3-2021

Architecting a One-to-many Traffic-Aware and Secure Millimeter-Wave Wireless Network-in-Package Interconnect for Multichip Systems

M Meraj Ahmed
ma9205@rit.edu

Architecting a One-to-many Traffic-Aware and Secure Millimeter-Wave Wireless Network-in-Package Interconnect for Multichip Systems

by

M Meraj Ahmed

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in Engineering

Kate Gleason College of Engineering

Rochester Institute of Technology
Rochester, New York
March, 2021

Architecting a One-to-many Traffic-Aware and Secure Millimeter-Wave Wireless Network-in-Package Interconnect for Multichip Systems

by

M Meraj Ahmed

Committee Approval:

We, the undersigned committee members, certify that we have advised and/or supervised the candidate on the work described in this dissertation. We further certify that we have reviewed the dissertation manuscript and approve it in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Engineering.

Dr. Amlan Ganguly Date
Dissertation Advisor

Dr. Cory Merkel Date
Dissertation Committee Member

Dr. Andres Kwasinski Date
Dissertation Committee Member

Dr. Panos P. Markopoulos Date
Dissertation Committee Member

Dr. Minseok Kwon Date
Dissertation Defense Chairperson

Certified by:

Dr. Edward Hensel Date
Ph.D. Program Director, Engineering Ph.D.

Architecting a One-to-many Traffic-Aware and Secure Millimeter-Wave Wireless Network-in-Package Interconnect for Multichip Systems

by

M Meraj Ahmed

Submitted to the
Kate Gleason College of Engineering Ph.D. Program in Engineering
in partial fulfillment of the requirements for the
Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

With the aggressive scaling of device geometries, the yield of complex Multi Core Single Chip (MCSC) systems with many cores will decrease due to the higher probability of manufacturing defects especially, in dies with a large area. Disintegration of large System-on-Chips (SoCs) into smaller chips called *chiplets* has shown to improve the yield and cost of complex systems. Therefore, platform-based computing modules such as embedded systems and micro-servers have already adopted Multi Core Multi Chip (MCMC) architectures over MCSC architectures. Due to the scaling of memory intensive parallel applications in such systems, data is more likely to be shared among various cores residing in different chips resulting in significant increase in chip-to-chip traffic, especially one-to-many traffic. This one-to-many traffic is originated mainly to maintain cache-coherence between many cores residing in multiple chips. Besides, one-to-many traffics are also exploited by many parallel programming models, system level synchronization mechanisms, and control signals. However, state-of-the-art Network-on-Chip (NoC)-based wired interconnection architectures do not provide enough support as they handle such one-to-many traffic as multiple unicast traffic using a multi-hop MCMC communication fabric. As a result, even a small portion of such one-to-many traffic can significantly reduce system performance as traditional NoC-based interconnect cannot mask the high latency and energy consumption caused by chip-to-chip wired I/Os. Moreover, with the increase in memory intensive applications and scaling of MCMC systems, traditional NoC-based wired interconnects fail to provide a scalable interconnection solution required to support the increased cache-coherence and synchronization generated one-to-many traffic in future MCMC-based High-Performance Computing (HPC) nodes. Therefore, these computation and memory intensive MCMC systems need an energy efficient, low latency, and scalable one-to-many (broadcast/multicast) traffic-aware interconnection infrastructure to ensure high-performance.

Research in recent years has shown that Wireless Network-in-Package (WiNiP) architectures with CMOS compatible Millimeter-Wave (mm-wave) transceivers can provide a scalable, low latency, and energy-efficient interconnect solution for on and off-chip communication. In this dissertation, a one-to-many traffic-aware WiNiP interconnection architecture with a starvation-free hybrid Medium Access Control (MAC), an asymmetric topology, and a novel flow control has been proposed. The different components of the proposed architecture are individually one-to-many traffic-aware and as a system, they collaborate with each other to provide required support for one-to-many traffic communication in a MCMC environment. It has been shown that such interconnection architecture can reduce energy consumption and average packet latency by 46.96% and 47.08% respectively for MCMC systems.

Despite providing performance enhancements, wireless channel, being an unguided medium, is vulnerable to various security attacks such as jamming induced Denial-of-Service (DoS), eavesdropping, and spoofing. Further, to minimize the time-to-market and design costs, modern SoCs often use Third Party IPs (3PIPs) from untrusted organizations. An adversary either at the foundry or at the 3PIP design house can introduce a malicious circuitry, to jeopardize a SoC. Such malicious circuitry is known as a Hardware Trojan (HT). An HT planted in the WiNiP from a vulnerable design or manufacturing process can compromise a Wireless Interface (WI) to enable illegitimate transmission through the infected WI resulting in a potential DoS attack for other WIs in the MCMC system. Moreover, HTs can be used for various other malicious purposes, including battery exhaustion, functionality subversion, and information leakage. This information when leaked to a malicious external attacker can reveal important information regarding the application suites running on the system, thereby compromising the user profile. To address persistent jamming-based DoS attack in WiNiP, in this dissertation, a secure WiNiP interconnection architecture for MCMC systems has been proposed that re-uses the one-to-many traffic-aware MAC and existing Design for Testability (DFT) hardware along with Machine Learning (ML) approach. Furthermore, a novel Simulated Annealing (SA)-based routing obfuscation mechanism was also proposed to protect against an HT-assisted novel traffic analysis attack. Simulation results show that, the ML classifiers can achieve an accuracy of 99.87% for DoS attack detection while SA-based routing obfuscation could reduce application detection accuracy to only 15% for HT-assisted traffic analysis attack and hence, secure the WiNiP fabric from age-old and emerging attacks.

Acknowledgments

At first, I would like to thank Almighty God who has given me this opportunity to pursue this Ph.D. which helped me to gather a small knowledge from the vast sea of Electrical and Computer Engineering. I am immensely indebted to so many people who kept me motivated throughout this journey and helped intellectually to make this dissertation better. I would like to take this opportunity to express my deepest gratitude towards them.

My advisor, Dr. Amlan Ganguly, is the first person who helped me to create a research mentality through his continuous mentorship and support. Throughout my Ph.D. he taught me how to identify a problem, analyze it, and come up with an acceptable solution. He also taught me to question my own solution so that eventually we could end up with a better one which I believe has helped me a lot to become a better and effective researcher academically at the end. Apart from academic excellence, I have received so many invaluable life lessons from him that have made me a better person from what I was five years ago and will continue to exercise them to improve myself day by day. I strongly believe the academic and life-lessons from him will enable me to contribute significantly in my professional and social life.

I am also very grateful to my Ph.D. dissertation committee members, Dr. Andres Kwasinski, Dr. Panos P. Markopoulos and Dr. Cory Merkel. Without their critical observations, suggestions, and overall feedback it would not have been possible to come up with an improved solution to the research questions answered in this dissertation. I thank them for managing time from their tight schedule for all of my research review meetings. I would also like to thank Dr. Miseok Kwon for serving as my dissertation defense chair. A very special thanks to Rebecca who helped me in so many ways that can not be expressed here.

My graduate life would have been very unproductive, had it not been for Dr. Naseef Mansoor. All the informal conversations over the phone and formal discussion in various meetings with him have kept me motivated and made me thoughtful about very small things in my research that otherwise I would have overlooked. I also thank Abhishek Vashist who was my lab mate, course mate, coffee mate, and research mate on various occasions, on various projects, and courses. Without his help it would not have been possible to have a better understanding of the security projects discussed in this dissertation.

I would like to thank my friends, Md Shahriar Shamim, Sayed Ashraf Mamun, Mohammad

Saidur Rahman, Usama Abdullah, Sajeed Mohammad Shahriat for their brotherly love and support throughout this entire Ph.D. journey. When I started this lonely journey, they were the people who made me feel like home and encouraged me in many ways after a long tiresome typical day of a graduate student. I'll never forget the time I spent with them.

I would also like to express my gratitude towards my beloved wife Sarrah Shahid with whom I have shared all the depressions of my graduate life and who tolerated everything without any complain. I am blessed to have her in my life and with that blessing, I have been gaining weights constantly. I wish all her dreams could come true and find her beside me till the end of my life. I also appreciate the support I received from my in-laws.

Finally, it is not possible for me to express my love and respect for my parents, M Mozammel Haque and Kamrunnahar for their unconditional love and support at every step in my life. Nothing is enough to appreciate the sacrifice they have made for me. I also thank my elder brother A.K.M. Kaisar Ahmed and sister Mahfuza Begum for their support and love. Everything in my life is meaningless without my family members.

The text of Chapter 3, 4, 5, and appendix A are in part a reprint of the material from the papers *M. Ahmed et al., "An asymmetric, one-to-many traffic-aware mm-wave wireless inter-connection architecture for multichip systems"* in IEEE Transactions on Emerging Topics in Computing, pp. 1–1, 2020, *M. M. Ahmed et al., "Aware-wi: A jamming-aware reconfigurable wireless interconnection using adversarial learning for multichip systems"* in Sustainable Computing Informatics and Systems, p. 100470, 2020, and *M Meraj Ahmed, Abhijitt Dhavlle, Naseef Mansoor, Purab Sutradhar, Sai Manoj Pudukotai Dinakarrao,Kanad Basu, Amlan Ganguly "Defense against On-Chip TrojansEnabling Traffic Analysis Attacks"* in EEE, AsianHost, 2020, *A Ganguly et al, "The Advances, Challenges and Future Possibilities of Millimeter-Wave Chip-to-Chip Interconnections for Multichip Systems". JLPEA, Vol. 28, Issue 8, No. 1:5,* Feb 2018 respectively. The dissertation author was the primary researcher and author, and the co-authors involved in the publication assisted the research which forms the foundation for that manuscript.

The research presented in this dissertation is supported in part by US National Science Foundation (NSF) CAREER grant CNS-1553264. This is also supported in part by US Defense Advanced Research Projects Agency (DARPA) grant HR00112020048, and does not necessarily reflect the position or the policy of the Government. Approved for public release; distribution is unlimited.

Dedication

To My Beloved Parents, Siblings, and Wife

For Their Unconditional Love, Sacrifice, Inspiration, and Support.

I Wish We Could Stay Together Forever.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Extensive computation requirements for various applications and their ever-increasing demand have driven chip designers towards massive integration arena for Complementary Metal Oxide Semiconductor (CMOS) devices. For decades, the silicon industry has exploited Moore's law to satisfy the exponential growth in the functionality required for application domains like weather forecasting, astrophysics, bioinformatics, computer graphics, and many others. To provide the required performance for such computation expensive applications, computing platforms have undergone a paradigm shift form Single Core Single Chip (SCSC) architecture to Multi Core Single Chip (MCSC) architecture due to the increased parallelism and low-power consumption offered by MCSC paradigm. As the design and integration complexity continue to increase in MCSC systems, recently, High Performance Computing (HPC) systems such as server blades and embedded systems have adopted Multi Core Multichip (MCMC) architecture over MCSC to ensure higher reliability, yield, and hence, lower cost.

Such MCMC architectures accompanied by the increase in memory intensive applications and device scaling have significant impact on design requirements for various components of the system. Therefore, to achieve the desired performance and meet the system requirements, memory subsystem, interconnect, integration, even the core micro architecture need to be redesigned accordingly. Moreover, due to the increase of Third Party IPs (3PIPs) for low cost and faster time to market, security has become a major concern in current MCMC systems. The research presented in this dissertation considers one-to-many traffic requirements, issues and challenges in a MCMC system and proposes a novel interconnection architecture to overcome them while ensuring a secure communication from various threats and vulnerabilities associated with the proposed interconnect.

In this chapter, some essential background starting from the SCSC system, interconnect evolution, MCMC systems have been discussed in Section 1.1, which justifies the need for MCMC system for future computing platforms. The issues in MCMC systems including one-to-many traffic requirement have been discussed in Section 1.3. These issues in MCMC have motivated this research to adopt wireless interconnect as a solution and discussed in Section 1.4. The challenges of the wireless interconnect especially the communication security have been presented in Section 1.4.2. The chapter concludes by describing the research objectives and contributions in Section 1.5 and Section 1.6 respectively.

1.1 Background

With technology scaling and increased operating frequency for higher performance, both static and dynamic power increase significantly for SCSC systems in advanced technology nodes. This upsurge in power consumption not only increases chip temperature but also decreases chip reliability and performance. On the other hand, increase in device density due to scaling has not only resulted in increased power consumption density, but also in the dramatic increase of global wires in a single chip. Unlike local wires and gate delays,

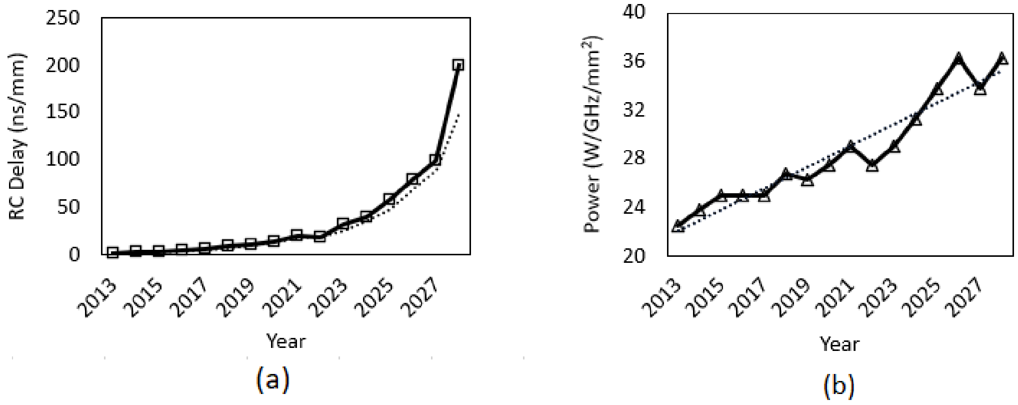


Figure 1.1: Interconnect (a) Delay and (b) Power consumption trend.

the global interconnect delay does not scale with the technology due to Resistive-Capacitive (RC) and inductive coupling [9] which also limits the maximum clock frequency of the system. Figure 1.1 shows the RC delay and power density trend according to International Technology Roadmap for Semiconductors (ITRS), 2013 [10] and also have been discussed in [11]. In a nutshell, the SCSC design lacks the speed and energy efficiency required by the future computation expensive system. Therefore, we need a power efficient interconnect solution that can satisfy the requirements of the future computation intensive HPC nodes.

The MCSC system has appeared as a feasible solution to address the power and frequency limitations of the uniprocessor system [12]. Multicore systems according to Flynn's taxonomy can be thought as an example of Multiple Instruction Multiple Data (MIMD) computing organization, where different cores execute different threads (Multiple Instructions), working on different parts of memory (Multiple Data) concurrently [13]. Therefore, a multicore system can exploit the parallelism by running more tasks in different cores in parallel. Because of the parallelism, cores can run at a lower frequency along with reduced operating voltage and still can produce significant speedup. Therefore, MCSC systems can reduce both dynamic and static power consumption while meeting the required performance for computation intensive systems. Intel's 80 core Polaris [14] and 48 core Single-chip Cloud Computer (SCC) [15], Tilera's 64 core Tile 64 [16] are some examples of MCSC systems.

However, as we integrate more numbers of cores in a system, due to the significant increase in Communication-to-Computation ratio (C2C), the communication architecture plays a vital role to determine the system performance. Traditional bus-based interconnection fabrics for System-on-Chip (SoC) like the ARM AMBA [17] and the IBM Core Connect [18] are based on a shared wired bus and therefore, cannot support multiple communication and require significant arbitration overhead. Moreover, with the increase in the number of connected cores to the bus, the parasitic capacitance of the interconnect increases significantly making the shared bus-based approach a non-scalable solution for future multicore SoC. Even after repeater insertion interconnect delay might exceed a single clock cycle. Bus splitting [19], various coding techniques [20] have been introduced to reduce power consumption and delay of the system. However, with an increase in the number of cores in the design, the increased complexity in implementing those methodologies fails to provide any viable solution. Given the limitations of traditional bus-based interconnect and importance of interconnects on overall system performance, the research focus in multicore systems have been shifted from computation to scalable communication architecture required for future HPC systems.

The Network-On-Chip (NoC) paradigm has emerged as an interconnection infrastructure enabling the integration of hundreds of cores on the same die [21]. The NoC separates the computational cores from the interconnection, hence, communication needs and provides a scalable plug-and-play network for data communication. In NoC interconnection architecture, cores within each chip communicate with each other through on-chip switches [22]. These switches are connected by a regular topology which makes the interconnect deterministic and scalable. The inter-switch distances are maintained carefully so that data can reach to the next core within a single cycle. Moreover, the switches usually follow a pipelined architecture to increase the system throughput even more. Unlike traditional circuit-switched wire-based system, NoC implements packet switched network where each packet is composed of many flow control units, flits. The NoC architecture can also be configured to provide a hybrid of packet switching and circuit switching methodology to achieve the desired performance. In short, various routing and switching protocol can be adopted to improve the system performance further. In NoC interconnection architecture, as more number of cores can communicate simultaneously, it significantly increases the throughput and reduces communication latency. Figure 1.2 shows a NoC interconnection architecture arranged in a regular mesh. Intel's 80 core Polaris [14] and 48 core SCC [15], Tilera's 64 core Tile 64 [16] multicore systems implement NoC as the communication fabric.

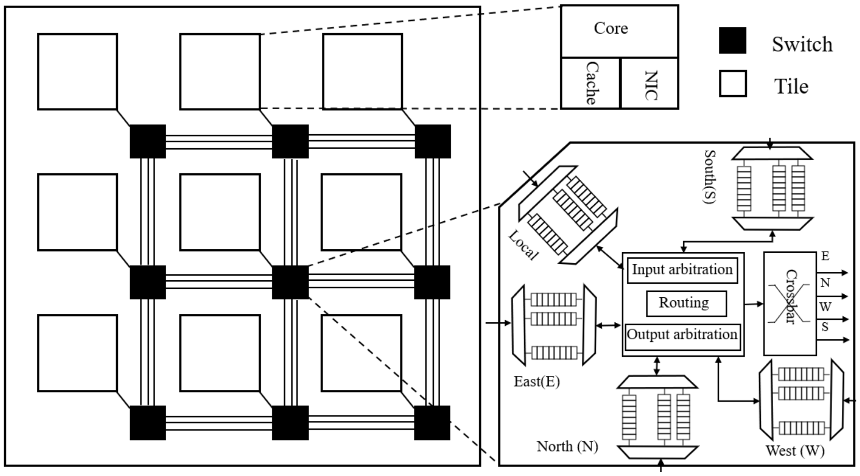


Figure 1.2: Regular mesh NoC topology with switch.

Table 1.1: Yield improvement with chip size reduction

Dies/package	**Cores/die**	**Die dimension (X $\times$ Y mm^2)**	**Good die/wafer**	**Yield (%)**
1	64	20×20	86	60.8
2	32	20×10	232	77.5
4	16	10×10	545	87.9
8	8	10×5	1186	93.7
16	4	5×5	2471	96.8

1.2 Multichip Multicore Systems: Motivations

As the demand for performance continues to grow up to 300×, according to ITRS [23], 100x cores need to be integrated to meet such performance. Though NoC can provide a scalable interconnection solution to integrate a large number of cores, integration of such a large number of cores in a single chip is faced with another massive challenge. In advanced process nodes, different factors such as sub-wavelength lithography, line edge roughness, and random dopant fluctuation can cause a wide process variation, which can result in higher fault density [24] and hence, lower yield. Figure 1.3 shows this phenomenon where we considered two wafers with two different size dies. The wafer with a smaller die size has a higher yield. Therefore, the disintegration of large multicore processors into smaller chips called, *chiplets* is used to alleviate the effect of higher fault densities in advanced technology nodes [25].

$$Y = [\frac{1 - e^{-AD}}{AD}]^2 \tag{1.1}$$

This is because the disintegration will decrease the area of individual chips and therefore, improve the yield of the individual chips. Table 1.1 shows the increase in yield for implementing a system with an increasing number of chips ranging from a single monolithic chip to 16 chips manufactured in a 300 mm round wafer. We consider a total of 64 cores in the system and a moderate fault density of $0.13/cm^2$ as reported by ITRS for current process nodes. The yield has been calculated using the Murphy model [26] as shown in equation 1.1. Where Y is the yield, A is the die area, D is the defect density. Disintegrating a large chip with 64 cores into smaller chips or chiplets will result in steadily increasing yield. These smaller chips or chiplets are integrated into a platform-based system and enable the integration of chiplets from heterogeneous process technologies or functionalities. This, in turn, offers both process and functional flexibility in the design while eliminating the design and manufacturing complexity of large SoCs. MCMC computing modules with multiple processors or chiplets can be found in a wide range of platform-based designs from servers to embedded systems. An example of such a MCMC module is the AMD EPYC series released in 2017 [27], which is a processor system designed for sophisticated servers. The EPYC Threadripper processor node is available as a 4-chip System-in-Package (SiP) with 8 cores in each chip, fabricated in 14 *nm* lithography technology. However, these new approaches require multiple chips to be interconnected efficiently to ensure desired performance with low cost. Moreover, the existing multichip systems can have processing chips such as multicore chips, CPUs, GPUs or a heterogeneous mix of such chips [28] (e.g., AMD's Fusion Accelerated Processor Units (APUs)) depending upon desired functionality. Moreover, disintegration enables a chip to be reused efficiently in different systems to provide the required functionality.

1.3 Challenges of Multichip Multicore Systems

Though disintegrating a large die into smaller chiplets improves yield and reduces cost per die, it also provides less functionality as it has now a fewer number of cores integrated into a small area. Aggregating more number of small chiplets in a single package can provide the similar functionality of a large chip having thousands of cores. The increased coordination among several chiplets results in a significant increase in chip-to-chip communication. Moreover, often these data packets need to encounter protocol conversion overhead while traveling chip-to-chip paths. Depending on interconnect, integration methodology used, and the nature of the traffics that exist in such MCMC environment, the design of the MCMC system

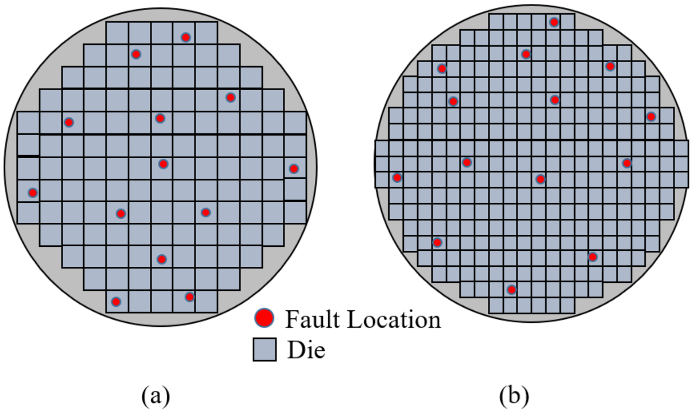


Figure 1.3: Fault for different die size (a) Large die (b) Small die.

can be faced with different challenges. However, it is obvious that, along with intra-chip communication, inter-chip communication needs to evolve at a rapid pace in a seamless manner to avoid being the performance bottleneck of the next generation HPC nodes.

1.3.1 Integration Issues with Wired Interconnect

Several metal-based integration and interconnection methodologies have been used to design the communication fabric in a MCMC environment. In traditional Two Dimensional (2D) multichip platforms, inter-chip communication happens through C4 bumps coupled with flip-chip packaging [29] or through Peripheral Component Interconnect (PCI) or PCI express (PCIe) which is a common local I/O bus standard. Recent trends according to the ITRS [23] predicts that the pitch of the wired I/O interconnects, solder bumps or pads in Integrated Circuits (ICs) is not scaling as fast as the gate lengths or pitch of on-chip interconnects as shown in Figure 1.4. This implies a gap in density and performance of traditional I/O systems relative to on-chip interconnections. The wiring complexity of both on-chip and off-chip interconnects exacerbates the problem by posing design challenges, crosstalk, and signal integrity issues [30]. All these factors along with protocol conversion overheads cause additional delay and reduce the energy efficiency for 2D multichip platforms.

On the other hand, in Three Dimensional (3D) integration technology, multiple chips can

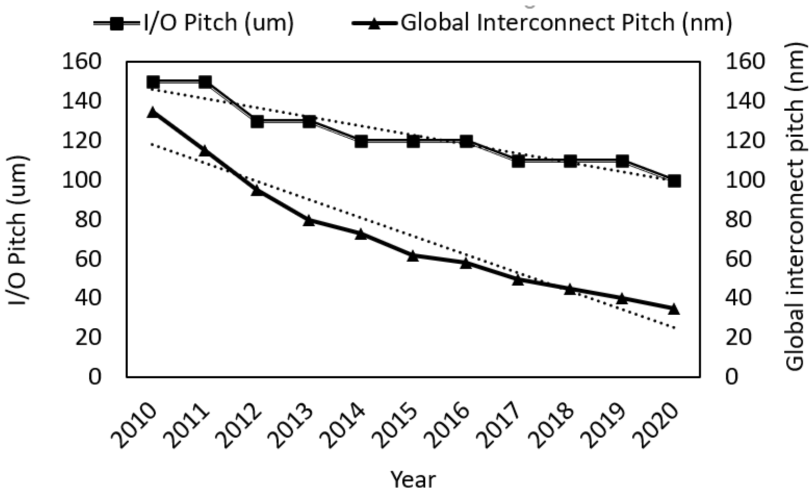


Figure 1.4: Scaling of I/O pitch and minimum global interconnect pitch.

be stacked on top of each other in different layers and hence can exploit the spatial proximity of the chips to provide low latency multichip communication [31]. Moreover, 3D integration can integrate CMOS with other disparate non-silicon technologies stacked in different layers to establish interconnection between them. However, complex thermal management techniques are required to address the higher power dissipation densities in 3D ICs due to smaller footprints. These thermal management techniques vary from dynamic power management methods such as Dynamic Voltage Frequency Scaling (DVFS), temperature-aware task migration [32] or microfluidic cooling channels for better heat circulation [33]. Moreover, die thinning for the fabrication of Through-Silicon-Vias (TSVs) to connect multiple layers results in lower yields.

In 2.5D integration technique, multiple chip modules are placed side by side on another die called interposer that contains several metal layers in it through which the multiple chips are connected to each other [34]. These metal layers are fabricated using the same Back-End-of-Line (BEOL) processes as the traditional metal interconnect inside dies. An array of μ-bumps are then used to provide an electrical connection between the dies and the top metal layers of the interposer using Redistribution Layers (RDL). Either TSVs or RDL can be used for the interposer to package interconnection to route signal, power, and ground to package C4 bumps. In 2.5D integration, the disintegration of processor chips will lower the total manufacturing cost considering the fact that smaller die size will result in higher yield and better packing of the rectangular die on a round wafer [35]. Moreover, in 2.5D integration, the processor chips do not need to undergo die thinning process alleviating the manufacturing woes of monolithic 3D IC. However, interposer metal layer usage has been limited by the edge-to-edge connection among neighboring chips making the available resources underutilized. Moreover, the number of chips that can be integrated using interposer is limited by the silicon die size used as an interposer.

1.3.2 Integration Issues with Alternative Interconnects

As conventional wire-based interconnection architectures limit the performance of future MCMC systems, an alternative emerging interconnect technology along with NoC framework have the potential to provide the required performance for future MCMC systems. Integrated inter and intra-chip photonic interconnections [36][2] is a promising solution to the off-chip interconnection challenges as it provides ultrahigh bandwidth, low power consumption, high noise immunity [37]. Moreover, the enormous bandwidth provided by optical interconnect can be leveraged through Wavelength Division Multiplexing (WDM) to increase the capacity and performance of the system. However, optical interconnect poses many new challenges such as huge static power consumption [38]. The laser source used for an optical signal generation has very low efficiency. The micro resonator used in the optical interconnect architectures needs thermal tuning as its wavelength selectivity varies with temperature. The photonic signal cannot be buffered directly and therefore requires electrical conversion and hence introduce additional delay. Most importantly, most of the nanophotonic devices lack compatibility with standard CMOS process.

The Zenneck Surface Wave Interconnect (SWI) is an emerging interconnect which is essentially an in homogeneous 2D electromagnetic wave (EM) supported by a surface. The surface is a designed waveguide that traps the EM in two-dimensional media [39]. As a result, the electrical-field decay rate in the SWI from the source horizontally along the boundary is around $\frac{1}{\sqrt{d}}$, where d is the distance from the source [41]. This low power dissipation allows the SWI to offer relatively linear Joules per bit scaling compared to the higher order scaling of regular global buffered wire interconnects. However, though the surface wave interconnect provides energy efficient multicast/broadcast message transmission the surface itself needs to be designed carefully to match the desired impedance. Therefore, designing such a surface with precise dimension and material is very challenging. Moreover, making the surface wave incident at the required Brewster angle for maximum transmission efficiency requires an additional transducer and hence increases the design complexity.

1.3.3 Challenges of One-to-many Traffic

Although different kinds of traffic exist in a MCMC environment, in this thesis, the importance of one-to-many traffic has been highlighted as a small portion of such traffic can degrade system performance significantly. State-of-the-art interconnects do not provide enough support for such traffic and hence, it is very challenging to come up with a one-to-many traffic-aware architecture. To overcome this challenge, it is critical to understand their sources and impact on MCMC systems as described in the following sections.

One-to-many Traffic Sources and Trend

Due to the scaling of parallel applications, the computation is likely to be distributed among various cores residing in different chips increasing the communication-to-computation ratio in MCMC systems. Also, due to application scaling, several applications can be mapped to different subsets of cores located in different chips. Moreover, as modern applications are memory intensive, such a SiP with in-package memory modules is very common. Therefore, traffic pattern in those compute nodes not only consists of the core to core unicast messages but also a memory to core multicast/broadcast messages. Since, multiple copies of the same data is shared among several caches across multiple chips, these one-to-many traffic patterns (multicast/broadcast) mainly originates from cache coherence protocols used in a multicore system. Cache coherence methods ensures that cores in all the chips see the same and the latest value of the shared variable during a read operation. To enforce this, communication between caches and hence multiple chips is required.

The directory or snooping based cache coherency is one of the primary sources of such multicast/broadcast messages in a shared memory SiP that uses multicast to invalidate a shared cache block or broadcast the updated block for the requesting cores. Although snooping-based cache coherence protocols are simple to implement in bus-based processors, they generally require additional broadcast-based support for global communication and hence, does

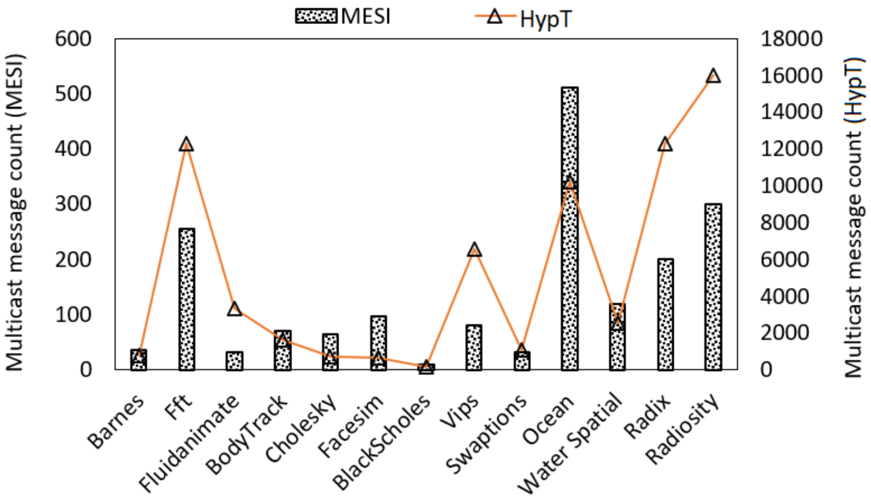


Figure 1.5: Number of multicast messages per 1M instructions in application specific traffic.

not represent a scalable solution in handling one-to-many traffic in a many core architecture. On the other hand, directory-based cache coherence protocols with few multicast and directories acting as ordering points reduce the overhead associated with global communication and ordered delivery. However, directory-based protocols incur additional area and power overhead to maintain directories and also lowers performance due to indirection. Irrespective of the cache coherence protocol used, due to scaling of parallel applications, transaction to manage shared data among multiple cores/chips in MCMC systems is increasing mainly as one-to-many traffic pattern. Figure 1.5 shows the number of multicast messages injected per one million instruction for MESI and AMD Hyper Transport (HypT) cache coherence protocol in a 64 core system with 64 KB, 2-way L1 cache (ID) and 512 KB of shared, 8-way L2 cache for PARSEC [40] and SPLASH2 [41] benchmark traffic patterns [42]. Though MESI protocol maintains a low multicast intensity, HypT, on the other hand being a broadcast based protocol injects a significant amount of traffic in the network. Moreover, the number of destinations for such multicast/broadcast messages varies from 5 to 27 in such multicore systems and consequently such one-to-many traffic can be as huge as 80% of the entire existing traffic in such systems [42]. Therefore, though the number of multicast message per one million instruction is small, the total amount of multicast traffic is significant, as each message needs to be delivered to multiple destinations.

Moreover, many control signals such as passing global states, power gating, and barrier synchronization require multicast/broadcast messages to be sent efficiently through the communication network. In message passing, widely employed collective primitives such as MPI_Allgather or MPI_Allreduce use multicast. Finally, novel computing paradigms such as spiking neural network [43], genetic algorithm-based computation [44] could be also multicast-driven. As the Last Level Cache (LLC) is distributed and the main memory is shared among all the cores in such multichip system, the number of one-to-many messages are significant as discussed in the previous paragraph.

State-of-the-art Wired Interconnect Support for One-to-many Traffic

Traditional NoC-based wired interconnection architectures are mainly designed to handle on-chip unicast traffic and therefore, the most simple mechanism to handle one-to-many traffic is to treat them as repeated unicast message. In this unicast-based mechanism the one-to-many message is replicated as many times as the number of destinations as each replica is served independently which makes the mechanism inefficient for large number of destination nodes. Other one-to-many traffic handling methodologies include either path-base or tree-based routing schemes [45]. In path-based routing, multiple replicas of the original message are send to several groups of destinations. Within each group, a copy of the message is consumed and the original flit is forwarded to the next destination. While in tree-based routing, a single original message is injected by the source which is replicated at the intermediate switches and delivered to the desired destinations following a spanning tree. However, both of the methodologies cause severe local and global congestion as they generate many replicas of the original message and therefore, incur significant power, area, and delay overhead.

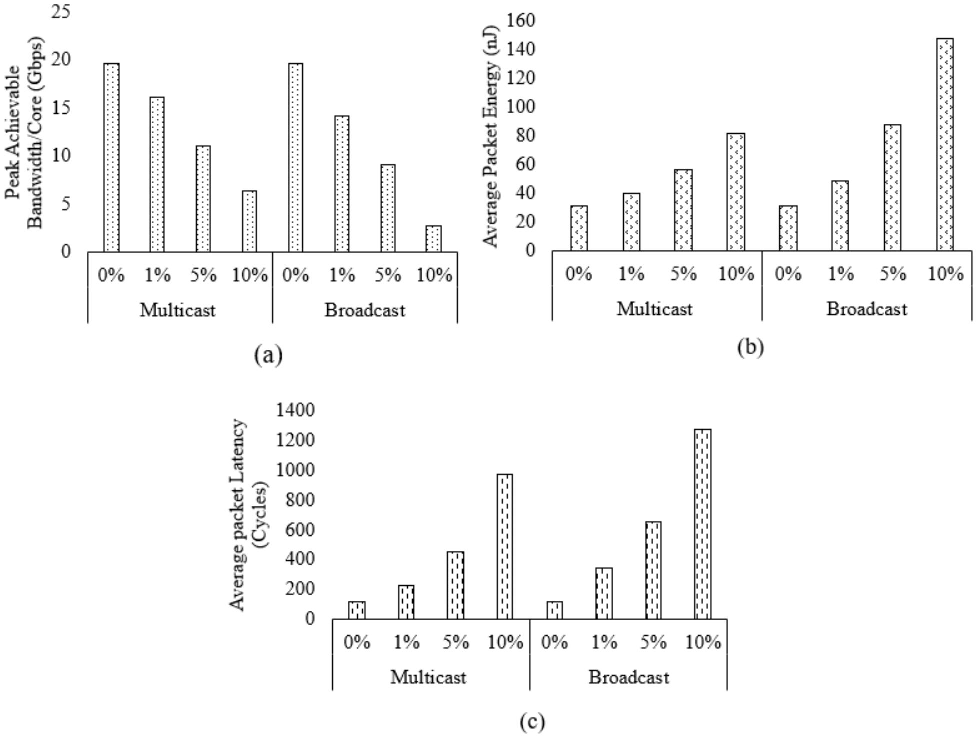


Figure 1.6: Performance (a) Bandwidth (b) Energy (c) Latency degradation due to one-to-many traffic in MCMC systems using wired interconnect.

Impact on MCMC System Performance

In addition, NoC being an on-chip interconnection network cannot mask the high latency and power overhead caused by chip-to-chip I/Os. Similarly, due to its multihop nature, in NoC interconnection architecture the multicast/broadcast latency increases with system size. As a result, even a small amount of such one-to-many traffic can introduce heavy local and global congestion, and thus can significantly reduce system performance irrespective of the adopted routing scheme [46]. Figure 1.6 shows the average bandwidth and average packet energy degradation for a 4 chip system with 16 cores in each chip using conventional wired interconnect for multichip communication. Figure 1.6 is clear representation of the consequence of having poor one-to-many traffic support on MCMC system performance. The MCMC performance becomes even worse as these systems are scaled. As one-to-many traffic represents collective communication, it can significantly slow down the execution time of cache coherent processors with no or limited support [47]. It has been shown in [48] that poor management of one-to-many traffic arising from synchronization can reduce system performance by 40%. Switch optimization [49], one-to-many traffic-aware topology [50] can be designed to reduce average communication latency for such systems. However, such architectural optimizations non-scalable and often come with huge complexities and overheads. Therefore, to ensure low-latency for one-to-many data transfer for such MCMC systems, a scalable, one-to-many traffic aware in-package interconnection architecture referred to Network-in Package (NiP) is required.

1.4 Millimeter-wave Wireless Interconnect: An Emerging Solution

Manufacturing and fabrication limitations associated with emerging interconnect technologies discussed in previous sections limits them from being implemented as a CMOS compatible and energy efficient communication backbone. Research in recent years has demonstrated that on-chip and off-chip wireless interconnects can establish radio communications within as well as multiple chips. On-chip antennas with multi Giga Hertz (GHz) bandwidths in millimeter-wave (mm-wave) bands, specifically, in the unlicensed 60GHz band, ranging up to 10m are fabricated and demonstrated [51]. Figure 1.7 shows a conceptual view of mm-wave wireless communication in a MCMC environment. Throughout this dissertation, such mm-wave NiP for MCMC will be called as Wireless NiP (WiNiP). While mm-wave interconnects for MCSC system will be termed as Wireless NoC (WiNoC). Recently, the feasibility of WiNiP in energy efficient chip-to-chip data communication has also been investigated [3]. Though on-chip antenna operating in Tera Hertz (THz) band using Carbon Nanotube (CNT) or Graphene-based structures can provide wireless channels with higher bandwidth [52][53], integration of these antennas with the CMOS process faces significant challenges and therefore cannot be adopted for intra and inter-chip communication. On the other hand, mm-wave wireless on-chip embedded antennas for intra-chip and inter-chip communication are designed and evaluated in [54]. However, such a wireless system has so many challenges to overcome. The advantages and challenges of mm-wave wireless interconnect architectures have been discussed in the following sections.

1.4.1 Advantages of Mm-wave Wireless Interconnects

Apart from the low latency and energy efficiency offered by mm-wave wireless interconnects, one of the key motivating factors of using mm-wave wireless interconnect is the CMOS compatibility. The mm-wave transceiver and antenna being CMOS compatible can be readily integrated with other components in the SoC. Moreover, there is no need for the layout of physical interconnects, neither metallic nor optical fibers. This implies that the designs will

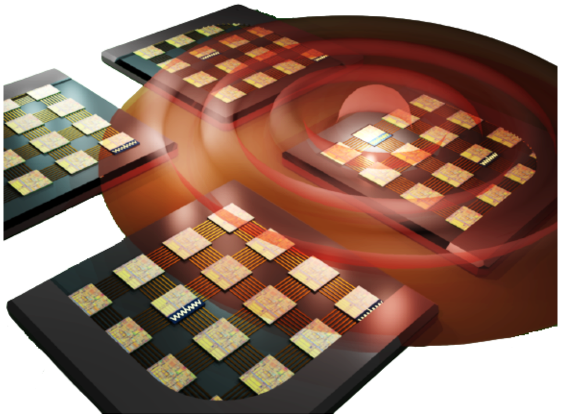


Figure 1.7: A conceptual view of MCMC mm-wave wireless communication.

be more flexible and modular. Therefore, systems can be designed in a plug-and-play manner where the interconnect routing and placement will not impose additional constraints on the system design. Eliminating interconnect place-and-route procedures for large and complex systems can result in a significant reduction in design, test, and verification times. Chiplets equipped with wireless transceivers can be just integrated on a physical platform without the need for physical interconnect layout, thus simplifying the design and integration process. This will reduce time-to-market for multichip systems, potentially reducing cost. The omnidirectional mm-wave antenna can be used for efficient multicast/broadcast transmission. This provides additional advantages for control and cache coherent traffic transmission. Moreover, the 60GHz spectrum is unlicensed and therefore mitigates channel acquisition cost.

1.4.2 Challenges of Mm-wave Wireless Interconnects

Mm-wave wireless interconnects have emerged as a CMOS compatible, energy-efficient, low latency intra and inter-chip interconnect solution. However, many researches have been going on to address different challenges of mm-wave wireless interconnect to make it the next generation interconnect for future HPC systems. Some of those challenges have been described briefly in the section below.

Energy-Efficient and High-Speed Transceiver Design

Inter-chip wireless links require extremely low energy consumption to be competitive with high-speed wired I/O and other emerging technologies such as photonic interconnects. The power consumption in data transmission over wireless medium mainly occurs at the transmitters and receivers. The choice of the transceiver is dependent on the choice of the physical layer and modulation technique. Modulation techniques such as Binary Phase Shift Keying (BPSK) [55], Quadrature Amplitude Modulation (QAM) requires sophisticated receiver design and often consumes high power. Therefore, many mm-wave interconnect designs consider using non-coherent On-off-Keying (OOK) modulation for its simplicity and energy efficiency. Non-coherent modulations eliminate power-hungry, high-frequency carrier recovery circuits such as Phase Locked Loops (PLLs) resulting in low power consumption in the transceivers. A 20Gbps OOK transceiver operating at 260GHz with a bit energy consumption of 58.65pJ/bit is designed in 65nm technology [56]. Due to the low voltage headroom in advanced technology nodes, the power output of the transmitter would be limited. However, Fully Depleted Silicon on Insulator (FD-SOI) processes in advanced nodes such as 45nm or 28nm can provide higher power-performance trade-offs due to faster transistor switching [57]. However, irrespective of the technology nodes, significant improvement in data rate is very challenging using OOK modulation technique.

Antenna Design

The most important criteria for the antenna is that it should be embedded on a die while supporting high bandwidths. Therefore, it needs a small footprint, which in turn dictates the choice of the carrier frequency as the antenna size is proportional to the wavelength of the carrier. In order to provide an energy-efficient alternative to the traditional inter-chip interconnects the wireless interconnects must also provide competitive data rates. Therefore, to enable multi-GHz bandwidths and a small antenna footprint, the minimum carrier frequencies need to be in the mm-wave range higher than a few tens of GHz. Many on-chip antennas like meander dipole, zigzag dipole, slot, loop, inverted F, bow-tie, and Yagi have been designed and investigated [58]. In order to improve directional gains of on-chip antennas fixed beam log-periodic antennas are also investigated [59]. Phased antenna arrays provide the advantage of high directive gain with beam steering although, with additional feeding circuitry. However, given the wavelengths of a few millimeters and die sizes of few millimeters across, it is challenging to design antennas arrays to enable beam-steering for intra and inter-chip wireless communication. A metal mm-wave zigzag antenna [60] has been demonstrated to possess the required characteristics for intra and inter-chip communication as they are more compact compared to other antenna structures such as a patch antenna. In addition, such mm-wave antennas fabricated using top layer metals are CMOS process compatible making them suitable for near-term solutions to the wired interconnect problem.

Robust Wireless Channel Modeling

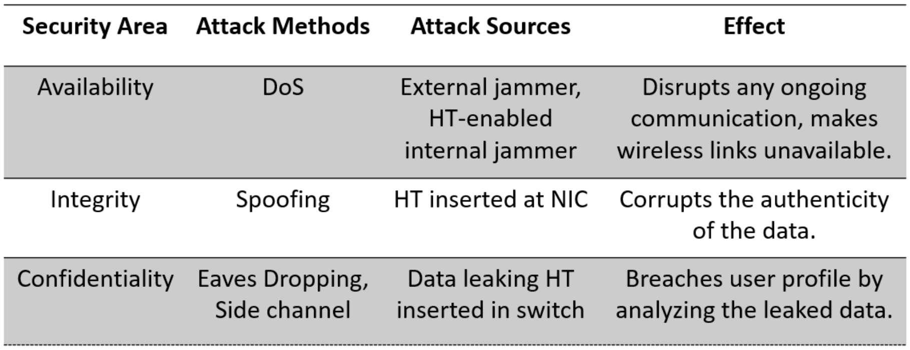
The medium of communication between chips is envisioned to be partially through the air between the chips and through a stack of silicon/silicon dioxide or packaging and encasement of the individual chips. Due to the wide variety of materials used in the fabrication of chips even more diversified by the integration of heterogeneous chiplets in a multichip system specific channel models need to be developed to fine-tune consequent antenna and transceiver design. This introduces challenges in having universal solutions that will fit all environments. The distance between the chips will be the dominant factor affecting the path loss. For normal outdoor and indoor communications, there are many channel models like log-normal or Saleh-Valenzuela (SV) model [61]. In modern processing platforms, chip encapsulant (epoxy mold), the substrate (organic, ceramic), underfill (epoxy composites), and other materials are used to securely pack the silicon die in an end-user package. So, it is required to characterize their complex permittivity i.e. dielectric constant and loss tangent in the mm-wave spectrum as in [62]. Furthermore, reflections from the metallic components in package and encapsulations such as heat-sinks, heat-spreaders, can cause multipath propagation which may cause delay spreads and therefore, create inter-symbol interference (ISI). The delay-spread and ISI in these environments also need to be captured with small-scale channel model for transceiver designs. Specific frequencies and structures make it difficult to generalize channel models to be used in other frequencies and scenarios. All these factors make the channel in a multichip environment quite hostile and extreme for wireless communication.

