are ordered from mostly local to mostly global. We observe that the best latency improvements are achieved for global traffic patterns using a threshold that favors the use of the wireless plane.

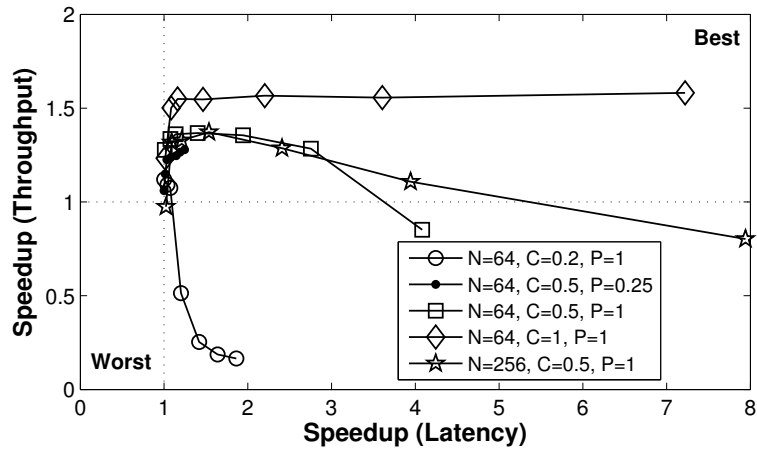
Figures 6.7(b) and 6.7(c) show the network throughput speedup for different values of distance threshold values and, again, with and without plane blocking. In this scenario, the outstanding difference in terms of aggregated bandwidth between the wired and wireless planes makes the use of plane blocking absolutely necessary to avoid premature congestion. When used, the throughput improves depending on the locality of the traffic profile. The case of *transpose* traffic is worth noting, as it causes a large load imbalance within NoCs with dimension order routing. OrthoNoC alleviates this effect and, as a result, performs remarkably better than the baseline.

### Scalability Analysis

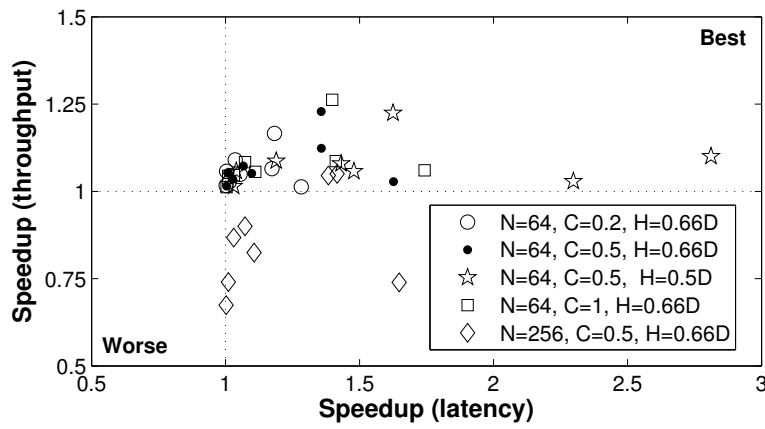
Results have not only shown significant latency and throughput speedups, but also revealed that interesting trade-offs between both metrics in different case scenarios. Now, let us analyze them together with the scalability of OrthoNoC with plane blocking.

Figure 6.8(a) is a representation of the speedups accomplished by the broadcast policy of OrthoNoC throughout the hybrid design space with increasing levels of broadcast traffic. Note that all lines tend to shift to the rightmost part as the broadcast percentage increases. We observe that both latency and throughput are affected by the wireless data rate, whereas increasing the network size not only largely maintains the throughput advantage but also increases the latency speedup. Reducing the wireless data rate too much can become a performance bottleneck unless advanced traffic steering techniques are applied.

Figure 6.8(b) represents the hybrid design space considering the global policy and different traffic profiles. Overall, improvements are more modest than in the broadcast policy. Results are mainly positive for  $N = 64$ , whereas the throughput is significantly reduced for higher system sizes. Increasing the threshold can help alleviate this issue but, in any case, the latency results are promising for the data rates considered.



(a) Broadcast Policy



(b) Global Policy

Figure 6.8: Scatter plot of the performance improvements of the two policies of OrthoNoC for different system sizes, wireless plane capacities, and input traffic characteristics.

## Chapter 7

# ARCH: Broadcast-Enabled Massive Multicore Architectures

Broadcast has been traditionally regarded as a prohibitive communication transaction in multiprocessor environments. Nowadays, such constraint largely drives the design of architectures and algorithms all-pervasive in diverse computing domains, directly and indirectly leading to diminishing performance returns as processors are scaled. We speculate that this trend can be reverted with the aid of the BoWNoC paradigm developed in this dissertation, given its potential for low-latency (a few cycles) and energy-efficient (a few pJ/bit) global communication as analyzed in previous chapters. In this one, we explore the different innovations that BoWNoC could open in terms of hardware architectures and algorithmic approaches, in the pathway of significantly improving the performance, energy efficiency, scalability and programmability of manycore chips.

This chapter divided in two main parts. Section 7.1 provides a holistic and qualitative view of the architectural design space that BoWNoC opens within the shared memory and message passing paradigms. Substantial coherence improvements can be introduced in shared memory systems if architects are freed from the imperative need of using methods that involve local communication, whereas programmability could be greatly enhanced in message passing environments if the cost of global coordination is indeed reduced. Between both extremes, as represented in Figure 7.1, a myriad of programming models could benefit of a lift of the global communication constraints that rule current manycore processors.

Section 7.2 has a less exploratory edge and, instead, delivers a specific architecture that makes use of a BoWNoC. With this, we aim to *1)* complement the traditional architecture-driven NoC design process (which has guided most of the works in the area) with a NoC-aware architecture development, and *2)* provide a first realization of the potential that broadcast-oriented architectures could achieve. The proposed architecture, called WiSync, implements fast synchronization through the use of a small memory module attached to a BoWNoC. The section explains the specific NoC and memory aspects of the architecture, describes the main implementation issues, and evaluates the speedups obtained with WiSync given different design profiles.

### 7.1 Potential Impact of BoWNoC on Future Manycores

Having an effective broadcast plane with low latency and total ordering relaxes a large number of constraints cast upon architects and parallel programmers. In shared memory,

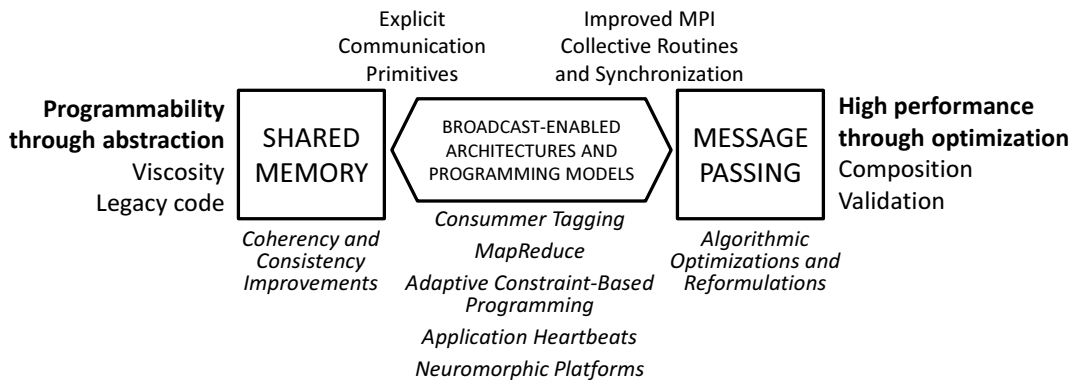


Figure 7.1: Representation of the many-core design space opened by broadcast between the shared memory and the message passing paradigms.

the introduction of such network plane is expected to provide scalable support for fine-grain data sharing, thereby bringing the manycore coherence solution space much closer to the area characterized by low architectural costs, low complexity, and high performance. The cost of synchronization will also be strongly reduced due to atomic nature of broadcasts: locks will be more effective and easier to implement, whereas barriers may take advantage of the peculiarities of RF signaling over a shared medium to virtually eliminate serialization in counting arrivals as explained in Section 3.2 [57]. This could lead to performance and programmability improvements stemming from both the exploitation of parallelism on a finer granularity and the reduction of the penalty of maintaining sequential consistency.

In message passing, all-to-all routines could be redefined seeking greater performance and lower cost. These improvements may enable the development of alternative approaches for widely used kernels and applications. The achievable speedup could be dramatic in applications where global sharing and coordination are the main bottlenecks. As an example, Fig. 7.2 shows the trace of communication actions within the so-called Deflated Conjugate Gradient (DCG) iterative solver for a specific example described in [244]. This solver performs much better than the classical Conjugate Gradient solver in terms of iterations, but exhibits a bottleneck due to an *allreduce* communication. For this particular example, the *allreduce* in question is of 4000b and its duration is that of a matrix-vector product. Although the iterative solver approach can achieve a speedup of ten in terms of CPU time in this case, the duration of the *allreduce* operation reduces the speedup to four. This demonstrates the great importance of such global communications in iterative solvers.

While the potential advantages are manifold, thus far there is little manycore architecture research revolving around advancements on efficient on-chip broadcast. A new breed of race-free coherence protocols based on the use of broadcast is proposed in [88]. Broadcasts are used to acquire fine-grained mutexes that serialize requests to conflicting addresses and, thus, eliminate race conditions. Other works have analyzed the impact of improved broadcast over the performance of traditional architectures rather than proposing new schemes. For instance, snooping and limited directory protocols have been evaluated using the SCORPIO prototype [25], revealing a speedup of 20% when using snooping coherence.

We believe that this scarce related work is just the tip of an iceberg of novel methods for manycore computing, which will be revealed once the feasibility of a globally shared medium is demonstrated. Next, we describe the advancements that the BoWNoC paradigm may bring towards greater performance and programmability in manycore chips.

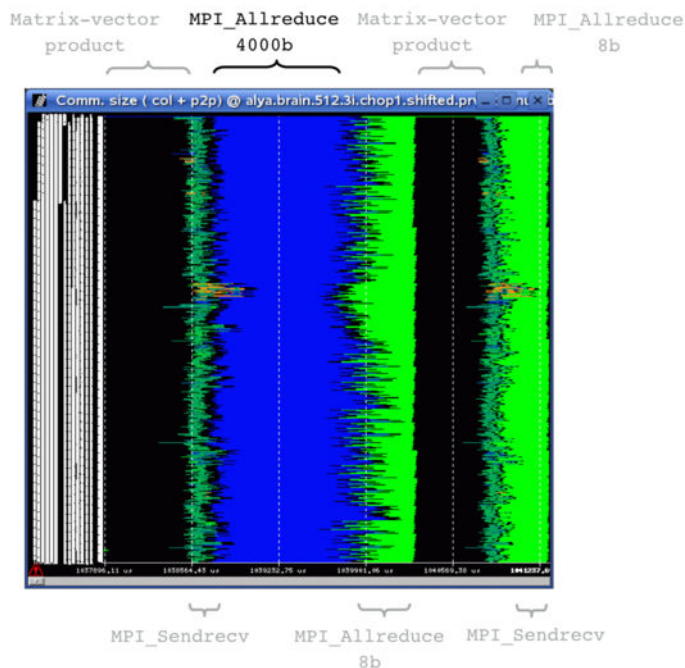


Figure 7.2: Graphical representation of inter-process communication in a DCG iterative solver over time. Each row represents a thread.

### 7.1.1 Software Architecture and Algorithms

Message passing systems explicitly define inter-thread communication, thereby shifting to the programmer the responsibility of carefully placing communication routines within the code. Collective communication primitives are an essential part of the message passing libraries, as they allow the transfer of data in an *one-to-all* and *all-to-all* fashion, functionalities that map well with the way several parallel programming approaches work. For instance, a typical implementation of Fast Fourier Transform (FFT) alternates block computation with the scattering and gathering of data [142]. Also, making data availability transparent from its location through the use broadcast primitives may reduce the complexity of certain algorithms. However, the traditionally high cost and poor scalability of these collective communication routines marginalizes their use, prompting programmers to come up with streamlined algorithms that use point-to-point routines instead. This is the case of computational mechanics solvers in general and iterative solvers in particular, where selected parts of the code generally show degraded scaling behavior at high core counts.

The advent of the BoWNoC paradigm will most likely require the revisit of the message passing routines that implement all-to-all communication beyond existing optimizations [245]. Native broadcast support will clearly benefit routines that are direct extensions of broadcast primitives, i.e. `MPI_Allgather` and `MPI_Allreduce`, whereas improving other all-to-all routines will demand a the careful orchestration of communication through the wired and wireless planes of the hybrid network. In this latter case, emphasis would be placed on the design of custom MAC protocols and traffic steering policies that would optimize latency and throughput in these collective communication patterns.

Once collective communication routines are improved, the next step would be to re-evaluate a set of representative algorithms. However, one can expect better speedups if algorithms are re-defined taking into consideration the presence of a natural broadcast mech-

anism. The programmer will need to account for optimizations that may involve the intense sharing of data among a large set of threads, or the use of oftentimes avoided synchronization primitives and all-to-all communication patterns. For example, after many OpenMP loops there is an implicit barrier that may be enhanced using broadcast-based synchronization mechanisms like the analyzed in Section 7.1.3. Even though substantial performance improvements could be obtained through optimization, it is worth noting that maximum gains will be only achieved by going back to the original problem and re-implementing a given algorithm without the constraints imposed by conventional on-chip communication. This is indeed a daunting task, yet cannot be totally discarded as the potential scalability, performance, or efficiency improvements may be huge.

### 7.1.2 Coherence Mechanisms

Chapter 1 explains how the capabilities and limitations of the on-chip interconnect have traditionally guided the design of shared memory architectures. Buses with ordered broadcast capabilities are feasible in processors with a handful of cores, situation that strongly suggests the use of snooping coherence due to its low complexity and high performance. As processors scale and the interconnect design shifts to switched NoCs, though, broadcast becomes a costly feature and the use of directory-based protocols becomes necessary. However, directory overheads and increased indirection limit the scalability of such schemes and encouraging the search for alternatives balancing interconnect cost and architectural cost (see Fig. 1.1). Some works even propose to adapt classical snooping protocols to modern switched NoCs [24, 25], attempting to regain their simplicity and performance.

Although the availability of inexpensive broadcast provided by BoWNoC may not represent the return of snoopy coherence, it may be beneficial to revisit existing techniques and assess their scalability. For instance, the protocol implemented by the Sun's Fireplane system would map especially well to the hybrid network proposed here, as it originally snooped addresses over a bus and transferred data through a switched network. The advent of BoWNoC could also greatly help to increase performance and reduce the complexity of existing protocols. By taking base on schemes like the proposed in Section 5.3, acknowledging at the coherence level can be omitted, greatly reducing the traffic in architectures such as HyperTransport [15] or in applications with a high rate of invalidations. Other examples that would benefit from BoWNoC include some variants of token coherence [16], which decouple performance and correctness (thereby reducing complexity), but that have not been used in manycore chips as they generate considerable broadcast. Also, new innovations such as race-free protocols [88] could be embraced. Finally, one can also envisage the use of explicit communication primitives within certain coherence protocols: for instance, some critical variables could be directly managed through the wireless medium to eliminate long delays in the critical path of L2 misses. These include several common idioms that require frequent global communication between processors, such as mutual exclusion to popular locations, producer-consumer communication, or reduction operations.

### 7.1.3 Synchronization and Control

Synchronization among a large set of cores is one of the functionalities that may benefit most from the BoWNoC paradigm. Locks and barriers are generally latency-critical and require significant amounts of global communication when involving several threads. Barriers are a particularly relevant case, as they normally have a chip-wide scope and can be easily imple-

mented with broadcast. Moreover, as explained in Section 3.2, the unique properties of the transmission of RF signals through a shared medium allow the design of alternative schemes for barriers. These aspects are actually investigated in Section 7.2, where we propose an architecture that exploits the BoWNoC uniquenesses to implement fast synchronization. Our study is limited to a speedup evaluation of unchanged versions of multithreaded applications, but the reality is that programmability will be also positively impacted by the cost reduction of global synchronization. The main reason is that maintaining sequential consistency, which is the most intuitive consistency model, becomes easier and less expensive. Finally, reducing the cost of synchronization may also have a transformative impact upon the design and programming of general-purpose GPUs, as barriers have been proposed as the basic blocks for inter-block communication within these systems [246].

Global communication and broadcast are also attractive features for control systems that operate at the network level or at upmost layers of the architecture. In the former case, congestion control generally makes use of distributed algorithms to notify congestion situations to distant cores [247,248], which could be replaced by a single broadcast message from the center of the congestion area. System-level control mechanisms can also take advantage of BoWNoC as demonstrated by Pande *et al*, which show that low-latency communication through wireless links increases the benefits of DVFS to reduce power consumption and thermal hotspot effects [154,161]. Such principles can be extended to other control systems that use distributed algorithms to reduce their cost, but that would be much more effective with fast and global communication.

#### 7.1.4 Programming Models

The differentiated management of critical variables has been already proposed in order to eliminate the bottlenecks caused by producer-consumer access patterns. In this approach, referred to as *Consumer Tagging*, the programmer tags the potentially conflicting addresses and the consumers associated to them. This represents a particular case of the “*always update through broadcast*” approach explained above, where the programmer does not need to know the consumer set. In related work, Psota *et al* discuss this case along with other new programming models that aim to combine the flexibility of shared medium and the hand-tuned performance of message passing, and that would benefit from an improved broadcast mechanism [249]. They analyze *Adaptive Constraint-Based Programming* and *Application Heartbeats*, both of which periodically broadcast information. The former uses broadcast to communicate or update a set of known constraints to worker cores; whereas, in the latter, worker cores broadcast their performance to scheduler cores, which later send instructions and performance goals to worker cores also through broadcast. Finally, we speculate that an efficient broadcast plane could be also used to improve the performance and redundancy properties of the well-known MapReduce model [160].

#### 7.1.5 Novel Computing Systems

Finally, it is worth remarking that the introduction of an effective broadcast plane may have a profound impact on the design of novel computing systems. A clear example is that of neuromorphic computing systems. Recent years have seen the rise of such brain-emulating systems, which model a potentially huge set of neurons within each core and communicate cores with multicast messages that simulate the behavior of neural spikes [250]. Given the multicast-driven nature of such communication and the strong scalability requirements of



this massive architecture, important improvements can be expected from the application of the BoWNoC paradigm. Current NoC approaches to serve neuromorphic traffic resort to hierarchical networks combining star and mesh topologies [251]. Within this context, BoWNoC could help to further scale the topology by reducing the cost of intra-cluster multicast communications.

Approximate computing [166] is another novel computing paradigm that matches with the characteristics of wireless on-chip communication. Approximate processors rely on the ability of some systems to tolerate certain loss of quality, relaxing the need for fully precise computations and substantially increasing the overall energy efficiency. Media processing and recognition, for instance, are applications that fall within this category. The WNoC paradigm in general and BoWNoC in particular are well-suited to this context due to the relatively high cost of ensuring a very high BER in chip-scale communications. By relaxing these requirements, wireless on-chip communication could achieve substantial performance improvements with much improved energy efficiency, strengthening the advantages of the approximate computing approach.

## 7.2 WiSync: An Architecture for Fast On-Chip Synchronization

In shared-memory programming, there are several common idioms that require fine-grain synchronization that generates frequent global communication between processors. One is mutual exclusion to popular locations, which involves repeatedly writing and reading locks and/or other variables. Another is frequently-accessed global barriers, which involve reading and writing counts and flags. Another pattern is producer-consumer communication which, in addition to the data communicated, often needs flags to coordinate writes and reads. Yet another idiom is broadcast and reduction operations.

Computer architectures have traditionally struggled to support these patterns efficiently. Some machines have provided advanced hardware support, such as the barrier network in Cray T3D [252], the collectives network in Blue Gene/L [253], and the fetch-and- $\Phi$  operations in the SGI Origin [254]. Also, there are multiple research proposals for advanced hardware support for synchronization (e.g., [22, 60, 62, 255–257]). Yet as technology scaling delivers larger and larger manycore chips, these patterns are expected to remain costly to support within the chip.

