AWARe-Wi: A Jamming-Aware Reconfigurable Wireless Interconnection using Adversarial Learning for Multichip Systems

M Meraj Ahmed[a], Amlan Ganguly[a], Abishek Vashist[a] and Sai Manoj PD[b]

[a]Rochester Institute of Technology, Rochester, NY, USA, 14623
[b]George Mason University, Fairfax, VA, USA, 22030

ARTICLE INFO

Keywords:
multichip
medium access control
mm-wave
jamming-aware
machine learning
network-on-chip
wireless

ABSTRACT

Performance of the compute-intensive multichip platforms such as micro-servers and embedded systems are limited by the latency and power hungry chip-to-chip interconnections. Millimeter wave (mm-wave) wireless interconnection networks have emerged as an energy-efficient and low-latency solution for such multichip system communication. We refer such multichip systems with in-package mm-wave wireless interconnect as Wireless Network-in-Package (WiNiP). Despite providing performance enhancements, wireless channel, being an unguided medium, introduces potential security vulnerabilities inherited from traditional wireless networks such as jamming induced Denial-of-Service (DoS) and eavesdropping. Securing the systems against such induced threats often introduce large overheads and performance penalties. To address these challenges, we propose a WiNiP architecture that reuses the in-built Design for Testability (DFT) hardware for securing against external and Hardware Trojans (HT) induced internal attacks. The proposed architecture is capable of securing against adversaries with a reconfigurable wireless interconnection (AWARe-Wi). We deploy machine learning (ML) classifier to detect the threats. In addition, for a robust threat detection, we introduce an Adversarial ML (AML)-based approach in this work. To enable sustainable multichip communication in such systems even under jamming attack from both internal and external attackers, we design a reconfigurable Medium Access Control (MAC) and a suitable communication protocol. 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 for application-specific traffic patterns.

1. Introduction

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 multicore-multichip (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 multi-chip compute systems, such as the AMD EPYC [15] 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 System-in-Package (SiP) of MCMC demands an efficient intra-chip as well as an inter-chip communication, as disintegration increases inter-chip traffic significantly. Although Network-on-Chip (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 [5][8].

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 [17] 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 Wireless Interfaces (WIs) operating at GigaHertz (GHz) bandwidth in millimeter-wave (mm-wave) bands can mask off-chip I/O delay by establishing single hop, energy-efficient chip-to-chip communication links [24][12]. We refer such MCMC systems with mm-wave wireless interconnect as Wireless Network-in-Pack- -age (WiNiP) here.

Although extensive research has been carried out towards improving performance and energy dissipation in WiNiPs [24], 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 Denial-of-Service (DoS), eavesdropping (ED), and spoofing. Although each of these attacks require its own detection and defense mechanism, in this work we focus on persistent jamming-based DoS attack as it is one of the most common, simple and yet powerful attack on wireless systems. To replicate such an attack, we consider an external attacker that produces a high energy electromagnetic (EM) radiation that causes interference in the wireless medium of the WiNiPs.

In addition, to address the complexities and vulnerabilities arising from hardware design and manufacturing process, we consider a hardware Trojan (HT) which is mali-

ORCID(s):

ciously 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 Medium Access Control (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 [20]. To address and meet such constraints, we propose to re-use the existing Design for Testability (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, we also develop a MCMC wireless communication protocol 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 we utilize an Adversarial Machine Learning (AML) and adversarial training for the deployed ML classifiers. 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.

2. Related Work

Although WiNiPs are seen to outperform wired MCMC systems and provide energy-efficient on- and off-chip communication, a very little attention has been drawn to address the security in on- and off-chip WiNiP communication. In [7], a small-world graph based wireless NoC architecture was proposed to mitigate DoS attacks. But, small-world irregular topologies have negative implications on design and verification efforts. In [13], a secure wireless NoC architecture that can protect against DoS, ED and spoofing has been proposed. 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 [26]. In case of external jamming, the authors in [26] 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.

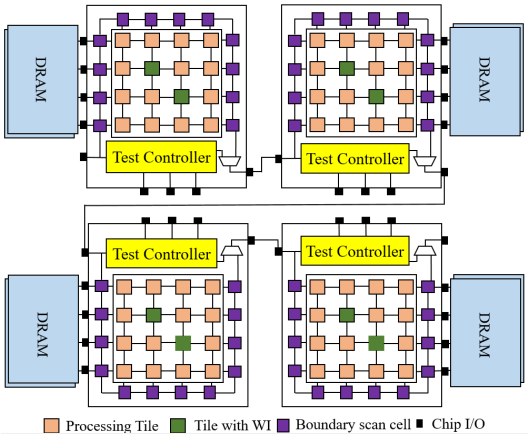

Figure 1: Proposed multichip WiNiP topology.