Medium Access Control Design

In order to improve performance, multiple wireless transceivers need to access the wireless medium to communicate with other wireless transceivers without interference. Efficient design of a Medium Access Control (MAC) layer is required to increase the channel utilization of the wireless interconnect. Several MAC mechanisms ranging from token passing based protocol to complex Code Division Multiple Access (CDMA) based mechanisms have been investigated [63][64][65]. Frequency Division Multiple Access (FDMA) based MAC can increase data rate through simultaneous wireless communication at different carrier frequencies using full channel access [66]. However, the FDMA based approach is non-trivial from the perspective of transceiver design and involves significant power, area, and control overheads. On the other hand, Carrier Sense Multiple Access (CSMA) based MAC suffers from higher contention delay at high injection loads and hence, reduce system performance [67]. Moreover, different MCMC systems have different traffic interactions and link utilization. Therefore, a traffic and utilization-aware MAC design is complicated which varies from system to system and therefore, has its own implementation challenges.

A Secure Wireless Interconnect Design

Although extensive research has been carried out towards improving performance and energy dissipation in MCMC systems using WiNiPs [3], relatively little attention has been given to the information integrity and security or privacy aspects of such wireless interconnects. Wireless being an unguided, shared transmission medium is vulnerable to many attacks such as spoofing, Denial-of-Service (DoS), and eavesdropping (ED) [68]. With continuous increase in device density, heterogenity and integration complexities new threats are evolving [69][70]. Although WiNiPs can have various kinds of threats from different sources, based on their effect they can be broadly classified as Confidentiality, Integrity and Availability (CIA) threatening attacks. As NoC security is a vast research area, this dissertation mainly talks about the availability and confidentiality issues in WiNiP caused by DoS attack and Hardware Trojans (HTs) respectively. Although, DoS and HT induced attacks can originate from various sources and affect the system in various ways, Figure 1.8 shows some attack sources, methods specially for WiNiPs and their corresponding effect.



Security Area	Attack Methods	Attack Sources	Effect
Availability	DoS	External jammer, HT-enabled internal jammer	Disrupts any ongoing communication, makes wireless links unavailable.
Integrity	Spoofing	HT inserted at NIC	Corrupts the authenticity of the data.
Confidentiality	Eaves Dropping, Side channel	Data leaking HT inserted in switch	Breaches user profile by analyzing the leaked data.

Figure 1.8: Various security threats in WiNiPs.

In a MCMC system using NoC as communication fabric, a DoS attack is launched to make the communication resources such as NoC links and switches unavailable to other computing elements resulting in partial or complete breakdown of the entire communication mechanism. The DoS attack on wireless interconnect in a WiNiP architecture can be launched very easily through either an external or internal attacker to breakdown the entire on and off-chip communication backbone. An external attacker can produce a high energy electromagnetic radiation that causes interference in the wireless medium used by the Wireless Interfaces (WIs) deployed in the MCMC system. Moreover, it is also possible that a HT implanted in the system from a vulnerable design and manufacturing process can cause a WI to transmit persistent jamming signals to cause DoS for other WIs. In this case, one of the WIs infected by a HT will send data over the wireless channel irrespective of whether it is enabled by the adopted MAC mechanism of the WiNoC. This will cause contention or interference with legitimate transmissions causing DoS on the remaining WIs. While well-known defenses exist against DoS attacks in large-scale wireless networks [71], those techniques are not directly applicable to the WiNoC scenario due to specific architecture and MAC constraints in WiNoCs. Moreover, on and off-chip wireless communication require different solutions in case of an external DoS attack for MCMC systems.

Due to various technical and economical constraints, most of the design houses have adopted Fabless manufacturing where the in-house designed IP is outsourced to third party company for manufacturing. Moreover, to have low cost and strict time-to-market constraints, many design houses procure 3PIPs from untrusted sources. These phenomena have increased the probability of inserting a HT during design or fabrication phase of the chip to break data confidentiality. HTs are malicious circuits that can alter the functionality of a chip or leak sensitive information to an attacker once it is triggered. As this trigger mechanism can be very complex and usually these HTs have very small footprint, it is very challenging to detect the presence of HTs in a design.

Considering the critical role played by the NoCs and increased use of 3PIPs in modern SoCs, a NoC embedded HT that exploits the interconnection backbone can reveal the communication patterns in the system. This information when leaked to a malicious attacker can reveal important information regarding the application suites running on the system, thereby compromising the user profile. This information in turn, can enable further more severe attacks not just on the multi/many-core processor infected with the HT, but on the systems on which they are deployed. For instance, an adversary obtaining secure military information through a HT deployed in a router can subvert the military backbone, thus leading to a compromise of the national security. Therefore, a security-aware wireless interconnection architecture can be thought of as the most important design challenge of modern HPC platforms.

Design for Testability for Wireless Interconnect

The mm-wave wireless transceivers enabling the on and off-chip communication are high-speed analog components and themselves need to be tested. The transceiver being an analog module makes it even harder to test the wireless interfaces and usually takes very long time for comprehensive testing. Therefore, efficient Design For Testability (DFT) architectures to test the components of wireless interfaces needs to be developed to provide high fault coverage with handful test vectors. A simplified fault model and detection technique need to be developed to detect faulty nodes in the system. Using the tested WIs as Test Access Mechanism (TAM) and interfacing with an external Automatic Test Equipment (ATE) to reduce die damage and faster test vector delivery across the die is another recent challenge that the research community is dealing with.

1.5 Research Objectives

Among all the challenges described in Sections 1.3 and 1.4.2, it is not only critical to design a one-to-many traffic-aware interconnection architecture for high performance but also ensure a secure, sustainable communication in case of various jamming or HT induced attacks. Hence, the objective of this dissertation is twofold as described in the subsequent subsections.

1.5.1 Design of a One-to-many Traffic-Aware WiNiP Architecture for MCMC Systems

With the increase in memory-intensive parallel applications and disintegration, the percentage of one-to-many traffic crossing chip boundaries is increasing. It is essential to provide low latency for such traffic mainly because of two reasons. First, as one-to-many traffic has many destinations in different chips, a small portion of such traffic can introduce huge local and global congestion to reduce system performance significantly. Second, state-of-the-art wired interconnection architectures do not provide a scalable solution to support one-to-many traffic and therefore, suffers huge latency and power overheads caused by off-chip I/Os.

On the other hand, mm-wave wireless interconnects using omnidirectional antennas have inherent support for one-to-many traffic. Moreover, the inherent support provided by such interconnect can be enhanced by making various components of the interconnection architecture individually one-to-many traffic aware. Therefore, one of the the goals of this dissertation is to design a WiNiP interconnect architecture that can efficiently handle one-to-many traffic patterns in a MCMC environment. This includes design and analysis of one-to-many traffic aware multichip topology, a starvation free hybrid MAC, a novel flow control, and eventually a complete interconnection architecture where all of its one-to-many traffic aware components (topology, MAC, flow control) can be integrated to provide the support for such traffic. In broad sense, the objective is to reduce the cost or constraints of one-to-many traffic in MCMC systems so that the performance of wide set of architectures and programming models are improved while opening a large set of unexplored possibilities.

1.5.2 Design of a Secure WiNiP Interconnection Architecture for MCMC System

Although mm-wave WiNiP interconnection architecture can provide low latency and energy-efficient communication for MCMC systems, wireless channel being an unguided and shared communication medium, is vulnerable to various security attacks such as DoS, ED and HT induced traffic analysis attacks as describes in Section 1.4.2. Therefore, even if we design a one-to-many traffic-aware WiNiP as our first objective, the entire MCMC communication will collapse if we fail to take defensive measures against such attacks on WiNiP. Securing the systems against such induced threats with traditional solutions such as encryption often introduce large area overheads and performance penalties. Therefore, such solutions can not be directly adopted in a WiNiP architecture. To address and meet such constraints, a secure WiNiP interconnection architecture needs to be designed that has the in-built intelligence to detect and defend against potential vulnerabilities in WiNiP with low overheads.

Many security attacks can happen on WiNiP interconnect in a MCMC system and each of these attacks require its own detection and defense mechanism. For this goal, in this dissertation we mainly focus on persistent jamming-based DoS attack as it is one of the most common, simple, and yet powerful attack on wireless systems. Moreover, DoS attack is yet to be explored for a multichip environment using wireless interconnect. In a WiNiP, DoS attack from internal and external sources can completely break down the entire communication backbone. To address such DoS attacks, we aim to design a persistent jamming-aware WiNiP interconnection for MCMC system that can detect such attackers as well as sustain on and off-chip communication even under internal and external jamming. Moreover, continuous increase in using 3PIPs from untrusted sources and Fabless manufacturing model, HT-enabled novel attacks are emerging. Such attacks can leak sensitive data to breach user profile and pose severe threat for national security. Analysis and mitigation of such novel threat is also required to ensure user privacy as well as advanced security for MCMC systems. Therefore,

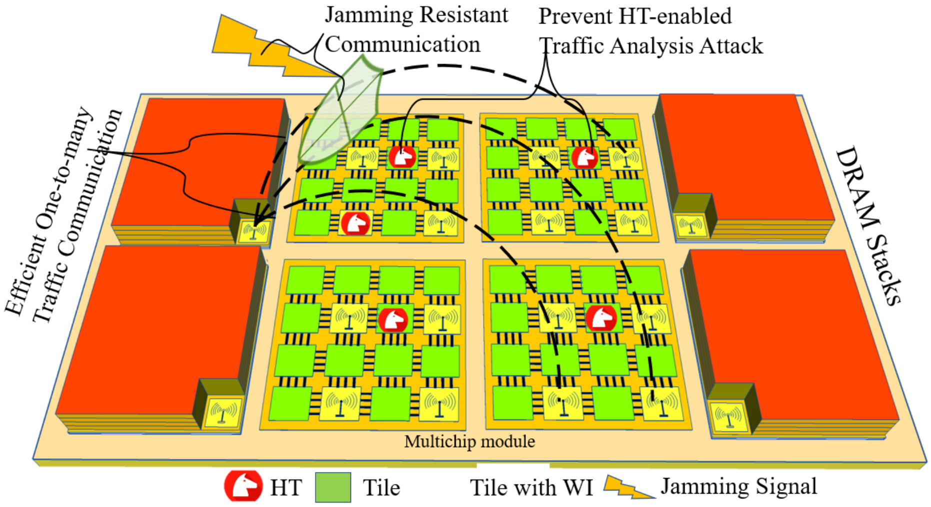

Jamming Resistant Communication
Prevent HT-enabled Traffic Analysis Attack
Efficient One-to-many Traffic Communication
DRAM Stacks
Multichip module
HT
Tile
Tile with WI
Jamming Signal

Figure 1.9: Schematic representation of the objectives of this dissertation.

this dissertation is also focused to analyze and design a DoS and HT-aware secure WiNiP interconnection architecture for MCMC system and promote adoption of such techniques for other systems such as Internet of Things (IoT) to enable a secure communication.

In a nutshell, if we combine the research objectives described in Sections 1.5.1 and 1.5.2, ***the ultimate objective of this dissertation is to design a one-to-many traffic-aware secure WiNiP interconnection architecture for MCMC systems.*** Figure 1.9 shows the schematic representation of all the objectives of this dissertation.

1.6 Research Contributions

In this dissertation, for both of the research objectives as described in Sections 1.5.1 and 1.5.2, various novel topologies, communication protocols, MAC mechanisms, flow control and routing methodologies have been explored which eventually guided us for architecting a one-to-many traffic-aware secure WiNiP communication for MCMC systems. The major contributions of each of the objectives will be describes in the next subsections. The peer-reviewed conferences, book chapter and journal papers are listed in appendix C.

1.6.1 Design of a One-to-many Traffic-Aware WiNiP Architecture for MCMC Systems

As part of this objective, first a one-to-many traffic aware MAC that prioritizes such traffic to ensure low latency was developed. Later, for the first time, a one-to-many traffic-aware asymmetric WiNiP topology that can be configured dynamically to provide efficient support for one-to-many traffic in MCMC system was also explored. Finally, a one-to-many traffic and power-aware flow control was also developed to design a complete one-to-many traffic-aware WiNiP architecture by integrating the asymmetric topology with the hybrid MAC and flow control. Specifically, the contributions of this research goal are

- An asymmetric WiNiP interconnection topology design to handle one-to-many traffic in MCMC systems.

- Design of a one-to-many traffic-aware, starvation-free hybrid MAC unit.
- Proposed a one-to-many traffic-aware WiNiP flow control that integrates the MAC and the asymmetric topology to create a complete one-to-many traffic aware WiNiP architecture.
- Conducted a comparative study between the proposed and other state-of-the-art WiNiP and wired interconnection architectures for both synthetic and application-specific traffic pattern.
- Also analyzed the performance of one-to-one traffic in the proposed one-to-many traffic-aware WiNiP and compared with other state-of-the-art interconnection architectures for MCMC systems.

1.6.2 Design of a Secure WiNiP Interconnection Architecture for MCMC System

As a part of this objective, we proposed a WiNiP architecture for MCMC communication that can detect and defend against persistent jamming-based DoS attack from internal and external sources using Machine Learning (ML). We re-used the existing DFT hardware to detect and defend against jamming attack. Moreover, under such jamming attack, specially for MCMC systems, it is non-trivial to synchronize and inform all other WIs about the presence of an adversary as chip-to-chip communication happens only through wireless medium which is itself vulnerable to the attack. We also developed a MCMC wireless communication protocol along with a reconfigurable MAC that can ensure robust and secure communication. To handle more intelligently crafted jamming attacks and ensure a robust, accurate detection and defense mechanism, an Adversarial Machine Learning (AML)-based detection mechanism have been exploited. A novel HT-based remote traffic analysis attack that can breach user profile has been also introduced. To combat such attack, a Simulated Annealing (SA)-based routing obfuscation methodology has also been proposed to enhance the security of WiNiP architecture. The major contributions of this goal are listed below.

- *To the best of our knowledge, this is the first work that proposes a solution for persistent jamming attack in MCMC systems by re-using DFT infrastructure for WiNiP.*

- Proposed a novel dynamically reconfigurable MAC and the corresponding synchronization mechanism for all WIs under jamming condition.

- Proposed an AML-based mechanism for high accuracy threat detection and recovery from persistent jamming-based DoS attacks.

- Developed a novel communication protocol necessary to ensure a robust communication even under a persistent jamming-based DoS attack.

- Analyzed the performance degradation of the proposed security mechanism for different WiNiPs and compare it with wired MCMC systems.

- *To the best of our knowledge, this is the first work, where, it is shown that by monitoring traffic patterns in a NoC through HTs, the user profile can be compromised.*

- A SA-based novel routing obfuscation methodology has been proposed to defended the system against such an attack.

1.7 Dissertation Organization

This dissertation is organized into six chapters. This chapter describes the motivation to go from a SCSC system to MCMC systems and the advantages and challenges of wired and emerging interconnect technologies. These challenges encouraged us to come up with novel solutions which is the ultimate goal of this research. Chapter 2 discusses the contemporary works on different MCMC interconnection architectures as well as one-to-many traffic-aware interconnect design. Moreover, many recent works on WiNiP design and associated security vulnerabilities along with proposed solutions have been also discussed. The inefficiency of state-of-the-art interconnects in handling one-to-many traffic has ultimately motivated us to design a scalable one-to-many traffic-aware WiNiP architecture for MCMC systems, which is presented in Chapter 3. The work presented in Chapter 3 is the first work that presents the concept of using an asymmetric topology, a reconfigurable MAC, and a one-to-many traffic-aware flow control to handle one-to-many traffic in the MCMC system. Through simulation we showed that the proposed WiNiP architecture outperforms other state-of-the-art wireless and wired interconnect architectures in terms of bandwidth, latency, and energy.

Chapter 4 discusses the design methodology of a secure WiNiP architecture to detect and withstand persistent jamming-based DoS attack in a MCMC system. To the best of our knowledge, this is the first WiNiP security work that uses the underlying DFT architecture to detect persistent jamming. Also, an AML-based detection methodology was used to make the detection more robust. Because of such detection architecture, novel communication protocol, and reconfigurable MAC the WiNiP architecture could ensure a secure MCMC communication even under internal and external jamming attack. A novel HT induced traffic analysis attack has been presented in Chapter 5. Chapter 5 showed how an external attacker can use the leaked information from a simple HT to infer the applications running in the system and hence breach user privacy. As a solution, a novel SA-based controlled dynamic routing obfuscation methodology has also been discussed in the chapter. Chapter 6 summarizes the important lessons learned while designing a secure and one-to-may traffic-aware WiNiP interconnection architecture for MCMC systems. Potential future research directions have also been discussed in Chapter 6.

Chapter 2

Literature Review

With the aim to provide necessary context to this dissertation, in this chapter, we briefly discuss the state-of-the-art research works in designing a multichip communication fabric using different interconnects. Starting from the age-old interconnect and integration methodologies, we review various NoC architectures, state-of-the-art works on MAC design, one-to-many traffic-aware interconnect design, and WiNiP security in this chapter. All these works have motivated us to conduct this research on architecting a one-to-many traffic-aware and secure WiNiP design for MCMC systems.

2.1 Intra and Inter-Chip Interconnects for Multichip Systems

In traditional multichip platforms, inter-chip communication happens through C4 bumps in flip chip package [29] or through PCI or PCIe which is a common local I/O bus standard. Recent trends according to the ITRS [23] predicts that the pitch of the wired I/O interconnects, solder bumps or pads in ICs is not scaling as fast as the gate lengths or pitch of on-chip interconnects. This implies a gap in density and performance of traditional I/O systems relative to on-chip interconnections. The wiring complexity of both on-chip and off-chip interconnects exacerbates the problem by posing design challenges, crosstalk, and signal integrity issues [30]. Such signal quality deteriorations due to microwave effects, crosstalk coupling effects, signal reflections, and frequency-dependent lines losses in the transmission line limit the number of concurrent, high-density inter-chip I/O [30]. Moreover, typically intra-chip and inter-chip interconnections are designed separately to provide design flexibility and modular design approach. However, switching between protocols is necessary if the off-chip communication protocol is different from the on-chip one incurring overheads in inter-chip data transfer. Therefore, the on-chip interconnect has seen a paradigm shift from bus-based design to NoC based interconnection architectures [72]. The NoC can reduce global wire delay and increase throughput through multiple simultaneous parallel transactions [22]. Though NoC can have several topologies (mesh, torus, butterfly), the grid-based mesh is the most commonly used topology as it is relatively easy to design, shows deterministic characteristics and testability [14][16][73]. However, to avoid its multi-hop nature, insertion of long-range links using conventional metal wires [74] or ultra low-latency and low-power express channels between communicating cores [75] have been proposed in the literature. However, such express virtual channel based architecture requires significant control overhead to assign a proper channel for each incoming packet and therefore prone to high contentions. The 3D integration methodology can reduce the communication latency of traditional metal interconnect because of its smaller footprint [31]. Various 3D NoC architectures have also been proposed as a solution for future high speed interconnect [1]. In 3D NoC, multiple dies are vertically connected through TSVs. Figure 2.1 [1] shows a MCMC system using a 3D integrated NoC where multiple dies are connected through vertical NoC links made from TSVs. TSVs are metallic interconnect that passes through a silicon sub-

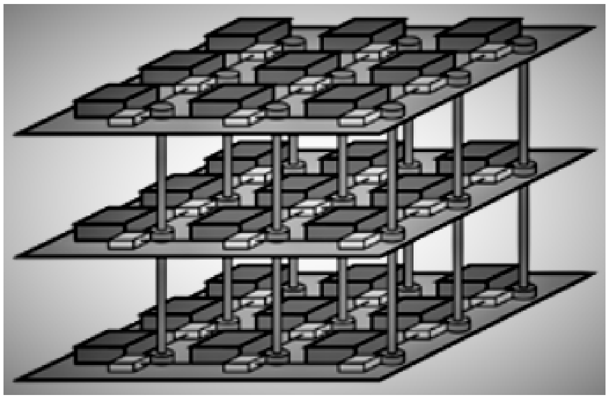


Figure 2.1: A 3D integrated MCMC system [1].

strate and can provide low latency due to its short length (20-100um). However, due to the vertical stacking of multiple dies, 3D integration suffers from high heat density and peak temperature [32]. Therefore it requires sophisticated thermal management. These thermal management techniques vary from dynamic power management methods such as DVFS, temperature-aware task migration [32] or microfluidic cooling channels for better heat circulation [33]. Microfluidic channels, while very effective at cooling chips, need fluid intake pipes through the packaging making the whole system complex. Moreover, die thinning for the fabrication of TSVs, particularly if co-existing with microfluidic channels, resulting in low yields. In the interposer-based 2.5D integration, conventional metal layers within the interposer are used to interconnect the dies of a multichip system [34]. The interposer itself is a relatively large bare silicon die with metal routing layers to provide interconnection between the dies. These metal layers are fabricated using the same BEOL processes as the traditional metal interconnect inside dies and placed at the top of the interposer to provide high-density horizontal connection between active dies. However, interposer metal layer usage has been limited by the edge-to-edge connection among neighboring chips making the available resources underutilized. Moreover, the number of chips that can be integrated using interposer is limited by the interposer die size. These limitations of metal interconnect have inspired researchers to look at emerging interconnect paradigms as a solution.

In [2] the authors proposed an all-optical hierarchical network for both intra chip and inter-chip communication. Figure 2.2 [2] shows the proposed communication architecture. For inter-chip communication, the architecture used segmentation for higher throughput. While traditional optical switching needs electrical control involving Electro-Optic (EO)/Opto-

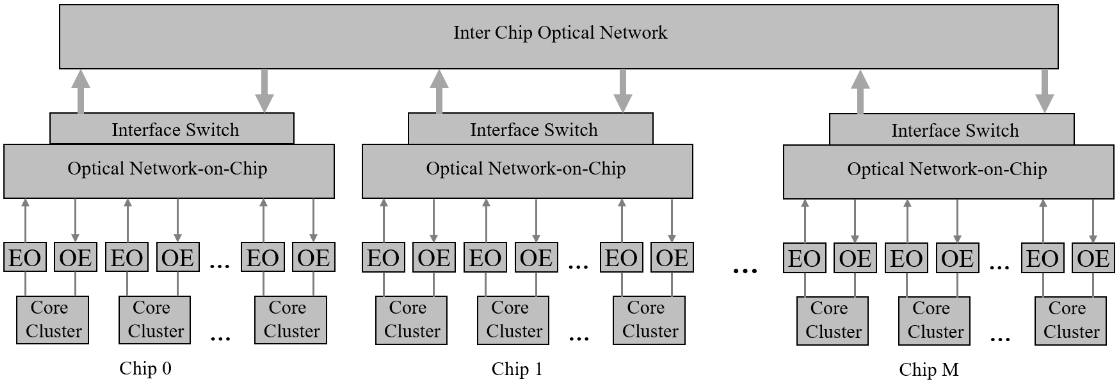


Figure 2.2: An optical interconnection architecture for multichip communication [2].

Electric (OE) conversion the architecture proposed here needs no conversion for inter-chip communication and thus saves significant power. Moreover, they proposed an adaptive power control for laser power for data transmission to reduce power consumption significantly. However, the proposed architecture is similar to shared bus-based wired interconnection architecture and does not provide support for one-to-many traffic. In [76] the authors proposed a 3D nanophotonic system many-core architecture called Corona that uses nanophotonic links for inter-core communication and off-stack memory communication. Dense Wave Division Multiplexing (DWDM) used in Corona has 256 wavelengths associated with each channel for unidirectional communication. Due to the spatial proximity of 3D stacking and huge bandwidth from optical interconnect the system performance improves significantly. In [77] a hybrid, hierarchical on-chip network architecture called Firefly consisting of clusters of nodes has been proposed. The architecture used a Reservation-assisted Single Write Multiple Reads (RSWMR) that enabled a significant reduction in energy consumption at the cost of additional latency. However, low laser efficiency, precise thermal control of the micro-ring resonators, and incompatibility with the existing CMOS process makes the optical interconnect more challenging to be integrated as a highly reliable interconnect. On-chip antennas from graphene, CNT based structures are predicted to provide high bandwidth wireless communication channels [52] [53][78]. On-chip antennas realized by Graphene-based structures are predicted to operate both as modulators as well as antennas providing high bandwidth wireless communication in the THz frequency channels [79]. However, the feasibility for such WiNiPs are limited by the physical implementation and precise fabrication environment control of CNT and graphene-based antennas. The Zenneck SWI interconnect is an emerging interconnect which is essentially an inhomogeneous 2D EM wave supported by a surface. The surface is a designed waveguide that traps the EM in two-dimensional media [39]. Designing such a surface with precise dimension and material is also very challenging. Moreover, making the surface wave incident at the required Brewster angle for maximum transmission efficiency requires an additional transducer and hence increase the complexity.

2.2 Mm-Wave Wireless Interconnect Architectures

Research in recent years has demonstrated that on-chip and off-chip wireless interconnects are capable of establishing radio communications within as well as between multiple chips.

On-chip antennas with multi GHz bandwidths in mm-wave bands, specifically, in the unlicensed 60GHz band, are fabricated and demonstrated [51][80]. Using such on-chip antennas embedded in the chip [60] or waveguides [81] WiNiP architectures are shown to improve energy efficiency and bandwidth of on-chip data communication in multicore chips. A detailed survey of WiNiP architectures and its various components has been presented in [82][83]. Regarding topology, keeping the minimum distance constraint in mind, small-world WiNiP architectures were proposed [52][65] and evaluated. Small world graphs are a type of complex network where both short distance, as well as long distance links, create a connected graph. The long distance, communication beyond a few millimeters happen through wireless links. In [84], the authors demonstrated that small world based WiNiP architecture is resilient to failure of CNT based wireless links. In [85], it has been shown that optimizing a small world WiNiP topology with respect to a particular metric can result in more homogeneous utilization of NoC links and switches consequently reducing temperature hotspots in the NoC. While irregular topologies have better connectivity and diameter properties, unequal wire lengths in irregular topologies like small world networks, make design, implementation, and verification very challenging. Other works have looked at the hierarchical design of WiNiPs where subnetworks communicate internally with wired links while inter-subnet data communication utilizes wireless links [60].

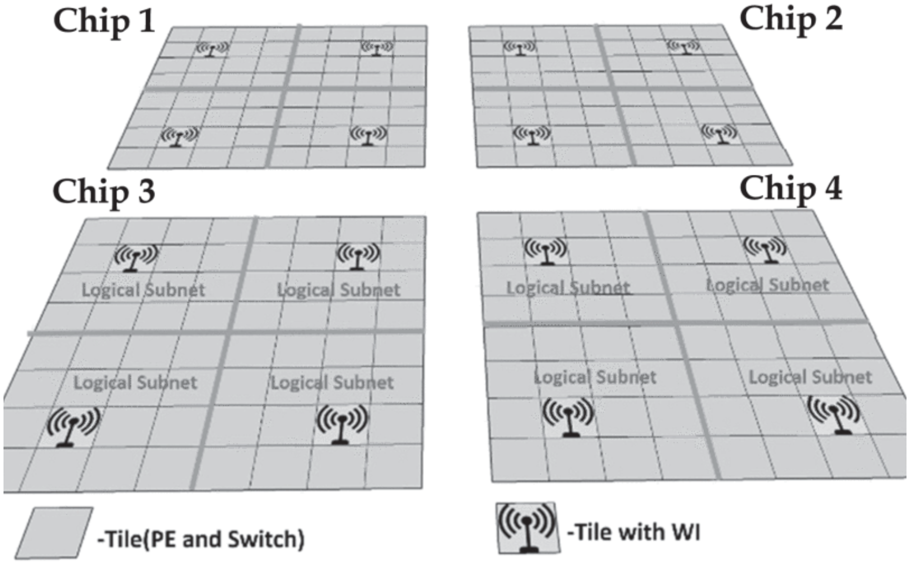


Figure 2.3: MCMC system topology using mm-wave WiNiP interconnect [3].

A hierarchical mm-wave WiNoC architecture [49], a 2D concentrated mesh-based WCube architecture using sub-THz wireless links [86] has been presented to explore the topological impact on system performance. Apart from topology, an energy efficient, dynamic MAC layer can improve WiNiP performance significantly. In the context of a wireless NoC, several MAC mechanisms have been proposed over the years. A Synchronous and Distributed MAC mechanism (SD-MAC) is proposed in [63] for the Ultra-Wide-Band (UWB) WiNiP where impulse based transceivers are used. However, the communication range for the WIs in such WiNiP is limited to a millimeter. Authors in [66] have proposed a WiNiP architecture with multiple non-overlapping channels to enable Frequency Division Multiple Access (FDMA)-based medium access. However, such an FDMA based approach is non-trivial from the perspective of transceiver design. A comparative performance evaluation of CSMA and token based MAC is presented in [87]. In [79], the authors discussed the performance of ALOHA and CSMA for graphene-based WiNiPs. However, due to contention-based retransmission, such MAC suffers from performance issues as demonstrated in [67]. In [88], a distributed token-based MAC arbitration mechanism has been proposed that utilizes orthogonal codes to request channel access. However, the code design does not consider realistic on-chip wireless channel models assuming only a single Line-of-Sight (LOS) path between communicating pairs. Moreover, token-based MAC limits the number of concurrently communicating pairs over a single channel to one. Authors in [3] proposed a WiNiP interconnection architecture for MCMC communication which uses a distributed token-based MAC. Figure 2.3 [3] shows their proposed MCMC topology. However, the proposed MAC did not consider providing priority to one-to-many traffic and focused on one-to-one traffic performance evaluations only. In [89], an efficient Radio Access Control Mechanism (RACM) is proposed for wireless NoCs. However, such MAC results in inefficient support for many-to-few hotspot traffic which is very common for state-of-the-art memory intensive systems. To address the spatial and temporal traffic variations in WIs, two demand aware, distributed dynamic token-based MAC has been proposed in [90]. In [91], authors present the MAC design challenges, requirements in the context of various on-chip systems, workload variations, and performance objectives. However, no literature has presented a one-to-many traffic aware MAC design that can also get support from the underlying topology.

2.3 Handling One-to-many Traffic in NoC Environment

Conventional wired NoC interconnection architecture handles one-to-many traffic as multiple unicast traffic which creates congestion and a severe queueing delay. In [92] authors proposed a hypercube based multichip interconnection architecture on silicon interposer to support one-to-many traffic. However, the interconnection complexity increases with system size scaling. A dynamic path multicast mechanism where the network is recursively divided into multiple small destination regions is proposed in [93]. To meet the requirements of the cache coherence communication, an XY-tree multicast NoC incorporating an ACK aggregation network is proposed in [94]. However, in these systems, the underlying NoC is still a mesh network. In a mesh NoC, the network latency is usually high due to the inherent multihop nature of the system. Multicast support using high-radix switches have been proposed in [95]. However, such switches involve high complexity and cost for implementation. A multicast-aware wired NoC with clockless repeater (referred to as SMART NoC) is also proposed [49]. The performance advantages of the SMART NoCs mainly come from the SMART control mechanism which involves more complex router design and high control overhead. A topology-aware one-to-many traffic support using 3D interconnect has been studied in [96]. The authors proposed different routing algorithms for regular and irregular 3D NoC topologies. However, as the proposed routing mechanisms still use dimension ordered routing, the local and global congestion due to one-to-many traffic can not be avoided in such system.

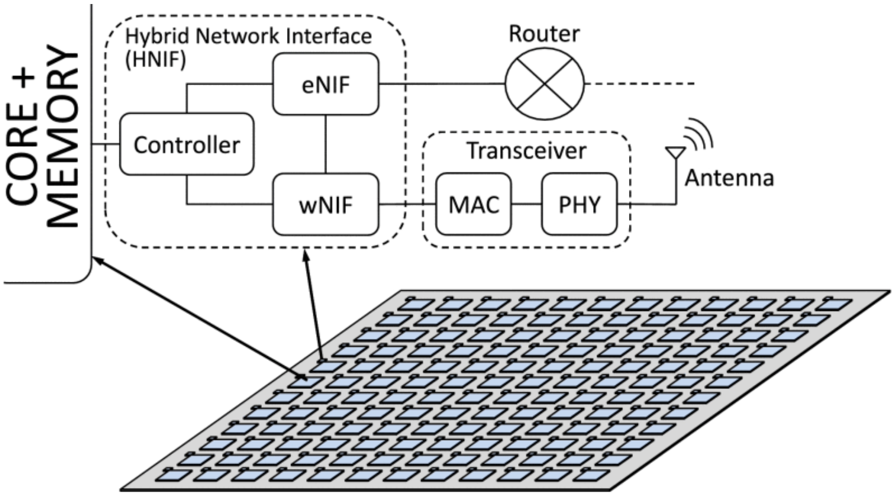


Figure 2.4: OrthoNoC: A broadcast-aware mm-wave WiNoC architecture [4].

A comparative study for multicast message communication using emerging technologies has been done in [97]. Photonic interconnects implementing Single Write Multiple Read (SWMR) architecture [98] inherently support one-to-many traffic. However, each connected node in optical waveguide add logarithmic losses and hence limit the scalability due to static laser power allocation. One-to-many traffic support using RF interconnect was studied in [99]. The obtained low latency for such traffic came with a huge cost of additional power and area overheads to implement the on-chip transmission line and hence does not provide a scalable solution. Broadcast message support with wireless interconnects has been investigated in [100]. However, in [100], the authors conducted only a study of the effect of system size, link capacity, broadcast percentage scaling on a single chip system and did not propose any optimized architecture for one-to-many traffic transmission. In [4], the authors proposed two separate communication planes for on-chip communication using wireless-wired hybrid interconnect. Where the wireless plane is used for multicast/broadcast and the rest of the NoC communication happens through wired interconnect. Because of using two orthogonal communication plane, they called the architecture as OrthoNoC as shown in Figure 2.4 [4]. The work only focused on the inherent broadcast capability of WiNoC and did not propose any architectural novelty to exploit the inherent broadcast capability. In [101], the authors proposed a congestion-aware multicast routing with network coding for WiNoC architecture to efficiently handle heavy multicast traffic. The performance benefit came with the cost of increased routing complexity due to additional wired links. Also, the authors did not take into account the associated area, power overhead with such routing and did not leverage the MAC for high performance.

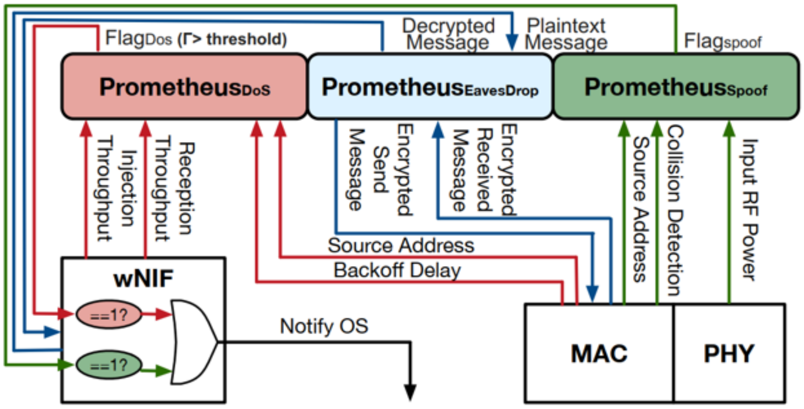


Figure 2.5: Prometheus: A DoS, ED and snooping-resistant WiNoC architecture [5].

2.4 Design of a Secure Mm-wave Wireless Interconnect Architecture

While securing wired NoCs have received attention from researchers in recent years [102][103], mechanisms to secure WiNiPs, in particular, have not been well developed. In [104], a small-world graph based WiNoC architecture was proposed to mitigate DoS attacks. However, small-world irregular topologies have negative implications on the design and verification effort. Moreover, the proposed architecture showed DoS resiliency due the inherent connectivity only and such solution is not suitable for DoS attack introduced by from wireless jamming. A hash-based authentication to prevent eavesdropping has been proposed in [105]. In [5] a secure WiNiP architecture has been proposed that can protect against DoS, eavesdropping, and spoofing. As shown in Figure 2.5 [5], three different modules were design to address each of the vulnerability. However, this work does not address the issue of jamming attack from an external attacker assuming that the packaging will protect against such attacks. This may not be true for all kinds of chips or packaging materials. Furthermore, the solution is too naive and not efficient to detect complex and sophisticated DoS attacks. Persistent jamming-based DoS attack for on-chip wireless interconnect has been addressed in [106]. In case of external jamming, the authors in [106] utilized the underlying wired NoC to sustain communication. However, such solution can not be adopted for WiNiP, as in WiNiP, off-chip communication happens only through wireless interconnect. Moreover, the ML model used in [106] is unaware of the adversarial conditions that an intelligent attacker can exploit to camouflage its presence. In [5], authors developed a spoofing detection and defense mechanism based on received signal power for on-chip wireless interconnect. However, the proposed mechanism in [5] imposes placement restrictions for WI nodes to distinctly identify the senders that are equidistant from the receiver. Such WI placement restrictions can have significant performance impacts and placement challenges. Moreover, such mechanism can not be extended for WiNiP systems specially in the presence of an internal or external jammer. Though persistent jamming attacks are less studied in WiNiPs, a vast amount of research is performed in Wireless Sensor Networks (WSN) for potential solutions. For instance, frequency hopping has been traditionally employed in order to overcome the presence of a jammer [107]. However, multiple jamming devices operating on different bands can effectively block the entire spectrum. Using a directional antenna can be another means to combat the jammer [108]. However, a directional antenna limits the multicast capability and limit WiNiP performance. In [109], authors developed a replay message based DoS attack and also proposed the solution for that. Though Advanced Encryption Standard (AES) has proven robustness against side-channels in the networking domain, adapting it for WiNiP communications adds large processing overheads and thus is not feasible [110].

Although many works have been done on energy-efficient and high-performance NoC architecture design, only a handful of works have explored routing as a solution to secure the NoC, especially against side-channel attacks that focuses on reverse engineering. In [111], authors proposed Region-based Routing (RBR) and Segment-based Routing (SBR) for a secure NoC communication. The proposed routing mechanism in [111] assumes the NoC to be partitioned into different security zones and minimizes the inter-zone traffic to ensure better NoC security. However, with increasing number of security zones, it is hard to reduce inter-zone traffic and requires complicated scheduling policy. In [6], authors proposed to use west-first and adaptive XY routing as they offer more paths for routing and reduce interference with attacker to improve NoC security. Figure 2.6 [6] shows the available paths between A,B and C,E using XY/YX and west-first routing respectively. However, all the routing algorithms implemented in [6] are only partially adaptive and therefore, vulnerable to

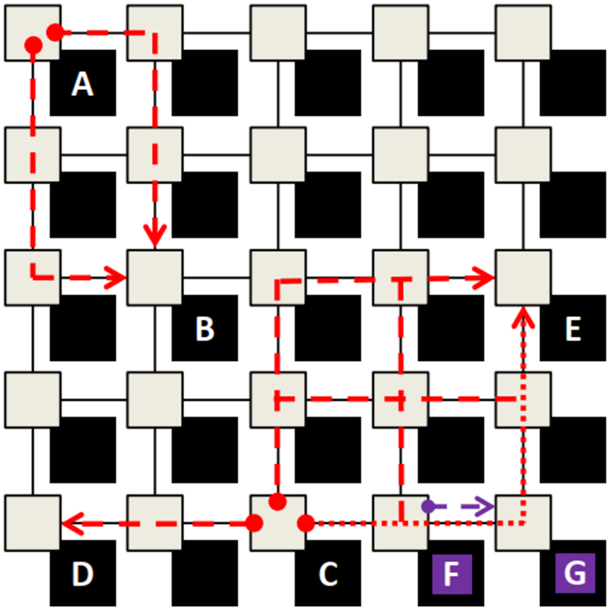


Figure 2.6: Pseudo adaptive (XY/YX (A,B) and west first (C,E)) routing for NoC [6].

an attacker exploring the routing paths. In [112], authors proposed Non-Interference Based adaptive (NIBR) Routing to secure NoCs from side channel and DoS attacks. Both 1-D and 2-D interference were considered to avoid timing channel attacks. However, the proposed NIBR is a hybrid routing mechanism mainly based on Dimension-ordered Routing, (DOR) and requires additional hardware overhead to make the routing decision. Moreover, being a priority-based routing mechanism, it might not be efficient to separate traffic flowing from applications having similar priority. In [113], a risk-aware NoC routing has been proposed for sensitive traffic. The risk of the path is evaluated at the destination interface. If the measured risk is above the pre-determined threshold, a new low-risk path is explored. However, all the four routing algorithms considered in the paper are constrained by the minimal path and hence restrict the search of low-risk paths. In [114] an overview of the reverse engineering attacks, their countermeasures have been discussed.

Chapter 3

A One-to-many Traffic-Aware WiNiP Architecture for Multichip Systems

Platform-based computing modules such as embedded systems and micro-servers are MCMC systems with in-package memory and processing chips. Such systems consist of both one-to-one (unicast) and one-to-many (broadcast/multicast) traffic patterns. State-of-the-art wired interconnection architectures such as NoC are specially designed to handle on-chip unicast traffic and cannot mask the high off-chip communication latency caused by the chip-to-chip I/Os. Moreover, with the increase in memory-intensive applications and hence one-to-many traffic in a multichip system, this scenario gets even worse as conventionally one-to-many traffic is handled as multiple unicast traffic in a multihop NoC infrastructure. Consequently, a small proportion of such one-to-many traffic increases energy consumption and message latency significantly for chip-to-chip communication. Therefore, to support such one-to-many traffic, these MCMC systems with in-package memory need one-to-many traffic-aware interconnection infrastructure. To address these issues, the design of one-to-many traffic-aware WiNiP architecture that utilizes a novel asymmetric WiNiP topology and a traffic-aware MAC mechanism has been discussed in this chapter. With cycle-accurate simulations, it has been demonstrated that the proposed WiNiP architecture reduces the energy consumption and latency up to 46.96% and 47.08% respectively for multichip data transfer compared to state-of-the-art wired NiPs for application-specific traffic.

3.1 Motivation

High-performance processors in computing nodes such as blade servers and embedded systems have already undergone a massive shift from traditional SCSC architecture to the MCSC paradigm. NoC has emerged as a scalable, modular interconnection architecture for such large parallel architectures [21] with hundreds of cores. However, the scalability of NoC-based multicore processors is limited as the number of cores continues to increase in a single chip. This is because, for a large single chip, different factors such as sub-wavelength lithography, line edge roughness, and random dopant fluctuation can cause a wide process variation, which can result in higher fault density and hence, lower manufacturing yield. Therefore, the disintegration of large and complex multicore processors into smaller chips [25] is used to alleviate the effect of higher fault densities in advanced technology nodes and thus reduces manufacturing cost per die. An example of such a MCMC processing platform is the AMD EPYC series released in 2017 [27]. The EPYC Threadripper processor node is available as a 4-chip SiP with 8 cores in each chip, fabricated in 14 nm lithography technology.

However, each chip in the SiP needs an efficient intra-chip as well as an inter-chip interconnection network as disintegration increases inter-chip traffic significantly. Therefore, the performance of the MCMC system is mostly limited by the high latency and power-hungry off-chip I/Os. Conventionally, C4 bumps coupled with in-package transmission lines or flip-chip packaging is used to interconnect chips within a multichip system [29]. However, signal quality deteriorations due to micro-wave effects, crosstalk coupling, and frequency-dependent line losses in the transmission line limit the number of concurrent, high-speed inter-chip I/O and hence chip-to-chip bandwidth. Therefore, to efficiently handle higher off-chip traffic resulting from disintegration, an in-package interconnection network called NiP needs to be designed that can mask the high latency and power consumption of off-chip I/Os in multichip systems.

Moreover, in such SiP with an in-package memory module, traffic pattern not only consists of the core to core unicast (one-to-one) messages but also significant memory to core multicast/broadcast messages. Directory or snooping based cache coherency protocols in a shared memory SiP use multicast to invalidate a shared cache block or broadcast the updated block for the requesting cores. 3.1 shows the number of multicast messages injected per one million instruction for MESI and AMD Hyper Transport cache coherence protocol in a 64 core system with 64 KB, 2-way L1 cache (ID) and 512 KB of shared, 8-way L2 cache for PARSEC and SPLASH2 benchmark traffic patterns [42]. However, the number of destinations for each multicast message varies from 5 to 27 in such multicore systems, resulting in a significant increase in network traffic [42]. These coherence transactions also increase with system size scaling. Moreover, many control signals such as passing global states, power gating, and barrier synchronization require multicast/broadcast messages to be sent efficiently through the communication network, NiP. Widely used parallel programming commands such as MPIallgather, MPIbcast, MPI allreduce, etc use multicast and broad-cast. Finally, novel computing paradigms such as spiking neural networks [43], genetic algorithm-based computation could be also multicast-driven.

Traditional NiP architectures are multihop in nature and provide very limited support for multicast/broadcast as they handle those traffics either as repeated unicast messages or follow a tree-based replication strategy. However, repeated unicast-based multicast/broadcast protocol introduces local congestion at the source node, global congestion in the network, and large power overhead. This is especially true for MCMC systems where traffic needs to go through off-chip I/Os. On the other hand, tree-based routing requires additional circuit/control overhead to replicate the message inside the network and hence leads to a significant delay, area, and power overhead. Therefore, a one-to-many traffic-aware NiP interconnection architecture is required to overcome the performance bottleneck introduced by off-chip I/Os. Various novel interconnect technologies such as vertically integrated 3D

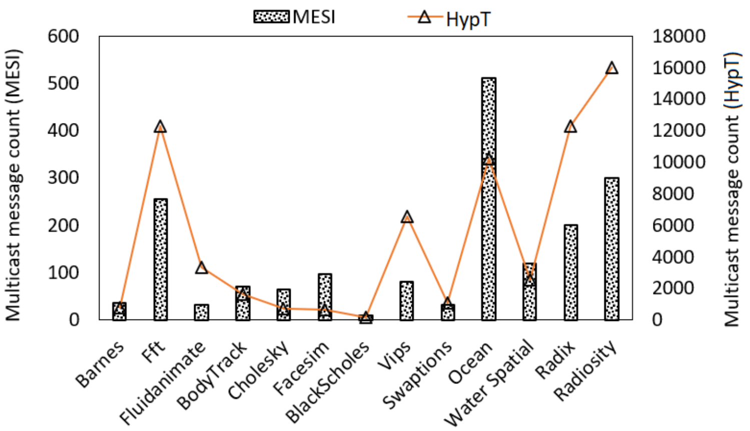


Figure 3.1: Number of multicast messages per 1M instructions in application specific-traffic.

integration [31], photonic interconnects [2], inductive or capacitive coupling based interconnects [115], utilization of metal layers in the interposer [34], and wireless interconnects [3], [116], [83] are being explored to mitigate the performance issues of conventional I/O based multichip systems.