The use of the BoWNoC paradigm developed in this dissertation provides two key supports for the idioms outlined above. The first one is the inherent broadcast support of BoWNoC, which satisfies the requirement for global communication. The second one is that the latency of such communication is around one order of magnitude lower than in current on-chip networks, as shown in Chapter 5, and hardly dependent on the distance between source and destination. Moreover, adding a BoWNoC does not complicate the chip layout because no additional wires are needed between cores and incurs into scalable area and power overheads.

Here, we present WiSync, an architecture that leverages BoWNoC in large manycores to perform fast synchronization. The following sections detail the main design aspects of WiSync, putting emphasis on the interface between memory and the employed BoWNoC, as well as on the explicit support for synchronization operations. It also presents an example instruction set architecture extension, and discusses multiprogramming, virtual memory, and context switching aspects. We then evaluate WiSync for different system sizes and

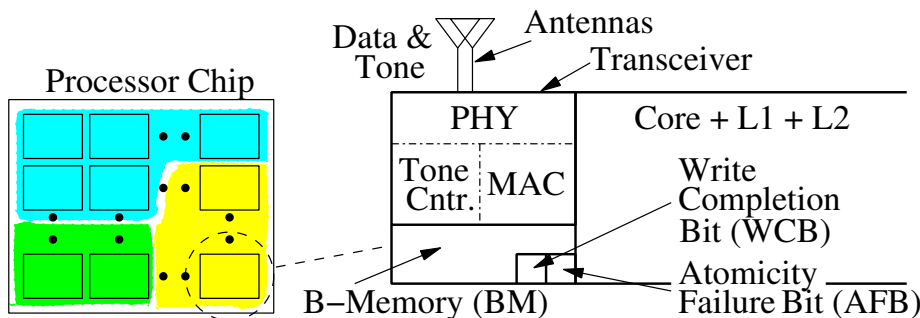


Figure 7.3: WiSync architecture. The different colors represent different programs running.

under different conditions, demonstrating that it can speed up synchronization operations by over an order of magnitude, and full applications by a factor of 1.12 in average.

### 7.2.1 Overview of WiSync

Figure 7.3 shows the WiSync architecture, the structure of which can be seen as a more detailed version of the architecture depicted in Fig. 6.3 of the previous chapter. Each tile in the manycore processor contains a core with its local instruction and data L1 caches, as well as a piece of the shared L2 cache. Besides this conventional setting, the tile contains a wireless RF transceiver, two antennas, a special memory called *Broadcast Memory (BM)*, and two single-bit registers called Write Completion Bit (WCB) and Atomicity Failure Bit (AFB). The transceiver has a module in charge of PHY actions, a module for MAC purposes, and a *Tone Controller*. The functionality and design decisions for each of the modules is detailed throughout the section.

Communication among cores occurs, on the one hand, via a regular NoC when variables stored in the cache hierarchy are accessed. On the other hand, wireless communication among cores takes place via a BoWNoC with two orthogonal channels when BM variables are accessed. First, we have a broadband channel that is used for plain data transfers and that is controlled by the MAC module, which implements the BRS-MAC protocol proposed in Chapter 5. We assume that the RF transceiver allows the transmission of  $\approx 80$  bits of data in a few nanoseconds, which is commensurate with the data rates achieved in state-of-the-art designs as discussed in Chapter 4. The second channel is much narrower than the Data Channel and is used to transfer tones. The main idea is that the channel inherently performs a collective OR of the tone signal as explained in Section 3.2, which enables a very efficient execution of barrier synchronization. This justifies the need for a second antenna and a Tone Controller. These aspects are further explained in Section 7.2.2.1.

From an architectural perspective, the key component of the WiSync is the per-core BM. Indeed, WiSync abstracts the wireless capability in the form of a per-core slice of memory that is used by variables declared of type *broadcast* in a given program. All BMs are directly connected to the wireless network, and they all contain the exact *same, replicated 64-bit variables*. When a core writes to a location in its BM, all the other BMs get automatically updated. Since there is only a single data transfer channel, all cores receive the update at the same time and with the same order of delivery, fact that ensures a total order of *writes* to the BM. Figure 7.4 shows this idea: three concurrent writes by different processors to variables  $x$  and  $y$  can result in different interleavings, but no update is lost

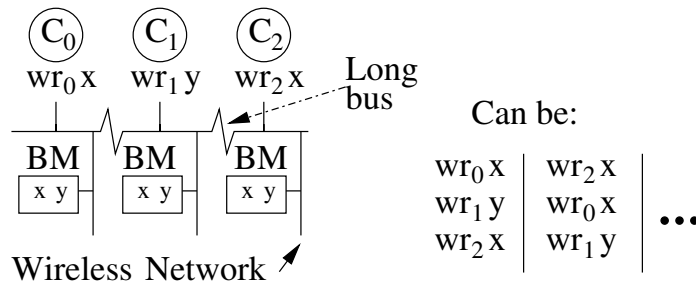


Figure 7.4: How global writes appear in WiSync. The jagged line means that the non-local BMs are physically far.

and all processors observe the same write interleaving. The core accesses its BM with plain loads and stores, using virtual addresses that are translated in the Translation Lookaside Buffer (TLB), but that bypass the L1-L2 hierarchy that contains regular variables. Conventional access-bit permissions in the TLB ensure that programs access only their own data, thus supporting multiprogramming. These and other aspects relative to the memory operation (e.g. context switching, process migration, and memory access protection) are further detailed in Section 7.2.2.2.

As mentioned above, core accesses its BM with plain loads and stores to a given address range. A load instruction (*ld R, BM\_addr*) reads from an address in the local BM into a register, which we assume it takes a few cycles. A store instruction (*st R, BM\_addr*) takes a register and stores its value to an address in the local BM and (using the wireless network) in *all the BMs*. To maintain coherence, no BM (including the local one) is updated until the wireless network is able to successfully convey the write to the remote processors. Besides plain loads and writes, WiSync supports bulk loads and writes involving a number of consecutive BM locations, as well as atomic read-modify-write (RMW) instructions such as Compare-and-Swap (CAS). A RMW instruction implies reading a local BM location, bringing the data to the pipeline, updating the data in the pipeline, and then writing and updating the result in all the BMs. To succeed, the instruction must guarantee atomicity from the time it reads from the BM until all the BMs are updated, possibly after several collisions. This is enforced through the WCB and AFB bits. More details on these facets are provided in Section 7.2.2.3.

## 7.2.2 Design Decisions and Implementation Issues

### 7.2.2.1 BoWNoC: Uses of the Wireless Channel

WiSync uses two different frequency channels to communicate. As shown in Fig. 7.5, a broadband channel is used for synchronization data transfers, whereas a narrower channel located at higher frequencies offloads the transfer channel by efficiently implementing the common barrier synchronization pattern. Even though the channels are at 60 and 90 GHz in our particular case scenario, we speculate that successive generations of WiSync will employ higher frequencies by virtue of the area and bitrate scalability requirements of the architecture (see Chapter 4 for more details).

**Data Channel:** although WiSync admits the use of several data channels to enhance performance in multiprogramming and heterogeneous computers, power and area reasons

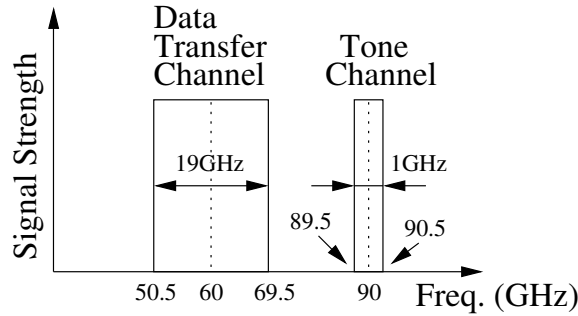


Figure 7.5: Transmission channels in WiSync.

advise against it. Instead, we use a single broadband channel for data transmission to keep implementation overheads reasonable. We time-slot the channel at the processor clock granularity — 1 GHz in our case. Typically, the Data Channel will be used for single writes, such as a write to a reduction variable or to a lock. This involves one BM location, which includes a 64-bit datum, its address, a Bulk Bit and a Tone Bit. The size of the BM determines the length of the address, which will be of 11 bits in this work for a total of 77 bits per transfer. As we will see, the Bulk and Tone Bits are used to indicate a long transmission or the beginning of a tone transaction.

In consonance with current transceiver implementations and future perspectives analyzed in Chapter 4, we will consider the data rate to be  $R = 19.25 \text{ Gb/s}$  so that the 77 bits can be transmitted in 4 cycles. Assuming a simple modulation with a spectral efficiency  $S_E = 1 \text{ bps/Hz}$ , this implies a bandwidth requirement of  $\sim 19 \text{ GHz}$ , achievable in the 60 GHz band. Given the trends obtained in Section 4.2 and the figures shown in Section 4.2.3, these frequency bands would be enough to accommodate one transceiver and antenna per core in 64-core systems with reasonable power and area overheads.

Access to the Data Channel is controlled by the MAC module, which implements the BRS-MAC protocol detailed in 5.3. Basically, cores listen to the medium and transmit a first piece of the packet in the next free slot. During the second cycle, the core waits for other nodes to complain in case there was a collision in the first cycle of transmission. If there was a collision, the transfer is aborted and rescheduled using an exponential backoff, and the channel is free in the third cycle. Otherwise, the rest of the 77-bit message will be sent during the next three cycles with the guarantee of no collisions. For all this, a successful transmission will take 5 cycles, whereas a collision will add a minimum delay of 2 cycles. To ensure a consistent order of delivery, BM writes are never redirected to the wired plane as suggested in Section 5.3.

WiSync supports bulk writes, which involve multiple BM locations within the same operation. This is possible by virtue of the variable-length transmission support delivered by the BRS-MAC protocol (see Section 5.3.1). Every message carries a Bulk Bit in the header that is set to one in the first chunk of a bulk transaction. When the MAC module reads this bit in a transmission that did not collide, it knows its length: if the Bulk Bit is zero, the message takes 5 cycles in total; otherwise, it will take 15 cycles. The reason why this transmission only takes a total of 15 cycles rather than  $4 \times 5 = 20$  cycles is that the second, third, and fourth words of the bulk message do not need to either check for collisions or carry the address, Bulk Bit, and Tone Bit. This is because 1) knowing the length of the transmission prevents collisions as nodes wait for the appropriate number of cycles, and 2) addresses are known, whereas the Bulk Bit and Tone Bit are not required in trailing words.

**Tone Channel:** WiSync also supports *tone barriers* based on the proposal of Oh et al. [57] for transmission lines. The implementation of a tone barrier requires the use of a *Tone Channel* orthogonal to the Data Channel and justifies the inclusion of a Tone Bit in every wireless message. As described in Section 3.2, nodes send a tone rather than actual data and interpret the presence or absence of signal as a collective OR: a '0' implies that no core is sending a tone, whereas a '1' implies that at least one core is sending a tone. In WiSync, this property is used to efficiently implement a barrier as follows. The first core that reaches the barrier sends a message through the Data Transfer channel with the Tone Bit set. On receiving the message, all the other nodes respond with a continuous tone in the Tone Channel. Then, as soon as a core reaches the barrier, its transceiver stops sending the tone. Hence, when the tone finally disappears, the absence of signal is interpreted as that all cores have arrived at the barrier.

To determine the required bandwidth, we assume that the Tone Channel is also slotted at the system clock granularity (1 GHz). A tone behaves like a 1-bit message, for a resulting transmission rate of 1 Gbps given the aforementioned slot length. This implies that the frequency band of the Tone Channel will be 1 GHz at most. With such a short band, the power needed to support it will be very small as compared to the power required at the Data Transfer channel. The reduced bandwidth and expected simplicity of the associated transceiver also advises the use of a frequency band above of that the Data Transfer channels.

Time-slotting the channel allows the Tone Channel not only to react to changes in the arrival of cores to a barrier, but also to support multiple barriers. Consider that there may be multiple tone barriers (each corresponding to a different BM address) that want to use the Tone Channel at any given time. In this case, slots are assigned round-robin to the tone barriers that are currently active, which is the period of time from the moment the first core arrives to the barrier until the last participating core arrives. Figure 7.6(a) shows an example of the Tone Channel distribution given 1, 2, or 3 active tone barriers.

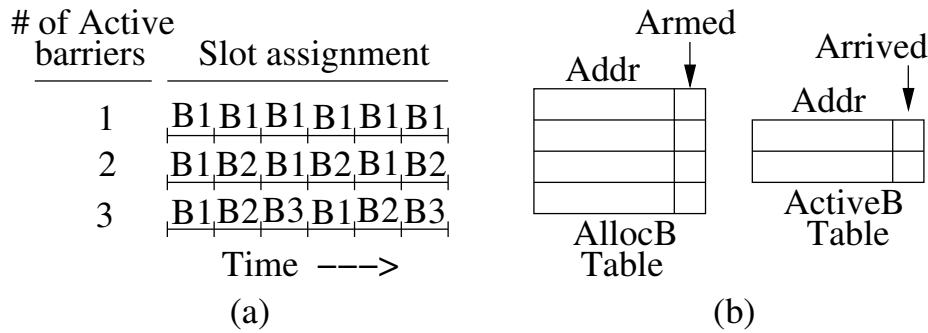


Figure 7.6: Sharing the Tone Channel among several barriers.

Access to the Tone Channel is controlled by the *Tone Controller* module. To support sharing, the controller in each node keeps two tables (Figure 7.6(b)): one with the *allocated* tone barriers (*AllocB*) and one with the *active* tone barriers (*ActiveB*). These tables contain the same addresses and in the same order in all the nodes of the chip. Such chip-wide consistency is required for correct assignment of slots to barriers, and it is easy to support thanks to the broadcast capabilities of the BoWNoC.

In AllocB, each entry has the BM address of a tone barrier variable plus an *Armed* bit. When a tone barrier variable is allocated by a program, an entry is created in the AllocB of

all the nodes in the chip. It is also at this point that the OS in each node sets the *Armed* bit to either 1 or 0. Finally, when the tone barrier variable is deallocated (possibly at program termination), its entry is removed from the AllocB of all nodes. At the same time, all the entries lower in the table are shifted up. Next section gives more details on the allocation and deallocation of BM variables and the interaction with the Tone Controller.

### 7.2.2.2 Memory Operation and Interface

The BM in a node contains space for all the allocated variables that are declared as *broadcast* type in the programs currently running on the chip. When a program allocates a broadcast variable, the variable is allocated in the local BM and in all the BMs, and is tagged with the process ID (PID) of the program for protection purposes. Such variables are replicated in all the BMs and their values are kept consistent at all times with the aid of the wireless network. When a variable is deallocated, it is removed from all the BMs.

The optimal size of the BM in each node is likely to be small, e.g., four 4-KB pages, for two main reasons. First, large memories require many address bits to be included in each wireless message. With 16KB, we already need 11 bits. Secondly, programs do not usually declare many broadcast variables. If the BM runs out of space for a variable, one can envision seamlessly allocating the variable in a page of regular memory, and access it through the wired network. Memory operation and interface aspects are explained more deeply in what follows.

***BM Entry Allocation and Protection:*** To access the BM, WiSync uses address translation based on a Translation Lookaside Buffer (TLB), so that programs do not have to manage memory and can benefit from access protection. However, traditional page-level assignment is not optimal, since the BM is small, and cannot afford the memory wastage due to page-level fragmentation when multiple programs want to share the BM. Indeed, if each program allocated a single broadcast variable but was assigned a full page, the BM would run out of space very soon.

To address this problem, WiSync uses page-level TLB translation but lets different programs use different chunks of the same physical BM page — tagging each chunk with the PID of the program that owns it. Different programs have virtual pages mapped to the same physical BM page, but each program only uses its own, non-overlapping chunks of the physical page. The smaller the chunk size is, the better the page can be shared, and there is less fragmentation. However, there is more bookkeeping and tag overhead. We do not examine these trade-offs in this work, but advocate for an effective use of the BM and, thus, propose the use of 64-bit chunks.

To allocate an entry in the BM, the core uses a special allocation instruction. The instruction broadcasts a message in the wireless network that contains the address (11 bits), PID (e.g., 8 bits), and a few miscellaneous bits. On reception, every node in the chip allocates a local entry and tags it with the PID. When this operation succeeds, the local BM allocates and tags an entry at the exact same address. While it appears inefficient that a variable takes space in all the BMs (even in the BMs of cores that do not run the relevant program), we do it for simplicity. Once allocated, when a core accesses a variable in the local BM, the address is first translated. Then, at the target BM location, the program's ID is compared to the PID tag (Figure 7.7). A mismatch is a protection violation.

The allocation of a variable that uses the Tone Channel proceeds in the same way. However, in addition, the OS in the receiving nodes records whether or not this variable

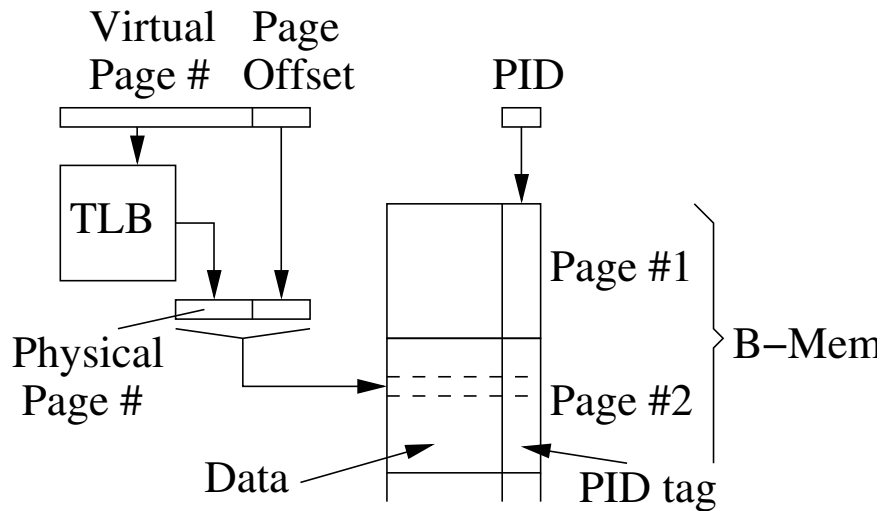


Figure 7.7: B-memory address translation.

is *armed*. Armed means that there is (or will be) a thread running on the local core that belongs to that program and, therefore, will participate in the tone barrier protocol for the variable. For example, if the variable is armed, when the node receives a message initiating a tone barrier for that variable, the Tone Controller will participate by locally starting a tone. For a correct operation of the tone barrier, the program or the compiler needs to specify which threads will participate. In case that participation be determined at runtime, the tone barrier cannot be used.

**Interface to Basic BM Instructions:** Loads to the local BM always succeed. Stores and RMW instructions, however, need special handling. For such treatment, WiSync uses the WCB and AFB bits (see Figure 7.3), which are set/reset in hardware and accessible to the software through a register. The WCB is set when the update operation of a write or a RMW operation completes, which means that both the global broadcast and the local BM update are finished. In the case of a RMW, if its atomicity fails, the instruction completes without the write and, at that time, WCB is set anyway. The atomicity failure of a RMW instruction is indicated through the AFB.

When a core executes a store instruction, the local transceiver first attempts to broadcast the update to all the remote BMs. If there is a collision, the transceiver keeps retrying until it succeeds. After it succeeds, the local BM is updated, the WCB gets set, and the pipeline receives the acknowledgment that the store is performed. No subsequent store from the local core can proceed to the global broadcast until the current one has finished. Subsequent loads from the local core may or may not be allowed to read any BM address while the current write is in progress, depending on the memory model desired. If loads are allowed to, we have a TSO memory model; if they are not allowed to read any BM address beyond the one being written, we have a sequential consistency model.