Research in recent years has demonstrated that intra- and inter-chip wireless interconnects are capable of establishing radio communications within as well as be-tween multiple chips [3], [54]. Wireless data communication links with integrated transceivers, with multi GHz bandwidths in mm-wave bands, are fabricated and demonstrated [51], [117]. In this chapter, we present the design of a traffic-aware WiNiP architecture with novel asymmetric topology, flow control, and a hybrid wireless MAC that can handle concurrent unicast, hotspot as well as multicast/broadcast messages. The novel, reconfigurable multicast/broadcast aware MAC, asymmetric topology, and flow control exploits each other to enhance the multicast/broadcast performance of the system. Through cycle-accurate simulations, it has been demonstrated that the proposed WiNiP architecture out-performs other state-of-the-art NiP architectures under synthetic as well as application-specific workloads. Hence, the contributions of the design are

- A novel asymmetric WiNiP architecture that can fully utilize the benefits provided by the proposed MAC and increase system performance in terms of peak achievable bandwidth per core and energy consumption per packet.
- A novel, starvation free, dynamically reconfigurable MAC, capable of handling unicast, and multicast/broadcast traffic
- A back pressure and one-to-many traffic-aware flow control for such WiNiP architecture.
- To study and compare the performance and energy benefits of such architecture with other state-of-the-art NiP architectures.

The rest of the chapter is organized as follows. Section 3.2 describes the proposed novel asymmetric topology, traffic-aware re-configurable MAC, and the overall flow control. Section 3.3 briefly describes the physical layer. The evaluation methodology, results, and implementation overheads are described in Section 3.4. Section 3.5 concludes the chapter.

3.2 Proposed WiNiP Architecture

The proposed one-to-many traffic-aware WiNiP architecture comprises of the asymmetric topology, traffic-aware MAC and flow control. In this section the fundamental components of the WiNiP architecture will be discussed.

3.2.1 Asymmetric Topology

In this section, the proposed asymmetric topology of the MCMC interconnection architecture will be discussed. As current and future applications are memory intensive, a system with multiple multicore processors along with in-package memory modules was considered. Each memory module is considered to be a stacked DRAM mounted on top of a base logic die. Such stacked RAM subsystems are becoming the solution for high-density memory technology [118]. The layers of the memory stacks are interconnected using TSVs mounted on top of the base logic die and to enable memory-core communication, this logic die is equipped with a WI as shown in Figure 3.2. Although each memory module is in close proximity to its adjacent chip, it can be accessed by any core of other chips through the wireless interconnects and thus represents a globally shared memory system.

Each tile in the multicore chips is composed of a processing core, a switch, L1 private cache, and a distributed shared LLC. In such multicore systems, the placement of the memory controller impacts the performance by ensuring low memory access latency between tiles and the memory subsystem. In [119], different memory controller placements, and their performance have been evaluated. Though optimal memory controller placement can be found through exhaustive simulation for specific systems, it is shown in [119] that a cluster of 4 processors with one memory controller provides, the least average hop count and uniformity of processor to memory traffic for small chips having less than 8x8 tiles. Therefore, we adopt this cluster-based memory controller placement where the tiles are connected with the memory controller in a regular mesh. Having a mesh topology rather than a star topology increases the scalability of the cluster if necessary and increases the reliability.

To provide fast multi-gigabit direct connections between all the memory controllers and the memory banks, each memory controller was equipped with a switch along with a WI having both transmitter and receiver modules and call it transmitter-receiver WI (TR-WI) as shown in Figure 3.2. These TR-WIs are not only used to establish a memory controller to memory communication but also to enable the cluster to cluster communication both inside and outside of the chip. Packet communication inside a cluster utilizes the intra-cluster mesh links whereas, tiles communicate through the TR-WIs for communication with cores outside the cluster. The routing algorithm required to enable this communication is explained in Section 3.2.4.

Each tile within a cluster was also equipped with a WI that has only the receiver module in it and hence call it a receiver only WI (R-WI). An R-WI serves the purposes of low latency multicast/broadcast message reception with a small area and power consumption compared to a TR-WI. The R-WIs are normally off during any unicast message transmission and only turned on during multicast/broadcast transmission based on a $\overline{SLEEP}$/AWAKE signal as discussed in Section 3.2.3. Having R-WI in each tile of the cluster node leverages the proposed MAC to support the efficient reception of multicast/broadcast messages by utilizing single-hop wireless communication. In addition to that, selective power on-off mechanisms for WIs provides higher power efficiency as shown in Section 3.4.3. Deploying R-WIs in every tile can result is significant area overheads. However, modern mm-wave transceivers and antennas have small form factors of a fraction of a square millimeter making the area overheads worth

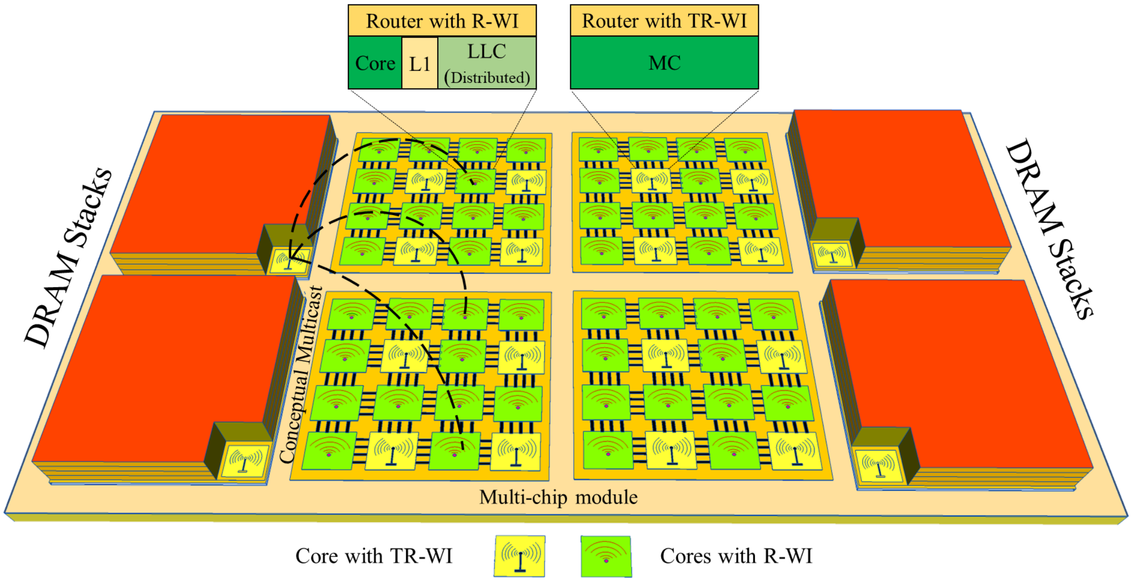


Figure 3.2: Proposed asymmetric WiNiP topology.

the improvement in performance. As some tiles in the proposed topology have TR-WIs and some have R-WIs, the proposed topology is therefore, called an asymmetric WiNiP topology.

3.2.2 Proposed Adaptive, Starvation-free Wireless MAC with Unicast and Multicast Support

For the one-to-many traffic-aware WiNiP, a hybrid adaptive MAC that can efficiently handle unicast, hotspot as well as multicast/broadcast traffic patterns has been proposed. The MAC is described in the following paragraphs.

MAC for Unicast/Hotspot Traffic

Token [3], [60] or CSMA [79] based TDMA and FDMA [66] based MAC limits the number of concurrent communicating pairs to one or only a few due to resource-limitations of an integrated transceiver. To provide concurrent communication links between multiple pairs of nodes in a WiNoC without requiring transceivers tuned to various non-overlapping frequencies, a CDMA based MAC was proposed in [65]. Due to the concurrent communication and higher channel utilization, the CDMA based interconnection provided better performance compared to token-based TDMA. Therefore, to handle one-to-one (uniform random) and many-to-few (hotspot) traffic, we consider a simplified code-division multiplexing scheme so that concurrent transmission can happen between multiple communicating pairs.

Moreover, to aid the many-to-few traffic, the T (transmitter)-protocol [120] was used in which each transmitter encodes the data according to a unique code. At the receiver, the received signal is correlated with all the code words to decode the transmissions from each transmitter. So, the T-protocol can operate normally under one-to-one or many-to-few type of traffic scenarios where each receiver can receive data from multiple transmitters at the same time on different code-channels specific to each transmitter. For this work, we choose Walsh-Hadamard codes as they have short length there-by minimally affecting the data rate on each wireless link. Walsh codes are not robust against loss of synchronization and can cause inter-channel interference when synchronization is lost between multiple transmitters. However, in [65] it is shown that the Signal to Interference and Noise Ratio (SINR) results in an acceptable Bit Error Rate (BER) of less than 10^{-12} in on-chip environments even with the loss of synchronization among multiple transmitters. Therefore, CDMA communication channel model was adopted from [120].

MAC for One-to-many Traffic

To provide low latency and faster transmission of the one-to-many (multicast/broadcast) messages, a TR-WI with such traffic switches the MAC protocol to a Distributed Arbitration-based Full Channel Access (DAFCA) scheme that acquires the whole wireless channel bandwidth during multicast/ broadcast communication. Figure. 3.3 briefly describes how the proposed MAC switches from CDMA to DAFCA and vice versa for multicast and unicast transmission respectively. As shown in Figure. 3.3, to initiate a multicast/broadcast communication, the sending TR-WI sends a multicast/broadcast request to all the other TR-WIs using its CDMA channel. The request for the multicast/broadcast message is sent as a header flit of the multicast/broadcast packet along with information regarding the sender, flit type, and the length of the message. Upon reception of the header, the TR-WIs decodes the header to identify if it is a multicast/broadcast request.

These requests are then sent to the Multicast Queue (MQ) as shown in Figure. 3.4. As multiple multicast/broadcast requests can originate at the same time, every TR-WI is equipped with an arbiter that grants the whole channel to the requesting TR-WIs. In DAFCA mode, starvation in granting whole channel access to any TR-WI with multicast/broadcast messages is avoided by granting next access to the TR-WI in the MQ, which is least recently

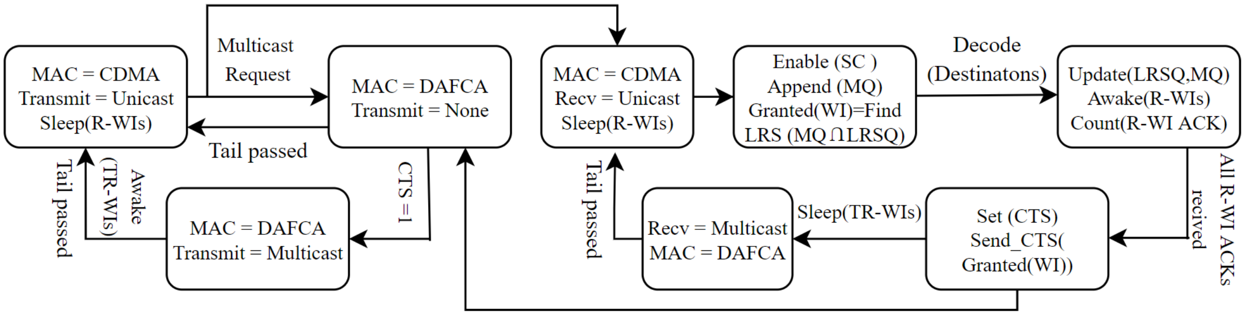


Figure 3.3: Simplified flow diagram for the proposed MAC and flow control.

serviced among the queued requesters. The Least Recently Served Queue (LRSQ) maintains the list of all TR-WIs that received whole channel access for multicast/broadcast messages where the most recently served TR-WI is enqueued at the top. Both the queues are updated after each multicast transmission in DAFCA mode as described in the next paragraph.

The LRSQ is initialized with all the TR-WI entries in order of their address. However, each entry in the LRSQ is updated using a counter-based policy once the channel access is granted to a TR-WI requesting multicast transmission. The old position of the transmitting TR-WI in LRSQ is removed and updated to the top of LRSQ. Any TR-WI entry above the old position is shifted one position down in the queue. Any TR-WIs below it is kept in the same position while updating LRSQ. The arbitration logic services the first TR-WI that is in MQ and holds the oldest served entry in the LRSQ. The served TR-WI and the corresponding request is removed from MQ by updating a tag indicating the invalid request.

For example, in Figure. 3.4(a), WI-3, being the bottom most entry in LRSQ, holds the Least Recently Served (LRS) position. The arbitration logic finds the LRS node from MQ LRSQ and therefore grants the next access to WI-3. Meanwhile, the LRSQ is updated by putting WI-3 as top entry indicating it as the Most Recently Service (MRS) node as shown in Figure. 3.4(b). MQ is also updated by removing the particular WI-3 request from the queue. Similarly, after WI-3, WI-1 will get the next channel access in DAFCA mode as it is the LRS node in the LRSQ and also present in MQ. As every TR-WI maintains the same MQ and LRSQ, therefore, each of them will grant access to the same requesting node even for multiple concurrent multicast requests and thus avoid channel contention using this distributed arbitration mechanism. Figure. 3.5(a) shows different functional blocks of the proposed MAC including LRSQ and MQ. Next, the proposed flow control mechanism to

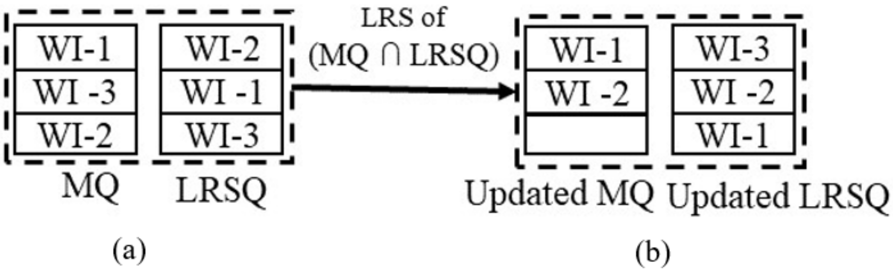
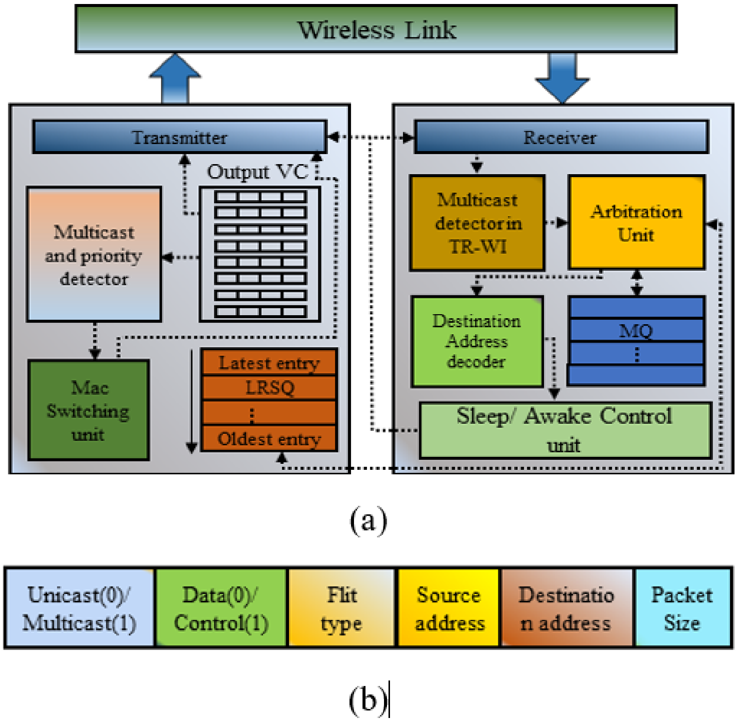


Figure 3.4: MQ and LRSQ entries (a) Before (b) After channel access.

support this hybrid MAC will be discussed.

3.2.3 Proposed Wireless Flow Control

The multicast/broadcast message is requested through the packet header. A field in the header of the multicast/broadcast packet as shown in Figure. 3.5(b) indicates that it is a multicast/broadcast packet. The type of any message is determined by the data/control field of the header. Moreover, the flit type (header, body, tail) detects the type of the flit to be transmitted. The header also has the source and destination addresses to ensure the appropriate R-WIs are awake. As it has been shown in [42] that the average and the maximum number of destinations for a 64 core system is 14 and 27 respectively and it increases with system size scaling, using an N bit destination address field for multicast messages reduce the header flit size significantly. In an N bit destination address, the Nth bit represents whether or not the Nth core is a destination. For unicast, we use the conventional



(a)

(b)

Figure 3.5: (a) Block diagram of the MAC unit (b) Header flit structure.

M bit address which is the most efficient way to represent each of the 2^M destinations for such messages.

One critical case might arise after receiving the multicast/broadcast header. The broadcast/multicast request might reach the TR-WI while it is in the middle of a unicast communication with other TR-WIs. In such a scenario, we prioritize the broadcast/multicast transmission to ensure the fast transmission of such traffic and propose a flow control called Store-and-Continue (SC) that can re-initiate the unicast communication at flit level granularity. In the proposed SC flow control, each output VC requires an additional register, R_{add} of the same size as that of a flit. The header flit of a unicast packet is temporarily stored in that register till the tail flit passes through the VC. When a TR-WI receives any broadcast/multicast message request during any ongoing unicast transmission, the ongoing unicast communication stops immediately once the current flit transmission is complete. The MAC switches to the DAFCA mode for the duration of the multicast/broadcast after which, the unicast transmissions resume by following the below process.

In order to resume the unicast message transfers, the output VCs use the previously stored header from R_{add}. The output wireless port of a sending TR-WI uses the same VC arbitration logic as that of the input VC arbiter in the input port of the receiving TR-WI. We adopt the VC allocation arbiter from [22]. Therefore, a packet in the output port of the sending WI is mapped to the same input VC of the receiving TR-WI as long as we have the header flit available in the output VC of the sending TR-WIs. This mirror arbiter in the output VCs of the TR-WIs is necessary as VC information cannot be back-propagated over wireless channels to enable traditional flow control available in wired NoCs. We consider 8 VCs for each output port with a unicast flit size of 64 bits which results in 8 R_{add} registers per TR-WI node for storing the header flit.

To enable backpressure flow control in the CDMA unicast messages, whenever a VC in the input buffers of a downstream WI is full or empty, the information is packetized into a small packet of a few bits and sent up-stream along the CDMA uplink. Any ongoing unicast data transmission will be halted using the above-mentioned SC protocol immediately before the VC overflow and the control packet will be prioritized to be sent upstream using the CDMA link between the downstream TR-WI. The flow control in the wireline links will follow the conventional backpressure flow control.

After receiving the multicast/broadcast request, the TR-WIs append the request to the MQ, and based on the entries in the LRSQ as discussed in Section 3.2.2, it grants the whole channel access to a particular TR-WI. The decoder logic determines the destination R-WIs from the header of the multicast/broadcast request packet and asserts an AWAKE signal to those R-WIs. On the other hand, after receiving the AWAKE signal each R-WI receiver sends back its acknowledgment to the TR-WI in its cluster using wired links of the control bus. Once all the destination R-WIs in a cluster acknowledge, each TR-WI sends the Clear to Send (CTS) signal to the arbitrated (granted) multicast/broadcast requesting node through their respective CDMA channel and goes into $\overline{SLEEP}$ mode itself. The CTS signal from all the destination TR-WIs enables the sender to switch the MAC mode to DAFCA to start the broadcast/multicast transmission as shown in Figure. 3.3 To switch back to the CDMA mode, the R-WIs monitor the multicast message for tail flit and the MQ queue.

If there is no other broadcast/multicast request in the MQ and tail flit of the transmitting multicast message has passed, then the R-WIs assert an AWAKE signal for the TR-WI and put themselves into $\overline{SLEEP}$ mode. With the all TR-WIs in AWAKE state, the system restores the CDMA mode. The $\overline{SLEEP}$/AWAKE mechanism of the WIs causes a latency overhead around 4-10 cycles [121] and have been considered in simulations. The broadcast transmitting node also monitors the multicast message for tail flit and the MQ. It changes the transmission mode to CDMA once it is done with all flit transmissions and if there is no request remaining in the MQ. However, if there is no recipient in a cluster, the TR-WI of that cluster wakes up based on a counter that counts up to the "packet size" defined in the header once the multicast transmission began.

Moreover, to avoid starvation for unicast messages, the number of consecutively served multicast requests was restricted to 8. This number represents the maximum number of multicast messages that can be present in such MCMC systems for a span of 10 cycles and obtained by analyzing the traffic traces generated by SynFull [122] for PARSEC and SPLASH2 benchmark suites used in Section 3.4.8. However, this is a design knob that affects MQ and VC implementation overheads and can be statically or dynamically optimized depending on the percentage of one-to-many messages in the system. The overall flow control of the proposed WiNiP architecture is summarized in Figure. 3.3 with a simplified flow diagram showing the MAC switching and on-off phases of R-WIs and TR-WIs.

3.2.4 Routing Protocol for Wireless Architectures

A multicast/broadcast message originating in any core and having destinations outside of its cluster is first routed to the TR-WI in its cluster. Then, as discussed in the previous section, the TR-WI requests access to the wireless channel for multicast/broadcast transmission. Upon acquiring the channel, the TR-WI transmits the packet to the R-WIs which are put in the wake mode by their respective TR-WIs. The proposed architecture requires control to turn on the receivers using $\overline{SLEEP}$/AWAKE signal during multicast/broadcast communication. In Section 3.4.9 the overheads for these mechanisms are shown to be negligible due to a simple logical implementation. In case there are multicast destinations within the same cluster, the message is replicated and copies are routed to the tiles within the cluster using the mesh-based wired links adopting X-Y dimension order routing.

In case of unicast packets, the originating core sends the packet to the TR-WI in its cluster, which then sends it to the TR-WI of the destination cluster using its transmitter code channel as discussed in the previous subsection. Upon receiving the packet at the destination TR-WI, it is routed to the final destination core in the cluster. To support the unicast message transmission within a cluster, we use the X-Y dimension order routing. As we consider each cluster to be a mesh connected network, such dimension order routing avoids deadlock without degrading the performance. Deadlock is avoided in this proposed routing mechanism as X-Y routing is adopted for message routing inside a cluster, which is known to be deadlock-free. Moreover, once a message leaves a cluster through the TR-WI it can never be addressed back to that cluster as all intra-cluster messages are rout-ed using the mesh-based links without engaging the TR-WI and the multicast messages with destinations inside the same cluster are split from the inter-cluster multicast message as discussed earlier. As a message can never return to the same cluster from where it is transmitted deadlock is avoided for the inter-cluster messages as well [123].

3.3 Physical Layer

Non-coherent OOK modulation allows relatively simple and low-power circuit implementation without power-hungry carrier recovery circuits like PLL. In addition to the OOK modulator and demodulator, a CDMA encoder and decoder are also designed specifically suitable for the OOK modulation. In the transmitter, the data bitstream is first encoded by XORing with the code specific for that channel and then modulated with a 60 GHz carrier generated from a Volt-age Control Oscillator (VCO) and amplified by a Drive Amplifier (DA) before being coupled to the on-die antenna as shown in Figure. 3.6(a). The wireless channel is assumed to be an additive multipath channel. This implies that individual transmissions encoded into different codes are added over the channel. At the receivers, the received signal is amplified by a Low Noise Amplifier (LNA). Then this signal is sent to an Envelope Detector (ED), which eliminates the carrier frequency. This signal is then amplified by a baseband amplifier (BA). In the R-WIs, as the whole channel is used for multicast this OOK receiver is sufficient for the R-WIs as shown in the grey box in Figure. 3.6(b).

In the TR-WIs, due to unicast messages and the adopted T-protocol for the CDMA based MAC, the receiver needs to have CDMA decoders for every code-channel. Therefore, in the

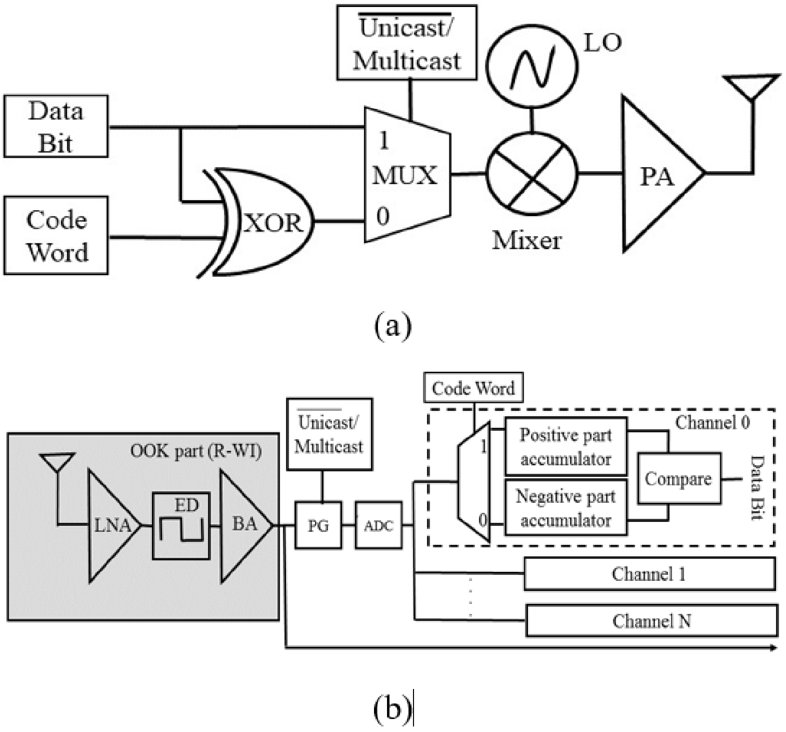


Figure 3.6: (a) Transmitter (b) TR-WI receiver block diagram with only R-WI receiver in grey box.

receivers of the TR-WIs, the output of the OOK demodulator is further sent to a CDMA receiver during unicast transmission. An Analog to Digital Converter (ADC) converts the received envelope from the additive multipath channel. Then, the signal is correlated with each codeword from the codebook to create separate receive channels corresponding to every codeword as shown in Figure 3.6(b). The code words being only in ‘0’ or ‘1’ enables us to adopt a digital correlator receiver that accumulates and compares the positive and negative part of the received symbols to compute the received digit for each channel [120] as shown in Figure 3.6(b). As each transmitter uses its predetermined specific code, a single receiver can receive transmissions from multiple transmitters at the same time forming a SWMR. This also provides support for hotspot or many-to-few kinds of unicast messages.

The R-WIs only has an LNA and an envelope detector [124] to only receive simply OOK modulated signals re-ceived via multicast or broadcast in DAFCA mode. The CDMA receiver is Power Gated (PG) during DAFCA mode as the entire channel is acquired by the transmitter for such messages. For transmitting such messages, the transmitter bypasses the CDMA encoder and simply modulates, amplifies, and broadcasts the message over the antenna. The on-die antenna is adopted from a design for similar multichip environments in [3]. Circuit level designs and implementation of the modulator, demodulator are outside the scope of this paper and have been already demonstrated to perform in on and off-chip realizations with bit error rates of less than 10^{-12} considering thermal noise at the receiver while consuming 23.6mW peak power with an active footprint of 0.12 mm^2 [124], [125].

3.4 Experimental Results and Analysis

In this section, the performance and energy consumption of the proposed WiNiP interconnection architecture will be evaluated in terms of peak achievable bandwidth per core, average packet energy consumption, average packet latency, and average Energy Delay Product (EDP). The peak achievable bandwidth per core is measured as the maximum sustainable data rate in number of bits successfully routed per core per second at network saturation with maximum injection load. Average packet energy is the energy consumed to transfer an entire packet from source to destination in the multichip system on an average. Average packet latency is the average number of clock cycles required between packet injection at

Table 3.1: Area, delay, and power overhead of the MAC Unit

Metric	**Multicast Detector Unit**	**MAC Switching Unit**	**Arbitration Unit**	**CDMA Decoder Unit**	**$\overline{\text{Sleep}}$/Awake Unit**	**Total**
Power(mW)	0.103	0.050	0.045	0.101	0.117	0.416
Area(um^2)	33.48	13.68	15.4	27.36	48.24	138.1
Delay (ns)	0.272	0.122	0.220	0.122	0.266	1.002

the source and absorption at the sink. The average packet latency involves both multicast and unicast packet latency. For multicast/broadcast packets, packet latency is defined as the average number of clock cycles required to absorb a packet in all destinations. EDP is measured by the multiplying average packet latency and average packet energy.

The in-package memory module was considered to be a High Bandwidth Memory (HBM) consisting of 4-layered vertically stacked DRAM mounted on top of a base logic die [118]. Each memory stack is assumed to have four channels. The base logic die works as an interface between the memory stacks and multicore chips. In this section, first the system will be evaluated for various types of synthetic traffic as well as with real application-based traffic. Synthetic traffic is used to illustrate the behavior of the system under various specific conditions followed by real application-based traffic.

3.4.1 Simulation Setup

The simulation of wireless interconnection requires the amalgamation of multiple simulation tools that are described here. Interconnection network switches are designed using Register Transfer Level (RTL)-level designs and extracting post-synthesis performance parameters using Application-Specific IC (ASIC) design flows such as Synopsys Design Compiler (DC) using 65 nm standard cell libraries from Chip Multi Projects (https://mycmp.fr/). The characteristics of the digital MAC units are obtained from post-synthesis RTL models as shown in Table 3.1. The delay and energy dissipation on the intra-chip wireline links is obtained through Cadence simulations considering the specific lengths of each link based on the NoC topology in each multicore die. The characteristics of the antenna, transceiver circuits are simulated in High-Frequency Structural Simulator (HFSS) [126] and Cadence Virtuoso respectively, and were adopted from [124], [125]. Table 3.2 shows the parameters used for the simulations.

The characteristics of the antennas, transceivers, routers, and wired links are annotated into a system-level, cycle-accurate simulator based on NOXIM [127]. However, NOXIM does not have the inherent capability to simulate MCMC systems and complex wireless interconnect architectures. Therefore, additional code has been plugged in to modify it to the necessary interconnect simulator and evaluated the performance of the proposed WiNiP system. Figure. 3.7 captures the interaction between simulators at various levels of abstraction. The proposed architecture was characterized with synthetic traffic as well as evaluate it with application-specific traffic patterns. As shown in Figure. 3.7, the synthetic traffic used in these experiments is uniform random spatial distribution and directly fed into the simulator. However, traffic traces for each of the PARSEC and SPLASH2 benchmark suite were generated using SynFull [122] and then the traffic traces were used in the cycle-accurate simulator for performance evaluation.

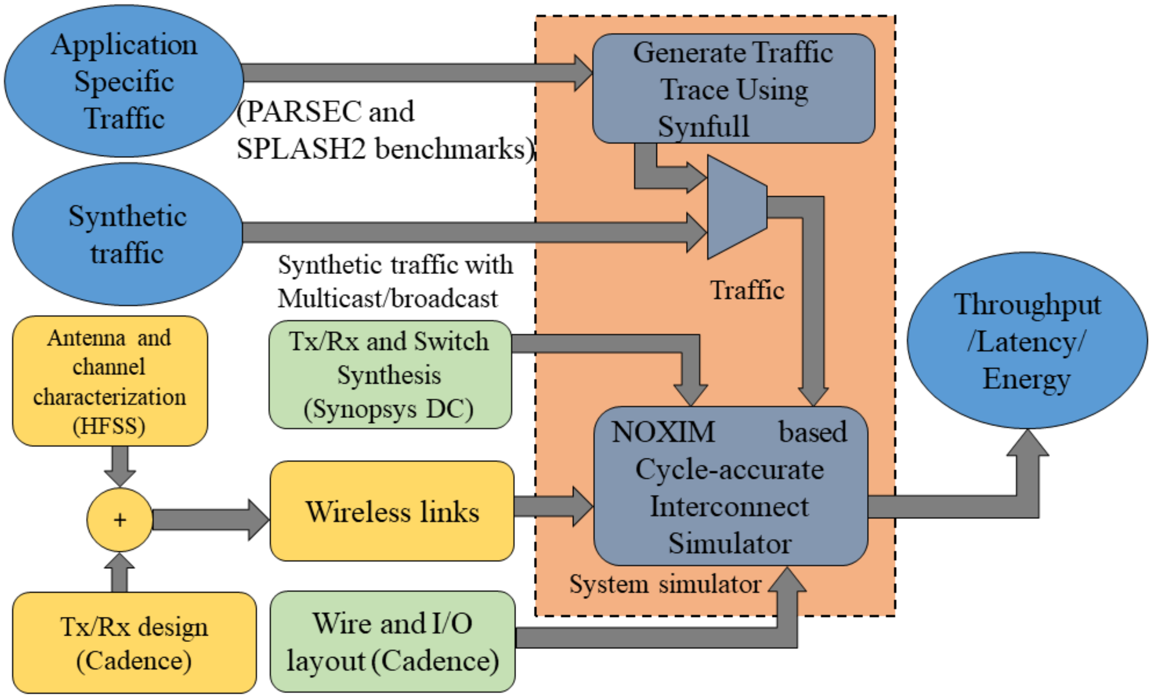

Application Specific Traffic
(PARSEC and SPLASH2 benchmarks)
Generate Traffic Trace Using Synfull
Synthetic traffic
Synthetic traffic with Multicast/broadcast
Traffic
Antenna and channel characterization (HFSS)
Tx/Rx and Switch Synthesis (Synopsys DC)
Throughput /Latency/ Energy
NOXIM based Cycle-accurate Interconnect Simulator
+
Wireless links
System simulator
Tx/Rx design (Cadence)
Wire and I/O layout (Cadence)

Figure 3.7: Overview of the simulation process.

Table 3.2: Component configuration for simulations

Components	**Configuration**
NoC Router	3 stage pipelined [37], 5 ports (except wireless)
Total VC	8, Each 4 flits deep
Flit width	64 bits
Wired NoC links	64-bit flits, single cycle latency, 0.2pJ/bit/mm
Wired Interposer links	64-bit flits, single cycle latency, 0.2pJ/bit/mm
OOK Wireless transceiver	16Gbps, 2.03pJ/bit, OOK modulated at 60GHz
CDMA encoder, decoder and ADC	16Gbps, 0.66pJ/bit, OOK modulated with ADC and CDMA decoder [33] at 60GHz
HBM link	128Gbps, 6.5pJ/bit [30]
Technology node	65nm, 1V supply, 1 GHz system clock

3.4.2 Architectures for Comparison

Several configurations for the MCMC systems were considered for the comparative performance evaluation. In all cases, a 4 chip system was considered with 16 processors arranged in a 4x4 grid, and the memory stacks are considered to be mounted on both sides of the processing chip array. Table. 3.3 lists the architectures and their properties.

The Asymmetric Token-only WiNiP (ATWiNiP), Asymmetric CDMA-only WiNiP (ACWiNiP), and Symmetric WiNiP (SWiNiP) architectures are used to evaluate the performance of the proposed hybrid MAC and the asymmetric architecture separately. On the other hand, the BDXNiP [25] and ONWiNiP [4] represents the state-of-the-art wired and wireless architecture to be compared with the proposed Multicast-Aware WiNiP (MAWiNiP) architecture. The BDXNiP topology combines the different topological aspects of Butterfly and Folded Torus topology and extends the interposer to multiple chips. In BDXNiP architecture, cores in every 2x2 grid share an interposer router. The misaligned topology also offsets the location of the boundary interposer routers that connect the edge cores of two adjacent chips. In ONWiNiP architecture, broadcast traffic is transmitted using the wireless plane where the unicast traffic uses the wired path. The plane selection, blocking, and switching is controlled by a hybrid controller integrated into Network Interface (NIF). The proposed hybrid interconnection architecture uses a CSMA based MAC proposed in [128] and a wireless link speed of 64Gbps having 2pJ/bit energy efficiency. Though the topology can be configured in various ways, we use Ort_1^4 configuration as it closely matches with other architectures considered in this section. The Wired NiP (WNiP) is our baseline architecture which is an extended mesh topology connecting multiple chips.

Table 3.3: Architecture configurations for performance comparison

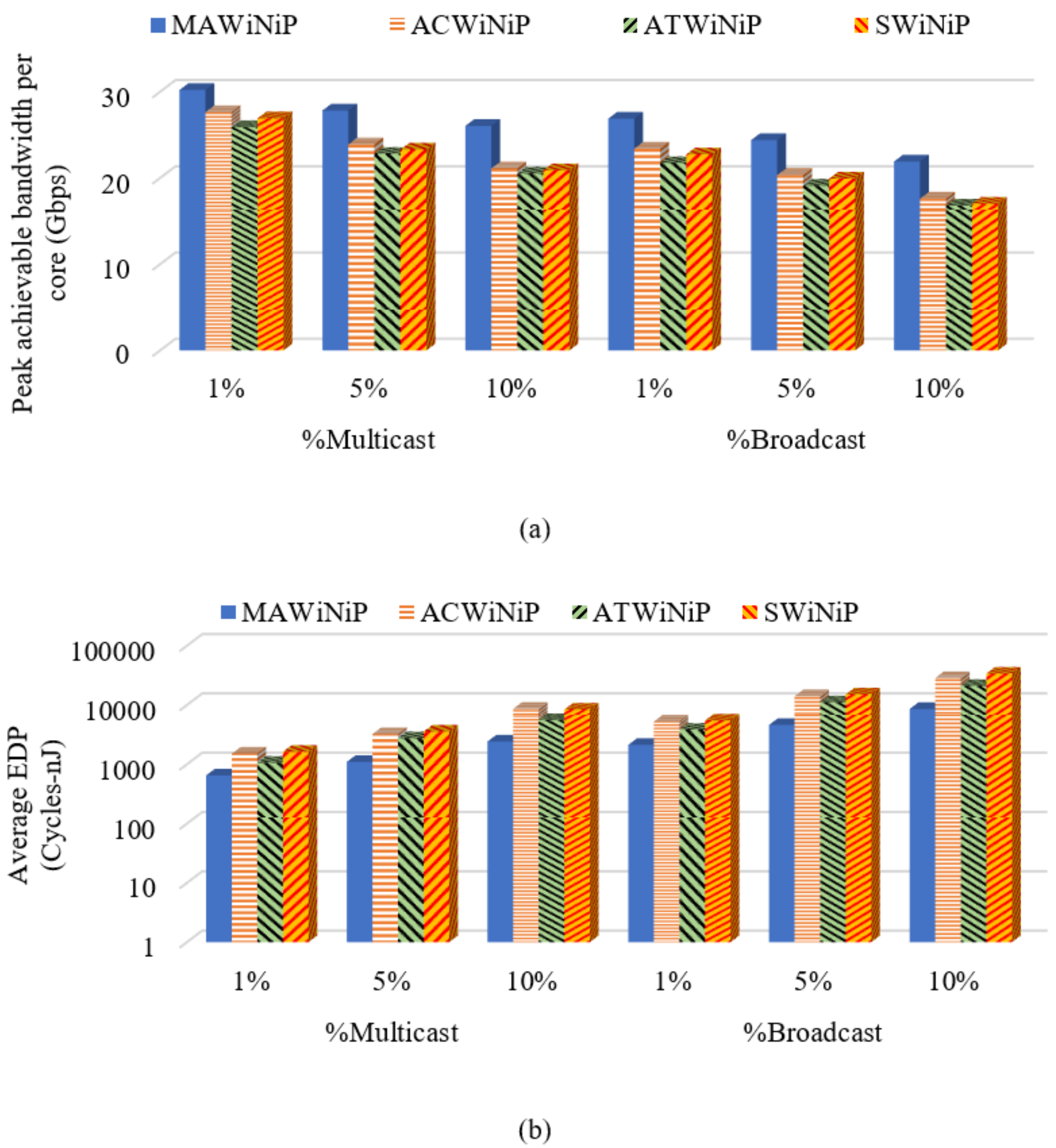
Architecture name	Intrachip wired topology	R-WIs	TR-WIs	MAC type	Interchip links
WNiP	Mesh	No	No	N/A	Wired
ATWiNiP	Mesh	Yes	Yes	Token Only	Wireless
ACWiNiP	Mesh	Yes	Yes	CDMA Only	Wireless
SWiNiP	Mesh	No	Yes	Proposed Hybrid MAC	Wireless
MAWiNiP	Mesh	Yes	Yes	Proposed Hybrid MAC	Wireless
BDXNiP	Butterfly+Folded torus	No	No	N/A	Active Interposer Links
ONWiNiP	Mesh	No	Yes	CSMA	Wired + Wireless

In the case of wireline configurations, the memory stacks are connected to the I/O modules of the processing chips through 4 channels each consisting of 128 bit (assuming µ-bump pitch of 50µm and 10mm die edge) [34] wide channel operating at 1GHz. Hence, this wide I/O provides a total bandwidth of 128Gbps per channel with its neigh-boring processing chip with an energy consumption of 6.5pJ/bit [118]. The 4 channels of the HBM memory are connected with this wide I/O to the adjacent 4 cores along the boundary of the adjacent chips. Table 3.2 shows bandwidth and energy consumption for wired and wireless links.

3.4.3 Performance Evaluation of the Proposed MAC and Topology with One-to-Many Traffic

In this section, the performance benefit of the proposed hybrid MAC and asymmetric topology of the proposed MAWiNiP architecture will be evaluated over other wireless multichip communication architectures discussed in Section 3.4.2. The one-to-many traffic pattern was synthetically modeled as multicast traffic originating primarily due to cache coherence and other control signals such as barrier synchronization. A certain percentage of the traffic is modeled as one-to-many (i.e. multicast), and the rest of the traffic is modeled as one-to-one unicast traffic. 30 cores were selected randomly as multicast destinations among all the multicore chips based on the findings of a study of multicast messages in such multicore systems [42]. For simulation, 1%, 5%, and 10% traffic packets originating from any core were considered to be multicast traffic along with 10 % core-memory traffic based on the traffic patterns investigated in [47]. The remaining traffic was modeled as one-to-one uniform random unicast traffic among cores in the system.

For this experiment, a 4 chip system with 4 in-package memory modules were considered where, each chip has 16 processing cores in it. The peak achievable bandwidth per core and average EDP for the wireless architectures under multicast traffic patterns are shown in Figure. 3.8. A maximum traffic load of 1 flit per core per cycle is used to observe the performance at maximum load. Among all the wireless architectures studied in Figure. 3.8, the proposed MAWiNiP interconnection architecture provides the highest peak achievable bandwidth and lowest EDP.



(a)

(b)

Figure 3.8: (a) Peak achievable bandwidth (b) Average EDP for different one-to-many traffic percentages.

ATWiNiP and ACWiNiP both being the asymmetric WiNiP topologies, the comparative results shown in Figure 3.8 reveals the performance benefit due to only the proposed hybrid MAC of the MAWiNiP architecture. The proposed multicast aware MAC prioritizes the one-to-many traffic. Switching to DAFCA mode during multicast transmission allows the multicast transmitter to capture the entire wireless channel bandwidth and along with support from the asymmetric topology, it ensures very low latency for multicast traffic. For unicast traffic, the proposed MAC exploits the multiple simultaneous wireless communication by switching the MAC to CDMA mode. Therefore, faster transmission of the one-to-many traffic in DAFCA mode coupled with the concurrent unicast transmission in CDMA mode increases the peak bandwidth with low EDP.

The ATWiNiP suffers from additional token round trip time while only one TR-WI has access to the medium at a particular point of time. In the ACWiNiP channel bandwidth is wasted during multicast traffic transmission as only one code-channel is used by each TR-WI. Therefore, both ATWiNiP and ACWiNiP provide lower performance compared to the hybrid MAC in MAWiNiP. To be accurate the proposed MAC of the MAWiNiP ar-chitecture achieves a bandwidth gain of 30.42% and 24.47% for 10% broadcast traffic compared to ATWiNiP and ACWiNiP architectures respectively.

The SWNiP architecture uses the proposed hybrid MAC but with a symmetric topology with no R-WIs in it. Therefore, the performance comparison of the MAWiNiP with the SWNiP presented in Figure 3.8 provides the performance gain only due to the asymmetric topology of the MAWiNiP architecture. The MAWiNiP architecture achieves 28.79% higher bandwidth and 74.75% lower EDP compared to SWiNiP architecture for 10% broadcast traffic. This is because, during multicast message trans-mission in DAFCA mode, the MAC can exploit the underlying asymmetrical topology by activating the R-WI of each multicast receiving node. Hence, it enables single-hop reception of the multicast flits in every destination node directly and reduces the overall network congestion, latency, and EDP.

In a nutshell, the proposed architecture enables faster and low latency transmission of the multicast messages that significantly reduces the network congestion and improves overall bandwidth and energy consumption by

- simultaneous packet transmission for unicast mes-sages.

- higher overall available bandwidth enabled by the proposed hybrid MAC in multicast transmission, and
- selective activation of the R-WI in each destination node during multicast ensuring single-hop multicast reception and low power consumption during only unicast message transfer.

As these one-to-many packets can be latency-critical, the average broadcast packet latency was also evaluated and compared for different broadcast traffic percentage with all the wireless architectures considered in this section. From Figure 3.9 it can be observed that the proposed MAWiNiP architecture has the lowest broadcast packet latency for different broadcast load compared to other wireless interconnection architectures. It can also sustain a higher volume of broadcast traffic compared to any other architecture. This is due to the symbiotic relationship among the asymmetric topology, broadcast aware MAC and the SC flow control, the MAWiNiP architecture prioritizes broadcast packets and grants immediate access to the entire channel to ensure low latency for broadcast traffic. The relative latency gain of the hybrid MAC and asymmetric topology can be measured separately by comparing the results of ATWiNiP, ACWiNiP, and SWiNiP respectively.

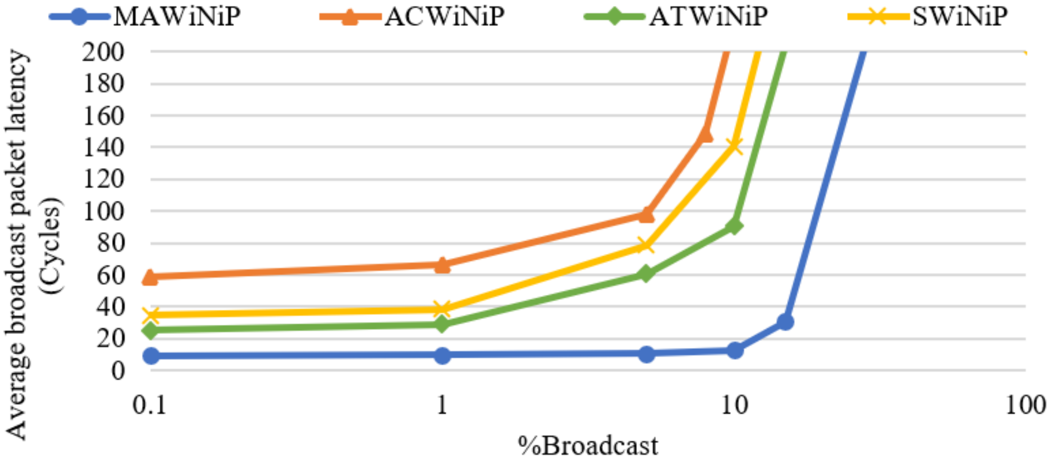


Figure 3.9: Average broadcast packet latency for different architectures with different broadcast traffic percentages.

3.4.4 Latency Comparison with State-of-the-art Architectures

Several state-of-the-art on-chip and multichip architectures have been proposed to handle one-to-many traffic resulting from cache coherence and system synchronization events. In this section, one-to-many traffic performance of our proposed MAWiNiP architecture with other state-of-the-art wired and wireless interconnection architectures will be compared in terms of average packet latency. As state-of-the-art wired architecture, we consider WNiP and BDXNiP architectures as described in Section 3.4.2 For wireless interconnection, we consider ONWiNiP as described in Section 3.4.2 and extend it for MCMC communication.

As one-to-many traffic affects the overall system latency, the average packet latency was considered as a performance metric. The considered architectures were evaluated for different injection load with 10% broadcast traffic and the rest of the traffic is considered to be uniform random. From Figure 3.10, it can be observed that the ONWiNiP inter-connection architecture outperforms the WNiP and BDXNiP wired architectures. This is because the ONWiNiP architecture provides a separate plane for broadcast communication and therefore the multicast/broadcast traffic does not have to contend for resources with unicast traffic which results in overall low latency for ONWiNiP architecture at low load. However, as the traffic load increases, most of the traffic is routed using the wired network to avoid wireless channel congestion which results in higher average packet latency. The BDXNiP wired architecture having less network diameter compared to WiNiP architecture provides lower

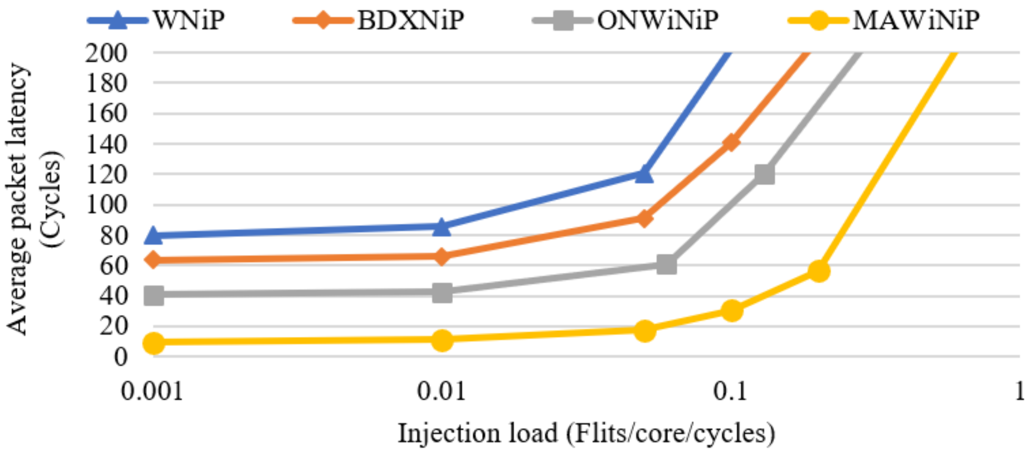


Figure 3.10: Average packet latency under various injection load.

latency than the WiNiP architecture. However, with increased congestion at higher injection load, the average broadcast packet latency increases. Finally, the proposed MAWiNiP architecture outperforms all the wired and wireless interconnection architecture in terms of average packet latency at different traffic loads. The one-to-many traffic prioritized by the pro-posed hybrid MAC, instant channel access provided by the SC flow control, and the underlying asymmetrical topology can efficiently support broadcast traffic. On the other hand, multiple simultaneous unicast transmission using CDMA mode enables the architecture to provide low latency even at a higher injection rate compared to other architectures.

The broadcast traffic percentages were also varied to measure the average packet latency gain of the MAWiNiP, BDXNiP, and ONWiNiP architectures with respect to the WNiP architectures. It is interesting to see from Figure 3.11 that at low broadcast traffic percentage, the ONWiNiP architecture has a lower speedup than BDXNiP. This is because, at a low broadcast percentage, as the unicast traffic uses the underlying wired interconnect, which is essentially a mesh for ONWiNiP architecture, the performance gain of using wireless links is masked by the high latency of the wired interconnect in the ONWiNiP compared with that of the BDXNiP. However, with the increase in broadcast traffic ONWiNiP outperforms the BDXWiNiP architecture due to the low latency separate wireless broadcast plane. MAWiNiP shows the highest latency speedup among all the architectures because of its broadcast awareness through the symbiotic relationship among MAC, topology, and flow control. However, at high broadcast load, as the network is already saturated the latency gain stabilizes

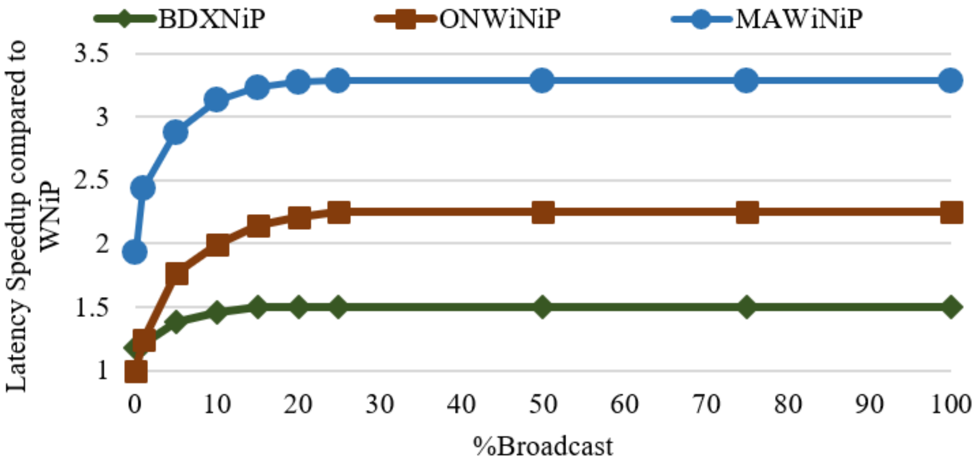


Figure 3.11: Average packet latency gain with higher broadcast traffic.

at a certain value for all the architectures.

3.4.5 Analysis of Unicast Message Latency

In a MCMC system, as most of the messages are unicast, it is essential to study the latency behavior of such messages in presence of one-to-many traffic. Therefore, in this section, the average unicast message latency of all the MCMC architectures considered in Sections 3.4.3 and 3.4.4, in presence of one-to-many traffic will be evaluated. For each MCMC system, 1%, 5%, and 10% broadcast traffic were considered, and the rest of the traffic was modeled as uniform random unicast traffic. Figure 3.12 shows the average unicast message latency of all the multichip architectures considered in this paper.

As unicast message queuing latency significantly increases with the increase in broadcast traffic percentage [88], for wired multichip architectures, such as WNiP and BDXNiP, the average unicast message latency in-creases drastically with the increase in broadcast traffic percentage. On the other hand, the hybrid ONWiNiP architecture provides limited support by using its wireless broadcast plane for a low percentage of broadcast traffic. However, for higher broadcast traffic percentage, the wired unicast plane gets quickly saturated by

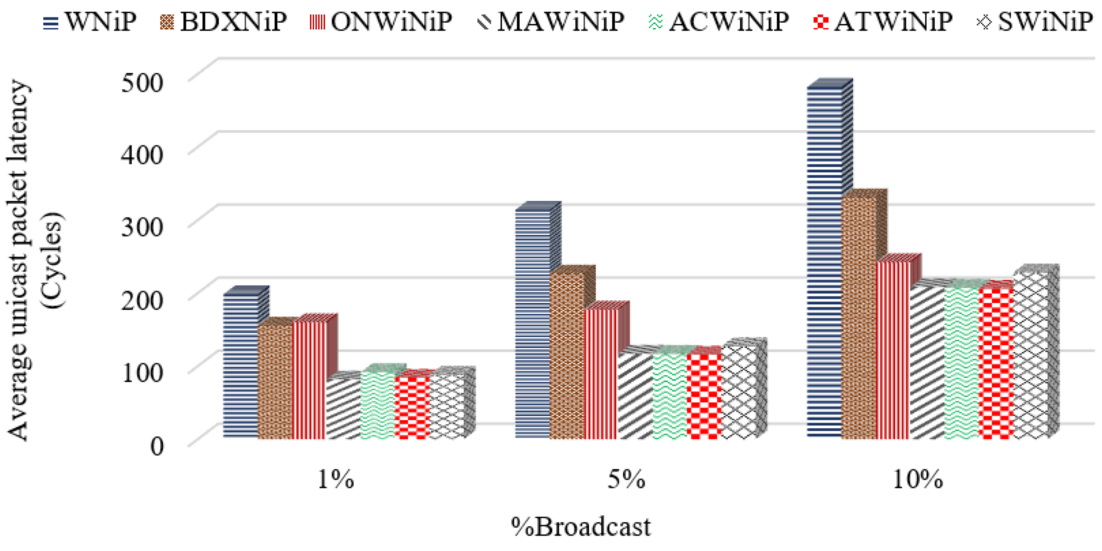


Figure 3.12: Average unicast latency for different multichip systems.

broad-cast traffic which results in a large average unicast latency for such systems.

ACWiNiP, ATWiNiP, and SWiNiP being the WiNiP architectures utilizing the asymmetric topology or the hybrid MAC, provide better average unicast latency compared to WNiP, BDXNiP, and ONWiNiP architectures. At lower broadcast traffic percentage (1%), because of the reconfigurable hybrid MAC, one-to-many traffic-aware SC protocol, and the underlying asymmetric topology, MAWiNiP can ensure faster transmission of one-to-many traffic to reduce overall network congestion. Therefore, it eventually helps to reduce the large queuing delay of unicast traffic and as shown in Figure 3.12, it ensures better average unicast latency compared to ACWiNiP, ATWiNiP, and SWiNiP architectures.

With the increase in broadcast percentage, large average broadcast latency due to only CDMA mode transmission in ACWiNiP and limited access to wireless channel along with the under utilization of bandwidth in ATWiNiP results in higher queuing latency for unicast messages in these systems. In MAWiNiP architecture, as the broadcast-aware SC protocol prioritizes more number of broadcast messages with the increase in broadcast traffic percentage, the average unicast latency for MAWiNiP system also increases. However, this increase is marginal compared to ACWiNiP and ATWiNiP architectures at a higher broadcast percentage. To be specific, for MAWiNiP architecture, the worst average unicast message latency increment is 1.02% compared to ATWiNiP architecture for 10% broadcast traffic. Due to the multihop wired path and similar MAC, average unicast latency for SWiNiP is always higher than MAWiNiP architecture.

There is no way to ignore the fact that some time-critical unicast messages exist in such systems and those messages can also be handled by the proposed flow control and reconfigurable hybrid MAC. The existing flow control can provide higher priority to such time-critical unicast messages by treating it as a multicast message with a single destination. The application layer can identify such messages to set the corresponding multicast and destination bits in the packet structure as shown in Figure 3.5(b) and the proposed flow will ensure low latency for the message. Moreover, multi-level priority can also be introduced to handle such messages with additional encoder and decoder overheads. However, it is also obvious that such messages are few and the ultimate goal of the paper is to introduce novel wireless interconnection architecture to achieve low latency for one-to-many traffic transmission which eventually leads to lower congestion, better unicast as well as overall network latency.

3.4.6 Evaluation for System Size Scaling

Scalability has always been an area of interest for on-chip and off-chip interconnection architectures. In this section, the performance of the proposed MAWiNiP architecture will be evaluated in comparison with the BDXNiP, WNiP, and ONWiNiP interconnection architectures as the number of multicore chips scale up. Such system scaling affects the chip-to-chip and most importantly chip-memory traffic as more chips will be sharing the in-package memories. Each system is evaluated in the presence of 10% broadcast along with 10% memory traffic from each core. The rest of the traffic is considered as one-to-one unicast traffic pattern between all cores. We consider three MCMC configurations of 4, 9, and 16 processing chips, with each chip having 16 cores. We considered 4 memory stacks in each configuration. A cluster size of 4 was considered including one memory controller in each cluster resulting in 4 clusters or 4 memory controllers in each chip.

As each of the memory controllers is equipped with a TR-WI, increasing the number of chips for MAWiNiP system introduces additional codeword overhead for CDMA mode which makes the wireless links slower. For the WNiP and BDXNiP wired architectures, increasing number of chips in the system increases off-chip traffic, which needs to travel through long multi-hop paths. Similarly, for ONWiNiP architecture, its wireless plane suffers from higher contention due to CSMA MAC protocol and wired plane encounters rapid latency increase similar to WNiP architecture. Therefore, the system bandwidth decays for all the

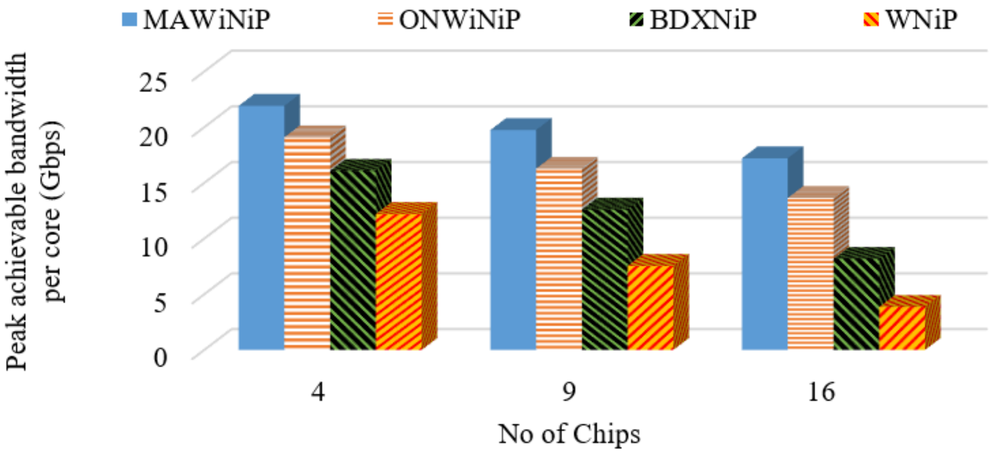


Figure 3.13: Peak achievable bandwidth with system size scaling.

architectures significantly with system size scaling as shown in Figure 3.13. It can be also observed from Figure 3.13 that, though MAWiNiP system also requires long codeword for unicast transmission with system size scaling, due to its efficient one-to-many traffic support it provides the highest overall system bandwidth among all the architectures.

3.4.7 Effect of Flit Size Variation

Different computing nodes and protocols might re-quire different flit sizes for inter, intra-chip, and chip-memory communication. The multicast/broadcast latency can vary due to flit size variation. Therefore, in this section, the effect of increasing flit width has been analyzed on average packet latency for MAWiNiP, ONWiNiP, and BDXNiP interconnection architectures compared to WNiP architecture. A 4 chip system with 4 in-package memory modules was considered for all the architectures and the relative latency gain with uniform random traffic patterns was measured with 10% broadcast traffic with respect to WNiP architecture.

For performance evaluation with variation in flit width, three different flit sizes of 32, 64, and 128 bits were considered. This is because as noted in [129], higher flit widths beyond 128 are shown to provide marginal gains in performance of even a conventional wired NoC based system while considerably increasing energy or power consumption of the data communication hardware. In the case of WNiP and BDXNiP wired architectures, an increase

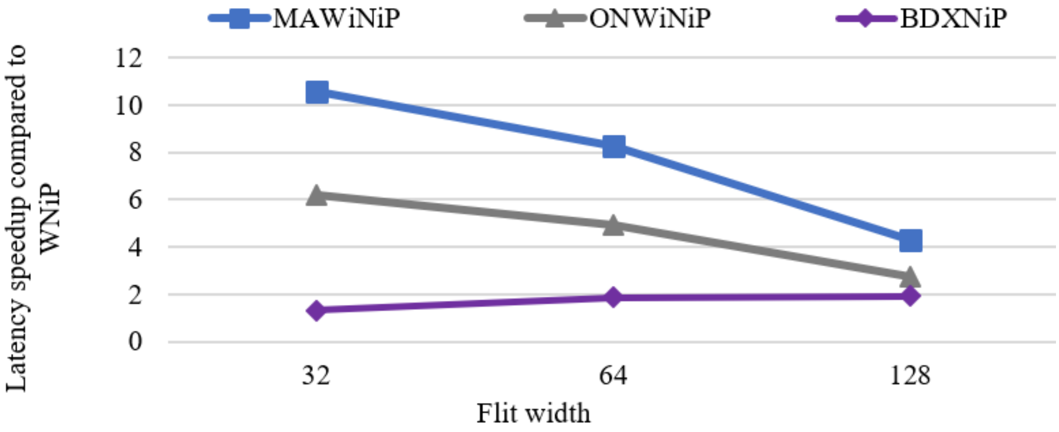


Figure 3.14: Average packet latency gain over WNiP for flit size variation.

Table 3.4: Core configuration for application-specific traffic simulation

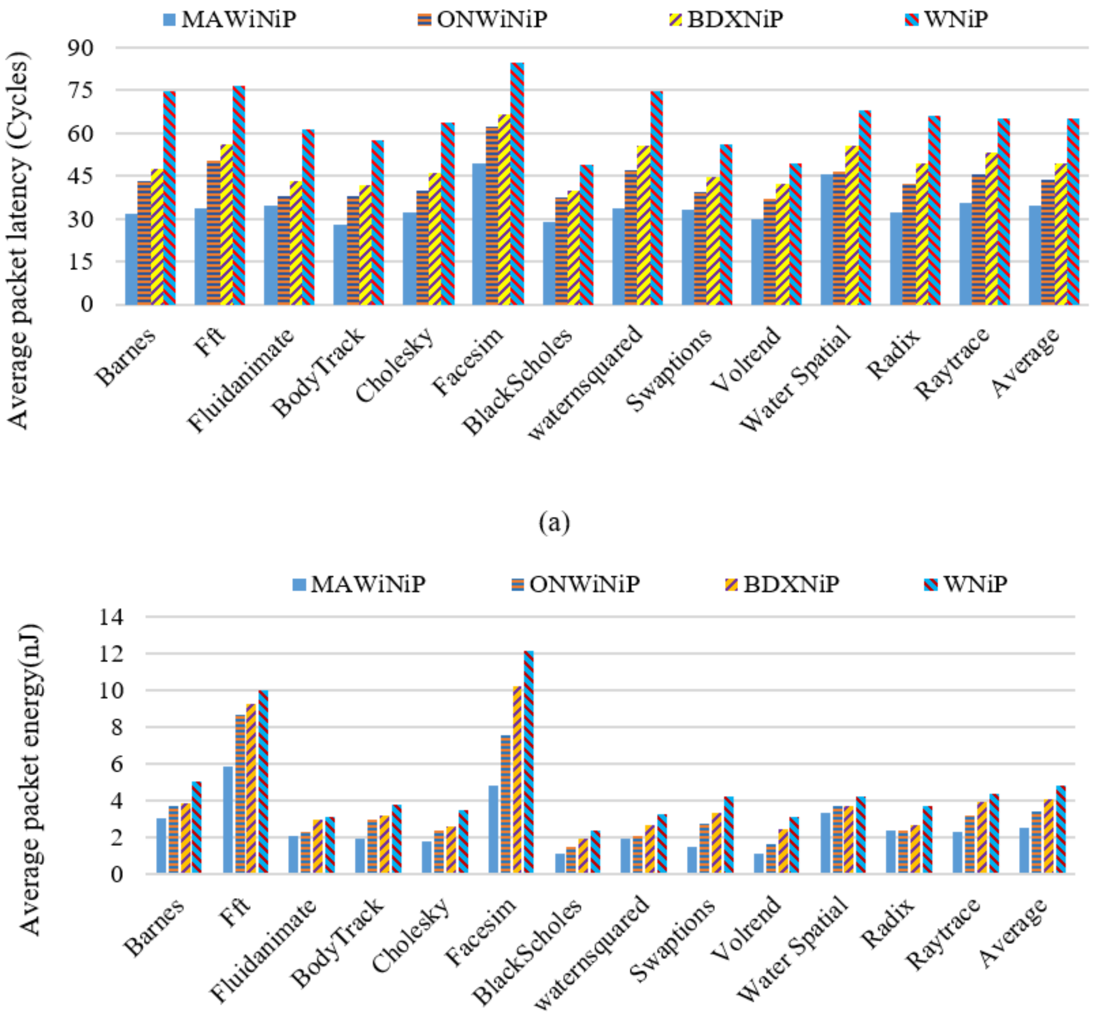
Component	**Configuration**
Cores	Out-of-Order, 16cores/chip, 1GHz
L1 Cache	32KB, 4-way, LRU policy, private
LLC (L2) Cache	512KB, 8-way, LRU policy, shared
Cache Coherency	Directory-based MOESI

in flit-width translates into increasing the band-width of the chip-to-chip and chip-memory links by increasing the number of parallel wires. On the other hand, the data rate of the wireless links is governed by the speed of the transceiver, modulation scheme, and band-width of the antennas, which do not change with flit width. Hence, while the wireline communication becomes faster with an increase in flit size in WNiP and BDXNiP architectures, the wireless communication speed remains constant in MAWiNiP and ONWiNiP architectures. This results in a reduction in latency gain for the wireless architectures with increased flit width as shown in Figure 3.14. On the other hand, BDXNiP, being completely wireline, shows an increment in latency gain with larger flit width as the link width increases with the increase in flit-size. However, as the wired network is already congested with 10% broadcast traffic, therefore, even with a flit width of 128 bits a relative average packet latency gain of 2.22 and 4.3 can be observed compared to BDXNiP and WNiP architectures respectively for MAWiNiP system.

3.4.8 Evaluation with Application-Specific Traffic

After evaluating the performance of the proposed architecture under various synthetic traffic scenarios, this section evaluates the performance of the MAWiNiP, BDXNiP, WNiP, and ONWiNiP multichip architectures with application-specific traffic patterns from PARSEC and SPLASH2 benchmark suites. For evaluation a 4 chip system with 4 in-package memory modules was considered. To generate the application-specific traffic patterns, a multicore chip with 16 out-of-order cores were considered whose features are listed in Table 3.4. These core configurations are then used to ex-tract the core-to-memory and cache coherency traffic for these applications when they are executed until completion using SynFull [122]. In order to map these traffic patterns to the MCMC environment, it was considered that multiple threads of the same application kernel running on the MCMC system where each processing chip executes a single thread, and the memory stacks are shared among threads.

The average packet latency and average packet energy of the wired and wireless architectures for different application-specific traffic patterns are shown in Figure 3.15. The latency best represents the performance in these cases as the interconnection network is not saturated in the steady-state. The average packet latency and average packet energy for the wireless MCMC systems vary between applications due to the variation in off-chip traffic patterns from different memory access patterns. However, for all application-specific traffic



(a)

(b)

Figure 3.15: Performance evaluation of the multichip system with application specific traffic (a) Average packet latency (b) Average packet energy.

patterns considered here, the performance of the proposed MAWiNiP multichip system is better than all other interconnection architectures studied in the paper. This is because of the proposed multicast-aware, dynamically reconfigurable MAC, and the support from the underlying asymmetric topology and flow control. For the MAWiNiP architecture, the average reduction in latency with respect to WNiP, BDXNiP, and ONWiNiP architecture is 47.08%, 30.12%, and 20.75% respectively. Similarly, the average reduction in packet energy with respect to WNiP, BDXNiP, and ONWiNiP systems is 46.96%, 15.23%, 25.61% respectively.

3.4.9 Area Overhead Analysis

This section discusses the area overhead for all the R-WIs and TR-WIs in a 10×10 mm^2 multicore chip in 65nm technology node. Each receiver in R-WI occupies 0.093mm^2 and each transceiver for TR-WIs takes 0.17mm^2 active area in silicon [124], [125]. Hence for a 16 core multicore chip, R-WIs and TR-WIs take up to 1.116 mm^2 and 0.68 mm^2 silicon area respectively with the smallest cluster size of 4. Therefore, all the R-WIs and TR-WIs together in a multicore chip take only 1.796% of the active silicon area of the multicore chip. The length of each antenna is 0.4 mm, but it does not occupy any active area being fabricated using upper layer metal processes [3]. Additionally, the area, delay and power overhead for the digital MAC unit components corresponding to each TR-WI have been shown in Table 3.1. Therefore, for each multicore chip, having 4 TR-WIs will have an additional area overhead of 552.4 μm^2 making the total overhead of the wireless interconnection less than 1.8 mm^2 per chip.

3.5 Chapter Summary

Modern computing platforms experience multicast/broadcast traffic due to their multicore, multiprocessor architecture as well as for the rapid growth of memory-intensive applications. In the near future it is obvious that for massive computing platforms consisting of thousands of cores/processors, this one-to-many traffic will be the major performance bottleneck if the architecture and the interconnect fail to provide the required support for such traffic.

Motivated by the limited support of the state-of-the-art wired interconnects, in this chapter a novel one-to-many traffic-aware asymmetric WiNiP topology along with a novel dynamically reconfigurable MAC and flow control has been proposed. The proposed topology exploits the MAC and the flow control through a symbiotic relationship and provides the required support not only for one-to-many traffic but also for many-to-few hotspot traffic. Through system-level simulation, we show that the proposed WiNiP architecture can reduce the system latency in presence of such one-to-many traffic by 47.08% compared to WNiP. Even with the scaling of system size or cluster size the proposed architecture outperforms the other wired and wireless architectures in terms of energy, bandwidth, latency and hence, represents a scalable interconnect solution for future MCMC-based HPC nodes.

Chapter 4

A Jamming-Aware Secure WiNiP Architecture for Multichip Systems

Mm-wave enabled wireless interconnection architectures have emerged as an energy-efficient, low-latency, and scalable solution for future MCMC-based HPC systems. Despite providing performance enhancements, wireless channel, being an unguided medium, introduces potential security vulnerabilities inherited from traditional wireless networks such as jamming induced DoS and eavesdropping. Securing the systems against such induced threats often introduce large overheads and performance penalties. To address these challenges, in this chapter, the architecture of a WiNiP architecture that reuses the in-built DFT hardware for securing against external and HT induced internal attacks has been discussed. The proposed architecture is capable of securing against adversaries with a reconfigurable wireless interconnection (AWARe-Wi). To ensure higher accuracy, ML classifier was deployed to detect the threats. In addition, for a robust threat detection against crafted attacks, an AML-based approach has been also introduced. Moreover, to enable sustainable multichip communication in such systems even under jamming attack from both internal and external attackers, design of a reconfigurable MAC and a suitable communication protocol have been discussed. The simulation results show that, the ML and AML classifiers can achieve an accuracy of 99.87% and 95.95% respectively for attack detection while the proposed WiNiP can sustain chip-to-chip communication even under persistent jamming attack with an average 1.44× and 1.56× degradation in latency for internal and external attacks respectively.

4.1 Motivation

High-performance computing nodes such as blade servers and embedded systems have already undergone a massive paradigm shift from single core, single chip architecture to MCMC architecture. This paradigm shift is justified as follows, for a large single chip, different factors such as sub-wavelength lithography, line edge roughness, and random dopant fluctuations can cause wide process variations, which can result in higher fault density and hence, reduces the manufacturing yield. Therefore, the disintegration of larger single chips into smaller chips, forming multichip compute systems, such as the AMD EPYC [27] series released in 2017, aid in alleviating the effect of higher fault densities in advanced technology nodes and eventually leading to reduced manufacturing cost per die. Despite the achieved yield enhancements, each chip in the SiP of MCMC demands an efficient intra-chip as well as an inter-chip communication, as disintegration increases inter-chip traffic significantly. Although NoC has emerged as a scalable, modular on-chip interconnection architecture, it cannot provide low latency for large systems due to its multi-hop nature [60][52].

On the other hand, the performance of the MCMC system is mostly limited by the high latency and power hungry off-chip I/Os. Conventionally, C4 bumps coupled with in-package transmission lines or flip-chip packaging [29] is used to interconnect chips within a MCMC system. However, signal quality deterioration due to microwave effects, crosstalk coupling effects and frequency-dependent line losses in the transmission line limit the number of concurrent, high-speed inter-chip I/O and hence chip-to-chip bandwidth. In recent literature is has been shown that WIs operating at GHz bandwidth in mm-wave bands can mask off-chip I/O delay by establishing single hop, energy-efficient chip-to-chip communication links [3][54]. We refer such MCMC systems with mm-wave wireless interconnect as WiNiP here.

Although extensive research has been carried out towards improving performance and energy dissipation in WiNiPs [3], [83], relatively little attention has been given to the information integrity and security or privacy aspects of WiNiPs. Wireless being an unguided, shared transmission medium is vulnerable to many attacks such as DoS, eavesdropping, and spoofing. Although each of these attacks require its own detection and defense mechanism, this chapter is focused on persistent jamming-based DoS attack as it is one of the most common, simple and yet powerful attack on wireless systems. Moreover, such attacks are yet to be analyzed for MCMC systems using wireless interconnect.

To replicate such an attack, an external attacker that produces a high energy EM radiation that causes interference in the wireless medium of the WiNiPs was considered. In addition, to address the complexities and vulnerabilities arising from hardware design and manufacturing process, a HT was considered to be maliciously embedded in MCMC system during the design or fabrication process. In this case, one of the WIs infected by a HT will transmit the data over wireless channel irrespective of whether it is enabled by the adopted MAC protocol of the WiNiP. This will cause contention or interference with legitimate transmissions causing DoS on the remaining WIs.

While well-known defenses exist against DoS attacks in large scale wireless networks, those techniques cannot be adopted and applied directly to the WiNiP scenario due to large power, area and timing overhead of the existing security implementations [108]. To address and meet such constraints, in this chapter, a persistent jamming-based DoS attack resistant architecture has been proposed that re-uses the existing DFT hardware to detect and defend against jamming attack in WiNiPs. Moreover, under such jamming attack, specially for MCMC systems, it is non-trivial to synchronize and inform all other WIs about the presence of an adversary as chip-to-chip communication happens through only wireless medium which is itself vulnerable to the attack. To address this issue, a novel MCMC wireless communication protocol was also developed along with a reconfigurable MAC that can ensure robust and secure communication under internal and external persistent jamming attack. To handle more intelligently crafted jamming attacks and ensure a robust, accurate detection and defense mechanism AML and adversarial training for the deployed ML classifiers have been utilized. The research contributions of this work can be outlined as follows:

- *To the best of our knowledge, this is the first work that proposes a solution for persistent jamming attack by re-using DFT infrastructure for WiNiP.*

- We propose a novel dynamically reconfigurable MAC and the corresponding synchronization mechanism for all WIs under jamming condition.

- We propose an AML-based mechanism for high accuracy threat detection and recovery from persistent jamming-based DoS attacks.

- We describe a novel communication protocol necessary to ensure a robust communication even under a persistent jamming-based DoS attack.

- We analyze the performance degradation of the proposed security mechanism for different WiNiPs and compare it with wired MCMC systems.

4.2 Fundamentals of DFT Architectures

Before discussing the proposed jamming-aware WiNiP interconnection architecture, a brief of the IEEE P1500 and IEEE 1149.1 test architectures will be discussed in this section. As such DFT architectures have been re-used to detect persistent jamming attack in this chapter, the fundamental of those DFT architectures is required to understand the secure WiNiP architecture described in the next section.

The IEEE P1500 standard has been developed to ensure a scalable, plug-and-play testing methodology for embedded cores in a SoC environment. IEEE P1500 test standard mandates a test wrapper around each core. The wrapper architecture is comprised of an Wrapper Instruction Register (WIR), Wrapper Boundary Register (WBR), Wrapper Bypass Register (WBY) along with Wrapper Interface Ports (WIPs) with the on-chip Test Access Mechanism (TAM). The WIPs in the wrapper can be classified as mandatory Wrapper Serial Ports (WSPs) and optional Wrapper Parallel Ports (WPPs). The instructions in WIR can configure the cells in WBR to parallelly or serially load the test data to perform the vector-based testing of the functional core. Figure 4.1(a) shows the IEEE P1500 architecture.

The IEEE 1149.1 standard or Joint Test Action Group (JTAG) was introduced by a group

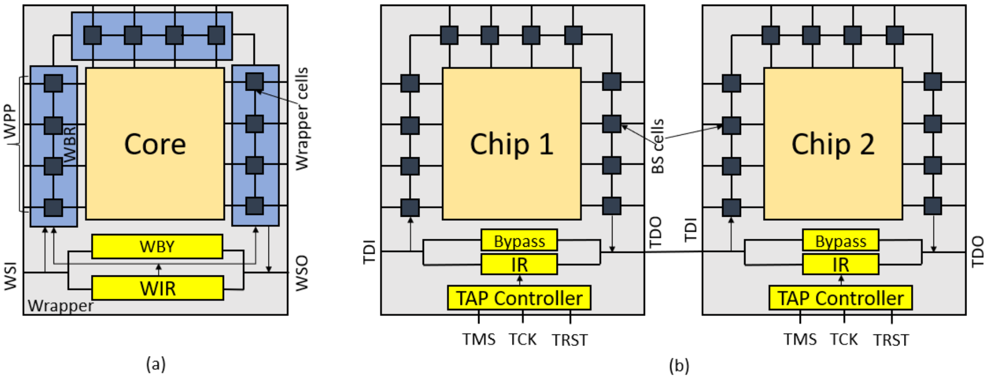


Figure 4.1: (a) IEEE P1500 (b) IEEE 1149.1 (JTAG) test architecture.

of European engineers to address the difficulties of so-called ”bed-of-nails” technique to test electronic devices on a printed circuit board. As the proposed solution involved serial shift register around the boundary of the chip, it is also known as boundary scan. The basic JTAG architecture is comprised of four mandatory test ports namely Test Data In (TDI), Test Data Out (TDO), Test Mode Select (TMS) and optional Test Reset (TRST) port. These ports are collectively referred to as Test Access Port (TAP). To apply the test vectors, Boundary Scan (BS) cells on the device primary input, output pins are placed serially along the boundary of the chip to form a scan chain. Moreover, the architecture also use Instruction Register (IR) and Bypass Register, which along with the TAP controller perform the intended tests on selected chips. Figure 4.1(b) depicts the JTAG test architecture with two chips.

The Built-in-Self-Test (BIST) is an on-chip test methodology proposed for faster and low-cost testing of the modern complex SoCs. In BIST, additional hardware is designed inside the chip to generate and apply the test pattern by itself for self testing. The pseudo-random pattern is mainly generated by a Linear Feedback shift Register (LFSR). LFSR implements a higher degree polynomial function to generate the pseudo-random pattern. Based on the initial value (seed) and the tapping points, LFSRs can produce random sequence of very long cycle. Such LFSRs have been re-used to secure the multichip wireless communication described in Section 4.3.3.

4.3 WiNiP Interconnection Architecture

This section discusses the proposed WiNiP interconnection architecture which covers the adopted topology, the proposed hybrid MAC and physical layer.

4.3.1 Adopted Multichip Topology

To meet the increasing memory demands for current and emerging applications and mimic real MCMC architectures, we consider an MCMC system with multicore processors and in-package memory modules. The memory modules are connected to the edge cores through wired interconnect. Each tile in the multicore chips is composed of a processing core, a switch, L1 private cache and a distributed shared LLC. Tiles in each chip are connected with each other through a regular wired mesh-based NoC. For inter-chip communication, in each chip, two NoC switches were equipped with WIs as shown in Figure 4.2. Keeping the number of WIs minimum for inter-chip communication helps to reduce the communication overhead during jamming attack for our proposed approach.

However, a minimum of two WIs are necessary for each chip to ensure connectivity and reliable communication with the rest of the system even if one of them is compromised by an internal HT. A higher number of HTs within a single chip is assumed to be unlikely as it will make HT detection easier. Typically, the footprint of HTs are minimal by design and hence a maximum of a single HT per chip was assumed in our analysis. Although inter-chip communication happens only through the WIs in functional mode, the MCMC system is compliant to JTAG test architecture where their boundary scans are daisy chained. We leverage this JTAG infrastructure for enhancing the security of MCMC system.

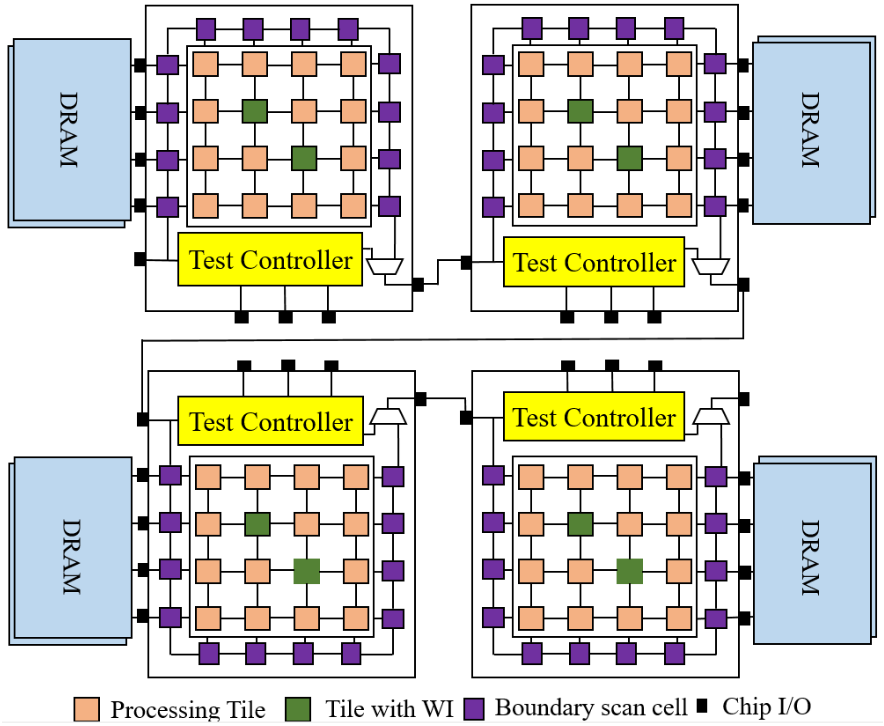


Figure 4.2: Proposed multichip WiNiP topology.

4.3.2 Persistent Jamming-Aware Reconfigurable MAC

A wireless MAC mechanism enables a contention-free communication over the shared wireless channel among multiple transceivers. So far, no MAC has been proposed which is jamming-aware and can sustain communication in both normal and attack scenario. Therefore, a reconfigurable MAC mechanism operating in two modes was proposed for sustainable communication even under persistent jamming attack. In the absence of persistent jamming attacks, a reservation-based MAC, termed as Normal MAC (NMAC) was considered active for MCMC communication. In NMAC, to get the channel access, each sender sends a non overlapping reservation request to all the receivers encoded by a Common (C) code. Figure 4.4 shows the structure of the reservation packet and is discussed in details in the next subsection. As each receiver is equipped with same arbitration logic, each of them grants access to the same transmitter that gets the whole channel access at a time. In NMAC, as one sender gets the whole channel access, it ensures a contention free, high bandwidth off-chip communication.

The above mentioned mode of WiNiP communication is unaware of any persistent external jamming. Therefore, the MAC is switched to Pseudo-random Noise (PN) encoded Asynchronous Code Division Multiple Access (ACDMA) during external jamming attack and call it Attack MAC (AMAC). In this work, by ACDMA we only refer to using PN sequences and not other protocol overheads present in ACDMA communication in mobile cellular network. Data encoded with PN sequence is jamming and eavesdropping resistant because of the spread spectrum technology where the transmitted signals appear as noise to every receiver, except the one that has the PN code which was used to encode the data

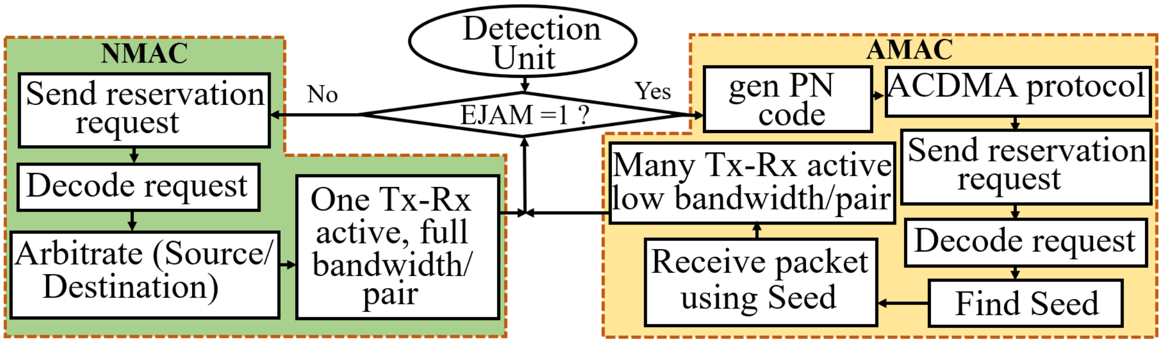


Figure 4.3: Overview of the reconfigurable MAC.

during its transmission. Therefore, any transmission not encoded with the same code appears as noise due to the weak cross-correlation, making this AMAC resilient to jamming. The PN codes used for ACDMA communication should have a strong auto-correlation and weak cross-correlation property. While maximal-length sequence (m-sequence) and Kasami sequence can be used to generate PN sequences, these sequences have worse cross-correlation property to Gold sequence [130]. Moreover, Gold sequence can also support more users than both Kasami and m-sequence. Therefore, PN codes were generated using Gold sequence.

The hybrid Transmitter-Common (TC) [120] PN code protocol was used to enable communication in AMAC mode where each transmitter have specific codes to encode packets they transmit and receivers have decoders for all channels to be able to receive data simultaneously from multiple transmitters. The common channel is used for arbitration and attack information propagation. The AMAC mode is not used in normal, attack free operation circumstances, as it reduces communication bandwidth of each link by its spreading factor. The focus of this paper is to ensure robust WiNiP communication in presence of persistent jamming attack on a high bandwidth WiNiP not to ensure high performance during such attack. Figure 4.3 shows the proposed reconfigurable MAC with the underlying operations.

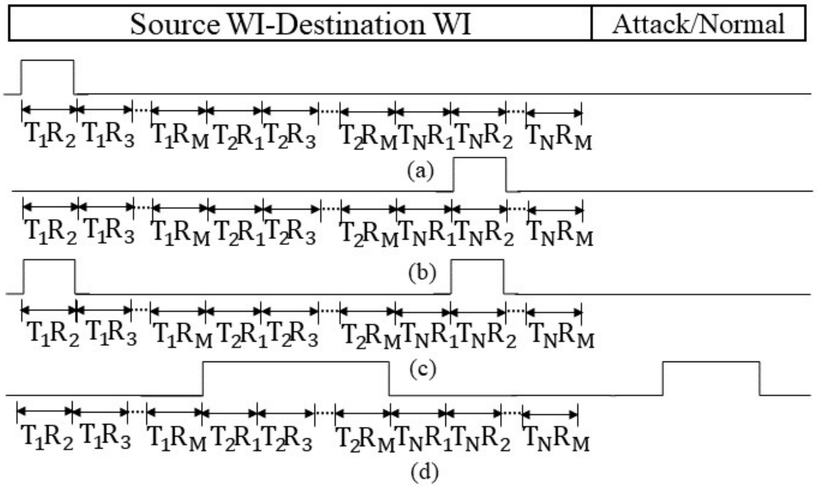

Source WI-Destination WI
Attack/Normal
T1R2 T1R3 T1RM T2R1 T2R3 T2RM TNR1 TNR2 TNRM
(a)
T1R2 T1R3 T1RM T2R1 T2R3 T2RM TNR1 TNR2 TNRM
(b)
T1R2 T1R3 T1RM T2R1 T2R3 T2RM TNR1 TNR2 TNRM
(c)
T1R2 T1R3 T1RM T2R1 T2R3 T2RM TNR1 TNR2 TNRM
(d)

Figure 4.4: Channel reservation process.

4.3.3 Flow Control and Communication Protocol

Some of the key challenges of such jamming-aware hybrid MAC is to ensure proper switching and synchronous operation across MCMC system for both NMAC and AMAC modes with low overheads. In this section, the proposed flow control that addresses these issues will be discussed.

To ensure low area and latency overhead, we adopt a VC-based wormhole switching protocol for routing data where packets are broken into smaller flow control units or flits for both wired and wireless links. A forwarding-table based routing over pre-computed shortest paths is adopted to minimize the packet latency. The routing tree is constructed using Dijkstra's algorithm, which extracts a Minimum Spanning Tree (MST) providing the shortest path between any pair of nodes in a graph. Consequently, deadlock is avoided by transferring packets along the shortest path routing tree, as it is inherently free of cyclic dependencies.