When a core executes a RMW instruction, the hardware reads the datum from the local BM into the pipeline, updates it in the pipeline, and then tries to write the result to the BM. As usual, the write involves performing the global broadcast first and, when it succeeds, updating the local BM. It is possible that, in between the read from the local BM and a successful global broadcast, a remote node manages to update the variable in the local BM.

This is detected in hardware by comparing the outgoing address to the incoming addresses. In this case, the atomicity of the instruction has failed. As soon as this occurs, the AFB bit gets set, and the update is aborted — i.e., the RMW instruction neither broadcasts its value nor it updates the local BM. The WCB also gets set because the RMW operation is now terminated. Consequently, a RMW instruction needs to be followed by a software check of the WCB and AFB bits. The instruction has executed atomically and, therefore, performed the write only when WCB=1 and AFB=0. If, instead, WCB=1 and AFB=1, the write never occurred, and the RMW instruction has to be re-executed. Fig. 7.8 in Section 7.2.2.3 gives some examples of such course of operation.

**Interface to Tone Channel Instructions:** Cores have special BM load and store instructions that enable the use of the Tone Channel for a particular BM address. They are *tone\_ld R, BM\_addr* and *tone\_st R, BM\_addr*. With these instructions, the Tone Channel can be used to implement a barrier very efficiently. Specifically, when a core reaches the barrier, it performs a *tone\_st* operation on the BM location (note that this is not an ordinary update of the location). Then, the core keeps reading the BM location using *tone\_ld*. The load will return a special code when all the participating cores have performed the *tone\_st*. In the meantime, the core may choose to do other work, while periodically executing *tone\_ld*. The implementation of these instructions relies on the Tone Controller as follows:

- On a *tone\_st*, the Tone Controller does not update does not update the BM location, but instead checks whether this address is currently issuing a tone in the Tone Channel. If so, the local core is not the first core to arrive at the barrier, and the controller stops issuing the tone; otherwise, the local core is the first one to arrive, and the controller sends a message in the Data Transfer channel with the address of the BM location and the Tone Bit set. The content of the 64-bit data field is immaterial.
- When the Tone Controller observes an incoming message in the Data Transfer channel with the Tone Bit set, the core starts issuing a continuous tone in the Tone Channel only if the local core participates in the barrier. The participation of the core can be known by checking the AllocB table.
- As soon as the Tone Controller detects that the Tone Channel has fallen silent, it automatically toggles the value of the local BM location as this means that all cores have arrived at the barrier. Such location can only take the values Zero or Non-zero.

With these steps, WiSync implements an efficient sense-reversing barrier [1]. When a core spinning with *tone\_ld* on the address of the barrier observes that the value has changed due to the action of the Tone Controller, it knows that the barrier has completed.

**Context Switching and Thread Migration:** the Data Transfer channel in WiSync is designed to operate correctly under context switching, thread migration, and multiple threads sharing the same core. Consider first a core that preempts a running thread. Even while the thread is preempted, updates from other cores to broadcast variables will reach the local node and update the local BM. When the thread is rescheduled again, it will observe the correct BM state.

A thread can also migrate to another core and resume execution there seamlessly. This is because the state of the BMs is identical in all the nodes. The non-BM state is not relevant to our discussion, and we assume that the cache coherence keeps it coherent. Finally, multiple threads can share the same core, and update the same or different BM variables.



The situation is different for programs that use the Tone Channel. The reason is that this channel is managed in hardware. In this case, threads can still be preempted, but cannot migrate or share the core with another thread that also uses the same tone barrier. Indeed, when a tone barrier is allocated, the OS either arms or disarms the local AllocB entry, depending on whether the local thread will participate or not. Migrating a thread would require somehow migrating this state, which is costly. Also, two threads on the same core trying to use the same tone barrier would result in incorrect operation.

### 7.2.2.3 Supporting Synchronization Operations

We now outline how WiSync supports some of the popular synchronization operations.

**Basic Read-Modify-Write Primitives:** Figure 7.8(a) shows the pseudo-code of the algorithm used to execute a basic RMW operation, which is fetch&increment in this particular example. After the core executes the RMW instruction, it checks the WCB register bit until it gets set and, then, it reads the AFB register bit. If it is set, the instruction has been aborted and it has to be retried; otherwise, the operation performed successfully. Similar pseudo-code is used for test&set and fetch&add.

<pre> retry: fetch&amp;inc R, BM_addr while (!WCB) {} if (AFB) {   jmp retry } else {   /* success */ } (a) </pre>	<pre> retry: ld R_old, BM_addr . . . if (!CAS(BM_addr, R_old, R_new)) {   jmp retry /* comparison already failed */ } else {   while (!WCB) {}   if (AFB) {     jmp retry /* atomicity failure */   }   else {     /* success */   } } (b) </pre>	<pre> for ( ) {   /* work */   barrier: local_sense = !local_sense            tone_st (addr)            spin until (tone_ld(addr)==local_sense) } (c) </pre>
	<hr style="width: 100%;"/>	
	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <hr style="width: 100%;"/> <p>Broadcast Writer</p> <pre> for ( ) {   local_sense = !local_sense   write data   count = N   release = local_sense   spin until (count==0) } </pre> </div> <div style="text-align: center;"> <hr style="width: 100%;"/> <p>Broadcast Readers</p> <pre> for ( ) {   local_sense = !local_sense   spin until (release==local_sense)   read data   fetch&amp;add (count,-1) } </pre> </div> </div>	
		<hr style="width: 100%;"/> <p>(d)</p>

Figure 7.8: Examples of code used for synchronization.

Figure 7.8(b) shows pseudo-code for CAS. In this case, we can determine that the comparison in the CAS failed before we check for atomicity failure. This speeds-up the execution and, by avoiding an unnecessary write and wireless update, reduces the load on the BoWNoC. The pseudo-code assumes that the CAS returns 0 if the content of the BM location was different than the CAS' second argument.

**Barriers:** There are two types of barriers: AND-barriers and OR-barriers. AND-barriers are the conventional ones and, in our implementation, we use the popular sense-reversing barrier algorithm [1] with fetch&increment. A single 64-bit BM entry contains the *Count* variable in one 32-bit word and the *Release* flag in the other. These variables are accessed with 32-bit loads, stores, and RMW operations. Alternatively, OR-barriers (also called Eureka) are triggered as soon as one of the participating processors detects a certain condition, e.g., overflow of a variable, the solution of a parallel search, or an exception. We implement them as a boolean variable in a BM location. All processors periodically read it; when one detects the condition, it changes the variable. We also use a sense-reversing implementation here to allow barrier reuse.

Note that AND-barriers are supported more efficiently with the Tone Channel. In accordance with the operation rules set in Section 7.2.2.2, each core executes the pseudo-code algorithm of Figure 7.8(c). A barrier with sense-reversing is implemented this way with minimal communication. While the first core to arrive sends a message through the Data Transfer channel, the other cores simply issue a colliding tone in the Tone Channel. Then, all cores spin in their local BM, offloading the Data Transfer channel and, thus, offering a scalable barrier implementation.

**Producer-Consumer Operation:** To support the single producer, single consumer pattern, we use a BM entry for the data and a BM entry for a flag. The producer writes the data to the BM address, sets the flag, and then spins on the flag until it is cleared. The consumer spins on the flag until it is set, reads the data, and then clears the flag. The process repeats. Often, the producer will use a *bulk\_st* instruction, which triggers 4 transfers in the Data Transfer channel and updates 4 consecutive BM locations.

**Reduction and Multicast/Broadcast:** The BM is perfectly designed to support reductions. For instance, when all the cores need to add to a single location, they can use *fetch&add R, BM\_addr*. To support different types of reductions, one can include other *fetch&Φ* instructions and, for scientific computations, floating-point versions of them. Very tight reduction loops are supported efficiently.

Multicast/broadcast is the single producer, multiple consumers pattern. It is also supported very efficiently through a write of the producer to a BM entry and reads of all consumers from it. To provide ordering between the writer and the readers, we can use an extension of the full-empty flag discussed above. Specifically, we use two additional variables in the BM: one variable is a count, and the other a toggling flag, effectively implementing a sense-reversing barrier. The pseudo-code is shown in Figure 7.8(d). The producer writes the data, sets the count to the number of readers  $N$ , toggles the flag, and then spins on the count until it is zero. Each reader spins on the flag until it toggles, reads the data, and then decrements the flag with *fetch&add*. The process repeats.

Alternately, a lower overhead solution is to use a tone barrier. After the producer writes the datum, it writes to another BM location using *tone\_st*, and spins on it using *tone\_ld*. Each reader, as it receives the message with the Tone Bit set, starts to issue a tone. Then, as each reader reads the datum, it stops its tone. When the Tone Channel becomes silent, *tone\_ld* in the producer returns a toggled value, and the next cycle can start.

### 7.2.3 Performance Evaluation

We use cycle-level execution-driven simulations to model a 64-core manycore with and without WiSync support. We use the Multi2sim [258] simulator, wherein we implemented the BM and the necessary modules for the cycle-accurate simulation of the wireless network, including the detection of collisions. We also embedded a hybrid controller (see Chapter 6) that forwards BM variables through the wireless plane and the regular variables through a conventional NoC. Table 7.1 shows the general parameters of the architecture and, in the lower part, those related to WiSync. We rely on realistic transceiver implementation trends to obtain the bandwidth of the wireless network (see Chapter 4 for more details).

We compare four manycore configurations, as shown in Table 7.2. *Baseline* is a plain manycore with no wireless hardware. For synchronization, it uses CAS (only writing when the location is found free) and a sense-reversing centralized barrier. *Baseline+* enhances *Baseline* hardware with: (1) virtual tree-based broadcast in the on-chip network with flit

Table 7.1: Architecture modeled. RT means round trip.

General Parameters	
Architecture	22nm manycore with <b>16–256 cores</b> (def: 64)
Core	Out of order, 2-issue wide, 1GHz, x86 ISA
ROB; ld/st queue	64 entries; 20 entries
L1 cache	Private 32KB WB, 2-way, 2-cycle RT, 64B lines
L2 cache	Shared with per-core 512KB WB banks
L2 bank	8-way, 6-cycle RT (local), 64B lines
Cache coherence	MOESI directory based
On-chip network	2D-mesh, 4 cycles/hop, 128-bit links
Off-chip memory	Connected to 4 mem controllers, 110-cycle RT
WiSync Parameters	
Per-core BM	16KB, 2-cycle RT, 64-bit wide entry
Tone Channel	1Gbps; 1-cycle transfer latency
Data Tran. Channel	19Gbps; 5-cyc transfer lat; collision detec. cyc 2
Collision handling	BRS-MAC (see Section 5.3.1)

Table 7.2: Architecture configurations compared.

Configuration	BM?	Broadcast HW	Locks	Barriers
Baseline	No	No	CAS	Centralized
Baseline+	No	Virtual Tree	MCS	Tournament
WiSyncNoT	Yes	Wireless	Wireless	Wireless (Data)
WiSync	Yes	Wireless	Wireless	Wireless (Data+Tone)

replication at the router crossbars [20], (2) MCS locks [259], and (3) Tournament barriers [259]. WiSyncNoT is WiSync without the Tone barrier support.

We evaluate the performance of WiSync for two sets of kernels and a set of applications as summarized in Table 7.3. First, we run a set of CAS-intensive kernels in order to obtain the number of CAS operations that can be executed per second in processors of up to 128 cores. Second, we analyze the performance of WiSync in a set of kernels that execute barriers in configurations with up to 128 cores. Finally, we obtain the execution time of the entire SPLASH-2 [10] and PARSEC [11] benchmark suites for 64 threads.

### 7.2.3.1 Compare-and-Swap Synchronization

The first set of kernels consists of three kernels that execute CAS operations on lock-free data structures. In the ADD kernel, there is a shared queue to which, through a CAS, threads attempt to insert nodes taken from their private memory pools. In the FIFO and LIFO kernels, threads both enqueue and dequeue nodes from the shared queue. Since the kernels analyzed here involve lock-free structures and do not use barriers, the results are independent of the lock and barrier implementation. Consequently, we simply compare WiSync (where CASes use the BM) to Baseline (where CASes use the cache hierarchy). In all cases, a CAS only writes when its comparison succeeds.

In all the kernels, a given number of instructions are executed between successive node insertions or queue accesses. This number is a parameter and is used to adjust the amount

Table 7.3: Kernels and applications executed.

CAS Kernels	FIFO, LIFO, ADD
Barrier Kernels	TightLoop, Livermore loops 2, 3, 6
Application Suites	SPLASH-2, PARSEC

of CAS contention. Figure 7.9 shows the throughput of the FIFO, LIFO, and ADD CAS kernels on the two architectural configurations as we change the number of instructions (small numbers are to the right). Top and bottom charts show the number of successful CASes per 1,000 cycles corresponding to 64-core and 128-core executions, respectively.

The figures show that WiSync is able to attain a much higher CAS throughput than a conventional architecture. For 64 cores, there is little or no difference between the architectures when the number of instructions between CASes is 8-16K or larger. However, the difference increases as the critical section becomes smaller and contention rises. By the time we have about 2K instructions, WiSync delivers a CAS throughput that is about one order of magnitude higher than Baseline. With 128 cores, at about 4K instructions in the critical section, WiSync delivers a one order of magnitude higher CAS throughput than Baseline.

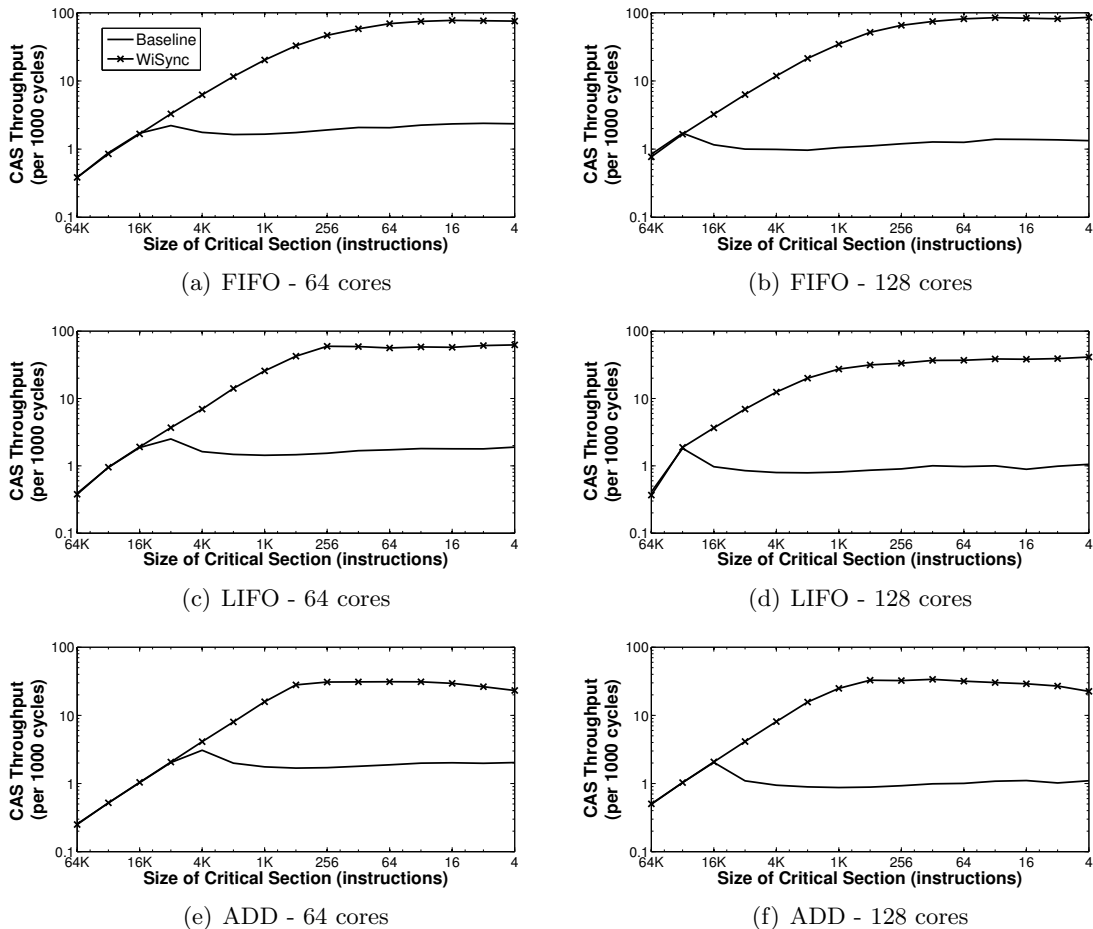


Figure 7.9: CAS throughput of three kernels on different architectural configurations for several critical section sizes and 64 or 128 cores. Higher is better.

Applications that use CASes to implement lock-free synchronization could benefit from the improved performance of WiSync. For instance, *mandelbrot* computes a Mandelbrot fractal set using a set of worker threads that pass their results to a collector thread via a lock-free stack. The finer the work granularity, the higher the CAS contention in the stack. We obtained speedups of up to 19% and 38% with 64 cores and 128 cores. Another example is *cannal*, from PARSEC, which handles a set of pointers through CAS. By using WiSync to perform those operations, we achieved a 38% speedup for 64 threads.

As we will see, this increase in terms of CAS throughput also benefits applications with high lock contention in settings where locks are implemented using CAS operations.

### 7.2.3.2 Barrier Synchronization

The second set of kernels consists of four loops that execute barriers. The first loop is *TightLoop*, which represents a very demanding environment, and in which each thread adds-up the contents of a 50-element array into a local variable and then synchronizes in a barrier. Figure 7.10 shows the number of cycles that each iteration of the loop takes on the different architectural configurations as we change the number of cores from 16 to 256. Note that the Y-axis in this plot and most of the others is *logarithmic*.

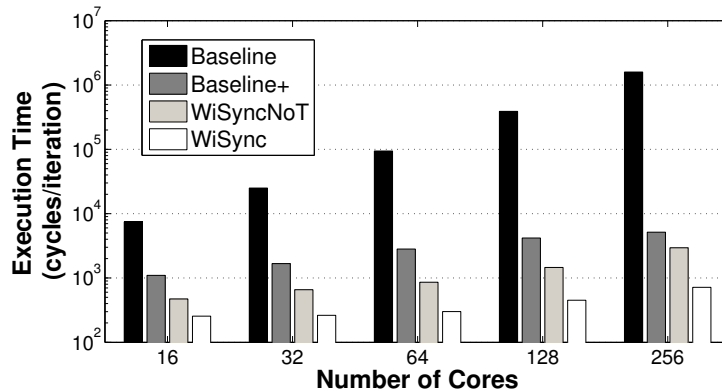


Figure 7.10: Execution time of *TightLoop* on different architectural configurations. Note that the Y-axis is *logarithmic*.

The figure shows a large difference between the configurations. As we increase the core count, the execution time of WiSync remains low, thanks to the Tone Channel. WiSync’s execution time is about one order of magnitude lower than Baseline+ (which uses a Tournament barrier), and two to three orders of magnitude lower than Baseline. On the other hand, WiSyncNoT takes 2–6x longer than WiSync to execute because of collisions in the Data Channel. Hence, the Tone Channel is useful for barrier-intensive codes. Notwithstanding this, WiSyncNoT’s execution time is 2x–4x lower than Baseline+.

The rest of kernels belong to the Livermore loops [260]. We focus on the evaluation of loops 2, 3, and 6 because, as Sampson et al. [256] argue, they are the representative ones with regard to fine-grained synchronization. Some other loops are embarrassingly parallel (loop 1), whereas others are serial (loops 5 and 20) or are very similar in structure to another loop. We parallelize the loops using the methodology detailed in [256], putting special attention on data alignment and other aspects to minimize coherence traffic [57].