For high bandwidth off-chip communication during NMAC, each WI sends its reservation signal encoded by a fixed common PN code to all other WIs. The PN code being common to every WI increases the chance of corrupting the source-destination addresses of the multiple simultaneous requests. Therefore, a non-overlapping/non-interfering source-destination representation was proposed. As shown in Figure 4.4(a) and (b), each transmitter has its own slot to define its intended receivers. The slots being non-overlapping and orthogonal does not create any interference with each other in their aggregate signal as shown by Figure 4.4(c). Hence, receivers can arbitrate among multiple requests and grant the channel to a single transmitter in NMAC mode. The adopted arbitration logic considers channel access starvation for WIs and provides priority to multicast traffic [131]. We re-use such non-overlapping signals to ensure synchronous operation even under jamming attack as discussed in the next paragraph.

When the MCMC system is under attack, all the WIs in the system changes its MAC to ACDMA mode and continue their communication, but with a reduced bandwidth as the data is now encoded with PN sequence. The attack condition is provided the highest priority which is indicated by the attack flag in Figure 4.4(d). After detecting a potential external jamming attack as described in Subsection 4.4.2, a WI uses such signaling encoded by fixed PN code to inform other WIs during external jamming. All the other WIs in MCMC system after receiving the attack signal switches to AMAC mode simultaneously due to the priority in attack bit. The PN sequence generation and AMAC communication are described in the next subsections.

PN Code Selection and Generation

The PN codes are binary sequences that appears to be random, but, they can be generated in a deterministic manner. However, to generate Gold sequence, two preferred m-sequences of the same length are required. In each of the transmitters, we configure two LFSRs according to the preferred polynomial pair and XOR their output to finally generate the desired Gold sequence. Figure 4.5 shows the LFSR configuration to generate a 32 bit gold code. Moreover, to generate a different PN sequence for each of the transmitter, different seed values were chosen for each of the transmitters.

ACDMA Communication Mechanism Under Attack

During any persistent jamming attack, all the WIs in the MCMC system change the MAC to ACDMA mode as discussed in Section 4.3.2. In ACDMA mode, the PN codes are managed using TC protocol. Before any transmission, similar to reservation assisted NMAC mode, the senders use a common PN code to send non overlapping send requests as shown in

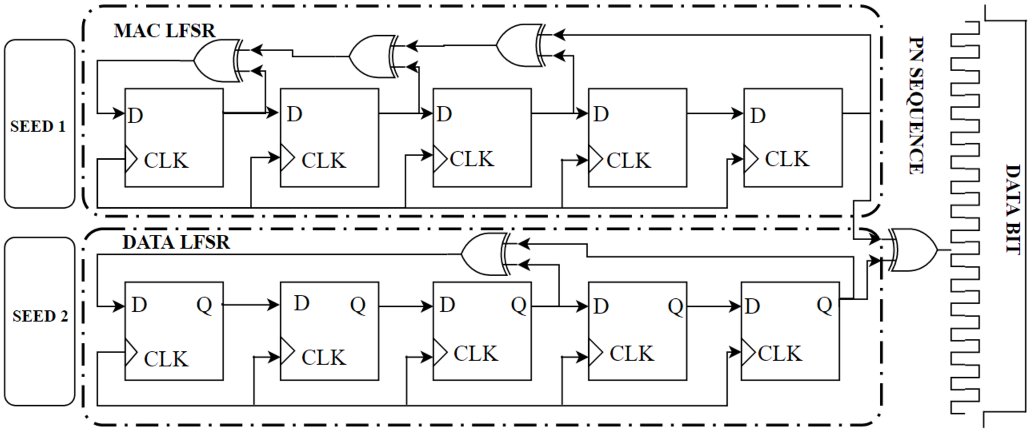


Figure 4.5: PN code generation using Data and MAC LFSR.

Figure 4.4. However, based on the received requests, multiple receivers can grant access to multiple transmitters as now communication happens through different ACDMA channels. We consider the LFSR length to be 5 so that each PN sequence repeats after 32 cycles which is exactly the same time duration of a single bit of the baseband signal. Therefore, each signal in a particular transmitter will be modulated by the same PN sequence. However, different transmitters use different codes of the same length because of having different seed values. Each receiver stores the seed values in a small tamper-proof Read-Only-Memory (ROM) where the address of the seeds matches their transmitter address. Therefore, the receiver already know which PN code to use for demodulation in a particular channel while granting the channel access through reservation requests. Hence, the additional delay for seed search does not have any impact on data transmission. To enhance security the seed values can be dynamically changed as commonly practiced in cellular networks [132]. The AMAC steps are also depicted in Figure 4.3. The transmitter and receiver architecture will be discussed in the next section.

4.3.4 Physical Layer

To combat the persistent jamming, physical layer implements the components required for both NMAC and AMAC protocols along with WIs. On-chip miniature zig-zag antennas operating in the unlicensed 60 GHz mm-wave band was used to establish direct communication channels between the WIs. Such antenna provides a bandwidth of 16GHz for both intra and inter-chip communications [3]. The transceiver design was adopted from [125] [124]. Non-coherent OOK-based transmitter and receiver design is chosen, as it allows a relatively simple and low-power circuit implementation without the need for power-hungry carrier recovery circuitry [3]. In addition to the OOK modulator and demodulator, a CDMA encoder and decoder is also designed for reservation and AMAC mode communication.

Irrespective of the NMAC or AMAC mode, as shown in Figure 4.6(a), each transmitter sends a C channel encoded reservation request to access the channel which is decoded in receiver's C channel decoder as shown in Figure 4.6(b). Then only base-band data or PN-encoded data is transmitted during NMAC and AMAC mode respectively. As only one transmitter is active during NMAC, the transmitted data is captured directly at the receiver. However, due to the adopted TC protocol for AMAC, the receiver needs to have CDMA decoders for every ACDMA code-channel. Therefore, in the receivers, the output of the OOK demodulator is further sent to a CDMA receiver during an external jamming attack.The signal is correlated with each regenerated PN code in the receiver side to create separate receive channels. The PN codes are regenerated by retrieving the seed for the sender from the ROM as soon as the receiver responds to the sender's reservation request and thus, hides the run time PN code regeneration latency.

4.4 Attack Model and Detection

This section discusses the attack model, proposed detection, and defense mechanism that ensures robust communication under external and internal jamming attack scenario using ML and AML approach.

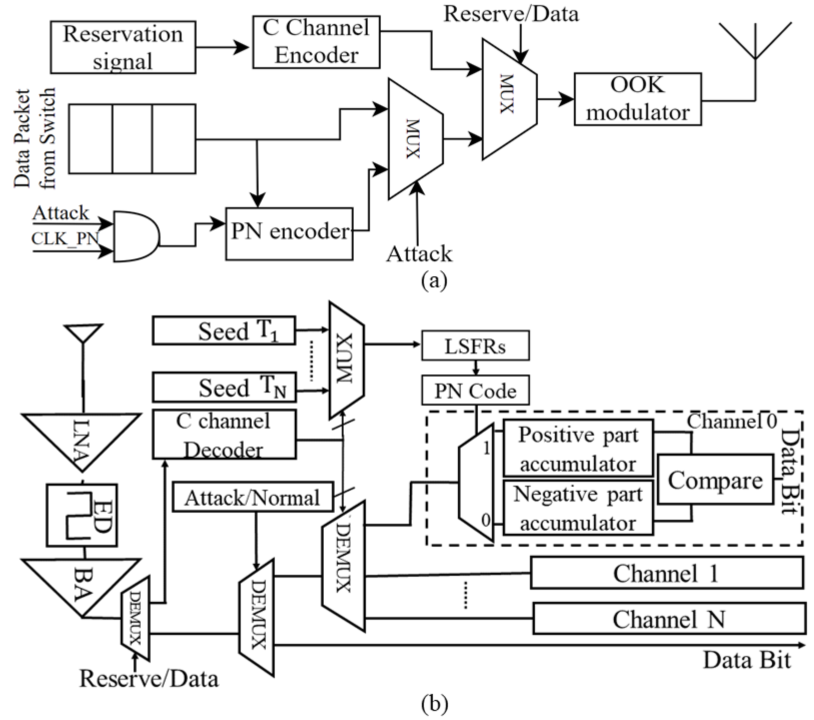


Figure 4.6: (a) Transmitter (b) Receiver block diagram

4.4.1 Attack Model

In this chapter, as aformentioned, a persistent jamming-based DoS attacks was considered on the wireless interconnections of a WiNiP. In the presence of such a persistent DoS jamming attack either from an external or internal attacker, there will be interference among the attacker and the legitimate transmitter. This interference will cause high error rates due to interference noise. Moreover, as the attack is persistent, it will cause errors in contiguous bits of flits resulting in burst errors. Over the duration of the attack, these errors will span multiple flits and therefore, cause burst errors in consecutive flits of a packet.

Burst errors in both wired and wireless links can happen as a random event as well such as, power source fluctuations, ground bounce or crosstalk [133]. However, the burst errors due to random events such as crosstalk will be relatively short lived, due to the data transition pattern in that cycle. On the other hand, burst errors resulting from jamming attacks could be sustained for longer duration as a shorter DoS attack is not effective. A few burst errors caused by a short-lived DoS can be corrected/detected by a burst error correction/detection (BEC) code or retransmissions. Therefore, to be truly effective as an attack, the jamming has to be persistent to cause enough flits to be in error such that the existing BEC mechanism either cannot correct it or causes a prohibitively large number of retransmissions. Hence, we consider persistent jamming attacks either from a single external attacker or a single internal HT which affect the WIs in the WiNiP. For the internal attacker, a single HT per chip was considered in the MCMC system as that is a smarter HT insertion approach because the probability of HT detection increases with increase in the number and footprint of HTs [134].

ML techniques were employed for attack detection as discussed in the next subsection. Despite ML being robust to random noises, it has been shown that ML techniques are vulnerable to crafted threats, termed as *adversarial samples* [135, 136]. Adversarial samples exploit the sensitive features in the input or the ML model, adding noise to which can lead to misleading the output of the ML model [137, 138]. In similar manner, in this chapter, adversarial attacker will be introduced who can attack the system by cognitively crafting the attack.

The first step to launch such an adversarial threat is to determine the model (and/or parameters). This is performed through reverse engineering process by iteratively sending in the data and obtaining the responses, similar to that in [139]. Once the reverse engineered model is built, then, the attacker tries to estimate the model and introduce the perturbations by incrementally increasing the noise to the input features that are sensitive similar to [140] to evade detection or to induce false alarms. In this work, we utilize Fast Gradient Sign Method (FGSM) attack [135] to craft such an adversarial attack. However, it needs to be noted that direct application of FGSM is not feasible, as it does not have a notion of relativity between individual features when crafting an adversarial sample. To combat such scenario, we introduce the relationship between different features such as number of errors not more than the total number of packets sent in the form of constraints.

4.4.2 Attack Detection Methodology

To detect a persistent jamming attack with less area overhead the JTAG test infrastructure has been re-used for probing the wireless interconnect. The architecture of the proposed security framework is shown in Figure 4.7. When the probe (PRB) signal is asserted to the security controller from the Attacker Detection Unit (ADU) as shown in Figure 4.7, the MCMC system suspends its normal WiNiP operations and enables an LFSR called MAC-LFSR to enter into the probe mode. Only a single MAC-LFSR is necessary for the entire MCMC system. The MAC-LFSR grants access of the wireless medium to each WI

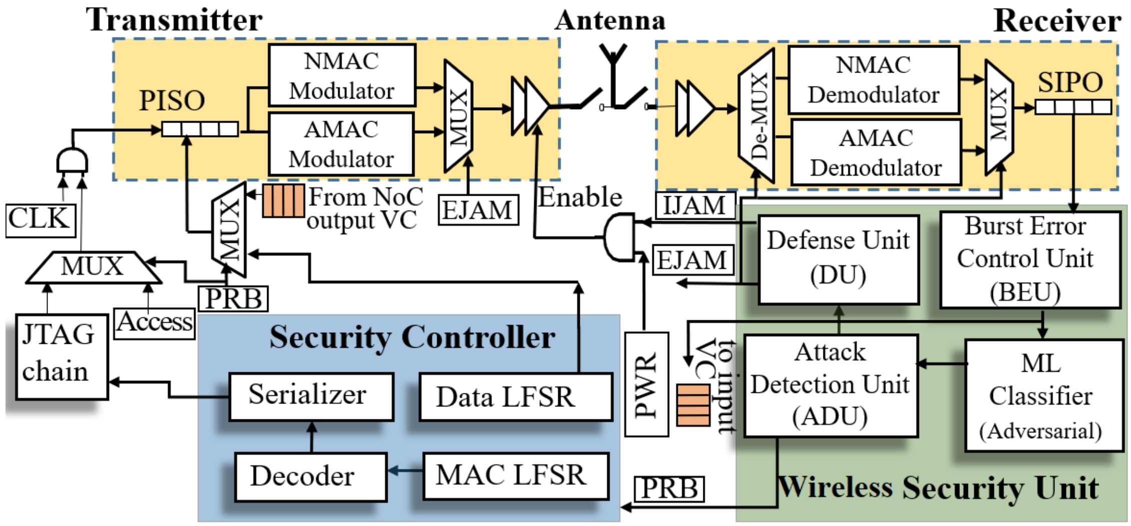


Figure 4.7: Proposed security framework.

in a pseudo-random pattern to transmit normal data packets such that performance is not impacted in the probe mode. The MAC-LFSR sets the MAC controller of each WI among various chips by utilizing the serial JTAG boundary scan chain as it is not vulnerable to wireless jamming. Each WI is equipped with a Data-LFSR. On being enabled by the MAC-LFSR, the Data-LFSR creates a packet with pseudo-random bits to be sent from the WI. This data includes the destination address of the target WI making the selection of the destination pseudo-random as well. Data padding is done to embed the source address of the sending WI in the packet. In addition, each receiver of a WI is equipped with a Wireless Security Unit (WSU) that will enable detection of persistent jamming from both internal HTs as well as external attacker. In the next subsection, the architecture of the WSU will be briefly discussed.

Architecture of WSU

In the normal mode of operation, the data flits are received at the deserializer buffer of a NoC switch equipped with a WI. Upon reception of flits at the receiver's buffer, flits are sent to the Burst Error Unit (BEU). The BEU employs the BEC proposed in [133] to detect burst errors. The corrected flits after burst error correction are sent to the input VCs of the NoC switch to be routed downstream in parallel to the error related information as discussed in the next subsection, being sent to the ML classifier, to remove the DoS detection mechanism from the critical path of the data transfer. If the ML classifier detects an attack as opposed to a random burst error, it asserts a flag to the ADU. The ADU receives the input from ML classifier and determines if the attack is internal or external as discussed below.

Machine Learning for Attack Detection

As aforementioned, the considered attacks in this chapter primarily result in causing continuous sustained burst errors in the flits (data corruption). This can be detected by observing the number of flits in error. In the proposed WiNiP, the output of BEU, which is the number of burst errors within a block, is fed to an ML classifier to detect and differentiate attacks. Multiple ML classifiers such as multi-layer perceptron (MLP), support vector machine (SVM), k-nearest neighbors (KNN), Decision tree (DT), and J48 were investigated to evaluate the robustness and efficiency of attack detection. For the MLP, a single hidden layer with 10 nodes is utilized, with two neurons in the output layer. A polynomial kernel based SVM has been utilized for detection, as it considers the combination of the input features as well as input features for classification. Similarly, experiment was also done with k-nearest neighbors with k=1 and 3 in this work. In addition, two variants of decision classifiers namely DT and J48 were tested, where J48 is an optimized version of DT with reduced search space [141].

In order to train the ML classifier, cycle accurate WiNiP simulator was modeled to operate in one of the three modes: normal, random burst errors and attack. In the normal mode, the wireless interconnects are assumed to work with the reliability level determined by the operation of the transceiver and their operating thermal noise. This type of noise is shown to result in a random BER of 10^{-10} or less [124]. The second mode (random burst errors) is modeled with higher BERs as the burst errors are contiguous bits of flits. BERs of 10^{-5} is used in this case [133]. Lastly, under DoS attack, a high BER of 0.5 is assumed as for identically and independently distributed (iid) data bits even a very high power jamming signal can cause errors only half of the time on an average. This is because the adopted modulation mechanism in these wireless interconnects is OOK, where on an average the data bits are represented as presence or absence of transmission. Therefore, a jamming signal will only cause errors when the transmission is supposed to be absent, which can be assumed to be half of the time for iid data.

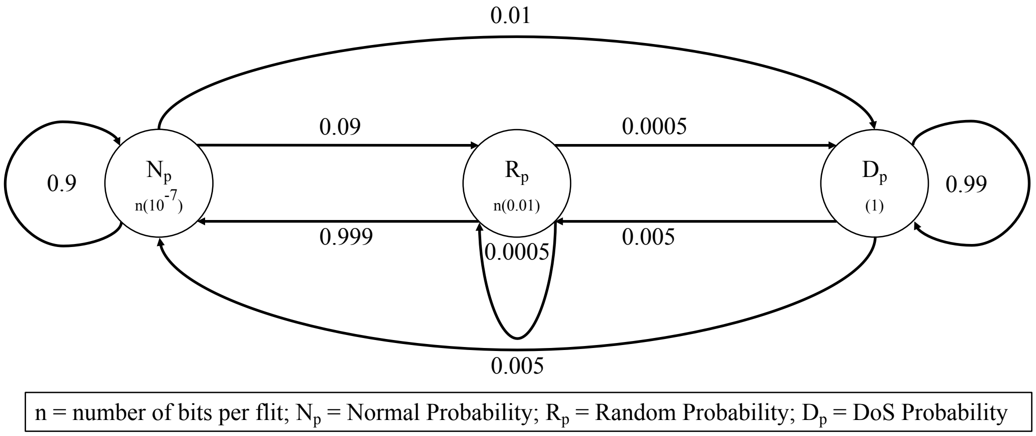

0.01
0.09
0.0005
0.9
N_p
$n(10^{-7})$
R_p
n(0.01)
D_p
(1)
0.99
0.999
0.0005
0.005
0.005
n = number of bits per flit; N_p = Normal Probability; R_p = Random Probability; D_p = DoS Probability

Figure 4.8: Markov Chain to generate training and test data.

The simulator is modeled to create flit errors based on these BERs, which are then assumed to be detected by the BEU. The simulator is made to operate in one of the three modes dynamically by using a Markov Chain driven process, as shown in Figure 4.8. The probability of staying in the attack mode, when already under attack is considered high, as a jamming attack is effective only when it is sustained for sufficiently long duration. The probability of staying in a random burst error mode when already in it is modeled low, as random burst errors are short-lived phenomena. The probability of transition into normal mode from a random burst error mode is therefore high. The specific probability values can be altered to model any particular scenario. This observed data (number of flit errors) along with the operating mode (attack class i.e., random or burst) is used to train ML classifiers.

Strengthening Attack Detection with Adversarial Learning

In order to craft the adversarial perturbations, a functionally reverse engineered ML classifier was considered i.e., a neural network with θ as the hyper parameters, x as the input to the model (communication information such as number of packets transmitted, packet errors), and y as the output for a given input x, and $L(\theta, x, y)$ as the cost function used to train the neural network. Then the perturbation required to misclassify the ongoing communication is determined based on the cost function gradient of the neural network (in this case). The adversarial perturbation generated based on the gradient loss, similar to the FGSM [135] is given by

$$x^{adv} = x + \epsilon sign(\nabla_x L(\theta, x, y)) \tag{4.1}$$

where ϵ is a scaling constant ranging between 0.0 to 1.0 is set to be very small such that the variation in x (δx) is undetectable. In case of FGSM the input x is perturbed along each dimension in the direction of gradient by a perturbation magnitude of ϵ. Considering a small ϵ leads to well-disguised adversarial samples that successfully fool the machine learning model. In contrast to the images, where the number of features are large, the number of features in our environment i.e., flit errors are limited. Thus the perturbations need to be crafted carefully and also ensured that they can be generated during runtime by the applications. For instance, a flit error higher than transmitted flits makes no sense and is impossible to implement. Hence, we include a lower bound on the adversary values that can be predicted.

Once the adversarial pattern is predicted or determined, the attacker crafts the attacks through induced errors or by spacing the attack in time so that the errors split over time as predicted. The attacker internal or external, is modeled to display the adversarial behavior as discussed above to create errors in the communicated flits only when the adversarial model allows rather than assuming constantly high BERs when in the attack state of the Markov Chain as in the previous subsection. Therefore, even when the simulator is in the attack stage, BERs may not be consistently high making the attack more sophisticated and decrease the likelihood of a detection. In order to defend against such threats, we incubate a hardener unit. The hardener unit predicts the adversarial samples, similar to the aforementioned attack and updates the ML classifier model through adversarial training [142]. The hardener is allocated off-chip (on a connected system), but it updates the weights of the ML classifier to robustify against the adversarial threats. One can argue that the adversarial training is inefficient in defending against wide range of crafted threats and large range of perturbations. However, in this given context, crafting too many vivid range of threats is not feasible due to the correlation between features. Further, large variations or perturbations can be easily caught, as large deviation in the errors clearly indicate the presence of anomaly.

Attacker Detection Unit

It is essential to differentiate between internal and external jamming attack as defense mechanism depends on attack type. The ADU takes as an input the signal from the ML classifier that detects the type of a jamming based DoS attack. On the detection of an attack through ML classifier, the ADU activates the probe mode and all the WIs operate according to the NMAC mechanism controlled by the MAC-LFSR. The MAC-LFSR generates an encoded signal which is decoded to create a one-hot signal and is sent over JTAG boundary scan chain to the transmitters of all the WIs. A parallel-load shift register is used to serialize this one-hot signal. At each transmitter this signal is ANDed with the CLK signal as shown in Figure 4.7. Thus, only one transmitter is enabled to transmits data flits over the WI in one instance.

The very first MAC is initialized as an all-zero signal to disable all WIs from transmitting. In this case, if any of the WIs still receives wireless transmission, it implies that the jamming source is an external attacker as none of the internal transmitters are powered on. An External Jamming (EJAM) flag is sent to the Defense Unit (DU). However, if in this case, there is no RF transmissions received, the MAC-LFSR progresses to further probing by cycling through the MAC-LFSR where, only one transmitter is powered on in each cycle. In these cases, where the enabled WI is not the internal attacker, there will be interference in received flits at the WIs due to continuous jamming from the attacker. Only in the case where the MAC-LFSR enables the attacker there will be no interference and correct reception will be received at the WIs. So, the algorithm declares the WI that is enabled by the MAC-LFSR in which case there is no interference, as the internal attacker. The ID of this WI and an Internal Jamming (IJAM) flag is passed to the DU. For external attacker an invalid (out of range) ID is sent.

4.5 Defense After Detection

DU implements different defensive measures based on the attack type. The ADU passes the address of the WI that is determined to be the attacker to the DU. If the address passed on to the DU indicates the address of an internal attacker, the DU sends IJAM signal to disable only the power supply to the indicated WI and updates the routing table of its NoC switch to prevent the use of the WI equipped port. Moreover, as there are at least 2 WIs in each chip, the WI that is not compromised will inform other WIs in the MCMC system to update their routing table for the compromised WI. Now, all the incoming packets at the compromised WI will be diverted to the other WI on the chip via wired links. Hence, only the HT infected WI is disabled and other WIs continue to use the wireless medium.

In case the attacker is an external agent, the DU of the enables the detecting WI to send control signal as shown in Figure 4.4(d) over the common reservation channel by setting the EJAM and attack flag on the transmitter side. The reservation channel like the other ACDMA channels is resilient to jamming. The attack condition is provided the highest priority and therefore, should be decoded immediately. As the signal has the attack flag set and is broadcast in nature, every WI in MCMC can switch the MAC mode to AMAC simultaneously and continue communication even under external persistent jamming attack. In AMAC mode, before any transmission, similar to reservation assisted NMAC mode, the

Table 4.1: Component configuration for simulation

Component	**Configuration**
System size	64 cores, Out-of-Order, 16cores/chip
Cache	32KB (private L1), 512KB (shared L2), MOESI
NoC router	3 stage pipelined 5 ports,0.078pJ/bit(wired)
Total VC	4, each 8 flits deep, 32 bits/flit
Wired NoC links	32-bit flits, single cycle latency, 0.2pJ/bit/mm
OOK transceiver	16Gbps, 2.03pJ/bit, 60GHz, 2WIs/chip
CDMA	encoder, decoder [65], 16Gbps, 0.66 pJ/bit
HBM links	128Gbps, 6.5pj/bit
Technology	65nm, 1V supply, 1GHz clock

senders use a common PN code to send non overlapping send requests. However, based on the received requests, multiple receivers can grant access to multiple transmitters as now communication happens through different ACDMA channels. As such AMAC mode uses TC protocol, different transmitters use different PN codes of the same length because of having different seed values. Each receiver stores the seed values in a small tamper-proof ROM where the address of the seeds matches their transmitter address. Therefore, the receiver already know which PN code to use for demodulation in a particular channel while granting the channel access through reservation requests and hence, reduce communication delay. Moreover, for an external attack, the ADU periodically probes the system to restore the system to NMAC mode once the external jammer is no longer active.

4.6 Results and Analysis

Here, the performance of the proposed secure WiNiP interconnection architecture under different attack scenarios has been evaluated. We also compare the performance of various ML classifiers for attack detection with and without adversarial learning. The section concludes with our study on the code length selection and system scaling.

4.6.1 Simulation Setup

Simulation of wireless interconnection requires a combination of multiple simulation tools. ASIC design flows have been utilized through Synopsys Design Compiler with 65nm CMP standard cell libraries (https://mycmp.fr/) to model the digital parts of the WiNiP such as NoC switches and the WSU. The BEU encoder and decoder is implemented as two pipelined stages in the WIs to accommodate their delay [133]. The characteristics of the antenna and the transceivers are simulated in HFSS and Cadence Virtuoso respectively. The delay and energy dissipation on the wireline links are obtained through Cadence simulations considering the specific lengths of each link based on the NoC topology assuming 20mm×20mm chips. The power and delay overheads of the NoC switches, wired and wireless links, and transceivers were considered during simulation. The power and delay overheads of the proposed WSU were also considered while running the cycle-accurate simulation. The simulation parameters are listed in Table 5.1.

The proposed system has been evaluated in terms of average packet latency and average packet energy for application-specific traffic patterns from PARSEC and SPLASH2 benchmark suites. We consider a 4 chip system with 4 in-package memory modules. The core configurations in Table 5.1 have been used to extract the core-to-memory and cache coherency traffic for these applications when they are executed until completion using SynFull [122]. In order to map these traffic patterns to the MCMC environment, multiple threads of the same application kernel were considered running on the MCMC system where each processing core executes a single thread and the memory stacks are shared among threads. Figure 4.9 shows the simulation process considering the traffic and Markov states. Before discussing the results for application specif traffic we first evaluate the performance of the ML classifiers.

4.6.2 Performance of the Attack Detection

Table 4.2 presents the accuracy and robustness of different ML classifiers when deployed to detect the DoS attacks. To compare the ML classifiers with a heuristic method as proposed in [5], a similar threshold-based approach was considered. For the neural network (MLP) a single hidden layer with 10 nodes is utilized. One can observe from Table 4.2, among different classifiers, KNN achieves high attack detection accuracy of nearly 99.87%, higher than other techniques. This behavior was anticipated, as no assumptions are made regarding the data during the training phase of KNN. Experiments were done with $k = 1, 3$ for KNN and have observed a similar performance, hence considered $k = 1$ due to its reduced complexity. Although SVM showed high accuracy, it is observed in experiments that it is not able to detect sporadic variations such as spontaneous random errors, and is hence not the best option. It can be argued that the hyper-parameters of other ML classifiers can be tuned to improve the performance, however optimizing the ML classifiers is not the focus nor contribution of this work.

During the runtime for the attack detection, the KNN classifier is fed with the information whether a flit is received or not and whether a burst error is detected or not, to detect the mode of operation of the system. The simulation data for a hundred thousand cycles was used to train each of the ML Classifiers and is then tested on a new hundred thousand cycles of simulation which were not used in training. The KNN classifier achieves a detection accuracy of 99.87% accuracy. Also, it has a Recall, F-score and Area Under the Curve (AUC) of 0.99, 0.99 and 0.99 respectively, showing high robustness. Furthermore, as shown in Table 4.2,

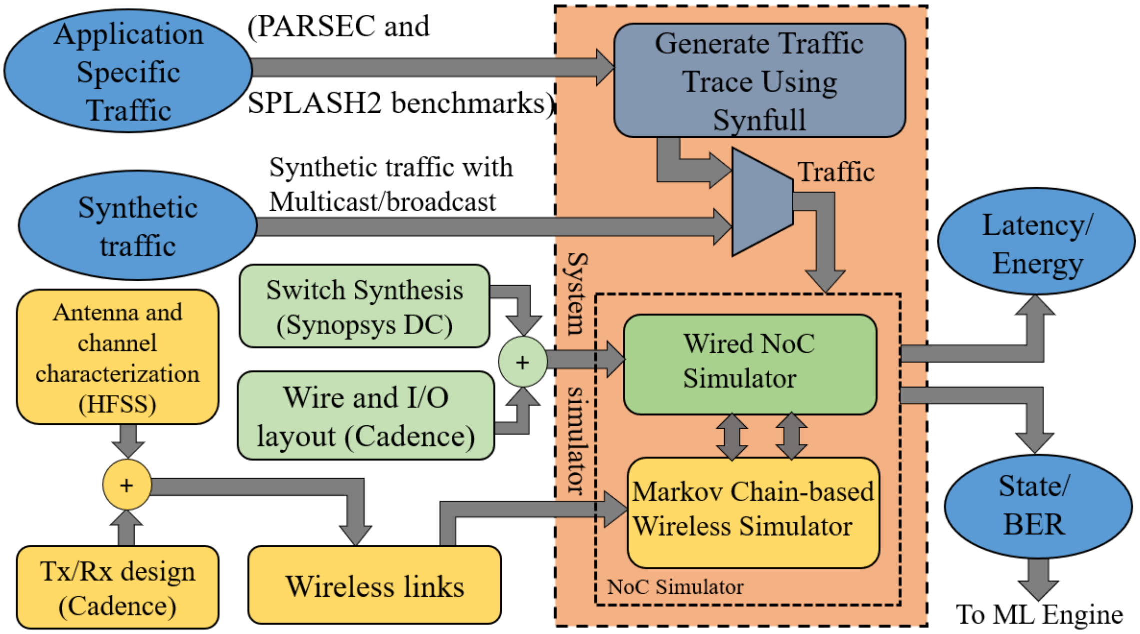

Application Specific Traffic
(PARSEC and
SPLASH2 benchmarks)
Generate Traffic Trace Using Synfull
Synthetic traffic with Multicast/broadcast
Synthetic traffic
Traffic
Latency/ Energy
System simulator
Switch Synthesis (Synopsys DC)
Wire and I/O layout (Cadence)
+
Antenna and channel characterization (HFSS)
Wired NoC Simulator
Markov Chain-based Wireless Simulator
NoC Simulator
+
State/ BER
To ML Engine
Tx/Rx design (Cadence)
Wireless links

Figure 4.9: Simulation process using Markov model.

the threshold based mechanism is not as accurate as the chosen machine learning (KNN) approach.

In this threshold-based approach, two thresholds are necessary, to separate between the attack mode, burst error mode and normal mode. The thresholds are computed based on the same data that was used to train the machine learning algorithms. The threshold between the attack mode and burst error mode is chosen to be equidistant from the average number of erroneous flits in burst errors and jamming induced errors. Likewise, the threshold to separate the burst error mode from the normal mode is chosen to be equidistant from the average number of flit errors in burst mode and normal mode. We further evaluate the detection accuracy in presence of adversarial attacks next.

4.6.3 Detection Accuracy with Adversarial Attacks

This section evaluates the impact of the crafted adversaries on the traditional ML-based threat detectors and the impact on the enhanced detector i.e., hardener unit trained with adversarial samples. Table 4.3 presents the performance of the traditional and hardener detectors. As one can observe that under normal threat conditions, the ML classifier (KNN) is able to achieve an accuracy of 99.87%. However, under the adversarial scenarios, the accuracy of the same KNN drops to 85.67%. A similar degradation in terms of performance is observed in other metrics too. Subsequently, through the adversarial training an improvement in the accuracy to 95.95% is observed with a similar trend in other performance metrics. One can observe that the performance with adversarial training makes the classifier to have lower accuracy compared to the normal classifier. However, it should be noted that in this case the system is under attack from a smarter attacker which has adversarial knowledge of the system and that without the adversarial training the WSU would be much less accurate.

In addition to the performance benefits, ML classifiers also incur silicon and resource overheads. To obtain these metrics, the post-synthesis models of the ML classifiers with 65nm standard cell libraries (`https://mycmp.fr/`) are designed using Synopsys. Table 4.4 presents the incurred overhead in terms of area, power and delay of the deployed ML Classifiers. In addition to KNN performing a good attack detection, KNN also incurs lowest area and power consumption, hence, we adopt the KNN Classifier for the evaluation of overall system. It

Table 4.2: Attack detection performance of ML classifiers

ML classifier	Accuracy (%)	Recall	F-score	AUC
ANN	47.86	0.48	0.65	0.47
SVM	98.96	0.98	0.98	0.99
KNN	99.87	0.99	0.99	0.99
DT	52.46	0.52	0.69	0.53

Table 4.3: Attack detection in presence of adversaries

	Accuracy (%)	Recall	F-score	Precision
After attack	85.67	0.94	0.86	0.79
W Adv. Training	95.95	0.97	0.97	0.97

can be argued that the delay of the KNN classifier is not the optimal, however, we choose KNN for attack detection, as the ML Classifier is not in the path of data transmission of the WiNoC, as shown in the proposed secure wireless architecture in Figure 4.7. Therefore, the ML classifier does not add latency overhead to the data transmission of the WiNiP. Despite having low latency, threshold-based approach has higher area and power consumption due to the involved floating point computations and comparisons, as shown in Table 4.4. As the hardener unit does not involve additional operations during inference rather than change in the weights of the classifiers, it does not incur additional overheads.

4.6.4 Performance Evaluation Under Persistent Jamming

As the proposed architecture takes different defensive measures for internal and external attack, in this subsection, the impact of such measures on system energy and latency has been studied using application-specific traffic patterns.

Table 4.4: Overhead analysis for different ML classifiers

Classifier	Area (μm^2)	Power (μW)	Timing (ns)
ANN	34448.79	6299.3	0.41
SVM	5412.01	8076.1	0.37
KNN	105.28	27.075	0.56
DT	127.32	41.12	0.23

Internal jamming

Disabling a compromised WI (CWI) in case of internal attack, forces the incoming flits to change its route toward the remaining WI for chip-to-chip communication. Therefore, it introduces congestion for other WI nodes and increases latency as well as energy consumption. Three scenarios were considered for our performance evaluation under internal attack. First, MCMC system with one CWI was considered for the entire (4 chip) system (1-CWI/sys). Second, an MCMC system having one CWI per chip (1-CWI/chip) was considered. These two scenarios Were compared with a wired-only MCMC system where the cores at the edges of each chip are connected to corresponding cores in the other chip with a mesh topology over high-speed I/Os.As a maximum of two WIs per chip was considered, a system having more than one CWIs in a chip indicate a complete system failure and JTAG chain can be used for MCMC communication with huge latency penalty It can be observed from Figure 4.10 that, although both the latency and energy consumption of the WiNiP increase with increasing number of compromised nodes, it is still lower than the wired MCMC system as each flit does not have to traverse through energy and latency-hungry NoC links and I/O modules. However, the average packet latency is 1.44× of the baseline system.

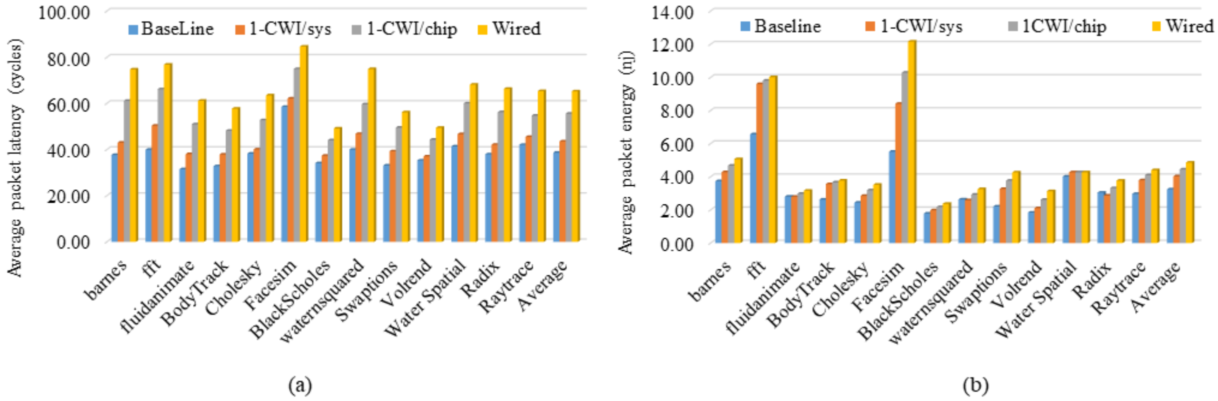


Figure 4.10: Performance evaluation (a) Latency (b) Energy under internal jamming attack for different systems.

External jamming

In the presence of an external persistent jamming attack, the MAC switches to ACDMA which ensures secure communication. However, it increases the average packet latency due to the encoding and decoding through PN sequence. Moreover, the runtime PN sequence generation through LSFRs and CDMA transceivers introduces additional energy overhead. The energy and latency overhead increases with the PN Code Length (CL). The relative performance degradation of ACDMA communication under external persistent jamming with respect to the baseline NMAC mode communication for different PN CL in bits (16b, 32b, 64b) has been shown in Figure 4.11. It can be observed from the figure that, using a higher CL increases latency and energy consumption while providing higher security.

4.6.5 Optimum CL Selection

In AMAC mode communication, the system performance and communication security are heavily depended on PN code length. Figure 4.11 shows the effect of PN CL on system latency and energy. This subsection analyzes the effect of CL on system security.

In ACDMA, all the simultaneous wireless transmissions appears as noise for a particular receiver. Moreover, the attacker can also introduce its interference noise and vary its output

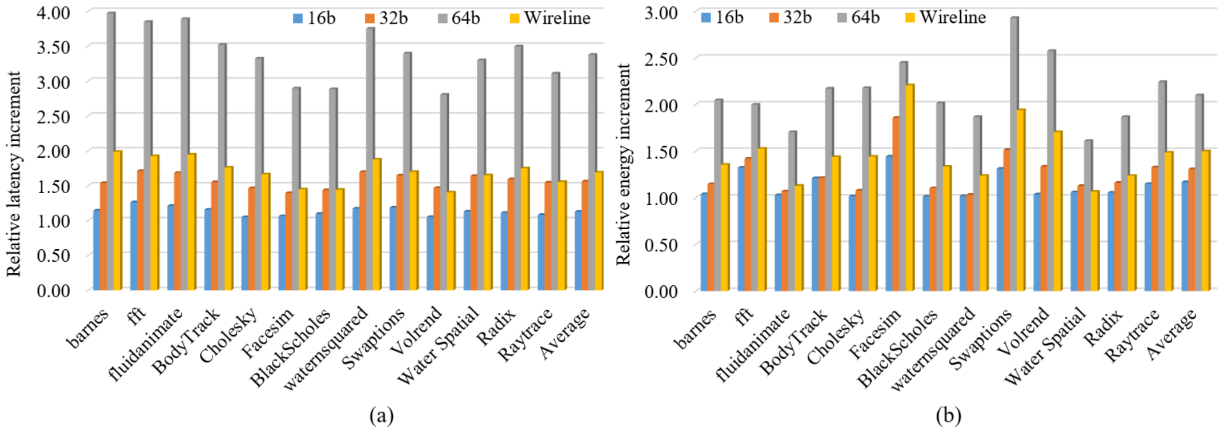


Figure 4.11: Performance evaluation (a) Latency (b) Energy under external jamming attack for different PN code length.

power to decrease the SINR. Therefore, we determine the maximum power of the attacker that can be tolerated for a reliable communication for each of the CL considered above. We target an SINR of 15dB [3] that results in a BER of 10^{-15} which is comparable of wired link's BER. For each transmitter and receiver pair we adopt the transmitter power of -23.93dBm, the noise floor of -69.43dBm and the path loss of 26.5dB [65]. For SINR calculation, one valid communicating WI pair was considered and other WIs' communication including the attacker transmission were modeled as noise in the receiver side. Figure 4.12 shows the SINR variation for various PN CL (in bits) in any receiver after considering the auto and cross-correlation among PN codes. The 16b PN code results in lower SINR although it showed better latency and energy performance in Figure 4.11. The 64b PN code though provide marginally better SINR than 32b PN, its latency and energy performance is worse than wireline interconnection architecture as shown in Figure 4.11. From Figure 4.11 it can be seen that the 32 bit PN sequence increases the average packet latency by 1.56× and average packet energy by 1.31× compared to baseline while still outperforming the wired counterpart and therefore, we choose the 32b PN code for the best trade-off between performance and security.

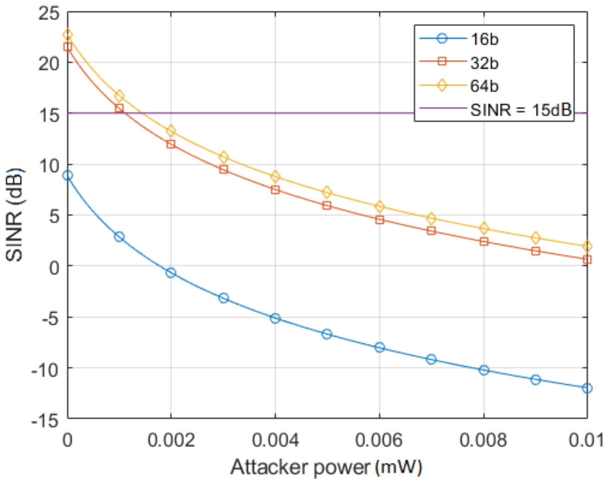


Figure 4.12: SINR value for different PN code length.

Table 4.5: Overhead analysis of WSU and Tx-Rx

Component	Area (mm^2)	Power (mW)	Delay (ns)
WSU	0.0047	1.01	1.12
Tx-Rx	0.12	23	0.0625

4.6.6 Overall Area Overhead

In the previous sections, the area overheads incurred by the ML classifiers have been discussed. Here, the overall overhead of other components of the WSU has been summarized. Based on post-synthesis RTL models in the 65nm technology node, the area overheads of the WSU is 0.0047 mm^2 per WI including the data, MAC LFSRs. The area overhead of the transceiver (Tx-Rx) is around 0.12 mm^2 [125][124], making the overhead only 3.9% of the transceivers. Table 4.5 summarizes the area, power and delay overheads of the WSU and transceiver. In WSU, only the BEU is in critical data path providing a delay of 0.8ns [133].

4.7 Chapter Summary

In this chapter, a persistent jamming-aware mm-wave wireless interconnection architecture for MCMC systems has been presented. With the proposed ML-based attack detection and defense scheme, the proposed WiNiP architecture can detect both external and internal persistent jamming-based DoS attack with an accuracy of 99.87%. Moreover, the proposed ML is also robust and shows an accuracy of 95.95% even in presence of adversaries. Most importantly, with the re-configurable MAC proposed in this chapter, the MCMC system could sustain on and off-chip communication even under persistent jamming attack with an average latency increment of 1.56× compared to baseline for a 32b PN code length. However, the secure WiNiP interconnection architecture outperformed the wired counterpart for both internal and external persistent jamming attack with very minimal area overhead.

Chapter 5

Securing WiNiP from HT-Enabled Emerging Attack on Wired NoC

In chapter 4, we talked about securing wireless communication fabric in a WiNiP interconnection architecture from persistent jamming attack. However, WiNiP interconnection architecture also has underlying wired interconnect fabric used for mainly on chip communication. The objective of designing a secure WiNiP architecture would be incomplete if we do not consider threats in its wired fabric. Due to the increased use of third-party IPs and fabless manufacturing, the underlying wired NoC fabric is highly vulnerable to new emerging threats from HTs. Given the criticality of the interconnects, the system can be severely subverted if the interconnection is compromised. The threat of HTs penetrating complex hardware systems such as multi/many-core processors are increasing due to the increasing presence of third party players in a SoC design. Even by deploying naïve HTs, an adversary can exploit the NoC backbone of the processor and get access to communication patterns in the system. This information, if leaked to an attacker, can reveal important insights regarding the application suites running on the system; thereby compromising the user privacy and paving the way for more severe attacks on the entire system. In this chapter, it has been demonstrated that one or more HTs embedded in the NoC of a multi/many-core processor is capable of leaking sensitive information regarding traffic patterns to an external malicious attacker; who, in turn, can analyze the HT payload data with machine learning techniques to infer the applications running on the processor. Furthermore, to protect against such attacks, a SA-based randomized routing algorithm in the system. The proposed defense is capable of obfuscating the attacker's data processing capabilities to infer the user profiles successfully. Our experimental results demonstrate that the proposed randomized routing algorithm could successfully reduce the accuracy of identifying user profiles by the attacker from >98% to <15% in multi/many-core systems.

5.1 Motivation

Computing demands of the modern digital world require powerful computing platforms such as multi/many-core processors or blade servers and embedded platforms with multiple processors. Interconnection networks in multi/many-core processors such as a NoC [72] that connects processing units to memory and peripherals impact the capabilities of the current systems, as efficiency of the data movement plays a pivotal role. Due to this important role played by the NoC, they form one of the largest surface areas for attack in the system both physically, due to use of use of global interconnects and functionally, as it is responsible for all data communication in the system between processors, caches and memory.

On the other hand, to alleviate the operating costs, many chip vendors are becoming fabless. Further, to minimize the time-to-market and design costs, modern SoCs use 3PIPs, which maybe procured from untrusted organizations. An adversary either at the foundry or at the 3PIP design house can introduce a malicious circuitry, to jeopardize a SoC, which is known as HT. [143]. HTs can be used for various malicious purposes, including information leakage, functionality subversion and battery exhaustion [143, 144, 145].