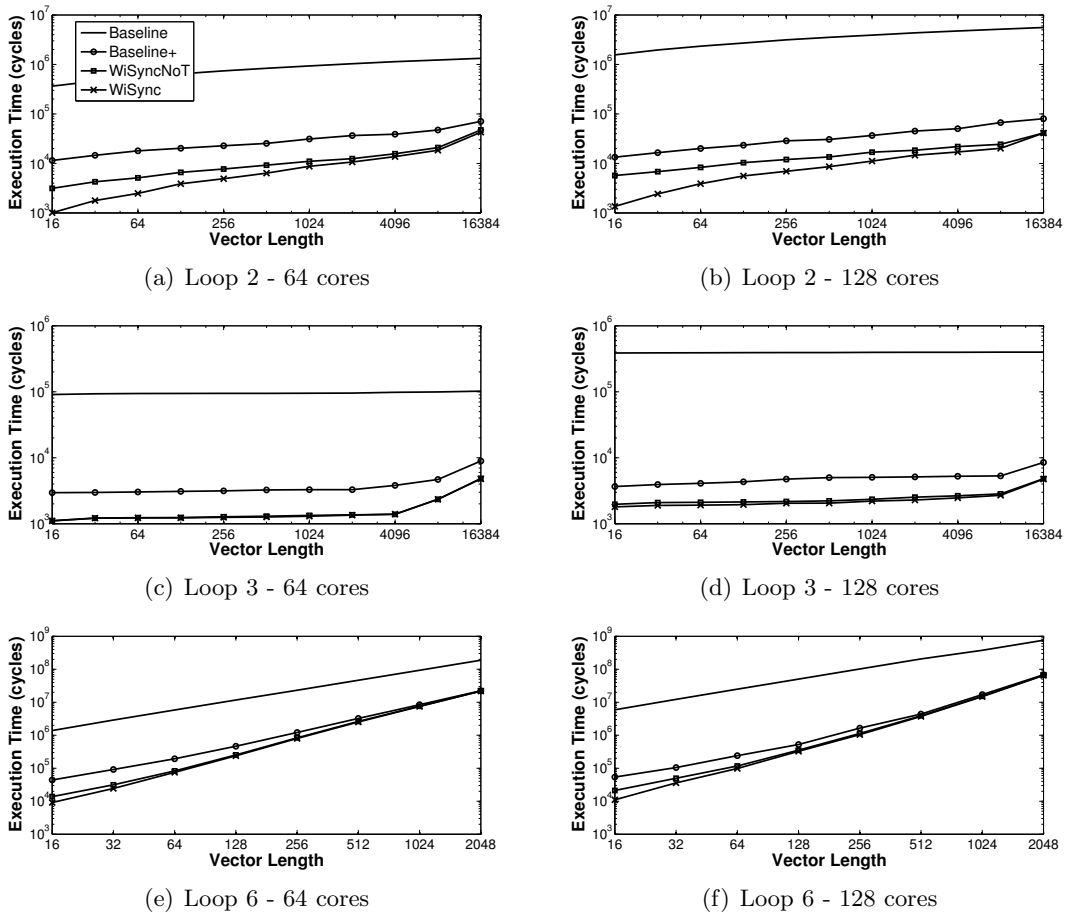


Figure 7.11: Execution time of the Livermore loops on different architectural configurations for several vector sizes and 64 or 128 cores.

Figure 7.11 shows the execution time of Livermore loops 2, 3, and 6 on the different architecture configurations as we change the vector length. The top and bottom plots correspond to 64-core and 128-core executions, respectively.

Starting with the three upper charts, we see that WiSyncNoT and WiSync are several times faster than Baseline+, and two orders of magnitude faster than Baseline. The gains are highest with small vector lengths, where the overhead of the barrier relative to the computation is most significant. As the vector lengths increase, the computation time becomes higher, and the barrier time (including any collisions) becomes relatively less important. As a result, Baseline+ tends to get closer to WiSyncNoT and WiSync. This is especially the case for Loop6, which has a large loop body. We see that WiSync is significantly faster than WiSyncNoT in Loops 2 and 6 for modest problem sizes. The reason is similar to that in TightLoop: a burst of arrivals causes collisions in the Data Channel in WiSyncNoT which, due to the modest duration of the loop, have a significant impact on latency. WiSync eliminates the collision by using the Tone Channel.

If we examine the lower plots, we see a wider gap between Baseline and the rest of configurations for 128 cores. The difference between WiSync, WiSyncNoT, and Baseline+ follows similar trends as for 64 cores.

Overall, we conclude that, while WiSync’s Data Transfer channel does well across the board, the Tone Channel is even better for some workloads.

Table 7.4: Lock (L) and barrier (B) calls in SPLASH-2 and PARSEC for 64 cores.

SPLASH-2 ( <i>default</i> )					
	<i>L</i>	<i>B</i>		<i>L</i>	<i>B</i>
barnes	139488	1088	radiosity	551248	640
cholesky	171506	256	radix	1502	704
FFT	128	768	raytrace	692084	64
FMM	96478	2176	volrend	153406	2560
LU	128	4288	water-nsq	137344	1280
ocean	26624	57600	water-sp	2434	1280
PARSEC ( <i>simsmall</i> )					
	<i>L</i>	<i>B</i>		<i>L</i>	<i>B</i>
blackscholes	0	64	fluidanimate	13183366	3200
bodytrack	12652	4568	freqmine	17690	0
canneal	272	2048	streamcluster	0	729408
dedup	160404	0	swaptions	0	0
facesim	17893	0	vips	39561	0
ferret	0	3514	x264	17832	0

### 7.2.3.3 Full Application Evaluation

We used the entire SPLASH-2 [10] and PARSEC [11] suites to evaluate the impact of WiSync on the execution speed of full applications. We consider 64 threads and use the standard input set sizes in SPLASH-2 and *simsmall* in PARSEC, to then obtain the performance of the parallel region of each application. Some applications have special considerations: for instance, *dedup* and *fluidanimate* declare arrays of locks larger than the 16KB BM used here. In those cases, we allocate the first four pages in BM and the rest in plain memory. Also, we had to modify the OpenMP libraries in order to evaluate *freqmine*.

To put the importance of synchronization in context, Table 7.4 contains the number of calls to locks and barriers for each and every SPLASH-2 and PARSEC application. Note the high number of barriers calls in *ocean* or *streamcluster*, and of lock calls in *radiosity*, *raytrace*, or *fluidanimate*.

Figure 7.12 shows the speedup of Baseline+, WiSyncNoT, and WiSync over the Baseline architecture for all the applications. The two rightmost set of bars show the arithmetic and the geometric mean, respectively.

Based on the geometric mean, WiSync delivers an average speedup of 1.23 over Baseline. Moreover, compared to the more advanced Baseline+ design that uses MCS locks and tournament barriers, WiSync delivers an average speedup of 1.12. These are significant improvements. We also see that WiSyncNoT performs about the same as WiSync. This is because the wireless Data Channel is not very utilized in these applications.

WiSync speeds-up nine applications. The others have too little synchronization for WiSync to make a difference. WiSync shows its best gains in applications that frequently use barriers, such as *streamcluster* ( $\sim 6X$ ) and *ocean* ( $\sim 2X$ ). In addition, significant speedups are also attained in a few lock-intensive applications such as *raytrace* (speedup close to 3) and *radiosity*. Many applications do not use fine-grain synchronization and, therefore, the improvements of WiSync make little difference. Baseline+ shows low speedups in some applications, due to the overhead of its more sophisticated synchronization implementations.

In *dedup* and *fluidanimate*, the locks did not fit in the BM, and we allocated a fraction of the locks in plain memory. However, simulations with an infinitely large BM did not yield any further speedup.

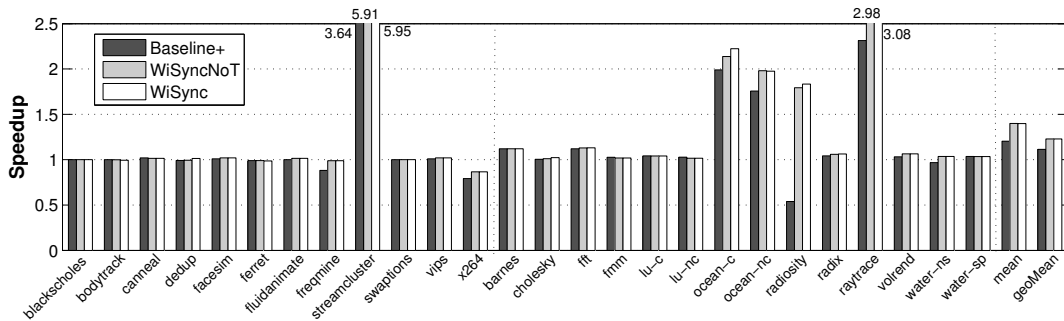


Figure 7.12: Speedup of the SPLASH-2 and PARSEC applications on different architectural configurations for 64 cores.

To understand these results better, Table 7.5 shows the percentage of the cycles in which WiSync and WiSyncNoT use the Data Channel. We show data for the most demanding applications and the geometric mean *of all* the applications. From these numbers, we see that WiSync and WiSyncNoT use the Data Channel for less than 0.1% and 0.2% of the time on average. In addition, there is little contention. It can be shown that, on average, the latency of a transfer in the Data Channel is 5.6 cycles. Such number goes up to 9.8 without the Tone Channel, mostly due to the latency of applications that use barriers frequently.

Table 7.5: Utilization of the Data Channel in WiSyncNoT and WiSync in % of the total cycles for the most demanding applications.

Arch	Str	Rsty	W/ns	Flu	Ray	O/c	O/nc	GeoM
WiSyncNoT	2.99	2.07	1.99	1.83	1.58	0.79	0.72	0.18
WiSync	0.01	2.06	1.97	1.83	1.55	0.26	0.23	0.06

### Sensitivity Study

To study the impact of the memory and network latencies on the SPLASH-2 and PARSEC speedups, we perform a sensitivity study with the configuration variants shown in Table 7.6. *Default* is the configuration we have used so far. *SlowNet* and *FastNet* increase and decrease, respectively, the network hop latency by two cycles. *SlowNet+L2* additionally makes the L2 cache slower. Finally, *SlowBMEM* makes the BM two cycles slower.

Figure 7.13 shows the geometric mean speedups of Baseline+, WiSyncNoT, and WiSync over Baseline for the different configurations. The results correspond to 64-core executions. We see that the speedups of WiSync and WiSyncNoT are higher when the on-chip network is slower, and lower when the network is faster. This is because Baseline (and Baseline+) locks and barriers are sensitive to network latency. The impact of the L2 latency is marginal, since all architectures are affected noticeably. Finally, the BM latency barely affects the performance of WiSync and WiSyncNoT, at least for the range considered.

Now consider the *FastNet* configuration as the baseline. To study the impact of the speed of the wireless channel upon the speedups, we evaluate WiSyncNoT with wireless transfer latencies between 1 cycle ( $> 77Gbps$ ) and 100 cycles ( $< 1Gbps$ ). Figure 7.14 shows the results for relevant applications, as well as the average and geometric mean of



Table 7.6: Memory and network configuration variants.

Configuration	L2 RT (cycles)	BM RT (cycles)	Net. Hop Lat. (cycles)
<i>Default</i>	6	2	4
<i>SlowNet</i>	6	2	6
<i>SlowNet+L2</i>	12	2	6
<i>FastNet</i>	6	2	2
<i>SlowBMEM</i>	6	4	4

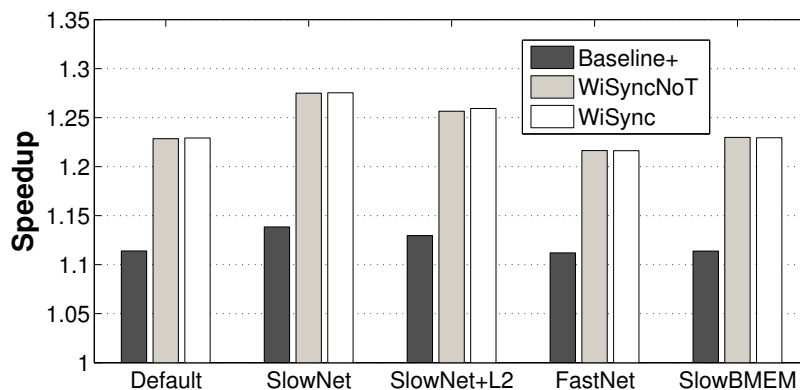


Figure 7.13: Geometric mean of the execution speedup for the different evaluation profiles.

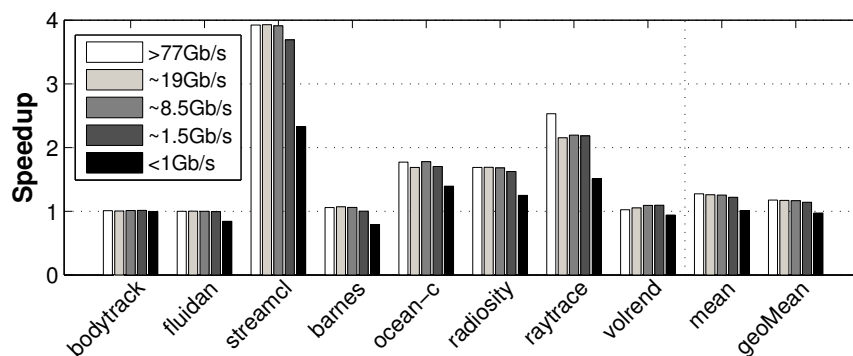


Figure 7.14: Speedup of WiSync for different wireless channel capacities.

all the benchmarks. We observe that speedups are quite consistent up until approximately  $C \approx 2Gbps$ . At this design point, the latency of the wireless plane would double (at best) the latency of the wired plane. Yet still, the speedups are maintained mainly because of the delays introduced by the architecture. In the baseline architecture, misses to those variables imply going through the cache hierarchy in a process that often takes several communication transactions to complete.

## Chapter 8

# Conclusion

The cost of broadcast has been constraining the design of manycore processors and of the algorithms that run upon them. However, advancements in CMOS RF, graphene RF and surface wave technologies could lead a paradigm shift, as native hardware support for low-latency and low-power broadcast could be implemented via wireless communication even in manycore chips. In shared memory, this approach would allow overcoming the so-called coherence wall by reducing complexity and increasing performance. It would also improve support for new programming models and computing systems. In message passing environments, the unprecedented availability of inexpensive broadcast could open the door to a wealth of optimization and reformulation opportunities.

This dissertation has presented the vision of *Broadcast-Oriented Wireless Network-on-Chip* (BoWNoC), which embodies and catalyzes the broadcast potential of the wireless on-chip communication technologies. This is achieved by integrating one antenna and transceiver per processing core and sharing a small set of broadband channels. Throughout the dissertation, we have explained the fundamental design principles and decisions of this novel paradigm and validated its feasibility from the implementation and networking perspectives. On top of this, we have qualitatively and quantitatively demonstrated that the impact of BoWNoC goes beyond simply improving the network performance of a multiprocessor. Instead, BoWNoC becomes a key enabler of unconventional hardware architectures and algorithmic approaches, in the pathway of significantly improving the performance, energy efficiency, scalability and programmability of manycore chips.

### 8.1 Lessons Learned

Taking base in the custom protocol stack proposed in Figure 3.3, each chapter of this thesis has delved into one of the design levels. In all cases, analytical and simulation works have provided deeper insight on the feasibility of the BoWNoC paradigm as summarized next.

Chapter 4 has provided an insightful view of the physical layer. The analysis of the unique characteristics of the scenario suggests the use of very simple modulations and a high carrier frequency to limit the area and power of BoWNoC. Through analytical models and a study of the state of the art of transceiver design, we confirmed this fact by observing that the area and power of wireless on-chip communication scale as  $O(N/C)$ . We also learned that while mmWave implementations may be enough in processors with tens of cores, frequencies up in the THz range might be required in the thousand-core scenario. For this prospective case, we have proposed the use of *graphennas* and evaluated their

potential with a new time-domain methodology that bridges nanoscale phenomena with communication performance. We proved that *graphennas* with high chemical potential and moderate carrier mobility can provide similar performance than metallic antennas. Finally, we demonstrated that molecular absorption effects present in the terahertz band do not represent an impairment at the chip scale, confirming that capacities of up to several Terabits per second could be reached in the long term.

Chapter 5 has revolved around the medium access control layer. The development of scalable, fair, and cost-effective MAC mechanisms is essential to maintain the competitive advantage provided by the inherent broadcast capabilities of the wireless technology. Existing WNoC works mostly employ channelization schemes that do not scale either because they ignore the unrealistic requirements of this approach at the manycore scale, or because scalability is not among their objectives. To address this, we first provided a much-needed context analysis that highlights the static landscape, the monolithic nature of the system, and the latency sensitivity as key features that will drive the design of MAC protocols for BoWNoC. Based on this analysis, we proposed a protocol based on carrier sensing with transmission jamming and implicit acknowledgment. Simulation results reveal severalfold latency improvements with respect to aggressive wireline designs as well as competitive throughput levels. We discovered that the performance of this scheme in the presence of temporal burstiness is a particular weakness to be addressed in future work. We have qualitatively studied some ways to make this possible.

Chapter 6 integrates a BoWNoC within OrthoNoC, a dual-plane hybrid architecture. This scheme differs from other wired-wireless architectures in that both network planes are decoupled and controlled by the NIF, which can implement adaptive traffic steering policies. With a broadcast policy and a realistic wireless capacity of half flit per cycle, OrthoNoC cuts latency up to 4X and increases throughput up to 41% with respect to an aggressive baseline. Using the long-range policy, OrthoNoC can provide up to 3X latency reduction and up to 25% throughput increase in favorable conditions.

Finally, Chapter 7 analyzed the impact that BoWNoC may have on the manycore architecture field. Algorithms, coherence mechanisms, synchronization systems, and programming models may see its performance increased and its complexity reduced by virtue of the scalable broadcast capabilities of BoWNoC. To quantify the potential improvements of the approach, we presented and evaluated WiSync an architecture that employs wireless on-chip communication to implement fast synchronization primitives. After detailing the main design decisions and implementation issues, we showed that WiSync outperforms a baseline architecture by one order of magnitude in terms of CAS throughput and barrier latency in 64-core and 128-core systems. Due to this superior performance, 64-threaded SPLASH-2 and PARSEC applications are sped up a 12% in average despite the reduced amount of fine-grained synchronization present in such benchmarks.

## 8.2 Future Avenues of Research

This dissertation only covers the initial steps of development of the BoWNoC paradigm, representing a point from which the vision can be materialized through different lines of research. As we aimed to capture in Figure 1.7, there are aspects at all levels of design that will need to be explored in future investigations, possibly leading towards a significant breakthrough in the field of computer architecture. Here, we detail some potential research avenues that could address the outstanding challenges of BoWNoC.

Challenges start at the the frontier between the technology and the physical layer of design. The actual development of transceivers and antennas capable of meeting the stringent area, power, and data rate requirements of the scenario remains as a key research objective. This requires great efforts mainly at the analog side, notably in the the design of compact and wideband antennas. Antennas could also be tunable with the help of graphene technology, opening a new degree of freedom in the upper layers of the stack that needs to be explored. Another key point is the design of fast and efficient variable gain amplifiers capable of achieving the readiness, flexibility, and energy efficiency required by the application. This concept would lead to transceivers that do not need to be fully active while the associated core does not have anything to transmit, or that save energy by transmitting only with the necessary power to meet a time-varying BER demand. To deliver the data rates required by manycore architectures, future work will also need to follow the relentless path of technology scaling or further inspect alternative technologies beyond what has been delivered in this dissertation. Last but not least, finding the optimal methodology to integrate all these elements within the manycore environment while minimizing interferences and undesired effects is also of crucial importance.

At the physical layer, there is a need for channel models that unify and improve existing results regarding propagation mechanisms, interferences from surrounding elements, and multipath. This is an essential tool to optimize the antenna and transceiver design processes in such a static environment, as well as to evaluate modulations with increased accuracy. IR techniques need to be carefully inspected as they may reduce the complexity of the transceiver at high frequencies. Coding-wise, the exploration correcting schemes with variable coding gain as an alternative to variable-gain amplifiers could be an interesting approach to meet the power and error resiliency demands of the architecture.

At the link layer of design, this dissertation attempts to open the door to new investigations through the context analysis presented in Section 5.2. The main objective is to develop more efficient and streamlined MAC protocols by taking advantage of the unique optimization opportunities offered by the scenario. Adaptivity and co-design with the architecture will arguably be of crucial importance to achieve unprecedented performance even at massive scales of integration.

At the network and system levels, research will continue towards the optimal integration of the wireless network over or within the wired network. Intelligent traffic steering with fine-grain reconfigurability is key to balance the load and improve performance even in the presence of bursty or hotspot traffic patterns. Other avenues of research include the exploration of network coding techniques to push the system throughput, the integration of the hybrid network within MPI collectives to enhance their performance, or the investigation of novel optical-wireless hybrid architectures.

Finally, the BoWNoC paradigm represents a game changer in terms of multiprocessor architecture design and programming. In fact, this dissertation has provided a specific design that leverages the inherent broadcast capabilities of BoWNoC to implement fast synchronization. However, this is the tip of the proverbial iceberg defined by the architectural design space that a cheap and fast broadcast could open. Broadcast-assisted coherence and enhanced message passing libraries would be first steps towards this set of new possibilities. From a programming perspective, new ways to identify the optimization opportunities stemming from the improved broadcast support are needed. In this respect, the review and refinement of profiling techniques could allow researchers to define new situations of interest and capture them in parallel applications on a phased basis.

## Appendix A

# Synthetic Traffic Generation

Network performance evaluations shown in Chapters 5 and 6 are done with PhoenixSim [235], a cycle-accurate NoC simulator based on Omnet++. Although its primary goal is to provide a simulation framework for nanophotonic NoCs, it also includes a complete set of modules to evaluate conventional NoCs as per *1)* comparison purposes *2)* the implementation of a support plane for the nanophotonic NoC. On top of this, we have implemented:

- Improved router models to support virtual cut-through routing, as well as the multicast support options explored in Chapter 5.
- The necessary modules for the simulation of wireless on-chip communication, including the MAC protocols studied in Chapter 5.
- The hybrid network controller policies studied in Chapter 6.
- The traffic generator used in all simulations. We are capable of controlling the temporal burstiness, the spatial concentration, the multicast intensity, or the hop distribution attributes of traffic.

Next, we provide some details on the implementation of the traffic generator within the simulation framework.

### From Characterization to Generation

Traffic characterization efforts have been shown in different parts of this dissertation, either to motivate it or to analyze the application context for MAC design purposes. Such work can be used to create models that faithfully capture the characteristics of unicast and multicast traffic in shared-memory multiprocessors [261]. With them, NoCs can be evaluated in realistic conditions without having to resort to lengthy traces or full-system simulations, as we did in most parts of this thesis.

Pseudocode in Box 1 below outlines the traffic generator used throughout this dissertation. The generator is inspired by works that use the Hurst exponent and a hotspot factor to model the spatiotemporal distribution of on-chip traffic [233]. Those works make no distinction between unicast and multicast flows, which is only acceptable in primitive NoCs with *unicast-based* multicast (see Chapter 1 for more details). Instead, our generator maintains these parameters and then leverages the knowledge on the destination set of multicasts to determine the number of destinations and their location. Still, our algorithm boils down to a unicast traffic generator if the number of destinations is set to 1. Mixed profiles,

```

input :  $\lambda$  (load; flits/cycle),  $H$  (Hurst exponent),  $\sigma$  (hotspot factor),  $D$ 
        (destinations),  $Hop$  (hop distribution),  $L$  (length distribution),  $N$  (number
        of cores), totalMsg

Calculate spatial distribution with  $\sigma$ ;
Calculate distribution of destinations with  $D$ ;
Calculate packet length with  $L$ ;
while numMsg < totalMsg do
    Identify source  $src$  using the  $\sigma$  distribution;
    Identify number of destinations  $N_{dests} < N$  using the  $D$  distribution;
    if  $N_{dests} == 1$  then
        Calculate destination using  $src$  and the  $Hop$  distribution;
    else
        for  $N_{dests}$  do
            Randomly select one destination;
        end
    end
    if  $burst > 0$  then
         $burst \leftarrow burst - 1$ ;
         $\tau \leftarrow T_{CLK}/N$ ;
    else
         $a_{ON} \leftarrow 3 - 2H$ ;  $b_{ON} \leftarrow 1$ ;
         $burst \leftarrow \text{pareto\_dist}(a_{ON}, b_{ON})$ ;
         $a_{OFF} \leftarrow a_{ON}$ ;  $b_{OFF} \leftarrow T_{CLK}b_{ON}(\lambda^{-1} - 1)$ ;
         $\tau \leftarrow \text{pareto\_dist}(a_{OFF}, b_{OFF})$ ;
    end
    Wait  $\tau$ ;
    Send message through  $src$ ;
end

```

**Algorithm 1:** Proposed multicast traffic generator.

required for the simulations performed in Chapters 5 and 6, can be created by having two independent generators with their own parameters.

We conceive the multicast traffic generator as a central module virtually connected to the NIF of each tile. This module calculates which tile should be sending each multicast message, to which destinations, and with which delay. The message is passed to the source NIF, which treats it according to the unicast or multicast communication policies and injects it into the NoC.

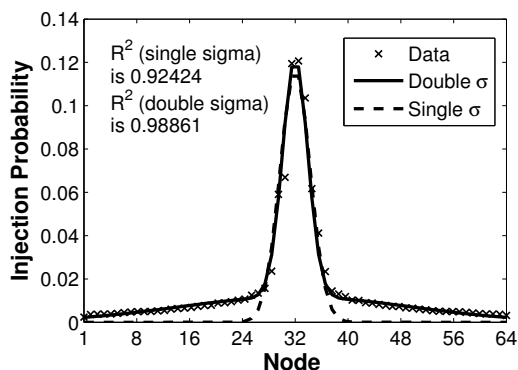
In the following, we provide further details on the specificities of the multicast traffic generator along with proof of its validity.

### Source

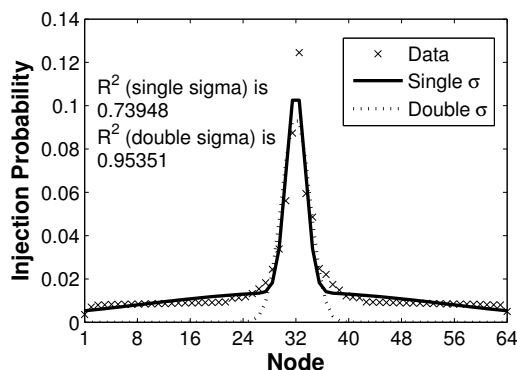
The work in [233] revealed that a gaussian standard deviation  $\sigma \in [0, \infty)$  may be enough to model the spatial distribution of the injection process in NoCs. A large value of  $\sigma$  represents an rather flat, uniform distribution among all cores; whereas a small value indicates that injection of traffic follows a hotspot distribution. To obtain the  $\sigma$  parameter, fitting methods are applied to the vector of injected multicasts per tile.

Coherence	Single- $\sigma$	Double- $\sigma$
MESI	0.9556	0.9841
HT	0.8039	0.9513
TokenB	0.7551	0.9799

(a) Geometric mean of all SPLASH-2 and PARSEC benchmarks



(b) MESI/*swaptions*



(c) TokenB/*fft*

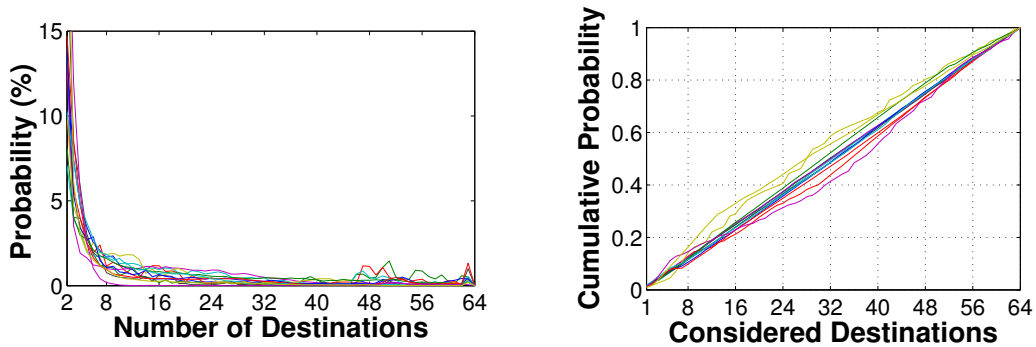
Figure A.1: Coefficient of determination ( $R^2$ ) of the spatial injection distribution, including the gaussian fitting in two relevant cases.

Given that multicast is a subset of all the on-chip traffic, its injection will likely follow a similar pattern. However, our simulations have revealed that using a normal distribution with a single  $\sigma$  may yield inaccurate results in some applications. As illustrated in Figure A.1(b) and A.1(c), some cases would benefit from a double- $\sigma$  fitting. To evaluate the confidence of both methods, we show their coefficient of determination  $R^2 \leq 1$  averaged over all the benchmarks and for the different coherence methods in Figure A.1(a). These results suggest that double- $\sigma$  distributions may be more appropriate to model the spatial injection distribution of multicast traffic, at the cost of slightly higher complexity.

## Destinations

The choice of the destination depends on the type of packet we are dealing with. Synthetic traffic models generally account for unicast traffic only and, in those cases, a variety of hop distributions can be used. The well-known neighbour, uniform random, transpose, or bit complement patterns select the destination as a function of the source with very simple rules. We refer the reader to [242] for more information in this regard. A more flexible and arguably realistic model is that of Rent's law, which generates traffic with a degree of locality determined by the Rent exponent. We implemented the methodology described in [243] for the generation of traffic following the Rent's law. Figure 6.5, in Chapter 6, plots the average Manhattan distance of the considered patterns.

For multicast traffic, the destination set needs to be chosen and existing models cannot be used to this end. This includes both the number of destinations per message and the destinations themselves. One possibility is to model the number of destinations and then to use existing approaches to independently choose the destinations of each message. More complicated schemes could try to correlate both aspects in order to faithfully characterize



(a) Probability distribution function of the number of destinations per multicast. (b) Cumulative distribution function of the multicast destinations.

Figure A.2: Statistical analysis of the multicast destinations in MESI (perfect tracking).

certain multicast flows that go from a small set of sources to a deterministic set of destinations. Due to its simplicity, we choose the first option to model multicast traffic and use MESI traces to confirm its effectiveness.

To model the number of destinations per message, a trivial approach would be to compute the average number of destinations for a given application. More accurate models can be obtained using fitting methods to the histogram of destinations. As shown in Figure A.2(a), the distribution of the number of destinations of most applications would be accurately modeled with a power function or a rational function. The coefficient of determination  $R^2$  is of 0.9774 and 0.9668 in average, respectively. Note that it may be worth to consider broadcast messages separately.

To model the destinations of each message, we consider the destination set to be independent of the source for simplicity. We collected the number of received multicast messages per NIF and performed initial modeling tests. Unlike the injection process, the spatial distribution of the received multicasts does not fit well with a gaussian distribution and exhibits a more uniform behavior instead. Figure A.2(b) plots the cumulative distribution function of the received multicasts, confirming that the destinations may be modeled using a uniform random variable. A linear fitting further validates this fact by achieving a coefficient of determination of 0.9964 in average.

## Delay

The modeling of self-similar traffic has been the subject of different studies in all areas of networking [262], including on-chip communication [233]. In light of the results shown in Chapter 5 regarding the temporal distribution of the injection of multicasts, knowledge obtained thus far can be used to model the burstiness of multicast traffic.

The most widespread method to generate self-similar traffic is through the alternate generation of ON and OFF periods using the Hurst exponent  $H \in [0.5, 1)$ . Burstiness varies from nonexistent ( $H = 0.5$ , exponential) to extremely bursty ( $H \rightarrow 1$ ). During the ON periods, the generator outputs at most  $N$  messages per clock cycle; whereas it remains silent the rest of the time. The length of both periods follow a Pareto distribution [262], which is a heavy tailed distribution with a probability density function



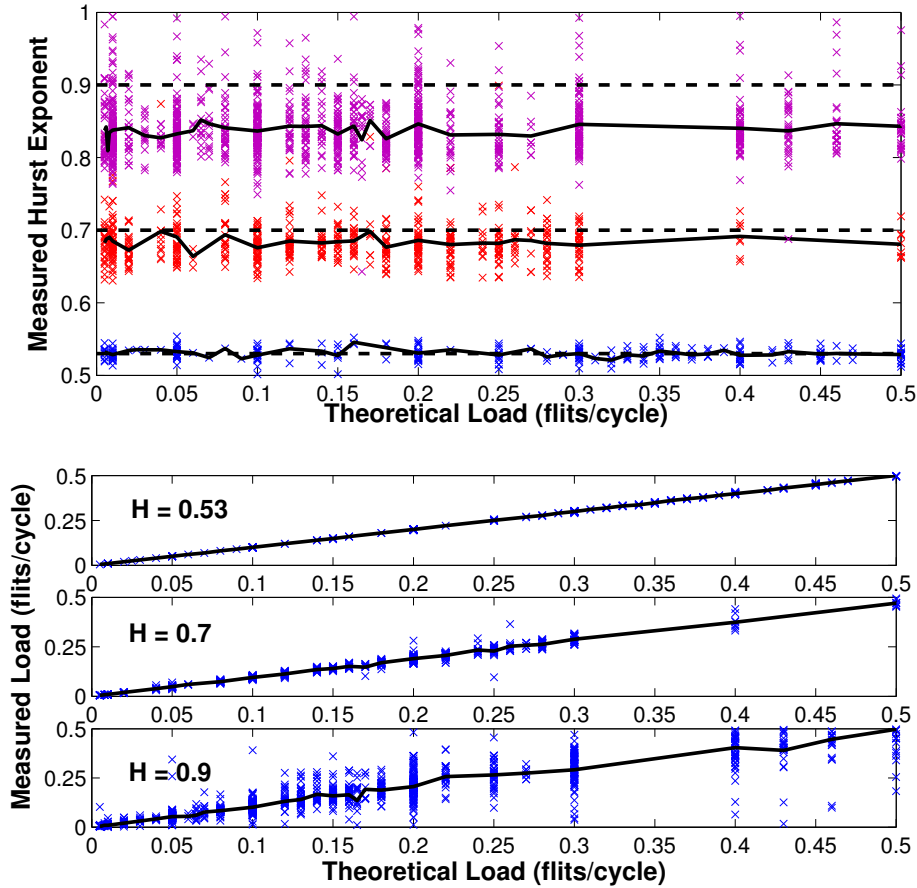


Figure A.3: Measured Hurst exponent (top) and load (bottom) as functions of the input load for  $H = \{0.53, 0.7, 0.9\}$ . Dashed and solid lines represent the theoretical value and geometric mean of the measured values, respectively.

$$f(x) = \frac{a \cdot b^a}{x^{a+1}} \quad x \geq b. \quad (\text{A.0.1})$$

The shape parameter  $a$  of the Pareto distribution is related with the Hurst exponent as

$$a_{ON} = a_{OFF} = 3 - 2H, \quad (\text{A.0.2})$$

whereas the location parameter  $b$  needs to be set at the minimum value of the distribution. In NoC environments, one can take  $b_{ON}$  as the equivalent to a burst of a single multicast, whereas  $b_{OFF}$  is scaled in order to fix the load to the desired  $\lambda$  value, as:

$$b_{OFF} = b_{ON} \left( \frac{1}{\lambda} - 1 \right). \quad (\text{A.0.3})$$

Using this method, we successfully created synthetic traffic with the desired  $H$  and  $\lambda$  characteristics. To prove the validity of the approach, we generated streams of bursty traffic of 100K messages, with  $H = \{0.53, 0.7, 0.9\}$  and different loads between 0 and 0.5 flits per cycle. The traces containing the timestamps of each generated message are then analyzed to obtain the real load and Hurst exponent, and then averaged over a variable number of repetitions. Figure A.3 shows the measured Hurst exponents and loads as functions of

the three analyzed burstiness levels. In both figures, it is observed that results become more random as the input Hurst exponent increases. Still, both the measured load and the measured Hurst exponent increase, in average terms, proportionally to the input load and Hurst exponent, respectively. Finally, it is important to note that the average error of the measured Hurst exponent increases with the input exponent. Therefore, corrective factors need to be applied as  $H$  values approach 1 and, therefore, ON and OFF periods become large. Increasing the length of the simulations also helps to reduce this error.

## Appendix B

# Scientific Production

The contributions of this thesis have been adapted and published both in journals and conferences. Part of the work has been performed within the project “Graphene-enabled Wireless Communications”, funded by SAMSUNG under the Global Research Outreach (GRO) program. The publications and awards related to the work of this thesis are as follows.

Awards:

- **INTEL Fellowship - Doctoral Student Honor Program:** which recognizes outstanding research done by PhD students and young researchers in the INTEL’s areas of interest. Fall 2013.
- **ASPLOS ’16 Travel Grant:** awarded to selected students to attend the 21st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Atlanta, April 2016.

Book chapters:

- S. Abadal, I. Llatser, A. Mestres, J. Solé-Pareta, E. Alarcón, A. Cabellos-Aparicio, “*Fundamentals of Graphene-enabled Wireless On-Chip Networking*”, to appear in **Modeling, Methodologies and Tools for Molecular and Nano-scale Communications**, Springer, 2014.

Journal Papers:

- S. Abadal, A. Mestres, E. Alarcón, M. Nemirovsky, A. González, H. Lee and A. Cabellos-Aparicio, “*Scalability of Broadcast Performance in Wireless Network-on-Chip*,” to appear in **IEEE Transactions on Parallel and Distributed Systems**, 2016.
- S. Abadal, R. Martínez, J. Solé-Pareta, E. Alarcón and A. Cabellos-Aparicio, “*Characterization and Modeling of Multicast Communication in Cache-Coherent Manycore Processors*,” to appear in **Computers and Electrical Engineering (Elsevier)**, 2016.
- 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, Oct. 2015.

- S. Abadal, M. Iannazzo, M. Nemirovsky, A. Cabellos-Aparicio, H. Lee, E. Alarcón, “*On the Area and Energy Scalability of Wireless Network-on-Chip: A Model-based Benchmarked Design Space Exploration*,” **IEEE/ACM Transactions on Networking**, vol. 23, no. 5, pp. 1501-1513, Oct. 2015.
- S. Abadal, I. Llatser, A. Mestres, H. Lee, E. Alarcón and A. Cabellos-Aparicio, “*Time-Domain Analysis of Graphene-based Miniaturized Antennas for Ultra-short-range Impulse Radio Communications*,” **IEEE Transactions on Communications**, vol. 63, no. 4, pp. 1470-1482, Apr. 2015.
- I. Llatser, A. Mestres, S. Abadal, E. Alarcón, H. Lee and A. Cabellos-Aparicio, “*Time and Frequency Domain Analysis of Molecular Absorption in Short-range Terahertz Communications*,” **IEEE Antennas and Wireless Propagation Letters**, vol. 14, pp. 350-353, Feb. 2015.
- S. Abadal, E. Alarcón, M. C. Lemme, M. Nemirovsky and A. Cabellos-Aparicio, “*Graphene-enabled Wireless Communication for Massive Multicore Architectures*,” **IEEE Communications Magazine**, vol. 51, no. 11, pp. 137-143, Nov. 2013.