Considering the critical role played by the NoCs, embedding a HT that exploits the interconnection backbone can reveal the communication patterns in the system. This information when leaked to a malicious attacker can reveal important information regarding the application suites running on the system, thereby compromising the user profile. This information in turn, can enable further more severe attacks not just on the multi/many-core processor infected with the HT, but on the systems on which they are deployed. For instance, an adversary obtaining secure military information through a HT deployed in a router can subvert the military backbone, thus leading to a compromise of the national security [146].

In this chapter, first a lightweight NoC-based HT has been introduced, which, in its simplistic form, is a simple counter, which, when inserted in one or a few switches of the NoC can count the number of packets traversing the specific switches over a time window. The HT can then periodically, packetizes this count and send it to an external attacker program for payload analysis, severely compromising user profile confidentiality. This packetized count, which is the HT payload, can be subsequently analyzed by the external attacker using data processing techniques to infer the applications running on the system. To analyze the retrieved information, the attacker trains a sophisticated Machine Learning ML algorithm, that can create training samples and maps packet traversal frequencies at specific switches to the application suites. The work presented in this chapter has been able to demonstrate that the application suites running in the system can be detected with only 4 or 8 counter-based HTs with more than 98% accuracy using ML techniques. This is possible because specific routing protocols are proposed for these particular system configurations, which when adopted result in application-specific traffic patterns. Therefore, observing the traffic patterns with the help of the HTs can enable inferring the application(s) being executed in the system.

In order to defend against such a HT, a novel SA-based randomized routing algorithm has been proposed for the NoC which can obfuscate the HT-based attack discussed above. SA is a type of genetic algorithm that allows sub-optimal traversal of the search space for optimization to avoid being stuck in local optima [147]. Random packet routing over the interconnection can severely degrade performance of the system due to packets not being routed over shortest paths. Therefore, instead of simply adopting random routing, a parameterized SA-based approach has been investigated that can be tuned to achieve a desired trade-off between the defense against the attack and loss in performance. Due to SA-based random routing, the path for each packet is unpredictable and therefore, makes the mapping of packet traversal frequency through specific switches and corresponding applications unreliable. It has been demonstrate through cycle-accurate simulations that this SA-based randomized routing can reduce the effectiveness of the attack. *To the best of our knowledge, the work presented in this chapter is a first of its kind where it is shown that by monitoring traffic patterns in a NoC through HTs the user profile can be compromised; and defended the system against such an attack with controlled random routing.*

5.2 Threat Model

The proposed threat model envisions a multi-user or multi-tenant server or data center where the processing engines are multicore processors connected with a NoC. Since the NoC may be procured from an organization different from the system integration designer, one or more HTs can be inserted in the routers of the NoC during the design and fabrication process. These HTs can be simple in functionality such as counting the number of packets traversing the switch over an observation window and sending that information to an external attacker in order to thwart detection. The attacker can then employ large-scale compute capabilities and algorithms to perform a traffic analysis attack on the packet count from one or more routers. This kind of attack can reveal the applications running on the system, since the traffic interaction in multi/many-core processors is always application dependent, and thus, compromise user privacy by revealing the applications that is being executed. Figure 5.1 shows the attack model discussed here. The functionality and design of the HT is described next.

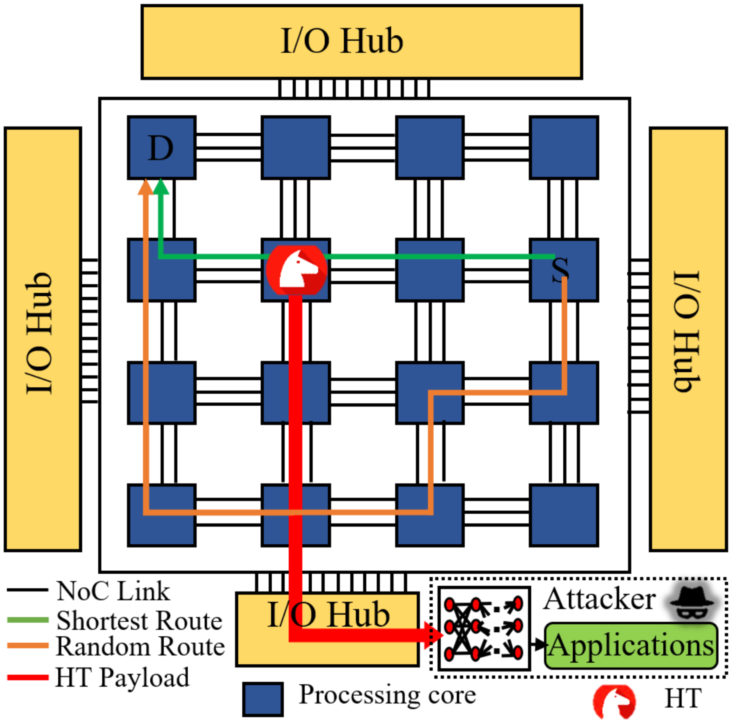


Figure 5.1: Multicore NoC with proposed threat model.

5.3 Hardware Trojan Design

This section describes various aspects of the particular attack model that have been studied. The HT in particular is a counter that is capable of counting the number of packets getting routed by the switch where it is inserted. It was assumed that a HT has been inserted inside the routing block of the NoC or interconnection switch. There maybe one or more such infected switches in the system. The functionality of this counter is that it will count up whenever a new packet accesses the router of the switch to get routed to its next destination. After counting for a pre-determined duration it packetizes this count by appending a destination address and other header information and inserts this packet into the NoC as the HT payload. Such processing systems are typically multi-user platforms where different parts of the processor are virtualized and allocated to various applications from myriad of users. As the attacker can be disguised as one of the multiple users of this shared virtualized multi/many-core processor hosting multiple users simultaneously, the destination of the payload will be a legitimate I/O port where the malicious program is hosted. Therefore, the packet is unlikely to be flagged by any other security measure in the system, as it does nothing anomalous compared to other packets in the system. The payload then gets analyzed by the external attacker.

The HT is a 16-bit counter that counts the number of packets being routed through the switch. This count is packetized every five thousand cycles and sent to the external attacker. The observation window is maintained by another 16-bit down counter that functions as the timer. The payload is launched from the HT when this timer expires. As the packets consist of multiple flits [22], taking several clock cycles to be routed through a switch [22], the maximum packet count will be less than the duration of the counting and hence the 16-bit counter will be sufficient. This particular HT also does not impact the data path of the legitimate packets getting routed in the NoC as it is not sequential to the routing logic and the counting happens in parallel to the routing. Therefore, timing analysis can not detect the HT(s). Moreover, even a few such counters of these moderate sizes are also undetectable in the large multi/many-core processors both in terms of area overhead or power consumption as even a single NoC switch with a size of ∼30-40K gates [22] is orders of magnitude more complex compared to the HT and also has many register files resembling counters.

5.3.1 Hardware Trojan Trigger Design

It was envisioned that the proposed HT does not lie in the data-path of the packet transfer mechanism over the NoC. It recognizes the flit-type and in case of a header, it increments a counter. This happens in parallel with the function of the routing block. Due to no impact on delay, the HT is difficult to detect based on timing analysis methods alone. Moreover, due to the nature of the payload as described below, the area and power footprint of the HT is negligible compared to the NoC or the entire processor. Therefore, the HTs can remain in always-on state without the need for a sophisticated trigger.

In lieu of an always on Trojan, a conditional trigger, like a counter-based trigger or an input combination-based trigger can be designed [143]. However, we do not use such an approach for two reasons: (1) The conditional Trojan incurs extra overhead in terms of hardware, power and area; which may lead to detection, and, (2) our proposed HT needs to collect data for a continuous time in order to analyze efficiently, as described in Section 5.3. Therefore, turning on the Trojan conditionally is not beneficial and can lead to information loss.

Since the proposed HT is always triggered, the probability of detection analysis framework is not applicable here [148]. The HT payload is the number of packets that traversed the particular router over the observation window.

5.3.2 Off-Chip Hardware Trojan Payload Analysis

An HT payload analyzer was crafted using ML techniques to demonstrate the threat that such an HT based attack can pose. For the purpose of training the machine learning classifier, an Artificial Neural Network (ANN) was chosen due to its ability to map complex patterns efficiently as well as its resilience to variations. Initially a dataset was built consisting of eighty different features (payloads) obtained from simulating a system with 64 cores and 16 memory controllers, with permutations of twelve different applications with no more than three running on the system simultaneously. It was assumed that a maximum of three applications running simultaneously in the system signifying up to three independent users hosted on the system executing three applications running simultaneously although a larger number of users can be easily accommodated only requiring creating training data with that assumption. This data was created in the same manner as an actual attacker who may be able to create using an emulator or a simulator of the real system. Specifically, we use a cycle-accurate simulator described in Section 5.5, which monitors the movement of packets broken down into flits in a NoC. Various permutations of the applications from a common parallel benchmark suites [122] were executed on the simulator to create traffic traces that were visible to the HTs and those traffic traces consisting of number of packets traversing HT infected switches were used as the training dataset.

As the traffic patterns i.e., packet counts depend on the executing applications and the on-chip traffic, as well as the mapping of the applications to the cores, the observed patterns will not be constant for a given set of applications. To capture such variations, a random noise following a negative binomial distribution has been added with the simulation data as a part of data augmentation and making the ANN robust to such variations. For generating this random noise, a M/M/1 queuing model was considered for the NoC routers and a Poisson arrival process for the incoming packets in each cycle. For such queuing model, the number of packets in the router buffers at every simulation cycle follows a geometric

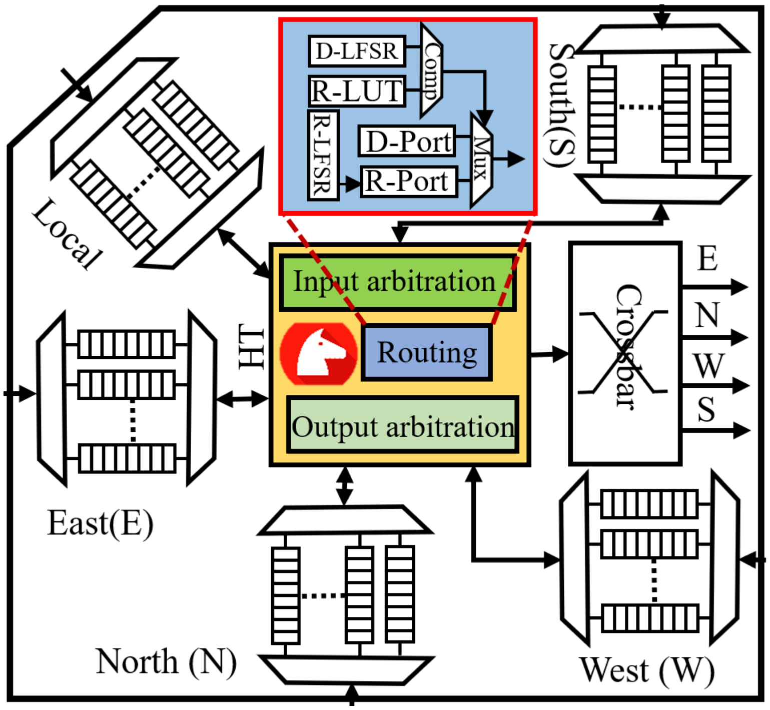


Figure 5.2: NoC switch components with inserted HT.

distribution and the total number of packets in the buffers over the simulation period can be computed as the convolution of the geometric distributions resulting in a negative binomial distribution [149], which is used for the random noise generation. Furthermore, the ANN model was trained with all the features available. This yields high accuracy in predicting the application(s) running on the system. However, the caveat here is that although the ANN could be trained on a large set of features, yet, it is not feasible to expect a HT to snoop on all switches simultaneously. Doing so would contribute a significant latency and overhead, eventually leading to HT detection. Thus, to reflect the real-world scenario, the attacker intends to insert a HT which focuses on the dominant features that still enable it to predict with high accuracy. We translated this approach by using correlation-based feature selection i.e., determine the routers in which the HTs need to be embedded (performed offline) that reduces latency without compromising the performance.

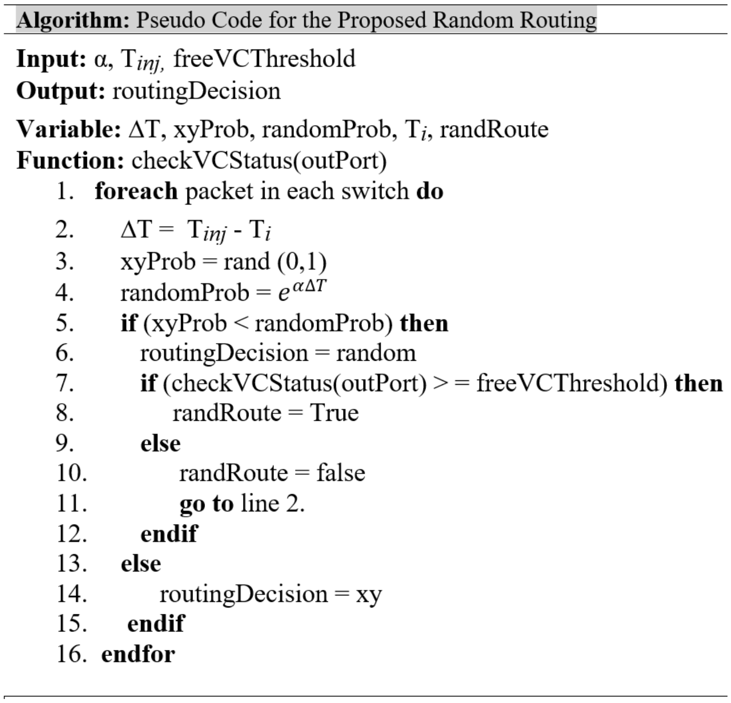
To be particular, a 5 layer ANN (N-800-500-200-64-12 neurons; N represents number of features at input) was deployed with ReLU as activation functions for hidden layers and softmax as the activation function for the output layer. There are twelve final outputs for the ANN, which are the number of individual applications used to build the dataset. To represent simultaneous execution of multiple applications, one-hot encoding is utilized. A 5-fold cross-validation is utilized to analyze performance, determined based on grid-search. As the ANN is deployed off-chip, the area and other overheads are not of concern, except the accuracy, precision, recall and F1-score.

5.4 Obfuscation using Random Routing

The proposed random routing should not only increase NoC resilience against aforementioned and similar attacks, but also ensure low latency for latency-sensitive messages. Additional issues such as deadlock, live-lock, and in-order packet delivery also need to be addressed. Here, we discuss the proposed routing methodology, live-lock and deadlock freedom.

5.4.1 Routing Methodology

The proposed defense mechanism against the HT-based traffic analysis attack, is based on obfuscating the traffic analysis mechanism by introducing controlled randomness in the routing. The underlying principle is that with deterministic routing, the number of packets through a particular router is highly correlated to the application or the architecture of the system. However, if the routing is based on random walk, the number of packets through any particular switch is essentially random, losing predictable correlation with underlying system parameters. However, completely random routing is shown to increase latency and negatively impact performance of the system. For the work presented in this chapter, a distributed random routing for NoC architectures based on SA heuristics has been proposed. Such SA-based random routing methodology enable designers to have a degree of controllability over the randomness of the routing decisions. According to SA heuristic, which is used to approximate the solution of an optimization problem, at the beginning of the annealing schedule, the probability of taking random non-optimal decisions are higher. However, the probability of accepting random non-optimal solution decreases as temperature is reduced



```
Algorithm: Pseudo Code for the Proposed Random Routing
Input: α, $T_{inj}$, freeVCThreshold
Output: routingDecision
Variable: ΔT, xyProb, randomProb, $T_i$, randRoute
Function: checkVCStatus(outPort)
  1.  foreach packet in each switch do
  2.     ΔT =  $T_{inj}$ - $T_i$
  3.     xyProb = rand (0,1)
  4.     randomProb = $e^{\alpha \Delta T}$
  5.     if (xyProb < randomProb) then
  6.        routingDecision = random
  7.        if (checkVCStatus(outPort) > = freeVCThreshold) then
  8.            randRoute = True
  9.        else
  10.           randRoute = false
  11.           go to line 2.
  12.       endif
  13.    else
  14.        routingDecision = xy
  15.    endif
  16. endfor
```

Figure 5.3: Pseudo code for the proposed random routing.

according to the annealing schedule. Similar to the SA, in the proposed routing algorithm, initially the probability of selecting a random output port for each new injected packet in the NoC fabric is high and decreases over the lifetime of the packet in the NoC. Therefore, with progression of time, the routing decisions are constrained to be optimal (shortest-path) with only occasional random sub-optimal decisions. Figure 5.1, shows the shortest (in green) and SA-based random path (in orange) for a particular Source, (S)-Destination, (D) pair. Similar to the Metropolis criterion in SA, the probability of taking a random output port for the next hop is defined as

$$R_i = e^{\alpha(T_{inj} - T_i)} \tag{5.1}$$

Where T_{inj} is the packet injection time and T_i is the current time in clock cycles. This time difference, ΔT is embedded in the packet header and incremented in each clock cycle. α is a designer parameter that controls the degree of randomness in the routing and will be described in later sections. As shown in Figure 5.2, any switch in the routing path of the packet generates a random number using a Linear Feedback Shift Register, (D-LFSR) which represents the probability of taking the shortest path. If this number is less than the probability stored in a Random Look-Up-Table, (R-LUT) based on equation 5.1 and current ΔT value, then the proposed routing chooses any of the ports of the switch that leads to non-optimal path with equal probability. The non-optimal random port, (R-Port) is generated by another LFSR, (R-LFSR). Otherwise, the optimal port is chosen by the router depending on the adopted shortest path algorithm.

However, according to equation 5.1, the probability of taking non-optimal, random paths decreases with the current time, T_i. Therefore, with elapsed time, each packet is more likely to take the optimal routing decision determined by the adopted optimal routing algorithm in the system. Thus, each packet is guaranteed to reach its destination as the time progresses and therefore, the proposed SA-based random routing is guaranteed to be live-lock free. For this work, shortest path routing based on Dijkstra's algorithm has been adopted in the proposed NoCs architectures as it is applicable to multiple NoC topologies such as trees, meshes and random topologies. Dijkstra's algorithm, extracts a MST providing the shortest path between any pair of nodes in a graph. We consider each switch to be equipped with a forwarding table having the list of pre-computed shortest path to reduce routing delays that will be caused by route computations once for every packet.

To ensure application performance, latency critical messages should experience less randomness in routing compared to other messages. Therefore, to control the degree of randomness in routing, the proposed methodology introduced the parameter α in equation 5.1. For latency critical messages a higher α value is preferred as it rapidly diminishes the probability of taking non-minimal paths. The latency critical messages then follow the shortest path determined by adopted shortest path algorithm in the system. Other latency-tolerant messages can have smaller α value to have more randomness in their routing and ensure better NoC resiliency against threats. The value of α was considered to be a tunable parameter to be determined by the designer keeping the security-performance trade-off, applications, and run environment in mind.

5.4.2 Deadlock Avoidance and In-order Packet Delivery

In any random or dynamic routing, deadlock is one of the most common and challenging issues to address. Although the proposed routing takes random paths for each new injected packet, it eventually follows the Dijkstra's path as the time elapses. Dijkstra's routing is inherently cyclic dependency free. Therefore, for the proposed routing, deadlock should be avoided only when the routing is more likely to take random paths. To avoid deadlock, the VC occupancy of the output ports have been considered before routing an incoming packet. If the VC occupancy of an output port is less than a pre-determined threshold, the packet is not routed immediately and waits in the input VCs to be rerouted. This ensures that a heavily utilized port causing a deadlock is avoided. Figure 5.3 describes this mechanism by using a pseudo-code for the proposed routing algorithm. Additionally, due to the exponential decaying probability of taking random paths, a packet is more likely to be routed toward the deadlock free port following the Dijkstra path as elapsed time increases.

As each packet follows a different path, such random routing also need to ensure in-order packet reception for each message in the system. For this work, the in-order packet delivery mechanism described in [150] has been adopted, where a lookup table entry is compared with the packet identifier of a message at the re-convergent switches. If the identifiers match, the packet is granted arbitration and the look-up table identifier value is incremented and thus in-order packet reception is ensured without significant re-order buffer overheads.

Table 5.1: Component configuration for simulation

Component	**Configuration**
System size	64 cores, Out-of-Order, 16cores/chip
Cache	32KB (private L1), 512KB (shared L2), MOESI
NoC router	3 stage pipelined 5 ports,0.078pJ/bit
Total VC	4, each 8 flits deep, 64 bits/flit
Wired NoC links	64-bit flits, single cycle latency, 0.2pJ/bit/mm
Technology	65nm, 1V supply, 1GHz clock

5.5 Evaluation

In this section, the performance of the proposed SA-based routing will be evaluated in terms of average packet latency for different α values as discussed in Section 5.4.1. Also the ML detection accuracy will be compared as a measure of security between our proposed and deterministic routing.

5.5.1 Experimental Setup

To evaluate the ML accuracy and average latecncy, a 64 core system arranged in a regular 8x8 mesh NoC fabric with 4 in-package memory modules was considered. The core configurations in Table 5.1 have been used to extract the core-to-memory and cache coherency traffic for PARSEC and SPLASH2 benchmark suites when they were executed until completion using SynFull [122]. In order to map these traffic patterns to the 64 core NoC environment, it has been considered that multiple threads of the same application kernel running on the system where each processing core executes a single thread and the memory stacks are shared among threads. Moreover, other NoC configurations such as number of VCs, topology, cache distribution/access using Uniform Memory Access, (UMA) and Non-Uniform Memory Access, (NUMA) have been varied to analyze corresponding ML results. Table 5.2 lists the NoC architectures considered in this chapter.

The evaluation framework used ASIC design flows with Synopsys Design Compiler with 65nm CMP standard cell libraries (https://mycmp.fr/) to synthesize the NoC switches. The

Table 5.2: Architectures for evaluation

Architecture name	**Topology**	**VCs**	**Cache access**
Mesh	Mesh	4	UMA
V2Mesh	Mesh	2	UMA
DMesh	Mesh	4	NUMA
Torus	Torus	4	UMA
FTorus	Folded Torus	4	UMA

delay and energy dissipation on the NoC links are obtained through Cadence simulations considering the specific lengths of each link based on the NoC topology assuming 20mm×20mm chips. The power and delay overheads of the NoC switches, NoC links were considered during simulation. In order to train the ML classifier, a cycle accurate simulator based on NoXim [127] was modeled to implement the proposed SA-based random routing and track the number of packets passing through each switch for each of the traffic traces. Figure 5.4 shows the flow diagram for performance evaluation and attacker training data generation.

5.5.2 Average Packet Latency with α Variation

Although it has been considered that the degree of randomness, α to be a designer's parameter, this section studies the latency distribution of the routed packets and variation in latency distribution for different values of α. For analysis, the value of α was considered to be 100, 0.01 and 0.005, where $\alpha = 100$ represents deterministic routing. The corresponding

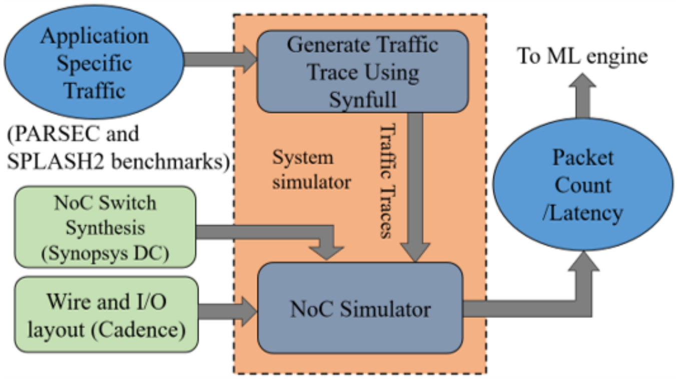

Application Specific Traffic
(PARSEC and SPLASH2 benchmarks)
Generate Traffic Trace Using Synfull
To ML engine
System simulator
Traffic Traces
Packet Count /Latency
NoC Switch Synthesis (Synopsys DC)
Wire and I/O layout (Cadence)
NoC Simulator

Figure 5.4: Simulation flow diagram for performance evaluation and attacker training.

effect of having different α values on latency has been observed on a 8x8 mesh connected NoC running Fast Fourier Transform, (FFT) traffic traces obtained from SynFull [122]. Following equation (1), higher α values result in a lower probability of taking more random paths which diminishes rapidly with elapsed time. Therefore, with a large value of $\alpha = 100$, the average packet latency is same as the average packet latency achieved using deterministic routing as it represents no randomness in the routing. From the latency distribution shown in Figure 5.5 (a), it is observed that the most of the packets were delivered to their destinations within 30-55 cycles using deterministic routing. On the other hand, smaller values of α increases the probability of taking more random paths and hence increases average packet latency along with its distribution spread. Figure 5.5 (b) and (c) show the distribution of packet latency for $\alpha = 0.01$ and $\alpha = 0.005$ respectively. From Figure 5.5 (b) and (c), it can be observed that the majority of the packets incurred a latency between 125-175 and 150-275

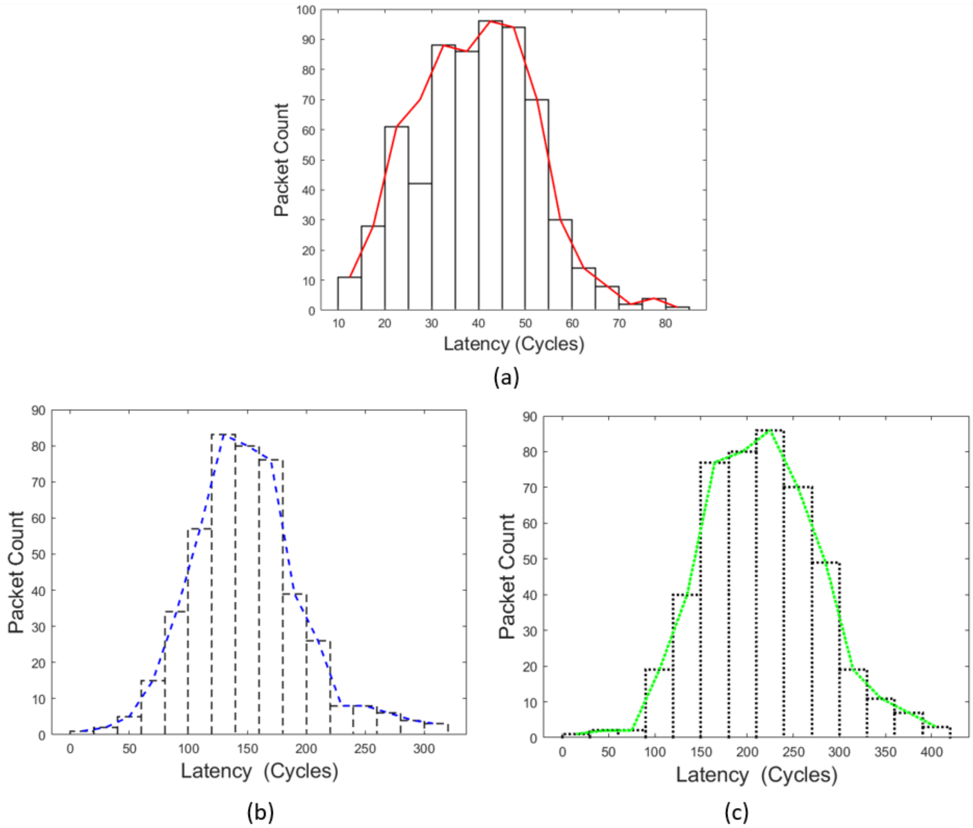


Figure 5.5: Latency variation for (a) $\alpha = 100$ (Deterministic) (b) $\alpha = 0.01$ and (c) $\alpha = 0.005$. .

cycles for $\alpha = 0.01$ and $\alpha = 0.005$ respectively. The average packet latency for $\alpha = 100$, $\alpha = 0.01$, and $\alpha = 0.005$ were found to be 38.98, 149.45, 215.56 cycles respectively for FFT traffic. This latency penalty can be thought as the cost of improved security provided by the proposed random routing. Keeping this security-performance trade off in mind, designers can choose α values that can provide meet system performance and security requirements making such NoC routing secure-by-design. It is also interesting to note that, with smaller α values, the standard deviation (σ), in latency also increases. The standard deviation for deterministic, $\alpha = 0.01$, and α= 0.005 was 12.52, 47.05, and 62.52 cycles respectively. The 99^{th} percentile of the latency distribution was approximately at 70, 270 and 360 cycles for deterministic routing, $\alpha = 0.01$ and α= 0.005 values respectively providing designers upper bounds of latency for their design guidelines.

5.5.3 ML Performance with Deterministic Routing

Here, the effectiveness of the proposed attack model will be measured for the architectures considered in Table 5.2 using Dijkstra's shortest path routing only. Dijkstra's routing has been considered as it can be generalized to all topologies and in the case of mesh topologies is identical to the dimension-order routing yielding same latency performance. The packet count is leaked by the inserted HT as payload to the ML-based attacker in a periodic interval of 5000 cycles. Therefore, as attack efficiency metric, the accuracy, F1-score, recall and precision of the ML engine have been considered which is placed on the attacker side. The number of observed features (number of inserted HTs) were also varied in the system and the change in the performance metric was measured. As shown in Figure 5.6, for a 64 core system, the attack efficiency increases with the increase in feature size. Even with a feature size of 4 i.e., with 4 HTs embedded, the attack accuracy varies between 85-98% for all the architectures considered in Table 5.2. A similar trend can be observed for the other metrics as F1 (98%), recall (97.9%), and precision (98%) with 16 features, but omitted for brevity. Thus, it can be concluded that the proposed attacker can efficiently interpret the user profile with very small footprint/overhead within the chip.

5.5.4 ML Performance with Proposed SA-based Routing

In this section, the performance of the proposed attack model with the proposed SA-based random routing has been discussed. As the degree of randomness for the proposed random routing can vary depending on α, we consider $\alpha = 100$ (deterministic), $\alpha = 0.01$, and $\alpha = 0.005$ and analyze the accuracy of the attacker with different feature sizes. Figure 5.7 shows the accuracy of the attacker on folded torus architecture for deterministic and random routing. It can be observed from Figure 5.7, that due to the higher routing obfuscation

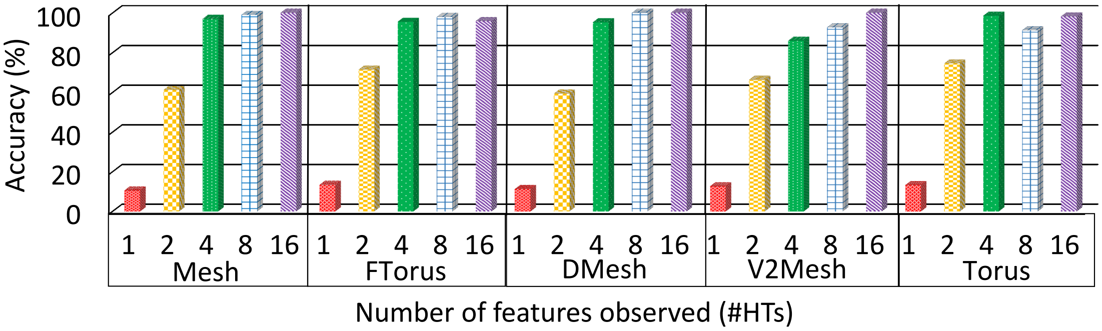


Figure 5.6: Plot of accuracy observed for deterministic routing with different features (number of HTs observed).

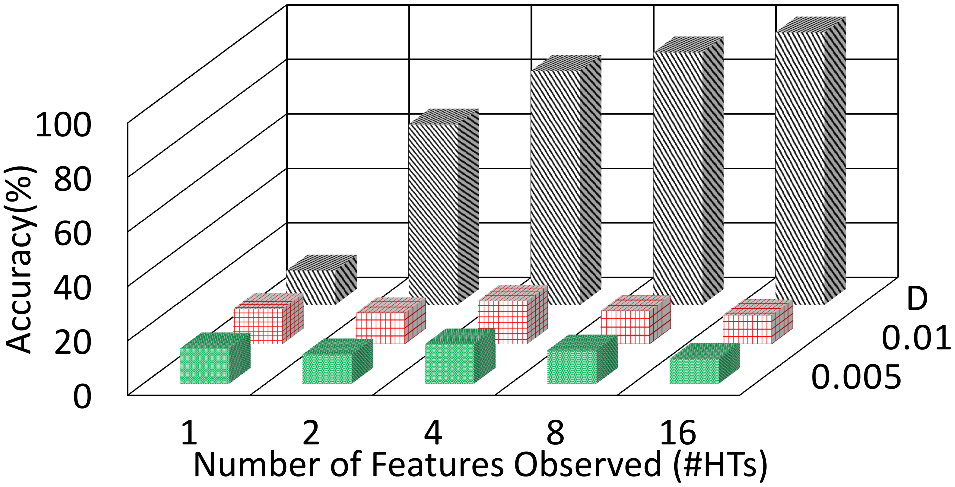


Figure 5.7: The performance with proposed routing with $\alpha = 0.01, 0.005$ and Deterministic routing (D).

introduced by decreasing α values in the proposed SA-based random routing, the accuracy of the attacker falls down significantly (<15%) compared to deterministic routing for smaller α values with higher feature sizes. Moreover, from Figure 5.7, it can also observed that, for smaller α values, increasing feature size, does not increase the accuracy significantly and thus, represents the robustness of the proposed random routing. If we combine Figure 5.7 and Figure 5.5, it is interesting to note that, the attacker accuracy is similar for both $\alpha = 0.01$ and $\alpha = 0.005$, however, using $\alpha = 0.005$ increases average system latency by 46.13% compared to $\alpha = 0.01$. Therefore, it can be concluded that, reducing α further to zero will realize a completely random routing, but the increase in obfuscation will be marginal while resulting in significant increase in average packet latency.

5.5.5 Evaluation of ML Classifiers for an Efficient Attacker Design

Multiple ML classifiers such as ANN, SVM, KNN, and DT were investigated to evaluate their accuracy for application detection to determine an efficient computing model for the proposed attacker. A 5 layer ANN (N-800-500-200-64-12 neurons; N represents number of features at input) was deployed with ReLU as activation functions for hidden layers and softmax as the activation function for the output layer. For simplicity, a linear kernel based SVM has been utilized for application detection. Similarly, experiment was also done with k-nearest neighbors with k=12 and 78 in this work. As k=12 and k=78 provide similar performance, we enlisted only the performance for k=12 in both Table 5.3 and Table 5.4 as it provides less delay and computation overhead. We considered 12 application-specific traffic obtained through SynFull [122] from PARSEC and SPLASH2 benchmark suites with maximum of 2 applications running simultaneously. For each of the ML model, a 16 feature input was considered. From the performance metrics listed in Table 5.3, it can be observed that the ANN, KNN, and DT classifiers have similar performance in detecting application suites running within a NoC using deterministic routing. Therefore, it can be argued that any of these classifiers can be used to design an efficient attacker to launch proposed traffic analysis attack. However, apart from the 12 base classes (for 12 applications), considering a maximum of 2 simultaneous running applications, the KNN, SVM, and DT classifiers required additional ${}^{12}C_2 = 66$ classes to be created during training to enable detection of 2 simultaneously running applications. Therefore, the complexity of training and computation

Table 5.3: Performance of the ML classifiers for deterministic routing

ML classifier	Accuracy (%)	Precision	Recall	F-score
ANN	98.18	0.98	0.98	0.98
SVM	74.3	0.72	0.72	0.74
KNN	98	0.98	0.98	0.98
DT	98.9	0.98	0.98	0.98

overhead of those ML classifiers increase as the attacker intends to detect larger number of simultaneously running applications. For example, for a similar SoC, to detect a maximum of 4 simultaneously running applications, the attacker needs to create $^{12}C_4 = 495$ additional classes to train those classifiers. The number of required classed also increases as the number of baseline applications which is 12 in our case, increases. Due to the increased training complexity and the huge time required by the classifiers like KNN and DT, an attacker with such ML classifiers does not represent an efficient attacker as it is not scalable in terms of detecting multiple applications in a typical ever increasing multi user environment. On the other hand, due to the softmax activation at the output layer and one-hot encoding, ANN can detect multiple applications without any additional complexities and hence best suites the attacker purpose for the proposed traffic analysis attack.

Moreover, from Table 5.4 it can also be observed that for a NoC employing SA-based routing, the ANN classifier provides higher application detection accuracy compared to other ML classifiers. As from attacker's perspective it is desired that the ML classifier should provide a better accuracy irrespective of the routing algorithm used in a NoC, from Table 5.4 it can be concluded that an attacker equipped with ANN classifier can provide better application detection accuracy compared to other ML classifiers even in presence of novel routing algorithms like SA-based routing proposed in this chapter. In summary, due the low complexity in detecting multiple simultaneous applications in a large multi user environment and better accuracy in presence of novel routing algorithms, the ANN classifier best serves the purpose of an attacker targeting such traffic analysis attack on a NoC-enabled SoC. Although it can also be argued that various other experiments can be done and the existing classifiers can be optimized to come up with an ultimate ML classifier that best suites the purpose of the attacker, however, it is necessary to understand here that the objective of this chapter is to present the vision of a novel HT-enabled traffic analysis attack and a probable solution, not optimization of attacker efficiency. It can be explored later as a future work.

Table 5.4: Performance of the ML classifiers for SA-based routing

ML classifier	Accuracy (%)	Precision	Recall	F-score
ANN	17.42	0.10	0.17	0.11
SVM	6.41	0.01	0.01	0.06
KNN	6.41	0.03	0.02	0.06
DT	1.28	0.003	0.001	0.01

5.5.6 Routing and HT Overheads

The increased security offered by the proposed SA-based random routing comes with additional overheads in routing logic as shown in the inset of Figure 5.2. The D-LFSR was considered to be 8 bit wide whereas the R-LUT had 300 entries with each having 8 bits for each of the α values. Considering a system having three α values, the routing logic consumes additional $7685.07um^2$ of area, $1.82uW$ of power and $0.93ns$ of delay in 65nm technology node. Moreover, each HT takes $1551.6um^2$ area, $125.88nW$ power and $0.2ns$ delay for its hardware realization in same technology node.

5.6 Chapter Summary

In this chapter a novel HT induced attack has been described. Such simple HT when inserted into NoC switches can monitor and leak traffic patterns of the system to an external attacker. Sophisticated ML mechanisms can be used by the attacker to analyze the HT payload to infer the application suites running in the system with high accuracy of more than 98%. As solution a genetic algorithm-based randomized routing approach was also proposed that offers a trade-off between performance of the system and the degree of security against such an attack. Through system-level simulations it has been demonstrated that the proposed routing is able to obfuscate this particular attack and reduce the accuracy of inferring the user profile to below 15%.

Chapter 6

Conclusion and Future Works

The inefficiency in handling one-to-many traffic by state-of-the-art interconnects has become a major constraint in designing future MCMC-enabled HPC systems. Similarly, the existing and emerging security vulnerabilities on communication fabric of a systems needs to be analyzed and resolved to ensure a sustainable communication. Therefore, the objective of this dissertation was to develop a one-to-many traffic-aware, secure WiNiP interconnection architecture for MCMC systems. This chapter concludes the dissertation by summarizing the significant contributions of this research. Moreover, based on this dissertation, some promising future research directions have also been discussed later in this chapter.

6.1 Conclusion

Device scaling trend according to Moore's law will no longer be able to satisfy the ever growing performance demand of the high-end sophisticated computational devices like servers and embedded systems. In fact, we have already entered the multicore-multiprocessor era to enable faster computation by exploiting thread level parallelism. However, such multicore systems suffer from scalability issue due to process dependent faults that result in lower yield. Therefore, a system with multiple chips known as the multichip system has no other alternative that can ensure the required scalability with lower faults to provide the necessary reliable infrastructure required by future computation expensive applications. However, the performance of the multichip system is limited by the chip-to-chip communication as traffic has to go through the latency and power hungry off-chip I/Os. The situation gets even worse for one-to-many traffic as traditional wired based interconnection architecture such as NoC is mainly designed for unicast traffic and therefore, does not provide the support to ensure low latency required for such traffic having many destinations. As this traffic can have many off-chip destinations and therefore a small amount of such traffic can introduce huge local and global congestion with significant power consumption while crossing through the chip boundaries. Moreover, due to increase in memory intensive parallel applications, collective communication in parallel programming models and disintegration, increase in such has traffic has become a significant design bottleneck for future HPC MCMC systems.

To address these one-to-many traffic challenges in MCMC system, in Chapter 3 a scalable one-to-many traffic-aware MAWiNiP interconnection architecture has been proposed. The proposed MAWiNiP wireless interconnection architecture can mask the high I/O latency and power consumption for chip-to-chip communication by establishing single hop communication among multiple chips through wireless links. Moreover, the proposed MAWiNiP architecture introduces a novel asymmetric topology with a hybrid MAC that can exploit each other through a supportive flow control to prioritize and ensure faster transmission of the one-to-many traffic. As different parts of the proposed wireless interconnection architecture are individually one-to-traffic aware, the proposed MAWiNiP architecture outperforms state-of-the-art wired and wireless interconnection architectures in terms of peak achievable bandwidth and energy consumption under both synthetic and application-specific traffic. To be specific, the proposed MAWiNiP system can reduce average system latency by 47.08% compared to the state-of-the-art wired interconnection architecture. Even with the scaling of system size or cluster size, the proposed architecture outperforms the other wired and wireless architectures in terms of energy, bandwidth, and latency.

Although Chapter 3 was mainly focused on improving performance and energy dissipation in WiNiPs in presence of one-to-many traffic, no attention has been given to the vulnerabilities affecting confidentiality, availability, and integrity of on and off-chip communication in WiNiPs. Wireless being an unguided, shared transmission medium is vulnerable to many security attacks such as DoS, ED, and spoofing. Although such attack can break down the entire on and off-chip communication in WiNiP, so far, various attack methodologies and their mitigation techniques have remained unexplored. For example, to introduce a DoS attack, an external attacker can produce a high energy EM radiation that causes interference in the ongoing wireless transmission. Moreover, it is also possible that a HT planted in the system from a vulnerable design and manufacturing process can cause a WI to transmit persistent jamming signals to cause DoS for other WIs. In this case, one of the WIs infected by a HT will send data over the wireless channel irrespective of whether it is enabled by the adopted MAC mechanism. This will cause contention or interference with legitimate transmissions causing DoS on the remaining WIs.

In Chapter 4, a mechanism to detect and recover from persistent jamming based DoS attacks that can disable the wireless communication in WiNiPs has been proposed. While designing such security architecture, the existing DFT hardware has been re-used and a ML classifier was deployed to detect and defend against persistent jamming-based DoS attack. To handle more intelligently crafted jamming attacks and ensure a robust, accurate detection, and defense mechanism an AML and adversarial training for the deployed ML classifiers have been used. Moreover, under such jamming attack, specially for WiNiP architectures, it is non-trivial to synchronize and inform all other WIs about the presence of an adversary as inter-chip communication happens through only wireless medium which is itself vulnerable to the attack. To address this issue, a novel MCMC wireless communication protocol along with a reconfigurable MAC that can ensure robust and secure communication under internal and external persistent jamming attack, have been developed. With the proposed ML-based attack detection and defense scheme, the proposed WiNiP architecture could detect both external and internal persistent jamming-based DoS attack with an accuracy of 99.87%. Moreover, the proposed ML is also robust and showed an accuracy of 95.95% even in presence of adversaries. Most importantly, with the re-configurable MAC proposed in Chapter 4, the MCMC system could sustain on and off-chip wireless communication even under persistent jamming attack with an average latency increment of 1.56× compared to baseline for a 32b PN code length. Most importantly, the secure WiNiP interconnection architecture outperformed the wired counterpart even under both internal and external persistent jamming attack with very minimal area overhead.

Moreover, the threat of HTs penetrating complex hardware systems such as multi/many-core processors are increasing due to the increasing presence of third party players in a SoC design. With increasing device density and design complexity, various novel HT induced attacks are also increasing. Even by deploying naïve HTs, an adversary can exploit the NoC backbone of the processor and get access to communication patterns in the system. This information, if leaked to an attacker, can reveal important insights regarding the application suites running on the system; thereby compromising the user privacy and paving the way for more severe attacks on the entire system. Such novel HT-based attacks need to be analyzed and mitigated in advance to ensure high degree of data confidentiality in MCMC systems. In Chapter 5, we demonstrate that one or more HTs embedded in the NoC of a multi/many-core processor is capable of leaking sensitive information regarding traffic patterns to an external malicious attacker; who, in turn, can analyze the HT payload data with machine learning techniques to infer the applications running on the processor. Furthermore, to protect against such attacks, we propose a SA-based randomized routing algorithm in the system. The proposed defense is capable of obfuscating the attacker's data processing capabilities to infer the user profiles successfully. Our experimental results demonstrate that the proposed randomized routing algorithm could successfully reduce the accuracy of identifying user profiles by the attacker from >98% to <15% in multi/many-core systems.

6.2 Future Works

The research work presented in this dissertation was focused on performance as well as communication security in multichip system which are two major topics that modern HPC systems are concerned with. This research work opens up many doors in NoC security and performance domains where future extensions of this work can be directed. Some of the research directions are briefly discussed in the following sections.

6.2.1 Advanced HT and Attacker Design to Extract more System Level Information

In Chapter 5, a simple HT-based traffic analysis attack has been discussed. The proposed HT leaked the packet count passing through the compromised switch to an external attacker which was analyzed to breach the user privacy by revealing the applications running in the system. The HT used to design such attack in that chapter was an always on HT which increases the chance of detection in case of low activity even it had low power and area overhead. Moreover, to launch the attack efficiently, the attacker needed to be someone within the organization. Sophisticated HT trigger mechanism can be designed to efficiently hide the presence of such HT in the system. Trigger mechanisms that are more controllable for attacker residing outside of the organization can also be designed to make the HT more stealthy. For example, recent trends of using multicore processors in cloud computing data centers expose these systems to new threats as different co-scheduled applications are forced to share the underlying hardware resources. Therefore, in cloud computing environment, HT in a compromised NoC can be triggered or controlled by an accomplice application by establishing a run-time covert channel with the HT compromised switch. Such trigger mechanism will make an HT more attacker friendly and efficiently hide its presence.