Conference Papers:

- 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 21st International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**), pp. 3-17, Apr. 2016.
- S. Abadal, M. Nemirovsky, E. Alarcón, and A. Cabellos-Aparicio, “*Networking Challenges and Prospective Impact of Broadcast-Oriented Wireless Networks-on-Chip*,” in Proceedings of the 9th IEEE/ACM International Symposium on Networks-on-Chip (**NOCS**), Sept. 2015.
- S. Abadal, A. Mestres, I. Llatser, E. Alarcón, and A. Cabellos-Aparicio, “*A Vertical Methodology for the Design Space Exploration of Graphene-enabled Wireless Communications*,” in Proceedings of the 2nd ACM International Conference on Nanoscale Computing and Communication (**NANOCOM**), Sept. 2015.
- S. Abadal, A. Mestres, R. Martínez, E. Alarcón, and A. Cabellos-Aparicio, “*Multicast On-Chip Traffic Analysis Targeting Manycore NoC Design*,” in Proceedings of the 23rd Euromicro International Conference on Parallel, Distributed and Network-based Processing (**PDP**), pp. 370-378, Mar. 2015.
- S. Abadal, A. Mestres, M. Iannazzo, J. Solé-Pareta, E. Alarcón, and A. Cabellos-Aparicio, “*Evaluating the Feasibility of Wireless Networks-on-Chip Enabled by Graphene*,” in Proceedings of the 7th International Workshop on Network-on-Chip Architectures (**NoCArc**), held in conjunction with the 47th annual IEEE/ACM International Symposium on Microarchitecture (MICRO-47), pp. 51-56, Dec. 2014.
- S. Abadal, R. Martínez, E. Alarcón and A. Cabellos-Aparicio “*Scalability-Oriented Multicast Traffic Characterization*,” in Proceedings of the 8th IEEE/ACM International Symposium on Networks-on-Chip (**NOCS**), pp. 180-181, Sept. 2014.
- G. Piro, S. Abadal, A. Mestres, E. Alarcón, J. Solé-Pareta, L. A. Grieco, G. Boggia, “*Initial MAC Exploration for Graphene-enabled Wireless Networks-on-Chip*,” in Proceedings of the 1st ACM International Conference on Nanoscale Computing and Communication (**NANOCOM**), May 2014

- I. Llatser, S. Abadal, A. Mestres, A. Cabellos-Aparicio and E. Alarcón, “*Graphene-enabled Wireless Networks-on-Chip*,” in Proceedings of the First International Black Sea Conference on Communications and Networking (**BlackSeaCom**), pp. 6973, Jul. 2013.
- S. Abadal, A. Cabellos-Aparicio, J. A. Lázaro, M. Nemirovsky, E. Alarcón and J. Solé-Pareta, “*Area and Laser Power Scalability Analysis in Photonic Networks-on-Chip*”, in Proceedings of the 17th International Conference in Optical Networks Design and Modeling (**ONDM**), pp. 131-136, Apr. 2013.
- S. Abadal, A. Cabellos-Aparicio, J. A. Lázaro, E. Alarcón and J. Solé-Pareta, “*Scalable NoC Architectures: Efficient and Low Energy Consumption Chip Communication*,” in Proceedings of the Photonics in Switching (**PS**), Sept. 2012.
- S. Abadal, A. Cabellos-Aparicio, J. A. Lázaro, E. Alarcón and J. Solé-Pareta, “*Graphene-enabled hybrid architectures for multiprocessors: bridging nanophotonics and nanoscale wireless communication*,” in Proceedings of the 14th International Conference in Transparent Optical Networks (**ICTON**), Jul. 2012.



# Bibliography

- [1] J. Hennessy and D. Patterson, *Computer architecture: a quantitative approach*. Morgan Kaufmann, 2012.
- [2] D. Culler, J. P. Singh, and A. Gupta, *Parallel computer architecture: a hardware/software approach*. Morgan Kaufman, 1999.
- [3] R. Kumar, T. G. Mattson, G. Pokam, and R. Van Der Wijngaart, “The case for message passing on many-core chips,” in *Multiprocessor System-on-Chip: Hardware Design and Tool Integration*, 2011, pp. 115–123.
- [4] L. Benini and G. De Micheli, “Networks on chips: a new SoC paradigm,” *IEEE Computer*, vol. 35, no. 1, pp. 70–78, 2002.
- [5] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. Brown III, and A. Agarwal, “On-chip interconnection architecture of the tile processor,” *IEEE Micro*, vol. 27, no. 5, pp. 15–31, 2007.
- [6] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, “An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, 2008.
- [7] T. Bjerregaard and S. Mahadevan, “A survey of research and practices of Network-on-chip,” *ACM Computing Surveys*, vol. 38, no. 1, pp. 1–51, 2006.
- [8] M. M. K. Martin, P. J. Harper, D. J. Sorin, M. D. Hill, and D. A. Wood, “Using destination-set prediction to improve the latency/bandwidth tradeoff in shared-memory multiprocessors,” in *Proceedings of the ISCA-30*, 2003, pp. 206–217.
- [9] S. Borkar, “Thousand Core Chips - A Technology Perspective,” in *Proceedings of the DAC-44*, 2007, pp. 746–749.
- [10] S. Woo, M. Ohara, E. Torrie, and J. Singh, “The SPLASH-2 programs: Characterization and methodological considerations,” *ACM SIGARCH Computer Architecture News*, vol. 23, no. 2, pp. 24–36, 1995.
- [11] C. Bienia, S. Kumar, J. Singh, and K. Li, “The PARSEC benchmark suite: characterization and architectural implications,” in *Proceedings of the PACT '08*, 2008, pp. 72–81.
- [12] 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.
- [13] W. Huang, K. Rajamani, M. Stan, and K. Skadron, “Scaling with design constraints: Predicting the future of big chips,” *IEEE Micro*, vol. 31, no. 4, pp. 16–29, 2011.

- [14] M. Kim, J. Davis, M. Oskin, and T. Austin, “Polymorphic on-chip networks,” in *Proceedings of the ISCA-35*, 2008, pp. 101–112.
- [15] P. Conway and B. Hughes, “The AMD Opteron Northbridge Architecture,” *IEEE Micro*, vol. 27, no. 2, pp. 10–21, 2007.
- [16] M. Martin, “Token Coherence: decoupling performance and correctness,” in *Proceedings of the ISCA-30*, 2003, pp. 182–193.
- [17] M. Daneshtalab, M. Ebrahimi, T. C. Xu, P. Liljeberg, and H. Tenhunen, “A generic adaptive path-based routing method for MPSoCs,” *Journal of Systems Architecture*, vol. 57, no. 1, pp. 109–120, 2011.
- [18] R. Boppana, S. Chalasani, and C. Raghavendra, “Resource deadlocks and performance of wormhole multicast routing algorithms,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 6, pp. 535–549, 1998.
- [19] N. Enright Jerger, L.-S. Peh, and M. Lipasti, “Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support,” in *Proceedings of the ISCA-35*, 2008, pp. 229–240.
- [20] T. Krishna, L.-S. Peh, B. Beckmann, and S. K. Reinhardt, “Towards the ideal on-chip fabric for 1-to-many and many-to-1 communication,” in *Proceedings of the MICRO-44*, 2011, pp. 71–82.
- [21] S. Park, T. Krishna, C.-H. Chen, B. Daya, A. Chandrakasan, and L.-S. Peh, “Approaching the theoretical limits of a mesh NoC with a 16-node chip prototype in 45nm SOI,” in *Proceedings of the DAC-49*, 2012, pp. 398–405.
- [22] C.-K. Liang and M. Prvulovic, “MiSAR: Minimalistic Synchronization Accelerator with Resource Overflow Management,” in *Proceedings of the ISCA-42*, 2015, pp. 414–26.
- [23] A. Agarwal, R. Simoni, J. Hennessy, and M. Horowitz, “An evaluation of directory schemes for cache coherence,” in *Proceedings of the ISCA-15*, 1988, pp. 280–289.
- [24] K. Strauss, X. Shen, and J. Torrellas, “Uncorq: Unconstrained snoop request delivery in embedded-ring multiprocessors,” in *Proceedings of MICRO-40*, 2007, pp. 327 – 342.
- [25] B. Daya, C.-H. O. Chen, S. Subramanian, W.-C. Kwon, S. Park, T. Krishna, J. Holt, A. P. Chandrakasan, and L.-S. Peh, “SCORPIO: a 36-core research chip demonstrating snoopy coherence on a scalable mesh NoC with in-network ordering,” in *Proceedings of the ISCA-41*, 2014, pp. 25–36.
- [26] 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.
- [27] E. Socher and M.-C. F. Chang, “Can RF Help CMOS Processors?” *IEEE Communications Magazine*, vol. 45, no. 8, pp. 104–111, 2007.
- [28] R. G. Beausoleil, P. J. Kuekes, G. S. Snider, S.-Y. Wang, and R. S. Williams, “Nanoelectronic and Nanophotonic Interconnect,” *Proceedings of the IEEE*, vol. 96, no. 2, pp. 230–247, 2008.
- [29] J. Kim and K. Choi, “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.



- [30] D. Schinkel and E. Mensink, “Low-power, high-speed transceivers for network-on-chip communication,” *IEEE Transactions on VLSI Systems*, vol. 17, no. 1, pp. 12–21, 2009.
- [31] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, “Kilo-NOC: A Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees,” in *Proceedings of ISCA-38*, 2011, pp. 401–412.
- [32] T. Krishna, C. Chen, W. Kwon, and L.-S. Peh, “Smart: Single-Cycle Multihop Traversals over a Shared Network on Chip,” *IEEE Micro*, vol. 34, no. 3, pp. 43–56, 2014.
- [33] L.-S. Peh and W. Dally, “A delay model and speculative architecture for pipelined routers,” in *Proceedings of the HPCA-7*, 2001, pp. 255–266.
- [34] R. Mullins, A. West, and S. Moore, “The design and implementation of a low-latency on-chip network,” in *Proceedings of the ASPDAC '06*, 2006, pp. 164–169.
- [35] H. Matsutani, M. Koibuchi, H. Amano, and T. Yoshinaga, “Prediction router: Yet another low latency on-chip router architecture,” in *Proceedings of the HPCA-15*, 2009, pp. 367–378.
- [36] J. Kim, J. Balfour, and W. J. Dally, “Flattened butterfly topology for on-chip networks,” in *Proceedings of the MICRO-40*, 2007, pp. 172–182.
- [37] D. Sánchez, G. Micheliogiannakis, and C. Kozyrakis, “An Analysis of On-Chip Interconnection Networks for Large-Scale Chip Multiprocessors,” *ACM Transactions on Architecture and Code Optimization*, vol. 7, no. 1, p. Art. 4, 2010.
- [38] U. Ogras and R. Marculescu, “‘It’s a small world after all’: NoC performance optimization via long-range link insertion,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 7, pp. 693–706, 2006.
- [39] N. Abeyratne, R. Das, Q. Li, K. Sewell, B. Giridhar, R. G. Dreslinski, D. Blaauw, and T. Mudge, “Scaling towards kilo-core processors with asymmetric high-radix topologies,” in *Proceedings of the HPCA-19*, 2013, pp. 496–507.
- [40] P. Abad, V. Puente, and J.-A. Gregorio, “MRR: Enabling fully adaptive multicast routing for CMP interconnection networks,” in *Proceedings of HPCA-15*, 2009, pp. 355–66.
- [41] F. A. Samman, T. Hollstein, and M. Glesner, “Multicast parallel pipeline router architecture for network-on-chip,” in *Proceedings of DATE '08*, 2008, pp. 1396–1401.
- [42] L. Wang, Y. Jin, H. Kim, and E. Kim, “Recursive partitioning multicast: A bandwidth-efficient routing for Networks-on-Chip,” in *Proceedings of the NoCS '09*, 2009, pp. 64–73.
- [43] S. Ma, N. Enright Jerger, and Z. Wang, “Supporting efficient collective communication in NoCs,” in *Proceedings of the HPCA-18*, 2012, pp. 165–176.
- [44] T. Krishna and L.-S. Peh, “Single-Cycle Collective Communication Over A Shared Network Fabric,” in *Proceedings of the NoCS '14*, 2014, pp. 1–8.
- [45] J. Burns, L. McIlrath, C. Keast, C. Lewis, A. Loomis, K. Warner, and P. Wyatt, “Three-dimensional integrated circuits for low-power, high-bandwidth systems on a chip,” in *IEEE ISSCC Dig. Tech. Papers*, 2001, pp. 268–269.
- [46] A. W. Topol, D. C. La Tulipe, L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini, and M. Jeong, “Three-dimensional

- integrated circuits,” *IBM Journal of Research and Development*, vol. 50, no. 4, pp. 491–506, 2006.
- [47] V. S. Nandakumar and M. Marek-Sadowska, “A Low Energy Network-on-Chip Fabric for 3-D Multi-Core Architectures,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 266–277, 2012.
- [48] Q. Gu, Z. Xu, J. Ko, and M.-C. F. Chang, “Two 10Gb/s/pin Low-Power Interconnect Methods for 3D ICs,” in *Solid-State Circuits Conference*, vol. 53, 2007, pp. 448–449.
- [49] N. Miura, Y. Kohama, Y. Sugimori, H. Ishikuro, T. Sakurai, and T. Kuroda, “A High-Speed Inductive-Coupling Link With Burst Transmission,” *IEEE Journal of Solid-State Circuits*, vol. 44, no. 3, pp. 947–955, 2009.
- [50] V. F. Pavlidis and E. G. Friedman, “3-D Topologies for Networks-on-Chip,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 10, pp. 1081–1090, 2007.
- [51] B. S. Feero and P. P. Pande, “Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation,” *IEEE Transactions on Computers*, vol. 58, no. 1, pp. 32–45, 2009.
- [52] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, “SunFloor 3D: A Tool for Networks on Chip Topology Synthesis for 3-D Systems on Chips,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 12, pp. 1987–2000, 2010.
- [53] C. H. Chao, K. Y. Jheng, H. Y. Wang, J. C. Wu, and A. Y. Wu, “Traffic- and thermal-aware run-time thermal management scheme for 3D NoC systems,” in *Proceedings of the NoCs ’10*, 2010, pp. 223–230.
- [54] X. Wang, M. Yang, Y. Jiang, M. Palesi, P. Liu, T. Mak, and N. Bagherzadeh, “Efficient multicast schemes for 3-D Networks-on-Chip,” *Journal of Systems Architecture*, vol. 59, no. 9, pp. 693–708, 2013.
- [55] M. Ebrahimi, M. Daneshalab, P. Liljeberg, J. Plosila, J. Flich, and H. Tenhunen, “Path-Based Partitioning Methods for 3D Networks-on-Chip with Minimal Adaptive Routing,” *IEEE Transactions on Computers*, vol. 63, no. 3, pp. 718–733, 2014.
- [56] M.-C. F. Chang, V. Roychowdhury, L. Zhang, H. Shin, and Y. Qian, “RF/wireless interconnect for inter- and intra-chip communications,” *Proceedings of the IEEE*, vol. 89, no. 4, pp. 456–466, 2001.
- [57] J. Oh, M. Prvulovic, and A. Zajic, “TLSync: support for multiple fast barriers using on-chip transmission lines,” in *Proceedings of ISCA-38*, 2011, pp. 105–115.
- [58] B. Beckmann and D. Wood, “TLC: transmission line caches,” in *Proceedings of the MICRO-36*, 2003, pp. 43–54.
- [59] J. L. Abellán, J. Fernández, and M. E. Acacio, “GLocks: Efficient support for highly-contended locks in many-core CMPs,” in *Proceedings of the IPDPS ’11*, 2011, pp. 893–905.
- [60] —, “Efficient Hardware Barrier Synchronization in Many-Core CMPs,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1453–1466, 2012.
- [61] M. F. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S.-W. Tam, “CMP Network-on-Chip Overlaid With Multi-Band RF-Interconnect,” in *Proceedings of the HPCA-14*, 2008, pp. 191–202.

- [62] J. Oh, A. Zajic, and M. Prvulovic, “Traffic steering between a low-latency unswitched TL ring and a high-throughput switched on-chip interconnect,” in *Proceedings of the PACT '13*, 2013, pp. 309–318.
- [63] A. Carpenter, J. Hu, J. Xu, M. Huang, and H. Wu, “A case for globally shared-medium on-chip interconnect,” in *Proceedings of the ISCA-38*, 2011, pp. 271–282.
- [64] A. Carpenter, J. Hu, O. Kocabas, M. Huang, and H. Wu, “Enhancing effective throughput for transmission line-based bus,” in *Proceedings of the ISCA-39*, 2012, pp. 165–176.
- [65] A. Brière, E. Unlu, J. Denoulet, A. Pinna, B. Granado, F. Pêcheux, Y. Louët, and C. Moy, “A Dynamically Reconfigurable RF NoC for Many-Core,” in *Proceedings of the GLSVLSI '15*, 2015, pp. 139–144.
- [66] C. Sun, M. T. Wade, Y. Lee, J. S. Orcutt, L. Alloatti, M. S. Georgas, A. S. Waterman, J. M. Shainline, R. R. Avizienis, S. Lin, B. R. Moss, R. Kumar, F. Pavanello, A. H. Atabaki, H. M. Cook, A. J. Ou, J. C. Leu, Y.-H. Chen, K. Asanović, R. J. Ram, M. A. Popović, and V. M. Stojanović, “Single-chip microprocessor that communicates directly using light,” *Nature*, vol. 528, no. 7583, pp. 534–538, 2015.
- [67] J. Xue, M. Huang, H. Wu, E. Friedman, G. Wicks, D. Moore, A. Garg, B. Ciftcioglu, J. Hu, S. Wang, I. Savidis, M. Jain, R. Berman, and P. Liu, “An intra-chip free-space optical interconnect,” in *Proceedings of the ISCA-37*, 2010, p. 94.
- [68] A. Novack, Y. Liu, R. Ding, M. Gould, T. Baehr-Jones, Q. Li, Y. Yang, Y. Zhang, K. Padmaraju, K. Bergmen, A. E.-J. Lim, G.-Q. Lo, and M. Hochberg, “A 30 GHz Silicon Photonic Platform,” in *Proc. SPIE - Integrated Optics: Physics and Simulations*, vol. 8781, 2013, pp. 1–4.
- [69] W. Cai, J. White, and M. Brongersma, “Compact, high-speed and power-efficient electrooptic plasmonic modulators,” *Nano Letters*, vol. 9, no. 12, pp. 4403–4411, 2009.
- [70] C. Manolatos, S. Johnson, S. Fan, P. Villeneuve, H. Haus, and J. Joannopoulos, “High-density integrated optics,” *Journal of Lightwave Technology*, vol. 17, no. 9, pp. 1682–1692, 1999.
- [71] J. Cardenas, C. Poitras, and J. Robinson, “Low loss etchless silicon photonic waveguides,” *Optics Express*, vol. 17, no. 6, pp. 4752–7, 2009.
- [72] A. Shacham, K. Bergman, and L. P. Carloni, “Photonic networks-on-chip for future generations of chip multiprocessors,” *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1246–1260, 2008.
- [73] N. Kirman and J. F. Martínez, “A Power-efficient All-optical On-chip Interconnect Using Wavelength-based Oblivious Routing,” *ACM Sigplan Notices*, vol. 45, no. 3, pp. 15–28, 2010.
- [74] S. Sahni, X. Luo, J. Liu, Y.-h. Xie, and E. Yablonovitch, “Junction field-effect-transistor-based germanium photodetector on silicon-on-insulator.” *Optics letters*, vol. 33, no. 10, pp. 1138–40, may 2008.
- [75] C. Chen and A. Joshi, “Runtime Management of Laser Power in Silicon-Photonic Multibus NoC Architecture,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 2, 2013.
- [76] L. Zhou and A. K. Kodi, “PROBE: Prediction-based optical bandwidth scaling for energy-efficient NoCs,” in *Proceedings of the NoCs '13*, 2013, pp. 1–8.