On the other hand, the work presented in Chapter 5 only focused on detecting applications running in the system. However, the packet count in each switch is also depended on the underlying NoC topology and cache organization. Therefore, using similar analysis on the attacker side the architectural information of the NoC can also be extracted. Figure 6.1 shows the attacker accuracy on detecting the various NoC topologies using deterministic and proposed SA-based routing in Chapter 5.

Moreover, the attacker proposed in Chapter 5 assumed the NoC uses a deterministic routing to route packet and trained the ANN engine with the packet count obtained from deterministic routing. However, the attacker can be intelligent enough to train the ANN engine with data obtained from a well know and widely used pseudo random routing such as

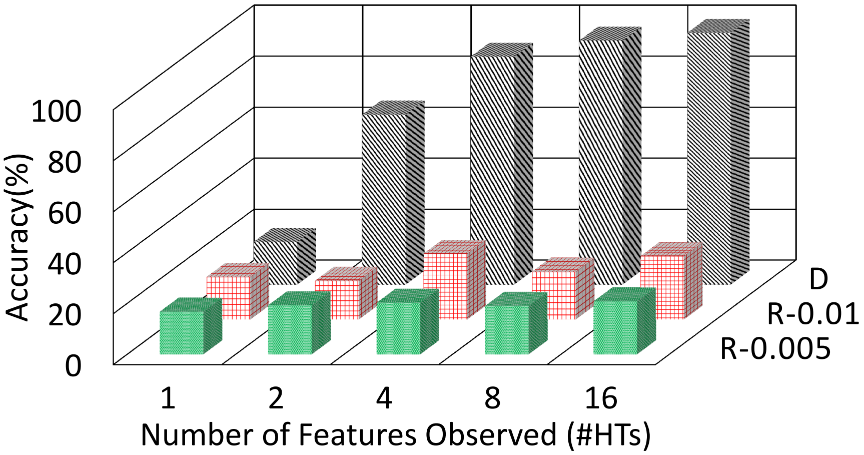


Figure 6.1: Attack accuracy with proposed routing for topology detection.

west-first routing. As the proposed SA-based routing had a some randomness in it, training the ANN with such pseudo-random routing might improve detection accuracy even the system is using SA-based random routing. In addition, designing a run time HT detection mechanism for the proposed HT will also be a very challenging work.

6.2.2 A DoS and QoS-Aware Fair Arbitration Mechanism for NoC

As the number of processing cores or the applications running in the system increases, they compete with each other for shared resources of NoC which might result in performance degradation. Different applications running in the system might require different level of services from the NoC. For example, a Guaranteed Service (GS) communication requires a consistent latency compared to fluctuating Best Effort (BE) communication. Therefore, to ensure the required service for different applications/traffic, Quality of Service (QoS) is often implemented in NoC. One of the common approach in VC-based NoC is to statically assign VC priority for particular application/traffic such as GS and serve those VCs prior to others in a particular port.

Besides, in a NoC enabled multicore/multiprocessor environment, depending on the applications running on different cores of the NoC, the message generated by those processing cores can have built-in higher priority than other messages existing in the communication fabric. Moreover, the priority of different messages from the same application on the same core can be different. For example, a request for memory access from any core should be prioritized from any other requests due to its latency sensitivity. Similarly, multicast enabled cache-coherence messages is likely to have higher priority and needs to be served faster to ensure data integrity while computation.

The priority based QoS implementation discussed in the previous segment can introduce vulnerability in the communication fabric and can easily cause DoS attack. It can be done either by running a malicious application or by an HT inserted compromised core to generate continuous high priority messages which would be served continuously by the communication fabric causing starvation for other nodes in the system and potentially leading to a DoS attack. Moreover, in a system implementing wormhole switching, the attacker can exploit such priorities and inject a packet without a tail flit to reserve a path for eternity. In worst case, these priority based QoS can be exploited to send undesired messages towards a particular destination from multiple compromised nodes causing a Distributed DoS (DDoS) attack on the destination node.

To combat such attack in NoC, we propose a DoS and QoS-aware arbitration mechanism. In the proposed DoS resilient VC arbitration we consider separate arbitration schemes for input VC and output VC arbitration. For, input VC arbitration we utilize a runtime priority generator. In this scheme the priority of other VCs requesting the same output port is increased if any particular VC has transmitted more flits than a certain threshold. We define this dynamic priority as fairness metric for arbitration and consider two seperate thresholds for GS and BE traffic respectively to ensure both DoS resiliency as well as better QoS for any incoming packet.

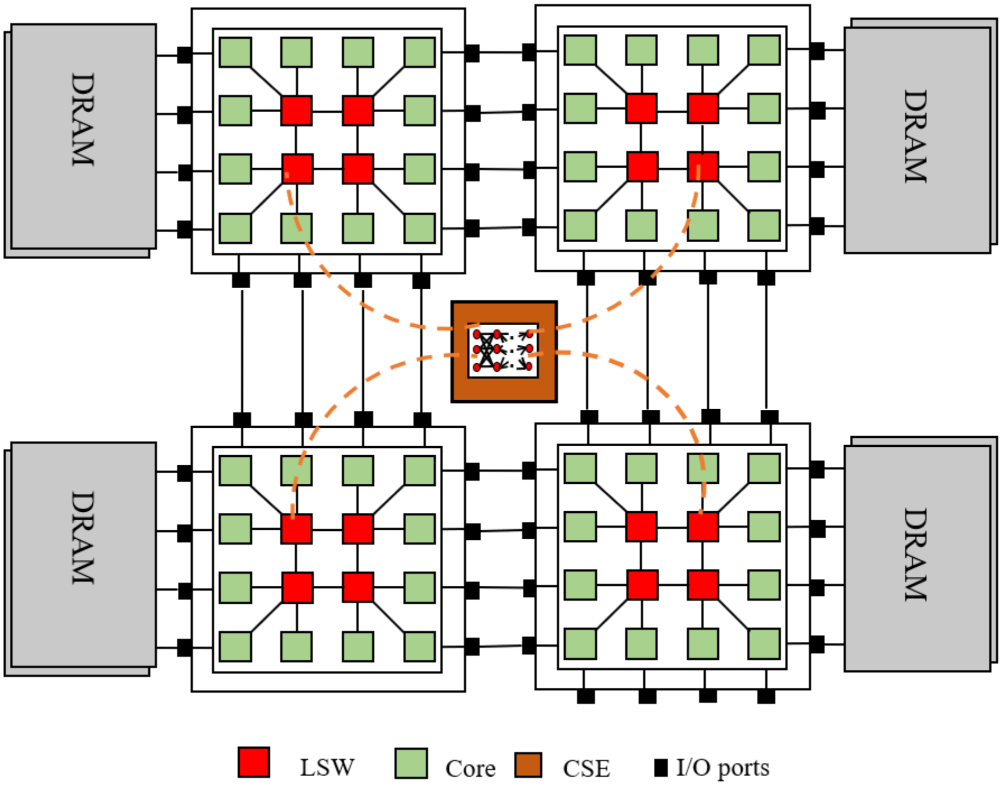


Figure 6.2: A hierarchical WiNiP security architecture for MCMC systems.

6.2.3 A Hierarchical Secure Interconnection Architecture for Runtime Anomaly Detection

Increasing use of untrusted 3PIPs, design houses or fabrication process has introduced various novel threats in modern SoCs and many threats are still evolving. In such SoCs using NoC as communication fabric, some of the attacks can be detected and mitigated with appropriate detection and prevention mechanism locally in each NoC switch. For example, if a malicious application running in a core injects too many packets to the network to flood the network, a security wrapper implanted locally in the switch can be enough to detect and prevent it. However, other kinds of attack such as packet misdirection or DDoS requires a global detection and prevention methodology.

We envision a hierarchical security architecture for MCMC communication as shown in Figure 6.2 where, we implement a Local Security Wrapper (LSW) at each switch to handle threats that can be prevented locally. However, all the LSW send data to a Central Security Engine (CSE) that can detect threats which requires data from multiple switches for accurate detection. The CSE is essentially a ML engine that tests the received data for the presence of any anomaly. The LSW/CSE communication happens through only WIs deployed in the system and we call this wireless communication plane as Security Plane as only security packet is exchanged through it. The regular on and off-chip packets are routed using wired interconnect. Such hierarchical architecture is inspired from Federated learning (FL) which is one of the recently introduced concepts by Google [151] for distributed learning while preserving the communication privacy. Using such FL methodology, a robust, training agnostic threat detection and defense mechanism can be developed.

6.2.4 Detection and Prevention of Jamming-based DoS Attack for WiNiPs Using Different MACs

In Chapter 4, the detection and prevention of persistent jamming-based DoS attack were discussed. We assumed a reservation-based MAC that uses the full channel access for wireless data transmission as our baseline multichip system. We characterized the DoS attack state based on the BER of the of received flits. However, DoS attack in a WiNiP architecture using a CSMA MAC can not be characterized by analyzing the BER of the received flits. This is because WIs in a multichip system using CSMA MAC would not transmit anything if it can sense the presence of a wireless transmission using the same carrier frequency. In such systems, DoS attack needs to be characterized by the delay of the wireless transmission. Therefore, it would be very challenging to develop an architecture to detect internal and external jamming attack for such WiNiPs using CSMA MAC.

Moreover, for CSMA MAC, along with the detection, a HT that can cause internal jamming attack on such systems needs to be designed which is nontrivial. Besides, as CSMA senses the carrier before transmission, it can be thought as an alternate solution for external jamming discussed in Chapter 4. The performance evaluation for external jamming with this solution can be compared with the previously discussed solution in Chapter 4. This also opens up the scope to analyze persistent and non-persistent jamming attack more closely and apply different solutions based on the type of jamming. Similarly, a solution to the jamming attack in WiNiPs using CDMA MAC needs to be developed as SINR in such systems are highly sensitive to attacker power.

6.2.5 A Congestion-Aware and Fault-tolerant WiNiP Interconnection Architecture for MCMC Systems

With continuous technology shrink and increase in device density, the reliability of modern devices in advanced nodes has become a significant concern. In advanced process nodes, different factors such as sub-wavelength lithography, line edge roughness, and random dopant fluctuation can cause a wide process variation, which can result in higher fault density. WIs being mostly composed of analog components are more vulnerable to faults. Besides, wired NoC links are also vulnerable to faults from electromigration, aging or any physical damage. Transient and intermittent failures can also occur due to crosstalk and noise in the signal, clock or power nets. If this faults are not addressed by the interconnection network, it can disable both on and off-chip communication. To address such faults in communication fabric, in [7], we proposed a fault-aware wireless interconnection architecture for multicore systems where we utilized a dynamic routing and two state MAC to continue communication in presence of faulty WIs. Figure 6.3 shows the topology of the proposed fault-tolerant architecture. However, research is still going on to integrate and differentiate such fault tolerant architecture with jamming-aware architecture proposed in Chapter 4 to come up with an ultimate jamming and fault-aware solution.

6.2.6 A Traffic-Aware and Secure Interconnection Architecture for Heterogeneous MCMC Systems

Heterogeneous multichip architectures have gained significant interest in HPC clusters to cater to a wide range of applications. In particular, heterogeneous systems with multiple multicore CPUs, GPUs, and memory have become common to meet application require-

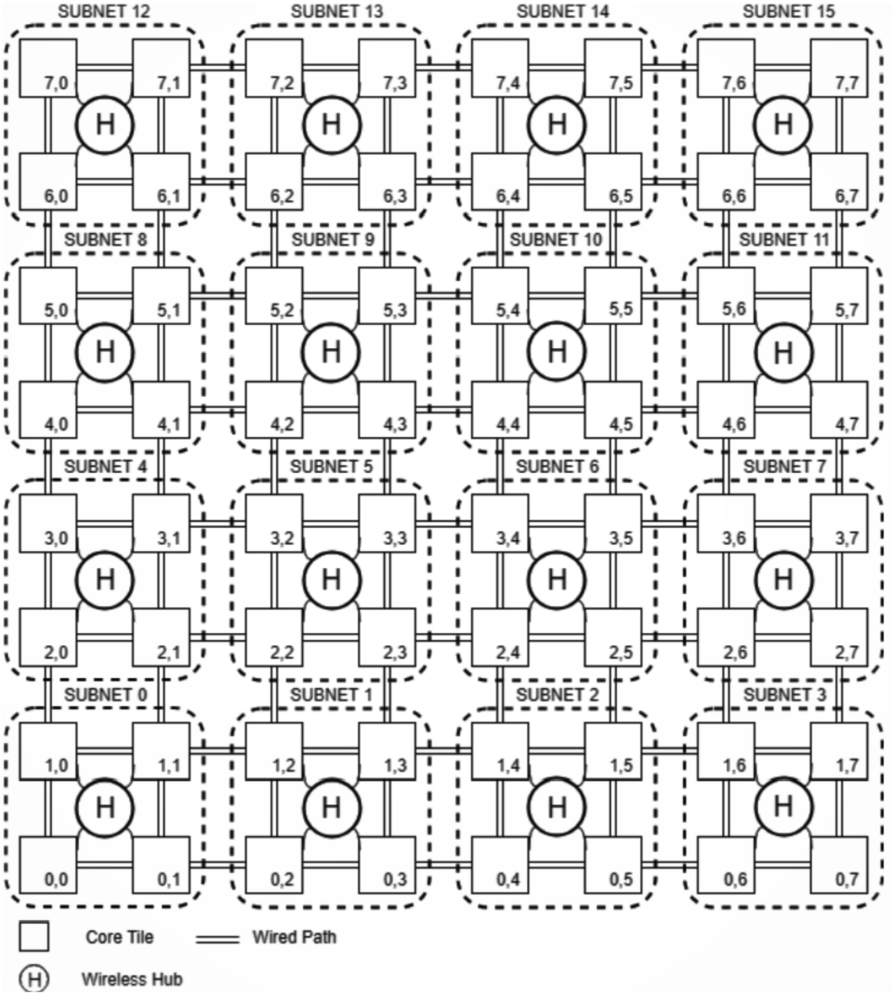


Figure 6.3: Proposed fault-tolerant WiNiP topology [7].

ments. In such CPU-GPU heterogenous multichip system, the memory access pattern significantly varies between CPU and GPU. CPU requires low latency memory access while high bandwidth memory access is required for efficient GPU computation. Keeping these different traffic requirements in mind, a heterogeneous WiNiP interconnection architecture using both directional and omnidirectional dual band antenna with hybrid MAC was proposed in [8]. Figure 6.4 [8] shows the proposed heterogeneous MCMC system. The omnidirectional antenna provided low latency communication between CPU/GPU, CPU/MC using asynchronous TDMA (aTDMA) MAC while the directional antenna provided direct high-bandwidth communication between GPU/MC using OFDMA MAC.

However, coherence traffic for GPU using directory based cache coherence protocol is significantly different than that of CPU [152]. As such multicast traffic can significantly decrease system performance, a detail analysis of coherence traffic needs to be done for such heterogenous system. Based on the analysis, implementation of one-to-many traffic-aware MAC developed in Chapter 3 can be tested or a new MAC can be proposed that improves the system performance. It will be interesting to look at how CPU and GPU utilizes the

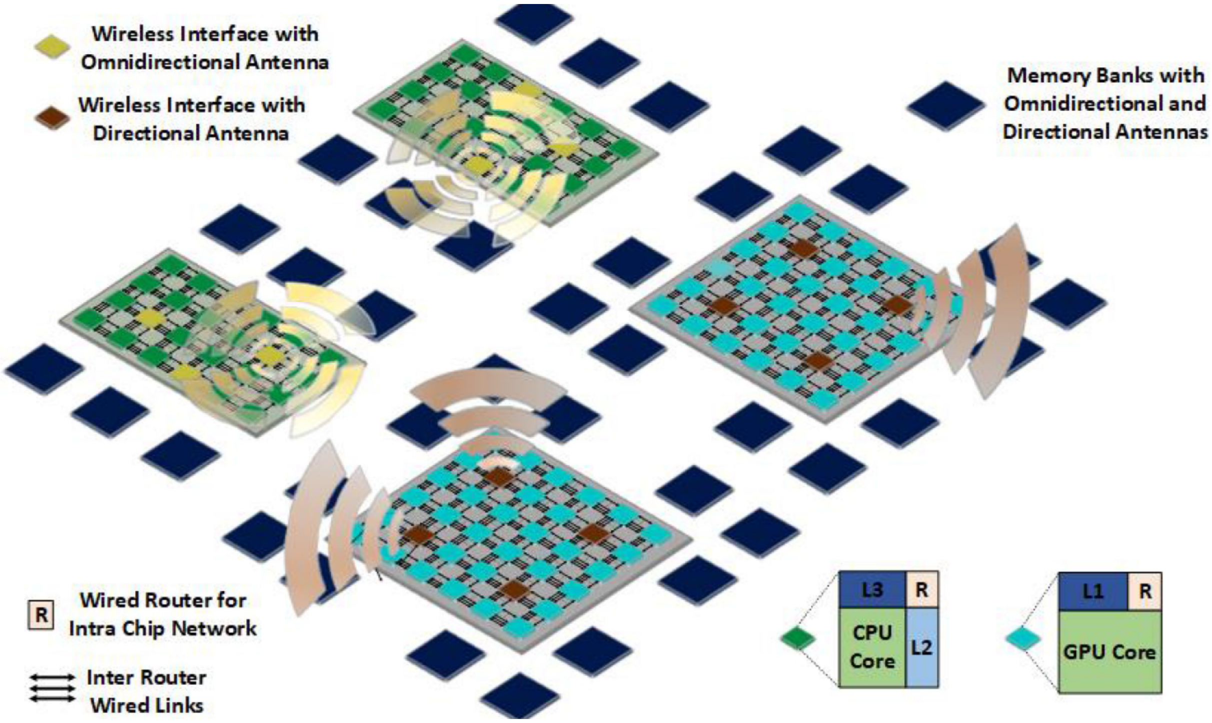

Wireless Interface with Omnidirectional Antenna
Wireless Interface with Directional Antenna
Memory Banks with Omnidirectional and Directional Antennas
Wired Router for Intra Chip Network
R
Inter Router Wired Links
L3
R
CPU Core
L2
L1
R
GPU Core

Figure 6.4: A WiNiP topology for heterogeneous (CPU-GPU) MCMC system [8].

MAC to handle their respective coherence traffic. Moreover, using heterogeneous chips in the system increases the attack surface of the whole system and hence, increase overall security vulnerability. The additional security vulnerability of the heterogenous system needs to be studied and a security-aware heterogeneous architecture needs to be developed.

6.2.7 High-speed Transceiver Design and Emerging Wireless Interconnects for Future WiNiPs

To satisfy the ever growing demand of the future HPC systems, it is critical to design the wireless transceivers in advanced technology nodes. However, designing wireless transceivers in advanced nodes is challenging due higher inductance and capacitance effect with lower voltage headroom. Appendix A shows a detail design of a OOK wireless transmitter in $45nm$ technology node which provides an energy efficiency of 1.2 pj/bit with 16Gbps link speed. Current work is going on to complete the layout of that transmitter and transceiver design in more advanced nodes will be explored in future.

Moreover, novel wireless interconnect technologies such as CNT or graphene-based nano-antennas have emerged as high-speed, energy efficient interconnect solutions for on and off-chip communication. Although such technologies are yet to overcome many integration and fabrication challenges, performance modelling with various MAC, topologies, communication protocols, and the associated vulnerabilities can be studied for future WiNiP interconnection architectures enabled by those high-speed interconnects for MCMC systems.

Appendices

Appendix A

A 60GHz Millimeter-Wave Transceiver and Link Budget Analysis

Inter-chip wireless links require extremely low energy consumption to be competitive with high-speed serial I/O and other emerging technologies like photonic interconnects. The power consumption in data transmission over wireless medium occurs at the transmitters and receivers. The choice of the transceiver is dependent on the choice of the physical layer and modulation technique. In the mm-wave wireless physical layer designs, most research adopt simple modulation techniques like non-coherent OOK as it eliminates power-hungry, high-frequency carrier recovery circuits such as PLLs resulting in low power design.

The wireless transceiver used for all of the experiments presented in this dissertation was designed in 65nm node providing a data rate of 16 Gbps with 2.03 pj/bit energy efficiency. Transceivers designed in advanced nodes can significantly improve data rate as well as energy efficiency. However, transceiver design in advanced node requires to overcome various additional challenges such as low voltage headroom, complex impedance matching. Here, as our continuous effort to improve wireless energy efficiency, design of a transceiver in 45nm technology node has been discussed. The fundamentals of wireless transceiver and channel modeling discussed here will also help the reader to understand the research work presented in this dissertation. In the next section, we present a non-coherent OOK transmitter designed at 45nm node for inter-chip wireless communication in the 60GHz band.

A.1 60GHz Transmitter Architecture

Figure A.1(a) [153] shows the block diagram of an OOK transmitter consisting of a VCO, modulator and Power Amplifier (PA). In this section, the design approaches for these components of the 60GHz OOK transmitter using 45nm technology have been briefly discussed.

A.1.1 Voltage Controlled Oscillator

The VCO is responsible for generating the 60GHz RF carrier wave for the OOK modulator. The maximum single-ended voltage swing with minimum phase noise can be achieved by an NMOS cross-coupled VCO as it has comparatively less phase noise compared to the other oscillators such as ring or LC oscillator [154]. In the cross-coupled NMOS oscillator shown in Figure A.2, the positive feedback is given via the M1, M2 transistors and the LC tank

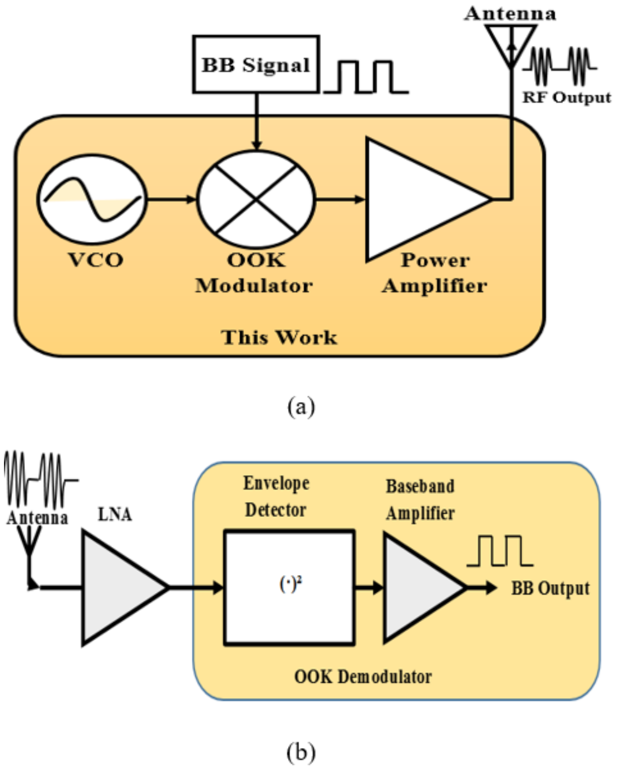


Figure A.1: Block diagram of the OOK (a) Transmitter, (b) Receiver.

circuit. The startup time is a critical part of the oscillator design. To avoid the startup time from limiting the data rate we do not use the VCO as a modulator. Therefore, the output of the VCO, the Local Oscillator (LO) signal, is coupled to a separate OOK modulator input via a transformer, X1 with a 1:1 turns ratio to ensure symmetrical swing at the input of the modulator. Transformer matching technique was adopted as it reduces overall power consumption by eliminating buffers or coupling capacitors.

A.1.2 OOK Modulator

The proposed modulator is shown in Figure A.2. The NMOS M6 works essentially as a pass transistor. The Base Band signal (BB) is fed into the gate terminal of M6. The BB signal when HIGH turns on the pass transistor allowing the carrier to pass to the drain of M6 from its source. Similar to the transformer X1 coupling the VCO to the modulator, another 1:1.5 transformer, X2 is designed to obtain single ended output from the modulator and for impedance matching of the modulator output with the input of the PA. NMOS M5 is used to drain the residual charge on the transformer when BB is LOW. Table A.1 shows

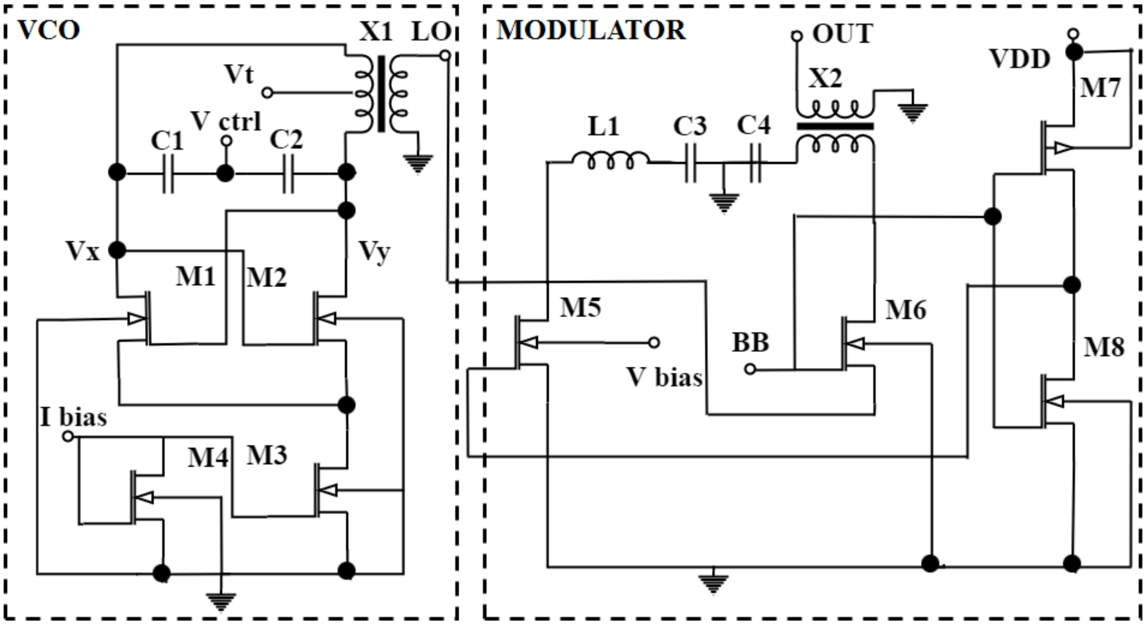


Figure A.2: Schematic of the proposed OOK modulator with NMOS cross-coupled VCO.

Table A.1: Component values of the proposed OOK modulator and VCO

X1, X2	**M1n, M2n**	**M3n**	**M4n**	**M5n, M6n**
1:1, 1:1.5	6um/65nm	60um/65nm	1um/65nm	2.25um/45nm
M7p	**M8n**	**C1,C2**	**C3,C4**	**L1**
480nm/45nm	450nm/45nm	10fF	20.1fF	350pH

Table A.2: Component values of the proposed PA

M1n, M2n	L_{d1}	L_{d2}	L_{g1}
10um/90um	350pH	420pH	600pH
L_{g2}	**C1**	**C2,C3**	**k1,k2**
350pH	1pF	14fF	0.088, 128

the component values used to design the OOK modulator.

A.1.3 Two Stage Power Amplifier

In the transmitter, power amplifier is one of the power-hungry blocks that affects the overall efficiency of the transmitter. In the design, a two-stage Common Source (CS) topology is used as it provides high frequency response and larger voltage swing compared to cascode designs. The two-stage PA proposed in this work is shown in Figure A.3. It uses a CS topology with drain-to-gate transformer-feedback neutralization technique, which creates an additional signal path that neutralizes the current flow through C_{gd} [125]. The transformers are formed using inductors at the gate and drain terminals of M1 (L_{g1} L_{d1}) and M2 (L_{g2} L_{d2}) as shown in Figure A.3 with coupling factors, k1 and k2 respectively. The values of the coupling factors are chosen such that they minimize the reverse transformation

$$Y_{12} = \frac{I_{gs}}{V_{ds}}, V_{in} = 0 \tag{A.1}$$

of the two stages of the amplifier. The inductor L_{d1} and L_{d2} at the drain of the transistors M1 and M2 resonates with the drain capacitance. This increases the overall gain and bandwidth at the center frequency. Transistor M1 and M2 are biased using a voltage divider based biasing circuit. The output of the power amplifier is matched with 50 Ohms by using L match impedance matching technique. Table A.2 shows the component values of the proposed PA.

A.1.4 Characteristics of the 60GHz transmitter

Here, the key characteristics of the transmitter has been discussed. Figure A.4(a) shows

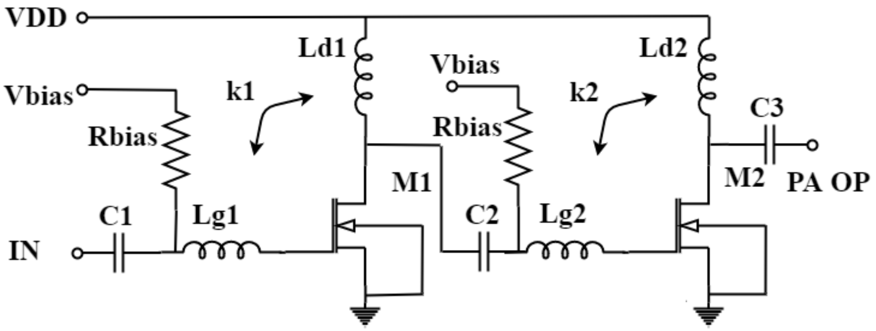
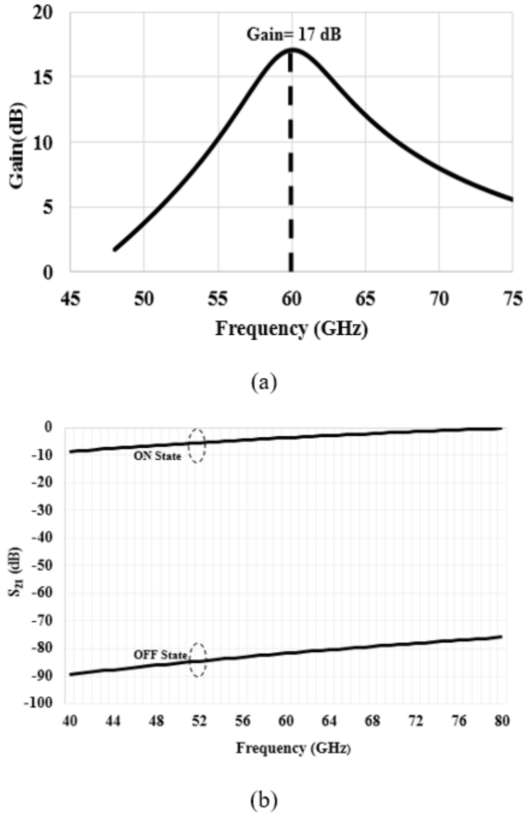


Figure A.3: Schematic of the proposed PA.



(a)

(b)

Figure A.4: Transmitter characterization (a) VCO oscillation frequency (b) ON-OFF state gain of the modulator.

the frequency response of the VCO. From Figure A.4(a), it can be observed that the VCO is centered at 60GHz. The on-off steady state gain of OOK modulator have been simulated using Cadence Virtuoso Analog Design Environment (ADE). At 60GHz the modulator has an "on" and "off" state gain of -3.6dB and -81.6dB respectively as shown in Figure A.4(b). Therefore, the on-off steady state gain is above 78dB for the frequency ranges from 45GHz to 75GHz.

Figure A.5 shows the S_{11}, S_{22}, and S_{21} for the proposed PA. The S_{11} is the ratio of reflected power to the incident power at the input port and is known as reflection coefficient or return loss. For the proposed PA design the value of reflection coefficient, S_{11} is found to be -30.15dB at 60GHz. Therefore, we conclude that the input is matched well with the 50 Ohm antenna impedance and hence we find negligible reflections at the input at 60GHz. Similarly, S_{22} is the ratio of reflected power to the incident power on the output port and it is found to be -15.42dB at 60GHz. S_{21} is the ratio of output power to the input power and represents the power gain for a well-matched PA. From Figure A.5, S_{21} is found to be 14.48dB.

The waveform shown in Figure A.6 shows the amplified OOK modulated signal at the output of PA with a pseudo-random sequence of logic 0s and 1s at baseband signal input.

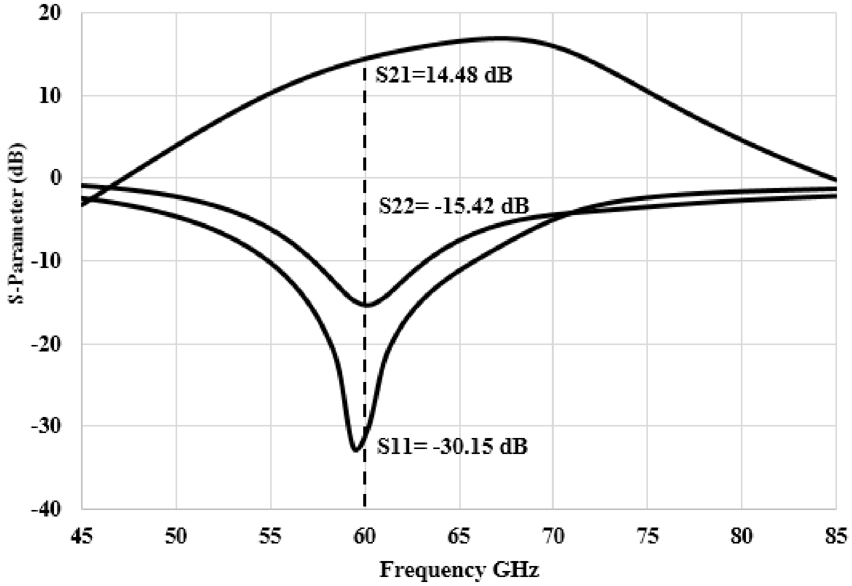


Figure A.5: S parameter of the PA.

As demonstrated 9 bits (000111100) are modulated between 1.06ns and 1.59ns indicating a data rate of nearly 17Gbps. The output waveform indicates a 450mVpp which translates to -3dBm output power and is achieved with a total DC power consumption of 3.9mW. This implies a bit energy efficiency of 0.23pJ/bit at 17Gbps on the transmitter side.

A.2 60GHz Receiver Design

Here, a 60GHz receiver proposed in [155] that can demodulate the signal received over inter-chip wireless interconnect transmitted from the above the transmitter, has been discussed. The Figure A.1(b) [155] shows the components of the receiver that consists of a high gain LNA and an OOK demodulator to demodulate the amplified received signal. The LNA

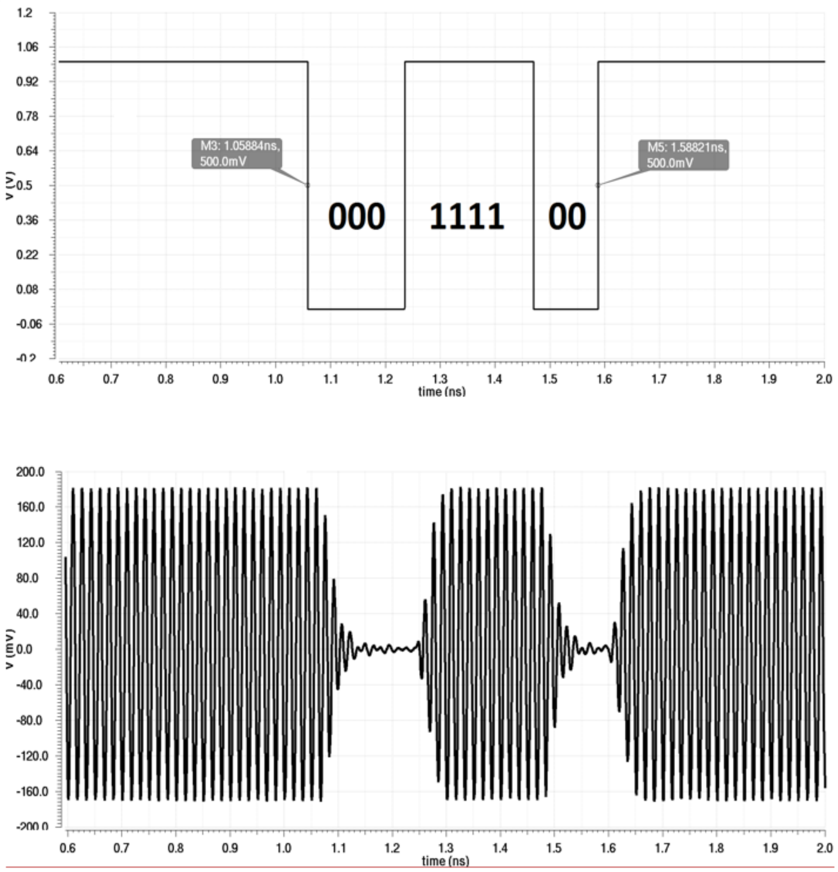


Figure A.6: BB signal (top) and transmitter output (bottom) for a pseudo-random sequence.

consists of a two-stage CS configuration as both the Common Gate (CG) and resistive feedback based topologies suffer from Noise Figure degradation due to the occurrence of noisy resistances in the signal path. Moreover, cascode structures, which are used commonly in low-frequency design for their high gains, are not suitable for the high frequency application. This is because the parasitic capacitances in the cascode transistors become dominant at higher frequencies, which reduces the inter stage impedance and hence, overall gain. The output of the LNA is then matched to the input of the demodulator. Since this information is modulated with a high frequency carrier signal, the OOK demodulator will exhibit a Low Pass Filter characteristic, removing the carrier wave and recovering the baseband digital signal. The proposed OOK demodulator consists of a source degenerated envelope detector at the input stage and a two-stage BB amplifier at the final output. The received waveform in Figure A.7 shows that the receiver is capable of achieving a data rate of 17 Gbps with 12mVpp at the input. The total DC power consumption of the OOK receiver is 6.1mW, which results in a bit energy- efficiency of 0.36pJ/bit [155] at a data rate of 17Gbps. The LNA of the receiver has a Noise Figure of 2.8dB making the noise floor -68.7dB at 300K for a bandwidth of 17GHz Therefore, together with the transmitter the transceiver consume a total of 0.59pJ/bit operating at 17Gbps.

A.3 60 GHz Link Budget Analysis

In this section, the wireless channel has been characterized in terms of BER that can be sustained by using the TX described here in typical MCMC environments. The BER has been evaluated for a range of path loss from 25 to 50dB corresponding to typical intra and inter-chip communication distances [3]. The received signal power is given by:

$$P_R = P_T - PL \tag{A.2}$$

Here, P_R and P_T are the received and transmitted power respectively and PL is the path loss. At the receiving side we assume a non-coherent OOK receiver as these are most power efficient [155]. Considering Additive White Gaussian Noise (AWGN) at the receiver, the overall Noise Floor, N_{Floor} of such a receiver is given by

$$N_{Floor} = 10log(kT) + 10log(BW) + NF \tag{A.3}$$

Where, k is the Boltzmann constant, T is the absolute temperature, BW is the bandwidth and NF is the Noise Figure of the receiver. We have considered an NF of 2.8dB for the LNA designed in [155]. Therefore, from (2) the N_{Floor} of the receiver is -67.8dBm at 300K for a BW of 16GHz. However, in addition to thermal AWGN, ISI can impact the reliability of the mm-wave link. Multipath channels existing in such environments as well as bandwidth limitation of the transceiver, the antennas and the channel, cause ISI. In the absence of thorough multipath channel models at 60GHz in multichip environments bandwidth limited

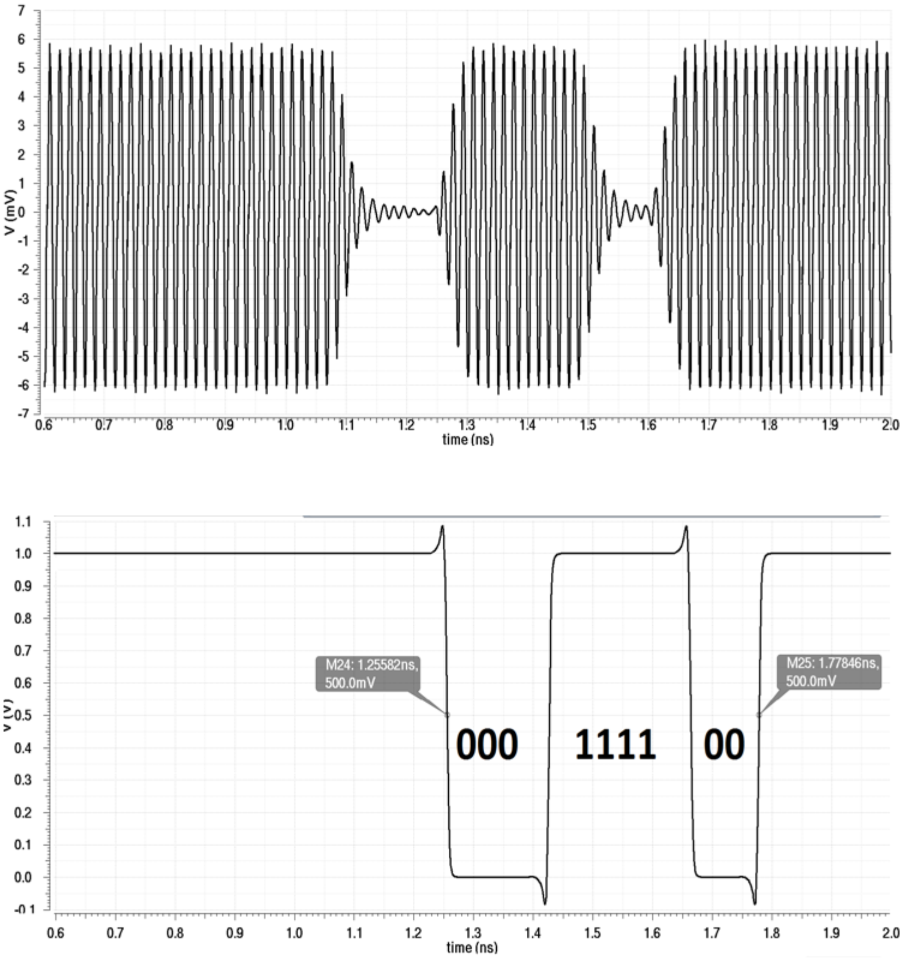


Figure A.7: LNA input of the receiver (top) and BB receiver output (bottom) for the same sequence.

ISI was considered only in our link budget analysis. From Figure A.6, the transmitter output is 450mVpp and 67mVpp when it is transmitting a '1' and a '0' respectively when measured at the center of the pulses. Therefore, the on-off ratio of overall OOK transmitter, R_{OOK} is 16.54dB. Therefore, the value of bandwidth-limited ISI noise, NISI is dependent on path loss and is given by,

$$N_{ISI} = P_T - R_{OOK} - PL \tag{A.4}$$

The received Signal to Interference plus Noise Ratio (SINR) is given by,

$$SINR = P_R - N_{Floor} - N_{ISI} \tag{A.5}$$

For non-coherent OOK demodulation, the BER is given by [124],

$$BER_{OOK} = \frac{1}{2}exp(-\frac{1}{2}SINR) + \frac{1}{4}erfc(\sqrt{\frac{1}{2}SINR}) \tag{A.6}$$

Where, erfc() is the complimentary error function. Figure A.8 shows the BER achieved with this transmitter for a wide range of path loss values with and without ISI (for comparison).

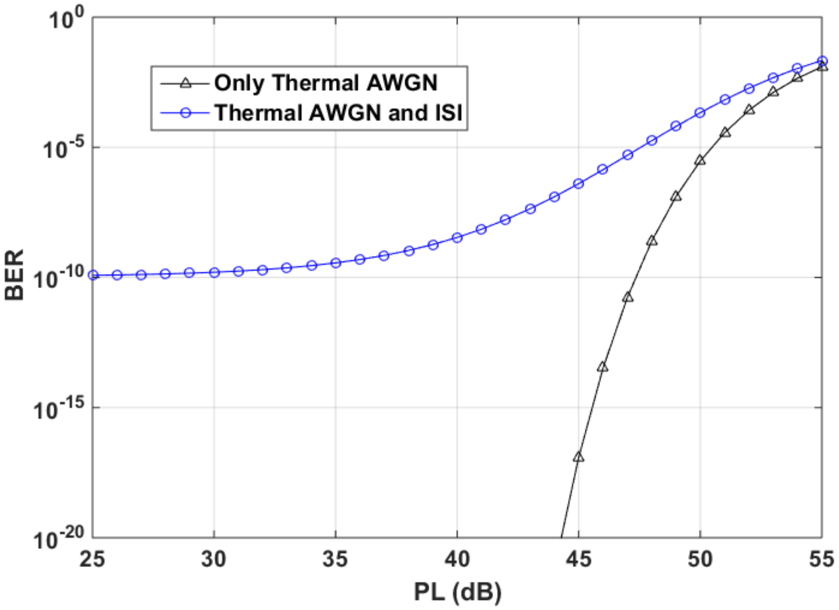


Figure A.8: BER variation vs path loss: with and without ISI.

Appendix B

Machine Learning and Neural Network Fundamentals

In Chapter 4, various ML classifiers were explored to detect the presence of an internal and external persistent jamming. Similarly, an ANN model was used in Chapter 5 from attacker's side to breach system security by accurately predicting the applications running in the system. As optimization of the ML or ANN models is not the purpose of this dissertation, here, we briefly discuss the training and testing procedures of the ML and ANN models used in those chapters with respect to the WiNiP architecture considered for MCMC environment.

B.1 Basics of the Used ML Classifiers

Several ML classifiers were investigated while detecting the presence of a persistent jamming in Chapter 4. In this section, some basics of those ML models and their training and testing process will be discussed.

A SVM is a supervised machine learning classifier with associated learning algorithm that can analyze data for classification. A SVM takes various data points as input and tries to find out a hyperplane that best separates the data point tags. This hyperplane is known as support vector. Once this hyperplane is optimized, based on the location, new data points are classified. Figure B.1 (a) shows the basic operation of a SVM classifier.