- [77] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi, "Phastlane: a rapid transit optical routing network," *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 441–450, 2009.
- [78] N. Kirman, M. Kirman, R. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins, and D. H. Albonesi, "Leveraging optical technology in future bus-based chip multiprocessors," in *Proceedings of the MICRO-39*, 2006, pp. 492–503.
- [79] G. Kurian, J. Miller, J. Psota, J. Eastep, J. Liu, J. Michel, L. Kimerling, and A. Agarwal, "ATAC: A 1000-Core Cache-Coherent Processor with On-Chip Optical Network," in *Proceedings of the PACT*, 2010, pp. 477–488.
- [80] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. Beausoleil, and J. Ahn, "Corona: System implications of emerging nanophotonic technology," *ACM SIGARCH Computer Architecture News*, vol. 36, no. 3, pp. 153–164, 2008.
- [81] G. Hendry, J. Chan, S. Kamil, L. Oliker, J. Shalf, L. P. Carloni, and K. Bergman, "Silicon nanophotonic network-on-chip using TDM arbitration," in *Proceedings of the HOTI-18*, 2010, pp. 88–95.
- [82] A. Joshi, C. Batten, Y. Kwon, S. Beamer, I. Shamim, K. Asanovic, and V. Stojanovic, "Silicon-photonic cros networks for global on-chip communication," in *Proceedings of the NoCs '09*, 2009, pp. 124–133.
- [83] J. Chan and K. Bergman, "Photonic Interconnection Network Architectures Using Wavelength-Selective Spatial Routing for Chip-Scale Communications," *Journal of Optical Communications and Networking*, vol. 4, no. 3, p. 189, 2012.
- [84] X. Zhang and A. Louri, "A multilayer nanophotonic interconnection network for on-chip many-core communications," in *Proceedings of the DAC-47*, 2010, pp. 156–161.
- [85] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: Illuminating Future Network-on-Chip with Nanophotonics," *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 429–440, 2009.
- [86] M. A. I. Sikder, A. K. Kodi, M. Kennedy, S. Kaya, and A. Louri, "OWN: Optical and Wireless Network-on-Chip for Kilo-core Architectures," in *Proceedings of the HOTI-23*, 2015, pp. 44–51.
- [87] R. Morris, E. Jolley, and A. K. Kodi, "Extending the Performance and Energy-Efficiency of Shared Memory Multicores with Nanophotonic Technology," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 83–92, 2014.
- [88] D. Vantrease, M. H. Lipasti, and N. Binkert, "Atomic Coherence: Leveraging nanophotonics to build race-free cache coherence protocols," in *Proceedings of the HPCA-17*, 2011, pp. 132–143.
- [89] B. A. Floyd, C.-M. Hung, and K. K. O, "Intra-chip wireless interconnect for clock distribution implemented with integrated antennas, receivers, and transmitters," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 5, pp. 543–552, 2002.
- [90] S. Deb, K. Chang, X. Yu, S. P. Sah, M. Cosic, P. P. Pande, B. Belzer, and D. Heo, "Design of an Energy Efficient CMOS Compatible NoC Architecture with Millimeter-Wave Wireless Interconnects," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2382–2396, 2013.

- [91] P. Russer, N. Fichtner, P. Lugli, W. Porod, J. a. Russer, and H. Yordanov, "Nanoelectronics-Based Integrated Antennas," *IEEE Microwave Magazine*, vol. 11, no. 7, pp. 58–71, 2010.
- [92] 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.
- [93] Y. P. Zhang, Z. M. Chen, and M. Sun, "Propagation Mechanisms of Radio Waves Over Intra-Chip Channels With Integrated Antennas: Frequency-Domain Measurements and Time-Domain Analysis," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 10, pp. 2900–2906, 2007.
- [94] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, and C. Teuscher, "Scalable Hybrid Wireless Network-on-Chip Architectures for Multi-Core Systems," *IEEE Transactions on Computers*, vol. 60, no. 10, pp. 1485–1502, 2010.
- [95] C. Wang, W.-H. Hu, and N. Bagherzadeh, "A load-balanced congestion-aware wireless network-on-chip design for multi-core platforms," *Microprocessors and Microsystems*, vol. 36, no. 7, pp. 555–570, 2012.
- [96] S.-B. Lee, S.-W. Tam, I. Pefkianakis, S. Lu, M.-C. F. Chang, C. Guo, G. Reinman, C. Peng, M. Naik, L. Zhang, and J. Cong, "A scalable micro wireless interconnect structure for CMPs," in *Proceedings of the MOBICOM '09*, 2009, p. 217.
- [97] 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.
- [98] D. Hou, Y.-z. Xiong, W. Hong, W. L. Goh, and J. Chen, "Silicon-based On-chip Antenna Design for Millimeter-wave / THz Applications," in *Proceedings of the EDAPS '11*, 2011, pp. 130–133.
- [99] B. Klein, R. Hahnel, D. Plettemeier, T. Morf, M. Despont, U. Drechsler, M. Kuhn, T. Toifl, D. Corcos, N. Kaminski, and D. Elad, "Design of a cloverleaf antenna for an antenna coupled bolometer for room temperature THz imaging," in *Proceedings of the ISCDG '13*, 2013, pp. 1–4.
- [100] T. S. Rappaport, J. N. Murdock, and F. Gutierrez, "State of the Art in 60-GHz Integrated Circuits and Systems for Wireless Communications," *Proceedings of the IEEE*, vol. 99, no. 8, pp. 1390–1436, 2011.
- [101] O. Markish, B. Sheinman, O. Katz, D. Corcos, and D. Elad, "On-chip mmWave Antennas and Transceivers," in *Proceedings of the NoCS '15*, 2015, p. Art. 11.
- [102] X. Yu, J. Baylon, P. Wettin, D. Heo, P. Pande, and S. Mirabbasi, "Architecture and Design of Multi-Channel Millimeter-Wave Wireless Network-on-Chip," *IEEE Design & Test*, vol. 31, no. 6, pp. 19–28, 2014.
- [103] N. Weissman and E. Socher, "9mW 6Gbps Bi-directional 85-90GHz Transceiver in 65nm CMOS," in *Proceedings of the EuMIC '14*, 2014, pp. 25–28.
- [104] J. Gorisse, D. Morche, and J. Jantunen, "Wireless transceivers for gigabit-per-second communications," in *Proceedings of the NEWCAS '12*, 2012, pp. 545–548.
- [105] K.-K. Huang and D. D. Wentzloff, "60 GHz On-Chip Patch Antenna Integrated in a 0.13- $\mu$ m CMOS Technology," in *Proceedings of the ICUWB '10*, 2010, pp. 4–7.

- [106] T. Deng, Z. Chen, and Y. P. Zhang, “Coupling mechanisms and effects between on-chip antenna and inductor or coplanar waveguide,” *IEEE Transactions on Electron Devices*, vol. 60, no. 1, pp. 20–27, 2013.
- [107] E. Laskin, P. Chevalier, B. Sautreuil, and S. Voinigescu, “A 140-GHz double-sideband transceiver with amplitude and frequency modulation operating over a few meters,” in *IEEE BCTM*, 2009, pp. 178–181.
- [108] N. Ono, M. Motoyoshi, K. Takano, K. Katayama, R. Fujimoto, and M. Fujishima, “135 GHz 98 mW 10 Gbps ASK transmitter and receiver chipset in 40 nm CMOS,” in *Proceedings of the VLSIC '12*, 2012, pp. 50–51.
- [109] S. Sankaran, C. Mao, E. Seok, D. Shim, C. Cao, R. Han, D. J. Arenas, D. B. Tanner, S. Hill, C.-M. Hung, and K. K. O, “Towards terahertz operation of CMOS,” in *Proceedings of the ISSCC '09*, 2009, pp. 202–204.
- [110] E. Seok, D. Shim, C. Mao, R. Han, S. Sankaran, C. Cao, W. Knap, and K. K. O, “Progress and challenges towards terahertz CMOS integrated circuits,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 8, pp. 1554–1564, 2010.
- [111] T. Kleine-Ostmann and T. Nagatsuma, “A Review on Terahertz Communications Research,” *Journal of Infrared, Millimeter, and Terahertz Waves*, vol. 32, no. 2, pp. 143–171, 2011.
- [112] S. Hu, Y.-Z. Xiong, B. Zhang, L. Wang, T.-G. Lim, M. Je, and M. Madihian, “A SiGe BiCMOS TX/RX Chipset With On-Chip SIW Antennas for Terahertz Applications,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 11, pp. 2654–2664, 2012.
- [113] J.-D. Park, S. Kang, S. Thyagarajan, E. Alon, and A. Niknejad, “A 260 GHz fully integrated CMOS transceiver for wireless chip-to-chip communication,” in *Proceedings of the VLSIC '12*, 2012, pp. 48–49.
- [114] B. Khamaisi, S. Jameson, E. Socher, and S. Member, “A 210–227 GHz Transmitter With Integrated On-Chip Antenna in 90 nm CMOS Technology,” *IEEE Transactions on Terahertz Science and Technology*, vol. 3, no. 2, pp. 141–150, 2013.
- [115] A. Lisauskas, S. Boppel, M. Mundt, V. Krozer, and H. G. Roskos, “Subharmonic Mixing With Field-Effect Transistors: Theory and Experiment at 639 GHz High Above fT,” *IEEE Sensors Journal*, vol. 13, no. 1, pp. 124–132, 2013.
- [116] R. Han and E. Afshari, “A High-Power Broadband Passive Terahertz Frequency Doubler in CMOS,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 3, pp. 1150–1160, 2013.
- [117] H. Rucker, B. Heinemann, and A. Fox, “Half-Terahertz SiGe BiCMOS Technology,” in *Proceedings of the SiRF '12*, 2012, pp. 133–136.
- [118] E. Öjefors, J. Grzyb, B. Heinemann, B. Tillack, and U. R. Pfeiffer, “A 820 GHz SiGe chipset for terahertz active imaging applications,” in *Proceedings of the ISSCC '11*, 2011, pp. 224–225.
- [119] R. A. Hadi, J. Grzyb, B. Heinemann, and U. R. Pfeiffer, “A Terahertz Detector Array in a SiGe HBT Technology,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 9, pp. 2002–2010, 2013.
- [120] U. R. Pfeiffer, J. Grzyb, H. Sherry, A. Cathelin, and A. Kaiser, “Toward low-NEP room-temperature THz MOSFET direct detectors in CMOS technology,” in *Proceedings of the IRMMW-THz '13*, 2013, pp. 1–2.

- [121] A. K. Geim and K. S. Novoselov, "The rise of graphene," *Nature materials*, vol. 6, no. 3, pp. 183–191, 2007.
- [122] F. Schwierz, "Graphene transistors," *Nature nanotechnology*, vol. 5, no. 7, pp. 487–96, 2010.
- [123] A. Vakil and N. Engheta, "Transformation optics using graphene," *Science*, vol. 332, no. 6035, pp. 1291–4, 2011.
- [124] 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.
- [125] I. Llatser, C. Kremers, A. Cabellos-Aparicio, J. M. Jornet, E. Alarcón, and D. N. Chigrin, "Graphene-based nano-patch antenna for terahertz radiation," *Photonics and Nanostructures - Fundamentals and Applications*, vol. 10, no. 4, pp. 353–358, 2012.
- [126] M. Tamagnone, J. S. Gomez-Diaz, J. R. Mosig, and J. Perruisseau-Carrier, "Reconfigurable terahertz plasmonic antenna concept using a graphene stack," *Applied Physics Letters*, vol. 101, p. 214102, 2012.
- [127] N. Behdad and K. Sarabandi, "A varactor-tuned dual-band slot antenna," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 2, pp. 401–408, 2006.
- [128] I. Llatser, C. Kremers, D. Chigrin, J. M. Jornet, M. C. Lemme, A. Cabellos-Aparicio, and E. Alarcón, "Radiation Characteristics of Tunable Graphennas in the Terahertz Band," *Radioengineering Journal*, vol. 21, no. 4, 2012.
- [129] J. M. Jornet and I. F. Akyildiz, "Graphene-based plasmonic nano-transceiver for terahertz band communication," in *Proceedings of the EuCAP '14*, 2014, pp. 492–6.
- [130] P. K. Singh, G. Aizin, N. Thawdar, M. Medley, and J. M. Jornet, "Graphene-based Plasmonic Phase Modulator for terahertz-band communication," in *Proceedings of the EuCAP '16*, 2016.
- [131] T. Palacios, A. Hsu, and H. Wang, "Applications of Graphene Devices in RF Communications," *IEEE Communications Magazine*, vol. 48, no. 6, pp. 122–128, 2010.
- [132] Y. Wu, D. B. Farmer, F. Xia, and P. Avouris, "Graphene Electronics: Materials, Devices, and Circuits," *Proceedings of the IEEE*, vol. 101, no. 7, pp. 1620–1637, 2013.
- [133] M. C. Lemme, T. J. Echtermeyer, M. Baus, and H. Kurz, "A Graphene Field Effect Device," *IEEE Electron Device Letters*, vol. 28, no. 4, pp. 282–284, 2007.
- [134] H. Wang, D. Nezich, J. Kong, and T. Palacios, "Graphene Frequency Multipliers," *IEEE Electron Device Letters*, vol. 30, no. 5, pp. 547–549, 2012.
- [135] H. Wang, A. Hsu, J. Wu, J. Kong, and T. Palacios, "Graphene-Based Ambipolar RF Mixers," *IEEE Electron Device Letters*, vol. 31, no. 9, pp. 906–910, 2010.
- [136] Y.-M. Lin, A. Valdes-Garcia, S.-J. Han, D. B. Farmer, I. Meric, Y. Sun, Y. Wu, C. Dimitrakopoulos, A. Grill, P. Avouris, and K. A. Jenkins, "Wafer-Scale Graphene Integrated Circuit," *Science*, vol. 332, no. June, pp. 1294–1298, 2011.
- [137] S.-J. Han, A. V. Garcia, S. Oida, K. A. Jenkins, and W. Haensch, "Graphene radio frequency receiver integrated circuit," *Nature communications*, vol. 5, 2014.
- [138] S. Vaziri, M. Ostling, and M. C. Lemme, "A Hysteresis-Free High-k Dielectric and Contact Resistance Considerations for Graphene Field Effect Transistors," *ECS Transactions*, vol. 41, pp. 165–171, 2011.

- [139] J. Hendry, "Isolation of the Zenneck Surface Wave," in *Proceedings of the LAPC '10*, 2010, pp. 613–6.
- [140] A. Karkar, T. Mak, N. Dahir, R. Al-Dujaily, K.-F. Tong, and A. Yakovlev, "Network-on-Chip Multicast Architectures Using Hybrid Wire and Surface-Wave Interconnects," *IEEE Transactions on Emerging Topics in Computing*, vol. 99, no. PP, 2016.
- [141] 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.
- [142] H. Jagode and J. Hein, "Custom assignment of MPI ranks for parallel multi-dimensional FFTs: Evaluation of BG/P versus BG/L," in *Proceedings of ISPA '08*, 2008, pp. 271–283.
- [143] H. Yordanov and P. Russer, "Area-Efficient Integrated Antennas for Inter-Chip Communication." in *Proceedings of the EuMC '10*, 2010, pp. 401–404.
- [144] E. Tavakoli, M. Tabandeh, S. Kaffash, and B. Raahemi, "Multi-hop communications on wireless network-on-chip using optimized phased-array antennas," *Computers and Electrical Engineering (Elsevier)*, vol. 39, no. 7, pp. 2068–2085, 2013.
- [145] M. Win and R. Scholtz, "Impulse radio: how it works," *IEEE Communications Letters*, vol. 2, no. 2, pp. 36–38, 1998.
- [146] J. G. Proakis and M. Salehi, *Fundamentals of Communication Systems*. Pearson, 2005.
- [147] A. Mineo, M. Palesi, G. Ascia, and V. Catania, "Exploiting antenna directivity in wireless NoC architectures," *Microprocessors and Microsystems*, vol. PP, no. 99, pp. 1–8, 2016.
- [148] V. Vijayakumaran, M. P. Yuvaraj, N. Mansoor, N. Nerurkar, A. Ganguly, and A. Kwasinski, "CDMA Enabled Wireless Network-on-Chip," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 10, no. 4, p. Art. 28, 2014.
- [149] K. Duraisamy, R. G. Kim, and P. P. Pande, "Enhancing Performance of Wireless NoCs with Distributed MAC Protocols," in *Proceedings of the ISQED '15*, 2015, pp. 406–411.
- [150] D. Zhao and Y. Wang, "SD-MAC: Design and Synthesis of a Hardware-Efficient Collision-Free QoS-Aware MAC Protocol for Wireless Network-on-Chip," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1230–1245, 2008.
- [151] P. Dai, J. Chen, Y. Zhao, and Y.-H. Lai, "A study of a wire-wireless hybrid NoC architecture with an energy-proportional multicast scheme for energy efficiency," *Computers and Electrical Engineering (Elsevier)*, vol. 45, pp. 402–416, 2015.
- [152] N. Mansoor and A. Ganguly, "Reconfigurable Wireless Network-on-Chip with a Dynamic Medium Access Mechanism," in *Proceedings of the NoCS '15*, 2015.
- [153] 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.
- [154] R. G. Kim, W. Choi, G. Liu, E. Mhandesi, P. P. Pande, D. Marculescu, and R. Marculescu, "Wireless NoC for VFI-Enabled Multicore Chip Design: Performance Eval-



- uation and Design Trade-offs,” *IEEE Transactions on Computers*, vol. 65, no. 4, pp. 1323–36, 2015.
- [155] D. Zhao, Y. Wang, J. Li, and T. Kikkawa, “Design of multi-channel wireless NoC to improve on-chip communication capacity,” in *Proceedings of the NoCs ’11*, 2011, pp. 177–184.
- [156] D. Zhao and R. Wu, “Overlaid Mesh Topology Design and Deadlock Free Routing in Wireless Network-on-Chip,” in *Proceedings of the NoCs ’12*, 2012, pp. 27–34.
- [157] A. M. Amorim, P. A. Oliveira, and H. C. Freitas, “Performance evaluation of single- and multi-hop wireless networks-on-chip with NAS Parallel Benchmarks,” *Journal of the Brazilian Computer Society*, vol. 21, no. 1, p. 6, 2015.
- [158] K. Duraisamy, P. P. Pande, and A. Kalyana, “High Performance and Energy Efficient Wireless NoC-Enabled Multicore Architectures for Graph Analytics,” in *Proceedings of the CASES ’15*, 2015, pp. 147–156.
- [159] T. Majumder, P. P. Pande, and A. Kalyanaraman, “Wireless NoC Platforms With Dynamic Task Allocation for Maximum Likelihood Phylogeny Reconstruction,” *IEEE Design & Test*, vol. 31, no. 3, pp. 54–64, 2014.
- [160] K. Duraisamy, R. G. Kim, W. Choi, G. Liu, P. P. Pande, and D. Marculescu, “Energy Efficient MapReduce with VFI-enabled Multicore Platforms,” in *Proceedings of the DAC-52*, 2015, p. Art. 6.
- [161] J. Murray, T. Lu, P. Wettin, P. P. Pande, and B. Shirazi, “Dual-Level DVFS-Enabled Millimeter-Wave Wireless NoC Architectures,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 10, no. 4, pp. 27:1–27, 2014.
- [162] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [163] “802.15.3c - Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs) - Amendment 2: Millimeter-wave-based Alternative Physical Layer Extension,” 2009.
- [164] D. Matolak, S. Kaya, and A. Kodi, “Channel modeling for wireless networks-on-chips,” *IEEE Communications Magazine*, no. June, pp. 180–186, 2013.
- [165] A. Ganguly, P. Pande, B. Belzer, and A. Nojeh, “A Unified Error Control Coding Scheme to Enhance the Reliability of a Hybrid Wireless Network-on-Chip,” in *Proceedings of the DFT ’11*, 2011, pp. 277–285.
- [166] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in *Proceedings of the ETS ’13*, 2013, pp. 21–27.
- [167] A. Mineo, M. Palesi, G. Ascia, and V. Catania, “Runtime Tunable Transmitting Power Technique in mm-Wave WiNoC Architectures,” *IEEE Transactions on VLSI Systems*, vol. PP, no. 99, 2015.
- [168] Z. Chen and Y. Zhang, “Inter-chip wireless communication channel: Measurement, characterization, and modeling,” *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 978–986, 2007.
- [169] P. Y. Chiang, S. Woracheewan, C. Hu, L. Guo, H. Liu, R. Khanna, and J. Nejedlo, “Short-Range, Wireless Interconnect within a Computing Chassis: Design Challenges,” *IEEE Design & Test of Computers*, vol. 27, no. 4, pp. 32–43, 2010.

- [170] Z. M. Chen, Y. P. Zhang, A. Hu, and T.-S. Ng, "Bit-error-rate analysis of UWB radio using BPSK modulation over inter-chip radio channels for wireless chip area networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 2379–2387, 2009.
- [171] L. Yan and G. W. Hanson, "Wave propagation mechanisms for intra-chip communications," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 9, pp. 2715–2724, 2009.
- [172] M. Sun and Y. Zhang, "Performance of intra-chip wireless interconnect using on-chip antennas and UWB radios," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 9, pp. 2756–2762, 2009.
- [173] J. Mehta, D. Bravo, and K. K. O, "Switching noise picked up by a planar dipole antenna mounted near integrated circuits," *IEEE Transactions on Electromagnetic Compatibility*, vol. 44, no. 2, pp. 282–290, 2002.
- [174] A. Oncu and M. Fujishima, "Millimeter-Wave CMOS Impulse Radio," in *Advances in Solid State Circuit Technologies*, 2010, pp. 255–288.
- [175] K. Witrissal, G. Leus, G. J. M. Janssen, M. Pausini, F. Troesch, T. Zasowski, and J. Romme, "Noncoherent ultra-wideband systems," *IEEE Signal Processing Magazine*, vol. 26, no. 4, pp. 48–66, 2009.
- [176] J. M. Jornet and I. F. Akyildiz, "Femtosecond-Long Pulse-Based Modulation for Terahertz Band Communication in Nanonetworks," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1742–54, 2014.
- [177] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: the energy-reliability tradeoff," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 6, pp. 818–831, 2005.
- [178] M. S. Rahaman and M. H. Chowdhury, "Energy efficiency of error control coding in intra-chip RF/wireless interconnect systems," *Microelectronics Journal*, vol. 41, no. 1, pp. 33–40, 2010.
- [179] S. L. Howard, C. Schlegel, and K. Iniewski, "Error Control Coding in Low-Power Wireless Sensor Networks: When is ECC energy-efficient?" *EURASIP Journal on Wireless Communications and Networking*, no. 2, pp. 29–46, 2006.
- [180] S. Abadal, M. Iannazzo, M. Nemirovsky, A. Cabellos-Aparicio, and E. Alarcón, "On the Area and Energy Scalability of Wireless Network-on-Chip: A Model-based Benchmarked Design Space Exploration," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1501–13, 2015.
- [181] S. Kawai, R. Minami, Y. Tsukui, Y. Takeuchi, H. Asada, and A. Musa, "Direct-Conversion Transceiver in 65-nm CMOS," in *Proceedings of the RFIC '13*, 2013, pp. 137–140.
- [182] A. Kahng, B. Li, L. Peh, and K. Samadi, "Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in *Proceedings of DATE '09*, 2009, pp. 423–8.
- [183] D. Ding and D. Pan, "OIL: a nano-photonics optical interconnect library for a new photonic networks-on-chip architecture," in *Proceedings of the SLIP '09*, 2009, pp. 11–18.

- [184] S. Xiao, M. Khan, H. Shen, and M. Qi, “Multiple-channel silicon micro-resonator based filters for WDM applications,” *Optics Express*, vol. 15, no. 12, pp. 7489–98, 2007.
- [185] J. Ahn, M. Fiorentino, R. G. Beausoleil, N. Binkert, A. Davis, D. Fattal, N. P. Jouppi, M. McLaren, C. M. Santori, R. S. Schreiber, S. M. Spillane, D. Vantrease, and Q. Xu, “Devices and architectures for photonic chip-scale integration,” *Applied Physics A*, vol. 95, no. 4, pp. 989–997, 2009.
- [186] K. Preston, N. Sherwood-Droz, J. S. Levy, and M. Lipson, “Performance Guidelines for WDM Interconnects Based on Silicon Microring Resonators,” in *Proceedings of the CLEO '11*, 2011, pp. 4–5.
- [187] “Intel Corporation, Intel Products. ark.intel.com,” 2015.
- [188] H. K. Mondal and S. Deb, “An energy efficient wireless Network-on-Chip using power-gated transceivers,” in *Proceedings of the SOCC '14*, 2014, pp. 243–248.
- [189] X. Yu, H. Rashtian, and S. Mirabbasi, “An 18.7-Gb/s 60-GHz OOK Demodulator in 65-nm CMOS for Wireless Network-on-Chip,” *IEEE Transactions on Circuits And Systems -I: Regular Papers*, vol. 62, no. 3, pp. 799–806, 2015.
- [190] Y. Wu, K. A. Jenkins, A. Valdes-Garcia, D. B. Farmer, Y. Zhu, A. Bol, C. Dimitrakopoulos, W. Zhu, F. Xia, P. Avouris, and Y.-M. Lin, “State-of-the-art graphene high-frequency electronics,” *Nano letters*, vol. 12, no. 6, pp. 3062–7, jun 2012.
- [191] M. Walther, D. Cooke, C. Sherstan, M. Hajar, M. Freeman, and F. Hegmann, “Terahertz conductivity of thin gold films at the metal-insulator percolation transition,” *Physical Review B*, vol. 76, no. 12, p. 125408, 2007.
- [192] I. Akyildiz, J. Jornet, and C. Han, “TeraNets: ultra-broadband communication networks in the terahertz band,” *IEEE Wireless Communications*, vol. 21, no. 4, pp. 130–135, 2014.
- [193] C. Zhang, C. Han, and I. F. Akyildiz, “Three Dimensional End-to-End Modeling and Directivity Analysis for Graphene-Based Antennas in the Terahertz Band,” in *Proceedings of the GLOBECOM '15*, 2015, pp. 1–6.
- [194] G. Hanson, “Fundamental transmitting properties of carbon nanotube antennas,” *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3426–35, 2005.
- [195] P. J. Burke, S. Li, and Z. Yu, “Quantitative Theory of Nanowire and Nanotube Antenna Performance,” *IEEE Transactions on Nanotechnology*, vol. 5, no. 4, pp. 314–334, 2006.
- [196] J. M. Jornet and I. F. Akyildiz, “Graphene-based nano-antennas for electromagnetic nanocommunications in the terahertz band,” in *Proceedings of the EuCAP '10*, 2010, pp. 1–5.
- [197] G. W. Hanson, “Dyadic Green’s Functions for an Anisotropic , Non-Local Model of Biased Graphene,” *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 3, pp. 747–757, 2008.
- [198] S. Amanatiadis and N. Kantartzis, “Design and analysis of a gate-tunable graphene-based nanoantenna,” in *Proceedings of the EuCAP '13*, 2013, pp. 4038–4041.
- [199] X. Zhang, X. Huang, T. Leng, G. Auton, and E. Hill, “Graphene Reconfigurable Coplanar Waveguide (CPW)-Fed Circular Slot Antenna,” in *Proceedings of the APS/URSI '15*, 2015, pp. 2293–2294.

- [200] M. Tamagnone, J. S. Gomez-Diaz, J. R. Mosig, and J. Perruisseau-Carrier, “Analysis and design of terahertz antennas based on plasmonic resonant graphene sheets,” *Journal of Applied Physics*, vol. 112, p. 114915, 2012.
- [201] A. Cabellos, I. Llatser, E. Alarcón, A. Hsu, and T. Palacios, “Use of THz Photoconductive Sources to Characterize Tunable Graphene RF Plasmonic Antennas,” *IEEE Transactions on Nanotechnology*, vol. 14, no. 2, pp. 390–396, 2015.
- [202] M. Jablan, H. Buljan, and M. Soljačić, “Plasmonics in graphene at infrared frequencies,” *Physical review B*, vol. 80, no. 24, p. 245435, 2009.
- [203] M. Tamagnone and J. Perruisseau-carrier, “Predicting Input Impedance and Efficiency of Graphene Reconfigurable Dipoles Using a Simple Circuit Model,” *IEEE Antennas and Wireless Propagation Letters*, vol. 13, pp. 313–316, 2014.
- [204] M. Dragoman, D. Neculoiu, A.-C. Bunea, G. Deligeorgis, M. Aldrigo, D. Vasilache, A. Dinescu, G. Konstantinidis, D. Mencarelli, L. Pierantoni, and M. Modreanu, “A tunable microwave slot antenna based on graphene,” *Applied Physics Letters*, vol. 106, no. 15, p. 153101, 2015.
- [205] C. H. Gan, H. S. Chu, and E. P. Li, “Synthesis of highly confined surface plasmon modes with doped graphene sheets in the midinfrared and terahertz frequencies,” *Physical Review B*, vol. 85, no. 12, p. 125431, 2012.
- [206] A. H. Castro Neto, F. Guinea, N. M. R. Peres, K. S. Novoselov, and A. K. Geim, “The electronic properties of graphene,” *Reviews of Modern Physics*, vol. 81, no. 1, pp. 109–162, 2009.
- [207] K. I. Bolotin, K. J. Sikes, Z. Jiang, M. Klima, G. Fudenberg, J. Hone, P. Kim, and H. L. Stormer, “Ultrahigh electron mobility in suspended graphene,” *Solid State Communications*, vol. 146, pp. 351–355, 2008.
- [208] H. Hirai, H. Tsuchiya, Y. Kamakura, N. Mori, and M. Ogawa, “Electron mobility calculation for graphene on substrates,” *Journal of Applied Physics*, vol. 116, no. 8, 2014.
- [209] R. Piesiewicz, T. Kleine-Ostmann, N. Krumbholz, D. Mittleman, M. Koch, J. Schoebel, and T. Kurner, “Short-Range Ultra-Broadband Terahertz Communications: Concepts and Perspectives,” *IEEE Antennas and Propagation Magazine*, vol. 49, no. 6, pp. 24–39, 2007.
- [210] J. M. Jornet and I. F. Akyildiz, “Channel Modeling and Capacity Analysis for Electromagnetic Wireless Nanonetworks in the Terahertz Band,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3211–3221, 2011.
- [211] J. Kokkonen, J. Lehtomaki, K. Umehayashi, and M. Juntti, “Frequency and Time Domain Channel Models for Nanonetworks in Terahertz Band,” *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 2, pp. 678–691, 2015.
- [212] C. Jansen, S. Priebe, and C. Moller, “Diffuse scattering from rough surfaces in THz communication channels,” *IEEE Terahertz Science and Technology*, vol. 1, no. 2, pp. 462–472, 2011.
- [213] C. Ronne, L. Thrane, P.-O. Astrand, A. Wallqvist, K. V. Mikkelsen, and S. R. Keiding, “Investigation of the temperature dependence of dielectric relaxation in liquid water by THz reflection spectroscopy and molecular dynamics simulation,” *The Journal of Chemical Physics*, vol. 107, no. 14, p. 5319, 1997.

- [214] W. Wiesbeck, G. Adamiuk, and C. Sturm, “Basic Properties and Design Principles of UWB Antennas,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 372–385, 2009.
- [215] E. G. Farr and C. E. Baum, “Time Domain Characterization of Antennas with TEM Feeds,” in *Sensor and Simulation Notes*, 1998, p. Note 426.
- [216] A. Shlivinski, E. Heyman, and R. Kastner, “Antenna characterization in the time domain,” *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 7, pp. 1140–9, 1997.
- [217] “EM Software and Systems, FEKO.” [Online]. Available: <http://www.feko.info>
- [218] A. F. Molisch, “Ultra-Wide-Band Propagation Channels,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 353–371, 2009.
- [219] D.-H. Kwon, “Effect of Antenna Gain and Group Delay Variations on Pulse-Preserving Capabilities of Ultrawideband Antennas,” *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 8, pp. 2208–2215, 2006.
- [220] C. Xu, Y. Jin, L. Yang, J. Yang, and X. Jiang, “Characteristics of electro-refractive modulating based on Graphene-Oxide-Silicon waveguide,” *Optics Express*, vol. 20, no. 20, pp. 22 308–22 405, 2012.
- [221] N. Abramson, “The ALOHA System – Another Alternative for Computer Communication,” in *AFIPS Fall Joint Computer Conference*, vol. 37, 1970, pp. 281–285.
- [222] B. Crow, I. Widjaja, L. Kim, and P. Sakai, “IEEE 802.11 Wireless Local Area Networks,” *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997.
- [223] L. Roberts, “ALOHA packet system with and without slots and capture,” *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.
- [224] 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.
- [225] R. M. Metcalfe and D. R. Boggs, “Ethernet: distributed packet switching for local computer networks,” *Communications of the ACM*, vol. 19, no. 7, pp. 395–404, 1976.
- [226] 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.
- [227] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, “A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges,” *Wireless Networks (Springer)*, vol. 21, no. 8, pp. 2657–2676, 2015.
- [228] P. Suriyachai, U. Roedig, and A. Scott, “A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 240 – 264, 2012.
- [229] K. Obraczka, “Multicast transport protocols: a survey and taxonomy,” *IEEE Communications Magazine*, vol. 36, no. 1, pp. 94–102, 1998.
- [230] N. Binkert, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, D. a. Wood, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, and T. Krishna, “The gem5 simulator,” *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, 2011.
- [231] W. Heirman and J. Dambre, “Rent’s rule and parallel programs: characterizing network traffic behavior,” in *Proceedings of the SLIP ’08*, 2008, pp. 87–94.

- [232] 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.
- [233] V. Soteriou, H. Wang, and L. Peh, “A Statistical Traffic Model for On-Chip Interconnection Networks,” in *Proceedings of MASCOTS '06*, 2006, pp. 104–116.
- [234] S. Sen, R. Roy Choudhury, and S. Nelakuditi, “CSMA/CN: Carrier Sense Multiple Access with Collision Notification,” in *Proceedings of the MobiCom '10*, 2010, pp. 25–36.
- [235] J. Chan, G. Hendry, A. Biberman, K. Bergman, and L. P. Carloni, “PhoenixSim: A Simulator for Physical-Layer Analysis of Chip-Scale Photonic Interconnection Networks,” in *Proceedings of DATE '10*, 2010, pp. 691–696.
- [236] J.-B. Seo, H. Jin, and V. C. M. Leung, “Throughput Upper-Bound of Slotted CSMA Systems with Unsaturated Finite Population,” *IEEE Transactions on Communications*, vol. 61, no. 6, pp. 2477–2487, 2013.
- [237] Y. C. Tay, K. Jamieson, and H. Balakrishnan, “Collision-minimizing CSMA and its applications to wireless sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1048–1057, 2004.
- [238] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, “Packet Reservation Multiple Access for Local Wireless Communications,” *IEEE Transactions on Communications*, vol. 37, no. 8, pp. 885–890, 1989.
- [239] A. Ephremides and O. Mowafi, “Analysis of a Hybrid Access Scheme for Buffered Users-Probabilistic Time Division,” *IEEE Transactions on Software Engineering*, vol. SE-8, no. 1, pp. 52–61, 1982.
- [240] E. Ebrahimi, O. Mutlu, C. J. Lee, and Y. N. Patt, “Coordinated control of multiple prefetchers in multi-core systems,” in *Proceedings of the MICRO-42*, 2009, pp. 316–326.
- [241] A. Rezaei, M. Daneshtalab, D. Zhao, F. Safaei, X. Wang, and M. Ebrahimi, “Dynamic Application Mapping Algorithm for Wireless Network-on-Chip,” in *Proceedings of the PDP '15*, 2015, pp. 421–424.
- [242] P. Gratz and S. Keckler, “Realistic workload characterization and analysis for networks-on-chip design,” in *Proceedings of the CMP-MSI '10*, 2010, pp. 1–10.
- [243] G. Bezerra, S. Forrest, M. Forrest, A. Davis, and P. Zarkesh-Ha, “Modeling NoC traffic locality and energy consumption with rent’s communication probability distribution,” in *Proceedings of the SLIP '10*, 2010, pp. 3–8.
- [244] R. Löhner, F. Mut, J. Cebal, R. Aubry, and G. Houzeaux, “Deflated Preconditioned Conjugate Gradient Solvers for the Pressure-Poisson Equation: Extensions and Improvements,” *International Journal for Numerical Methods in Engineering*, vol. 87, no. 1-5, pp. 2–14, 2011.
- [245] J. Matienzo and N. Enright Jerger, “Performance analysis of broadcasting algorithms on the Intel Single-Chip Cloud Computer,” in *Proceedings of the ISPASS '13*, 2013, pp. 163–172.
- [246] S. Xiao and W. C. Feng, “Inter-block GPU communication via fast barrier synchronization,” in *Proceedings of the IPDPS '10*, 2010, pp. 1–12.
- [247] P. Gratz, B. Grot, and S. W. Keckler, “Regional congestion awareness for load balance in networks-on-chip,” in *Proceedings of the HPCA-14*, 2008, pp. 203–214.

- [248] G. Nychis, C. Fallin, and T. Moscibroda, “On-chip networks from a networking perspective: congestion and scalability in many-core interconnects,” in *Proceedings of the SIGCOMM*, 2012, pp. 407–18.
- [249] J. Psota, J. Miller, G. Kurian, H. Hoffman, N. Beckmann, J. Eastep, and A. Agarwal, “ATAC : Improving Performance and Programmability with On-Chip Optical Networks,” in *Proceedings of the ISCAS '10*, 2010, pp. 3325–3328.
- [250] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, “The SpiNNaker project,” *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.
- [251] S. Carrillo, J. Harkin, L. J. McDaid, F. Morgan, S. Pande, S. Cawley, and B. McGinley, “Scalable hierarchical network-on-chip architecture for spiking neural network hardware implementations,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2451–2461, 2013.
- [252] D. Adams, “CRAY T3D System Architecture Overview (Technical Report),” 1993.
- [253] A. Gara, M. A. Blumrich, D. Chen, G. L.-T. Chiu, P. Coteus, M. E. Giampapa, R. A. Haring, P. Heidelberger, D. Hoenicke, G. V. Kopcsay, T. A. Liebsch, M. Ohmacht, B. D. Steinmacher-Burow, T. Takken, and P. Vranasi, “Overview of the Blue Gene/L System Architecture,” in *IBM Journal of Research and Development*, 2005.
- [254] J. Laudon and D. Lenoski, “The SGI Origin: A ccNUMA Highly Scalable Server,” in *Proceedings of the ISCA-24*, 1997, pp. 241–251.
- [255] C. Beckmann and C. Polychronopoulos, “Fast barrier synchronization hardware,” *Proceedings SUPERCOMPUTING '90*, 1990.
- [256] J. Sampson, R. González, J. F. Collard, N. P. Jouppi, M. Schlansker, and B. Calder, “Exploiting fine-grained data parallelism with chip multiprocessors and fast barriers,” in *Proceedings of the MICRO-39*, 2006, pp. 235–246.
- [257] W. Zhu, V. C. Sreedhar, Z. Hu, and G. R. Gao, “Synchronization state buffer: supporting efficient fine-grain synchronization on many-core architectures,” in *Proceedings of the ISCA-34*, 2007, pp. 35–45.
- [258] R. Ubal, P. Mistry, D. Schaa, H. Ave, and D. Kaeli, “Multi2Sim: A Simulation Framework for CPU-GPU Computing,” in *Proceedings of the PACT '12*, 2012, pp. 335–344.
- [259] J. M. Mellor-Crummey and M. L. Scott, “Algorithms for scalable synchronization on shared-memory multiprocessors,” *ACM Transactions on Computer Systems*, vol. 9, no. 1, pp. 21–65, 1991.
- [260] F. H. McMahon, “The Livermore Fortran Kernels: A Computer Test Of The Numerical Performance Range,” Lawrence Livermore National Laboratory, Livermore, California, Tech. Rep., 1986.
- [261] M. Badr and N. Enright Jerger, “SynFull: Synthetic Traffic Models Capturing Cache Coherent Behaviour,” in *Proceedings of ISCA-41*, 2014, pp. 109–120.
- [262] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of Ethernet traffic (extended version),” *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