DT is a supervised learning method used for classification and regression. It continuously breaks down a data set into smaller data sets based on some if-else like decision rules and therefore, forms a tree-like structure. The final result is a tree with decision nodes and leaf nodes. The decision nodes are intermediate points where further splitting happens. The leaf nodes represent the final classification results. The topmost decision node is called the root node. Figure B.1 (b) shows the basic operation of a DT classifier.

KNN is a simple supervised ML algorithm used for classification and regression problems. For object classification, the object is classified based on majority votes out of its K number of nearest neighbouring samples. Typically the distance between the nearest K training samples are measured and sorted in ascending order. The object is assigned the label according to the mode of the K nearest labels. Figure B.1 (c) shows the basic operation of a KNN classifier.

The considered jamming attacks in Chapter 4 primarily result in causing continuous sustained burst errors in the flits (data corruption). This can be detected by observing the number of flits in error. In the proposed WiNiP, the output of BEU, which is the number of burst errors within a block, is fed to an ML classifier to detect and differentiate attacks. In order to train the ML classifier, cycle accurate WiNiP simulator was modeled to operate in one of the three modes; normal, random burst errors, and attack mode governed by the Markov Chain process as described in Chapter 4. In the normal mode, the wireless interconnects are assumed to work with the reliability level determined by the operation of the transceiver and their operating thermal noise. The second mode (random burst errors) is

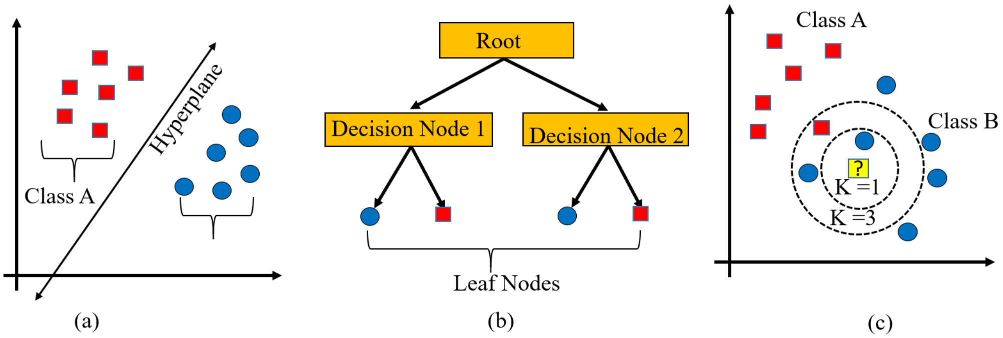


Figure B.1: Overview of (a) SVM (b) DT (c) KNN classifiers.

modeled with higher BERs as the burst errors are contiguous bits of flits. Lastly, under DoS attack, a high BER of 0.5 is assumed as for identically and independently distributed (iid) data bits even a very high power jamming signal can cause errors only half of the time on an average.

The simulator is modeled to create flit errors based on these BERs, which are then assumed to be detected by the BEU. The simulator is made to operate in one of the three modes dynamically by using a Markov Chain driven process with the assigned state transition probability discussed in Chapter 4. The specific probability values can be altered to model any particular scenario. This observed data (number of errors, flits transmitted and received) along with the operating mode as encountered in each WI is used to train the ML classifier at that WI. As the duration of the individual states are determined by the Markov Chain randomly, each specific instance of the states have varying duration, resulting in a diverse training data set. For the inference i.e., attack detection, the ML classifiers are fed runtime data such as, whether a flit is received or not, and whether a burst error is detected or not to detect the mode of operation of system. Training and testing of ML classifiers were performed with a hundred thousand cycles of data. The training and testing process of the ML classifiers are shown in Figure B.2.

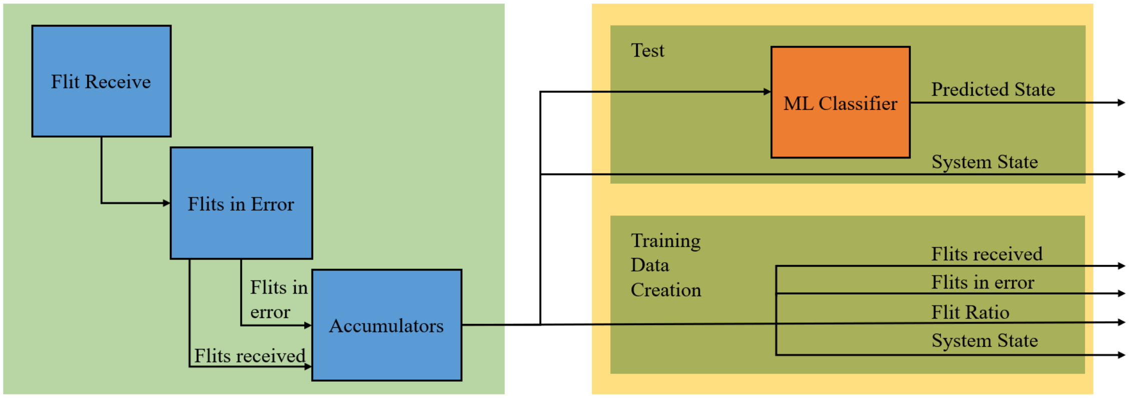


Figure B.2: Training data creating and testing of the ML classifiers.

B.2 Description of the Used ANN Classifier

ANNs are biological neural network inspired computational system. Like biological neurons, ANN is composed of many connected computing nodes that process an input signal and can send an output signal to the connected neurons once enabled by the activation function just like action potential in neurons. Each signal in ANN has an associated weight which is optimized during learning phase to reduce the prediction error. Typically the neurons are arranged in multiple layers where each layer may perform different functions. Figure B.3 shows a basic ANN architecture.

Although many kinds of ANN models exist, here we have limited our discussion to the architecture, training, and testing process of the Deep Neural Network (DNN) used in Chapter 5. DNN is an ANN with multiple hidden layers between the input and the output layers as shown in Fig B.3. We initially built a dataset consisting of eighty different features (payloads) obtained from simulating a system with 64 cores and 16 memory controllers, with combinations of twelve applications with no more than three running simultaneously.

We consider a maximum of three applications running simultaneously in the system sig-

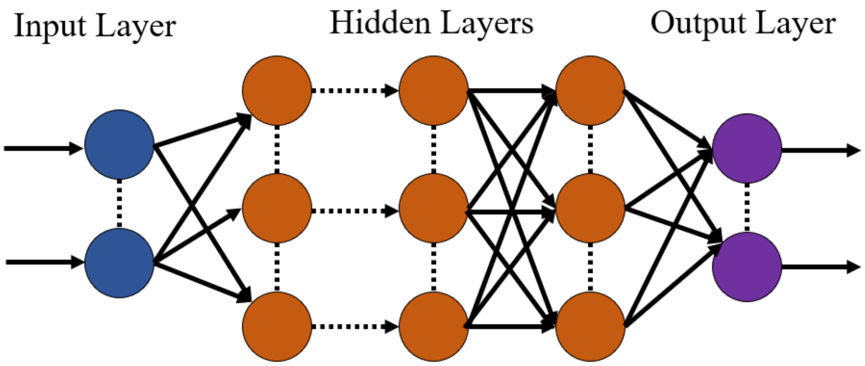


Figure B.3: Basic ANN architecture.

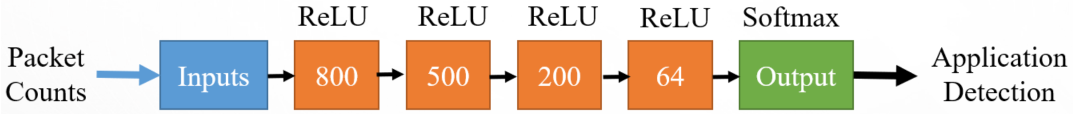


Figure B.4: DNN layers and neuron numbers with activation functions.

nifying up to three independent users hosted on the system executing three applications running simultaneously. Although a larger number of users can be easily accommodated by only creating training data with that assumption. We create this data in the same manner as an actual attacker who may be able to create using an emulator or a simulator of the real system. Specifically, we use a cycle-accurate simulator which monitors the movement of packets broken down into flits in a NoC. Various permutations of the applications from a common parallel benchmark suites [122] were executed on the simulator to create traffic traces that were visible to the HTs and those traffic traces consisting of number of packets traversing HT infected switches were used as the training dataset.

Furthermore, we train our ANN with all the features available. This yields high accuracy in predicting the application(s) running on the system. However, the caveat here is that although the ANN could be trained on a large set of features, yet, it is not feasible to expect a HT to snoop on all switches simultaneously. Doing so would contribute a significant latency, area and power overhead, eventually leading to HT detection. Thus, to reflect the real-world scenario, the attacker intends to insert a HT which focuses on the dominant features that still enable it to predict applications with high accuracy. We translated this approach by using correlation-based feature selection i.e., determine the routers in which the HTs need to be embedded (performed offline) that reduces latency without compromising the performance. We deploy a 5 layer ANN (N-800-500-200-64-12 neurons; N represents number of features at input) with ReLU as activation functions for hidden layers and softmax as the activation

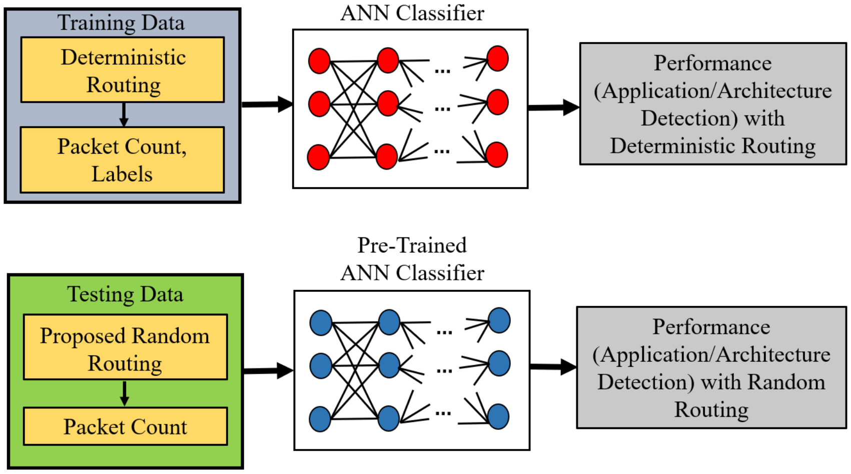

Training Data
Deterministic Routing
Packet Count, Labels
ANN Classifier
...
Performance (Application/Architecture Detection) with Deterministic Routing
Testing Data
Proposed Random Routing
Packet Count
Pre-Trained ANN Classifier
Performance (Application/Architecture Detection) with Random Routing

Figure B.5: Overall training and testing process.

function for the output layer. The DNN layer configuration is depicted in Figure B.4.

There are twelve final outputs for the ANN, which are the number of individual applications used to build the dataset. To represent simultaneous execution of multiple applications, one-hot encoding has been utilized. A 5-fold cross-validation was utilized to analyze performance, determined based on grid-search. As the ANN is deployed off-chip, the area and other overheads are not of concern, except the accuracy, precision, recall, and F1-score. The DNN model was trained using the packet count obtained from deterministic routing while the testing was done for the packet count obtained from the proposed SA-based routing in Chapter 5. The entire test and train process for DNN in shown in Figure B.5.

Appendix C

Related Publications

The contributions for each of the research objectives have been published in several peer-reviewed conferences and journals. The details of our research contributions and methodologies can be found from the following conference and journal papers.

C.1 Publications on a One-to-many Traffic-Aware WiNiP Architecture Design

C.1.1 Journals

1. **M. M. Ahmed**, N. Mansoor, and A. Ganguly, "An asymmetric, one-to-many traffic-aware mm-wave wireless interconnection architecture for multichip systems,"IEEE Transactions on Emerging Topics in Computing, pp. 1–1, 2020.

2. **M. M. Ahmed**, N. Mansoor and A. Ganguly, "A one-to-many traffic-oriented mm-wave wireless network-in-package interconnection architecture for multichip computing systems,"Sustainable Computing: Informatics and Systems, vol. 26,p. 100379, 2020.

3. S. H. Gade, **M. M. Ahmed**, S. Deb and A. Ganguly, "Energy efficient chip-to-chip wireless interconnection for heterogeneous architectures,"ACM Transactions on Design

Automation of Electronic Systems (TODAES), vol. 24, no. 5, pp. 1–27, 2019.

4. A. Ganguly, **M. M. Ahmed**, R.S. Narde, A. Vashist, M.S. Shamim, N. Mansoor, T. Shinde, S. Subramaniam, J. Venkataraman, and M. Indovina, “The advances, challenges and future possibilities of millimeter-wave chip-to-chip interconnections for multi-chip systems,”Journal of Low Power Electronics and Applications, vol. 8, no. 1, p. 5,2018.

C.1.2 Conferences

1. **M. M. Ahmed**, N. Mansoor, and A. Ganguly, “An asymmetric, energy efficient one-to-many traffic-aware wireless network-in-package interconnection architecture for multichip systems,” in2018 Ninth International Green and Sustainable Computing Conference (IGSC). IEEE, 2018, pp. 1–8.

2. **M. M. Ahmed**, A. Ganguly, S.M. Shariat, H. Pruswani, and N. Mansoor, “A one-to-many traffic aware wireless network-in-package for multichip computing platforms,” in 2018 31st IEEE International System-on-Chip Conference (SOCC).IEEE, 2018, pp.284–289.

3. **M. M. Ahmed**, M.S. Shamim, N. Mansoor, S.A. Mamun, and A. Ganguly, “Increasing interposer utilization: A scalable, energy efficient and high bandwidth multicore-multichip integration solution,” in2017 Eighth International Green and Sustainable Computing Conference (IGSC). IEEE, 2017, pp. 1–6.

4. M. S. Shamim, **M. M. Ahmed**, N. Mansoor, and A. Ganguly, “Energy-efficient wireless interconnection frame-work for multichip systems with in-package memory stacks,” in 2017 30th IEEE International System-on-Chip Conference (SOCC).IEEE, 2017, pp. 357–362.

5. A. Ganguly, N. Mansoor, M.S. Shamim, **M. M. Ahmed**, R.S. Narde, A. Vashist, and J. Venkataraman, “Intra-chip wireless interconnect: The road ahead,”in Proceedings of the 10th International Workshop on Network on Chip Architectures, 2017, pp. 1–6.

6. S. Saxena, D.S. Manur, **M. M. Ahmed**, and A. Ganguly, “Energy-efficiency in interconnection fabrics for inter and intra-chip communication using graphene-based

thz-band antennas,"in 2017 Eighth International Green and Sustainable Computing Conference (IGSC). IEEE, 2017, pp. 1–6.

7. T. Shinde, S. Subramaniam, P. Deshmukh, **M. M. Ahmed**, M. Indovina, and A. Ganguly, "A 0.24 pj/bit, 16gbps ook transmitter circuit in 45nm cmos for inter and intra-chip wireless interconnects," in Proceedings of the 2018 on Great Lakes Symposium on VLSI, 2018, pp. 69–74.

C.2 Publications on Secure WiNiP Interconnection Architecture Design

C.2.1 Journal

1. **M. M. Ahmed**, A. Ganguly, A. Vashist, and S.M. Pudukotai, "Aware-wi: A jamming-aware reconfigurable wireless interconnection using adversarial learning for multichip systems,"Sustainable Computing: Informatics and Systems, p. 100470, 2020

C.2.2 Conferences

1. **M. M. Ahmed**, A. Vashist, S. M. Pudukotai Dinakarrao and A. Ganguly, "Architecting a Secure Wireless Interconnect for Multichip Communication: An ML Approach," 2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), Kolkata, India, 2020, pp. 1-6, doi: 10.1109/AsianHOST51057.2020.9358256

2. **M. M. Ahmed**, A. Dhavlle, N. Mansoor, P. Sutradhar, S. M. Pudukotai, K. Basu, and Amlan Ganguly, "Defense Against on-Chip Trojans Enabling Traffic Analysis Attacks," 2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), Kolkata, India, 2020, pp. 1-6, doi: 10.1109/AsianHOST51057.2020.9358250.

3. **M Meraj Ahmed**, Abhijitt Dhavlle, Naseef Mansoor, Sai Manoj Pudukotai Dinakarrao, Kanad Basu, Amlan Ganguly "What Can a Remote Access Hardware Trojan do to a Network-on-Chip?" (accepted paper at IEEE, ISCAS, 2020)

C.2.3 Book Chapter

1. **M. M. Ahmed**, A. Vashist, A. Keats, A. Ganguly, S.M. Pudukotai, ”Security Framework for On-Chip Wireless Interconnection Networks in Network-on-Chip Security and Privacy”. Springer, 2021

Bibliography

[1] 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, 2008.

[2] X. Wu, Y. Ye, J. Xu, W. Zhang, W. Liu, M. Nikdast, and X. Wang, “Union: A unified inter/intrachip optical network for chip multiprocessors,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 5, pp. 1082–1095, 2013.

[3] M. S. Shamim, N. Mansoor, R. S. Narde, V. Kothandapani, A. Ganguly, and J. Venkataraman, “A wireless interconnection framework for seamless inter and intra-chip communication in multichip systems,” *IEEE Transactions on Computers*, vol. 66, no. 3, pp. 389–402, 2016.

[4] S. Abadal, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, “Orthonoc: A broadcast-oriented dual-plane wireless network-on-chip architecture,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 03, pp. 628–641, 2018.

[5] B. Lebiednik, S. Abadal, H. Kwon, and T. Krishna, “Architecting a secure wireless network-on-chip,” in *2018 Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*. IEEE, 2018, pp. 1–8.

[6] L. S. Indrusiak, J. Harbin, and M. J. Sepulveda, “Side-channel attack resilience through route randomisation in secure real-time networks-on-chip,” in *Int. Symp. on Reconfigurable Communication-centric SoC (ReCoSoC)*, 2017.

[7] S. M. Shahriat, “Global congestion and fault aware wireless interconnection framework for multicore systems,” 2019.

[8] S. H. Gade, M. M. Ahmed, S. Deb, and A. Ganguly, "Energy efficient chip-to-chip wireless interconnection for heterogeneous architectures," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 24, no. 5, pp. 1–27, 2019.

[9] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 490–504, 2001.

[10] "International technology roadmap for semiconductors (itrs), 2013."

[11] S. Abadal Cavallé, "Broadcast-oriented wireless network-on-chip: fundamentals and feasibility," 2016.

[12] S. Borkar, "Obeying moore's law beyond 0.18 micron [microprocessor design]," in *Proceedings of 13th Annual IEEE International ASIC/SOC Conference (Cat. No. 00TH8541).* IEEE, 2000, pp. 26–31.

[13] M. J. Flynn, "Very high-speed computing systems," in *Readings in computer architecture.* Morgan Kaufmann Publishers Inc., 2000, pp. 519–527.

[14] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, and Y. Hoskote, "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.

[15] J. Held and S. Koehl, "Introducing the single-chip cloud computer," *Intel White Paper*, vol. 1, no. 4, pp. 5–3, 2010.

[16] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, L. Bao, J. Brown, M. Mattina, C.-C. Miao, C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, and J. Zook, "Tile 64-processor: A 64-core soc with mesh interconnect," in *2008 IEEE International Solid-State Circuits Conference-Digest of Technical Papers.* IEEE, 2008, pp. 88–598.

[17] D. Flynn, "Amba: enabling reusable on-chip designs," *IEEE micro*, vol. 17, no. 4, pp. 20–27, 1997.

[18] R. Hofmann and B. Drerup, "Next generation coreconnect/spl trade/processor local bus architecture," in *15th Annual IEEE International ASIC/SOC Conference.* IEEE, 2002, pp. 221–225.

[19] C.-T. Hsieh and M. Pedram, "Architectural energy optimization by bus splitting," *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol. 21, no. 4, pp. 408–414, 2002.

[20] S. R. Sridhara and N. R. Shanbhag, "Coding for system-on-chip networks: a unified framework," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 13, no. 6, pp. 655–667, 2005.

[21] L. Benini and G. De Micheli, "Networks on chips: A new soc paradigm," *computer*, vol. 35, no. 1, pp. 70–78, 2002.

[22] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE transactions on Computers*, vol. 54, no. 8, pp. 1025–1040, 2005.

[23] B. Hoefflinger, "Itrs: The international technology roadmap for semiconductors," in *Chips 2020*. Springer, 2011, pp. 161–174.

[24] S. Bhunia, S. Mukhopadhyay, and K. Roy, "Process variations and process-tolerant design," in *20th international conference on VLSI design held jointly with 6th international conference on embedded systems (VLSID'07)*. IEEE, 2007, pp. 699–704.

[25] A. Kannan, N. E. Jerger, and G. H. Loh, "Exploiting interposer technologies to disintegrate and reintegrate multicore processors," *IEEE Micro*, vol. 36, no. 3, pp. 84–93, 2016.

[26] B. T. Murphy, "Cost-size optima of monolithic integrated circuits," *Proceedings of the IEEE*, vol. 52, no. 12, pp. 1537–1545, 1964.

[27] K. Lepak, G. Talbot, S. White, N. Beck, and S. Naffziger, "The next generation amd enterprise server product architecture," *IEEE hot chips*, vol. 29, 2017.

[28] J.-B. Missilany, "[22] amd fusion apu era begins. available online: Url: http://www.amd.com/en-us/press-releases/pages/amd-fusion-apu-era-2011jan04.aspx," Handed out at O'Hare, Oct. 1984, this is a full MISC entry.

[29] R. Mahajan, D. Mallik, R. Sankman, K. Radhakrishnan, C. Chiu, and J. He, "Advances and challenges in flip-chip packaging," in *IEEE Custom Integrated Circuits Conference 2006*. IEEE, 2006, pp. 703–709.

[30] K. C. Yong, W. C. Song, B. E. Cheah, and M. F. Ain, "Signaling analysis of inter-chip i/o package routing for multi-chip package," in *2012 4th Asia Symposium on Quality Electronic Design (ASQED)*. IEEE, 2012, pp. 243–248.

[31] A. W. Topol, D. 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. Leong, "Three-dimensional integrated circuits," *IBM Journal of Research and Development*, vol. 50, no. 4.5, pp. 491–506, 2006.

[32] A. Annamalai, R. Kumar, A. Vijayakumar, and S. Kundu, "A system-level solution for managing spatial temperature gradients in thinned 3d ics," in *International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2013, pp. 88–95.

[33] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler, "3d-ice: A compact thermal model for early-stage design of liquid-cooled ics," *IEEE Transactions on Computers*, vol. 63, no. 10, pp. 2576–2589, 2013.

[34] N. E. Jerger, A. Kannan, Z. Li, and G. H. Loh, "Noc architectures for silicon interposer systems: Why pay for more wires when you can get them (from your interposer) for free?" in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 2014, pp. 458–470.

[35] G. H. Loh, N. E. Jerger, A. Kannan, and Y. Eckert, "Interconnect-memory challenges for multi-chip, silicon interposer systems," in *Proceedings of the 2015 international symposium on Memory Systems*, 2015, pp. 3–10.

[36] R. Hendry, D. Nikolova, S. Rumley, and K. Bergman, "Modeling and evaluation of chip-to-chip scale silicon photonic networks," in *2014 IEEE 22nd Annual Symposium on High-Performance Interconnects*. IEEE, 2014, pp. 1–8.

[37] 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.

[38] A. Joshi, C. Batten, V. Stojanović, and K. Asanović, "Building manycore processor-to-dram networks using monolithic silicon photonics," in *High Performance Embedded Computing (HPEC) Workshop*, 2008.

[39] A. J. Karkar, J. E. Turner, K. Tong, A.-D. Ra'ed, T. Mak, A. Yakovlev, and F. Xia, "Hybrid wire-surface wave interconnects for next-generation networks-on-chip," *IET Computers & Digital Techniques*, vol. 7, no. 6, pp. 294–303, 2013.

[40] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, 2008, pp. 72–81.

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

[42] S. Abadal, A. Mestres, R. Martinez, E. Alarcon, and A. Cabellos-Aparicio, "Multicast on-chip traffic analysis targeting manycore noc design," in *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing.* IEEE, 2015, pp. 370–378.

[43] C.-K. Lin, A. Wild, G. N. Chinya, Y. Cao, M. Davies, D. M. Lavery, and H. Wang, "Programming spiking neural networks on intel's loihi," *Computer*, vol. 51, no. 3, pp. 52–61, 2018.

[44] Y. Xue and P. Bogdan, "User cooperation network coding approach for noc performance improvement," in *Proceedings of the 9th International Symposium on Networks-on-Chip*, 2015, pp. 1–8.

[45] M. Palesi and M. Daneshtalab, *Routing algorithms in networks-on-chip.* Springer, 2014.

[46] A. Karkar, N. Dahir, K. Tong, T. Mak, and A. Yakovlev, "Hybrid wire-surface wave architecture for one-to-many communication in networks-on-chip," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE).* IEEE, 2014, pp. 1–4.

[47] T. Krishna, L.-S. Peh, B. M. Beckmann, and S. K. Reinhardt, "Towards the ideal on-chip fabric for 1-to-many and many-to-1 communication," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, 2011, pp. 71–82.

[48] C.-K. Liang and M. Prvulovic, “Misar: Minimalistic synchronization accelerator with resource overflow management,” *ACM SIGARCH Computer Architecture News*, vol. 43, no. 3S, pp. 414–426, 2015.

[49] T. Krishna and L.-S. Peh, “Single-cycle collective communication over a shared network fabric,” in *2014 Eighth IEEE/ACM International Symposium on Networks-on-Chip (NoCS)*. IEEE, 2014, pp. 1–8.

[50] M. M. Ahmed, M. S. Shamim, N. Mansoor, S. A. Mamun, and A. Ganguly, “Increasing interposer utilization: A scalable, energy efficient and high bandwidth multicore-multichip integration solution,” in *2017 Eighth International Green and Sustainable Computing Conference (IGSC)*. IEEE, 2017, pp. 1–6.

[51] J.-J. Lin, H.-T. Wu, Y. Su, L. Gao, A. Sugavanam, J. E. Brewer, and K. K.O., “Communication using antennas fabricated in silicon integrated circuits,” *IEEE Journal of solid-state circuits*, vol. 42, no. 8, pp. 1678–1687, 2007.

[52] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, and C. Teuscher, “Scalable hybrid wireless network-on-chip architectures for multicore systems,” *IEEE Transactions on Computers*, vol. 60, no. 10, pp. 1485–1502, 2010.

[53] S. Abadal, E. Alarcón, A. Cabellos-Aparicio, M. C. Lemme, and M. Nemirovsky, “Graphene-enabled wireless communication for massive multicore architectures,” *IEEE Communications Magazine*, vol. 51, no. 11, pp. 137–143, 2013.

[54] S. Laha, S. Kaya, D. W. Matolak, W. Rayess, D. DiTomaso, and A. Kodi, “A new frontier in ultralow power wireless links: Network-on-chip and chip-to-chip interconnects,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 186–198, 2014.

[55] C. Lee, T. Yao, A. Mangan, K. Yau, M. Copeland, and S. Voinigescu, “Sige bicmos 65-ghz bpsk transmitter and 30 to 122 ghz lc-varactor vcos with up to 21% tuning range,” in *IEEE Compound Semiconductor Integrated Circuit Symposium, 2004*. IEEE, 2004, pp. 179–182.

[56] J.-D. Park, S. Kang, S. V. Thyagarajan, E. Alon, and A. M. Niknejad, “A 260 ghz fully integrated cmos transceiver for wireless chip-to-chip communication,” in *2012 Symposium on VLSI Circuits (VLSIC)*. IEEE, 2012, pp. 48–49.

[57] C.-H. Jan, M. Agostinelli, H. Deshpande, M. El-Tanani, W. Hafez, U. Jalan, L. Janbay, M. Kang, H. Lakdawala, J. Lin, Y.-L. Lu, S. Mundanai, J. Park, A. Rahman, A. Rizk, W. Shin, K. Soumyanath, H. Tashiro, C. Tsai, P. VanDerVoorn, J.-Y. Yeh, and P. Bai, "Rf cmos technology scaling in high-k/metal gate era for rf soc (system-on-chip) applications," in *2010 International Electron Devices Meeting.* IEEE, 2010, pp. 27–2.

[58] H. M. Cheema and A. Shamim, "The last barrier: on-chip antennas," *IEEE Microwave Magazine*, vol. 14, no. 1, pp. 79–91, 2013.

[59] M. S. Shamim, N. Mansoor, A. Samaiyar, A. Ganguly, S. Deb, and S. Sunndar Ram, "Energy-efficient wireless network-on-chip architecture with log-periodic on-chip antennas," in *Proceedings of the 24th edition of the great lakes symposium on VLSI*, 2014, pp. 85–86.

[60] K. Chang, S. Deb, A. Ganguly, X. Yu, S. P. Sah, P. P. Pande, B. Belzer, and D. Heo, "Performance evaluation and design trade-offs for wireless network-on-chip architectures," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 8, no. 3, pp. 1–25, 2012.

[61] A. A. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE Journal on selected areas in communications*, vol. 5, no. 2, pp. 128–137, 1987.

[62] A. P. Toda and F. De Flaviis, "60-ghz substrate materials characterization using the covered transmission-line method," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 3, pp. 1063–1075, 2015.

[63] 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.

[64] D. DiTomaso, A. Kodi, S. Kaya, and D. Matolak, "iwise: Inter-router wireless scalable express channels for network-on-chips (nocs) architecture," in *2011 IEEE 19th Annual Symposium on High Performance Interconnects.* IEEE, 2011, pp. 11–18.

[65] V. Vijayakumaran, M. P. Yuvaraj, N. Mansoor, N. Nerurkar, A. Ganguly, and A. Kwasinski, "Cdma enabled wireless network-on-chip," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 10, no. 4, pp. 1–20, 2014.

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

[67] N. Mansoor and A. Ganguly, “Reconfigurable wireless network-on-chip with a dynamic medium access mechanism,” in *Proceedings of the 9th International Symposium on Networks-on-Chip*, 2015, pp. 1–8.

[68] A. Ganguly, M. Y. Ahmed, and A. Vidapalapati, “A denial-of-service resilient wireless noc architecture,” in *Proceedings of the great lakes symposium on VLSI*, 2012, pp. 259–262.

[69] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, A. Fogh, J. Horn, S. Mangard, P. Kocher, D. Genkin, Y. Yarom, and M. Hamburg, “Meltdown: Reading kernel memory from user space,” in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 973–990.

[70] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, and Y. Yarom, “Spectre attacks: Exploiting speculative execution,” in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1–19.

[71] B. Wu, J. Chen, J. Wu, and M. Cardei, “A survey of attacks and countermeasures in mobile ad hoc networks,” in *Wireless network security*. Springer, 2007, pp. 103–135.

[72] W. J. Dally and B. Towles, “Route packets, not wires: on-chip inteconnection networks,” in *Proceedings of the 38th annual Design Automation Conference*, 2001, pp. 684–689.

[73] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Pailet, S. Jain, T. Jacob, S. Yada, S. Marella, P. Salihundam, V. Erraguntala, M. Konow, M. Riepen, G. Droege, J. Linderman, M. Gries, T. Apel, K. Henriss, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, R. V. D. Wijngaart, and T. Mattson, “A 48-core ia-32 message-passing processor with dvfs in 45nm cmos,” in *2010 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2010, pp. 108–109.

[74] U. Y. 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.

[75] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 150–161, 2007.

[76] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. G. Beausoleil, and J. H. Ahn, "Corona: System implications of emerging nanophotonic technology," *ACM SIGARCH Computer Architecture News*, vol. 36, no. 3, pp. 153–164, 2008.

[77] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: Illuminating future network-on-chip with nanophotonics," in *Proceedings of the 36th annual international symposium on Computer architecture*, 2009, pp. 429–440.

[78] M. Dragoman, A. Muller, D. Dragoman, F. Coccetti, Plana, and R, "Terahertz antenna based on graphene," *Journal of Applied Physics*, vol. 107, no. 10, p. 104313, 2010.

[79] G. Piro, S. Abadal, A. Mestres, E. Alarcón, J. Solé-Pareta, L. A. Grieco, and G. Boggia, "Initial mac exploration for graphene-enabled wireless networks-on-chip," in *Proceedings of ACM The First Annual International Conference on Nanoscale Computing and Communication*, 2014, pp. 1–9.

[80] F. Gutierrez, S. Agarwal, K. Parrish, and T. S. Rappaport, "On-chip integrated antenna structures in cmos for 60 ghz wpan systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1367–1378, 2009.

[81] M. O. Agyeman, Q.-T. Vien, A. Ahmadinia, A. Yakovlev, K.-F. Tong, and T. Mak, "A resilient 2-d waveguide communication fabric for hybrid wired-wireless noc design," *IEEE Transactions on Parallel and Distributed systems*, vol. 28, no. 2, pp. 359–373, 2016.

[82] 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.

[83] A. Ganguly, M. M. Ahmed, R. Singh Narde, A. Vashist, M. S. Shamim, N. Mansoor, T. Shinde, S. Subramaniam, S. Saxena, J. Venkataraman, and M. Indovina, "The advances, challenges and future possibilities of millimeter-wave chip-to-chip interconnections for multi-chip systems," *Journal of Low Power Electronics and Applications*, vol. 8, no. 1, p. 5, 2018.

[84] A. Ganguly, P. Wettin, K. Chang, and P. Pande, "Complex network inspired fault-tolerant noc architectures with wireless links," in *Proceedings of the fifth ACM/IEEE International Symposium on Networks-on-Chip*, 2011, pp. 169–176.

[85] M. S. Shamim, A. Mhatre, N. Mansoor, A. Ganguly, and G. Tsouri, "Temperature-aware wireless network-on-chip architecture," in *International Green Computing Conference*. IEEE, 2014, pp. 1–10.

[86] S.-B. Lee, S.-W. Tam, I. Pefkianakis, S. Lu, M. 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 15th annual international conference on Mobile computing and networking*, 2009, pp. 217–228.

[87] J. Chen and Y.-H. Lai, "A study of csma-based and token-based wireless interconnects network-on-chip," in *2014 IEEE International Conference on Communiction Problem-solving*. IEEE, 2014, pp. 205–209.

[88] K. Duraisamy, R. G. Kim, and P. P. Pande, "Enhancing performance of wireless nocs with distributed mac protocols," in *Sixteenth International Symposium on Quality Electronic Design*. IEEE, 2015, pp. 406–411.

[89] M. Palesi, M. Collotta, A. Mineo, and V. Catania, "An efficient radio access control mechanism for wireless network-on-chip architectures," *Journal of Low Power Electronics and Applications*, vol. 5, no. 2, pp. 38–56, 2015.

[90] N. Mansoor, S. Shamim, and A. Ganguly, "A demand-aware predictive dynamic bandwidth allocation mechanism for wireless network-on-chip," in *2016 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP)*. IEEE, 2016, pp. 1–8.

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

[92] M. M. Ahmed, M. S. Shamim, N. Mansoor, S. A. Mamun, and A. Ganguly, "Increasing interposer utilization: A scalable, energy efficient and high bandwidth multicore-multichip integration solution." in *IGSC*, 2017, pp. 1–6.

[93] L. Wang, Y. Jin, H. Kim, and E. J. Kim, "Recursive partitioning multicast: A bandwidth-efficient routing for networks-on-chip," in *2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*. IEEE, 2009, pp. 64–73.

[94] M. Lodde, J. Flich, and M. E. Acacio, "Heterogeneous noc design for efficient broadcast-based coherence protocol support," in *2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip*. IEEE, 2012, pp. 59–66.

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

[96] 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.

[97] 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.

[98] G. Kurian, J. E. Miller, J. Psota, J. Eastep, J. Liu, J. Michel, L. C. Kimerling, and A. Agarwal, "Atac: A 1000-core cache-coherent processor with on-chip optical network," in *2010 19th International Conference on Parallel Architectures and Compilation Techniques (PACT)*. IEEE, 2010, pp. 477–488.

[99] 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 22nd International Conference on Parallel Architectures and Compilation Techniques*, 2013, pp. 309–318.

[100] S. Abadal, A. Mestres, M. Nemirovsky, H. Lee, A. González, E. Alarcón, and A. Cabellos-Aparicio, “Scalability of broadcast performance in wireless network-on-chip,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3631–3645, 2016.

[101] K. Duraisamy, Y. Xue, P. Bogdan, and P. P. Pande, “Multicast-aware high-performance wireless network-on-chip architectures,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 1126–1139, 2016.

[102] S. Evain and J.-P. Diguet, “From noc security analysis to design solutions,” in *IEEE Workshop on Signal Processing Systems Design and Implementation, 2005.* IEEE, 2005, pp. 166–171.

[103] C. H. Gebotys and R. J. Gebotys, “A framework for security on noc technologies,” in *IEEE Computer Society Annual Symposium on VLSI, 2003. Proceedings.* IEEE, 2003, pp. 113–117.

[104] A. Ganguly, M. Y. Ahmed, and A. Vidapalapati, “A denial-of-service resilient wireless noc architecture,” in *Proceedings of the great lakes symposium on VLSI*, 2012, pp. 259–262.

[105] F. Pereñíguez-García and J. L. Abellán, “Secure communications in wireless network-on-chips,” in *Proceedings of the 2nd International Workshop on Advanced Interconnect Solutions and Technologies for Emerging Computing Systems*, 2017, pp. 27–32.

[106] A. Vashist, A. Keats, S. M. Pudukotai Dinakarrao, and A. Ganguly, “Securing a wireless network-on-chip against jamming-based denial-of-service and eavesdropping attacks,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2781–2791, 2019.

[107] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, “Using channel hopping to increase 802.11 resilience to jamming attacks,” in *IEEE INFOCOM.* IEEE, 2007, pp. 2526–2530.

[108] G. Noubir, “On connectivity in ad hoc networks under jamming using directional antennas and mobility,” in *International Conference on Wired/Wireless Internet Communications.* Springer, 2004, pp. 186–200.

[109] T. Boraten and A. K. Kodi, "Mitigation of denial of service attack with hardware trojans in noc architectures," in *2016 IEEE international parallel and distributed processing symposium (IPDPS)*. IEEE, 2016, pp. 1091–1100.

[110] W. Zhao, Y. Ha, and M. Alioto, "Aes architectures for minimum-energy operation and silicon demonstration in 65nm with lowest energy per encryption," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2015, pp. 2349–2352.

[111] R. Fernandes, C. Marcon, R. Cataldo, J. Silveira, G. Sigl, and J. Sepúlveda, "A security aware routing approach for noc-based mpsocs," in *Symp. on Integrated Circuits and Systems Design*, 2016.

[112] T. H. Boraten and A. K. Kodi, "Securing nocs against timing attacks with non-interference based adaptive routing," in *IEEE/ACM Int. Symp. on Networks-on-Chip (NOCS)*, 2018.

[113] J. Sepulveda, D. Flórez, R. Fernandes, C. Marcon, G. Gogniat, and G. Sigl, "Towards risk aware nocs for data protection in mpsocs," in *Int. Symp. on ReCoSoC*, 2016.

[114] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, 2014.

[115] H. Matsutani, M. Koibuchi, I. Fujiwara, T. Kagami, Y. Take, T. Kuroda, P. Bogdan, R. Marculescu, and H. Amano, "Low-latency wireless 3d nocs via randomized shortcut chips," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2014, pp. 1–6.

[116] M. A. I. Sikder, A. Kodi, W. Rayess, D. DiTomaso, D. Matolak, and S. Kaya, "Exploring wireless technology for off-chip memory access," in *2016 IEEE 24th Annual Symposium on High-Performance Interconnects (HOTI)*. IEEE, 2016, pp. 92–99.

[117] W. Yang, K. Ma, K. S. Yeo, and W. M. Lim, "A 60ghz on-chip antenna in standard cmos silicon technology," in *2012 IEEE Asia Pacific Conference on Circuits and Systems*. IEEE, 2012, pp. 252–255.

[118] J. Kim and Y. Kim, "Hbm: Memory solution for bandwidth-hungry processors," in *2014 IEEE Hot Chips 26 Symposium (HCS)*. IEEE, 2014, pp. 1–24.

[119] D. Abts, N. D. Enright Jerger, J. Kim, D. Gibson, and M. H. Lipasti, “Achieving predictable performance through better memory controller placement in many-core cmps,” *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 451–461, 2009.

[120] X. Wang, T. Ahonen, and J. Nurmi, “Applying cdma technique to network-on-chip,” *IEEE transactions on very large scale integration (VLSI) systems*, vol. 15, no. 10, pp. 1091–1100, 2007.

[121] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, “Improving the energy efficiency of wireless network on chip architectures through online selective buffers and receivers shutdown,” in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 668–673.

[122] M. Badr and N. E. Jerger, “Synfull: Synthetic traffic models capturing cache coherent behaviour,” *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3, pp. 109–120, 2014.

[123] R. Holsmark, S. Kumar, M. Palesi, and A. Mejia, “Hira: A methodology for deadlock free routing in hierarchical networks on chip,” in *2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*. IEEE, 2009, pp. 2–11.

[124] X. Yu, H. Rashtian, S. Mirabbasi, P. P. Pande, and D. Heo, “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.

[125] X. Yu, S. P. Sah, H. Rashtian, S. Mirabbasi, P. P. Pande, and D. Heo, “A 1.2-pj/bit 16-gb/s 60-ghz ook transmitter in 65-nm cmos for wireless network-on-chip,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 10, pp. 2357–2369, 2014.

[126] J.-B. Missilany, “[116] ansys hfss. available online. http://www.ansys.com/products/electronics/ansys-hfss,” Handed out at O’Hare, Oct. 1984, this is a full MISC entry.

[127] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, “Noxim: An open, extensible and cycle-accurate network on chip simulator,” in *2015 IEEE 26th interna-*

tional conference on application-specific systems, architectures and processors (ASAP). IEEE, 2015, pp. 162–163.

[128] A. Mestres, S. Abadal, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "A mac protocol for reliable broadcast communications in wireless network-on-chip," in *Proceedings of the 9th International Workshop on Network on Chip Architectures*, 2016, pp. 21–26.

[129] J. Lee, C. Nicopoulos, S. J. Park, M. Swaminathan, and J. Kim, "Do we need wide flits in networks-on-chip?" in *2013 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2013, pp. 2–7.

[130] M. A. Abu-Rgheff, *Introduction to CDMA wireless communications*. Academic Press, 2007.

[131] M. M. Ahmed, A. Ganguly, S. M. Shahriat, H. Pruswani, and N. Mansoor, "A one-to-many traffic aware wireless network-in-package for multi-chip computing platforms," in *IEEE SOCC*, 2018.

[132] B. Wu, J. Wu, E. B. Fernandez, and S. Magliveras, "Secure and efficient key management in mobile ad hoc networks," in *19th IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2005, pp. 8–pp.

[133] B. Fu and P. Ampadu, "Burst error detection hybrid arq with crosstalk-delay reduction for reliable on-chip interconnects," in *2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*. IEEE, 2009, pp. 440–448.

[134] I. H. Abbassi, F. Khalid, S. Rehman, A. M. Kamboh, A. Jantsch, S. Garg, and M. Shafique, "TrojanZero: Switching activity-aware design of undetectable hardware trojans with zero power and area footprint," in *Design and Test Europe Conference*, 2019.

[135] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[136] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations (ICLR)*, 2014.

[137] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy (Euro S&P)*, 2016.

[138] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *International Conference on Learning Representations (ICLR)*, 2017.

[139] K. N. Khasawneh, M. Ozsoy, C. Donovick, N. Abu-Ghazaleh, and D. Ponomarev, "EnsembleHMD: Accurate hardware malware detectors with specialized ensemble classifiers," 2018.

[140] P. D. S. Manoj, S. Amberkar, S. Bhat, A. Dhavlle, H. Sayadi, S. Rafatirad, and H. Homayoun, "Adversarial attack on microarchitectural events based malware detectors," in *Design Automation Conference*, 2019.

[141] J. R. Quinlan, "C4.5: Programs for machine learning," *Machine Learning*, vol. 16, no. 3, pp. 235–240, Sep 1994.

[142] U. Shaham, Y. Yamada, and S. Negahban, "Understanding adversarial training: increasing local stability of neural nets through robust optimization," *ArXiv e-prints*, 2015.

[143] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, no. 1, pp. 1–23, 2016.

[144] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware trojan: Threats and emerging solutions," in *IEEE Int. High Level Design Validation and Test W.*, 2009.

[145] Y. Jin, N. Kupp, and Y. Makris, "Experiences in hardware trojan design and implementation," in *IEEE Int. W. on Hardware-Oriented Security and Trust*, 2009.

[146] J. Cruz, F. Farahmandi, A. Ahmed, and P. Mishra, "Hardware trojan detection using atpg and model checking," in *Int. Conf. on VLSI Design*, 2018.

[147] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated annealing: Theory and applications.* Springer, 1987, pp. 7–15.

[148] M. Shayan, K. Basu, and R. Karri, "Hardware trojans inspired ip watermarks," *IEEE Design & Test*, vol. 36, no. 6, pp. 72–79, 2019.

[149] R. V. Hogg, J. McKean, and A. T. Craig, *Introduction to mathematical statistics.* Pearson Education, 2005.

[150] S. Murali, D. Atienza, L. Benini, and G. De Micheli, "A multi-path routing strategy with guaranteed in-order packet delivery and fault-tolerance for networks on chip," in *Design Automation Conf.*, 2006.

[151] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[152] I. Singh, A. Shriraman, W. W. Fung, M. O'Connor, and T. M. Aamodt, "Cache coherence for gpu architectures," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA).* IEEE, 2013, pp. 578–590.

[153] T. Shinde, S. Subramaniam, P. Deshmukh, M. M. Ahmed, M. Indovina, and A. Ganguly, "A 0.24 pj/bit, 16gbps ook transmitter circuit in 45-nm cmos for inter and intra-chip wireless interconnects," in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*, 2018, pp. 69–74.

[154] T. Dellsperger, C. Kromer, and G. Sialm, "Design of a 5 ghz vco in cmos," *Swiss Federal Institute of Technology Zurich: Zurich, Switzerland*, 2002.

[155] S. Subramaniam, T. Shinde, P. Deshmukh, M. S. Shamim, M. Indovina, and A. Ganguly, "A 0.36 pj/bit, 17gbps ook receiver in 45-nm cmos for inter and intra-chip wireless interconnects," in *2017 30th IEEE International System-on-Chip Conference (SOCC).* IEEE, 2017, pp. 132–137.