Moreover, the ML model used in [26] is unaware of the adversarial conditions that an intelligent attacker can exploit to camouflage its presence. In [14], authors developed a spoofing detection and defense mechanism based on received signal power for on-chip wireless interconnect. However, the proposed mechanism in [14] 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 [19]. 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 [20]. However, a directional antenna limits the multicast capability and limit WiNiP performance. Therefore, in this work we design a novel attack-aware MAC for the WiNiP.

3. WiNiP Interconnection Architecture

In this section, we discuss the proposed WiNiP interconnection architecture which covers the adopted topology, the

proposed hybrid MAC and physical layer.

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 last level cache (LLC). Tiles in each chip are connected with each other through a regular wired mesh-based NoC. For inter-chip communication, in each chip, we equip two NoC switches with WIs as shown in Fig. 1. 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 we assume a maximum of a single HT per chip in our analysis. Although inter-chip communication happens only through the WIs in functional mode, the MCMC system is compliant to Joint Test Action Group (JTAG) test architecture where their boundary scans are daisy chained. We leverage this JTAG infrastructure for enhancing the security of MCMC system.

3.2. Persistent Jamming-Aware Reconfigurable MAC

A wireless medium 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, we propose a reconfigurable MAC mechanism operating in two modes for sustainable communication even under persistent jamming attack. In the absence of persistent jamming attacks, we consider using a reservation-based MAC, termed as Normal MAC (NMAC) 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 3 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, we switch the MAC to Pseudo-random Noise (PN) encoded Asynchronous Code Division Multiple Access (ACDMA)

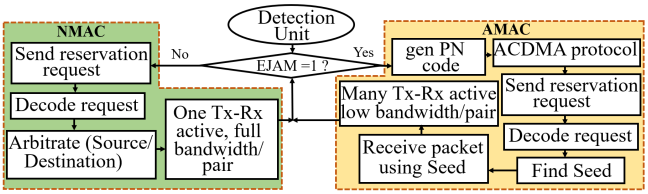

Figure 2: Overview of the reconfigurable MAC.

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 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 [2]. Moreover, Gold sequence can also support more users than both Kasami and m-sequence. Therefore, we consider generating PN codes using Gold sequence.

We use the hybrid Transmitter-Common (TC) [28] PN code protocol 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. We do not use AMAC 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 2 shows the proposed reconfigurable MAC with the underlying operations.

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, we discuss our proposed flow control that addresses these issues.

To ensure low area and latency overhead, we adopt a Virtual Channel (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 ex-

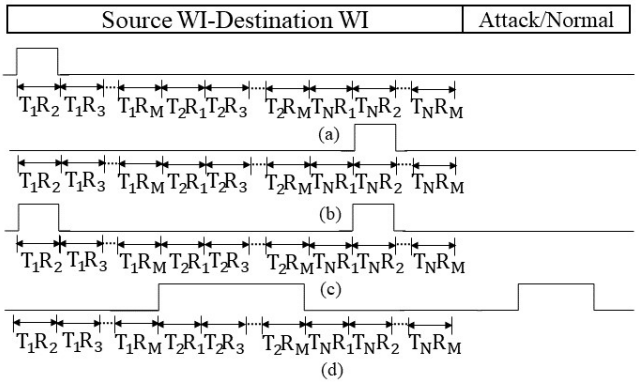

Figure 3: Channel reservation process.

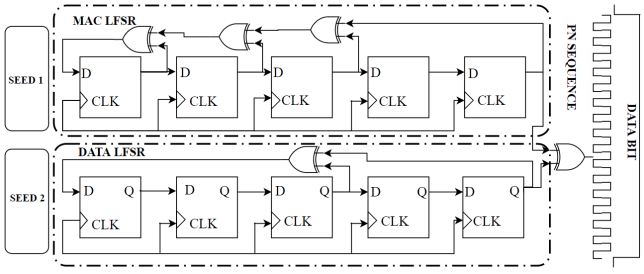

Figure 4: PN code generation using Data and MAC LFSR.

tracts 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, we propose a non-overlapping/non-interfering source-destination representation. As shown in Fig. 3(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 Fig. 3(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 multi-cast traffic [3]. 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. We consider providing the highest priority to attack conditions which is indicated by the attack flag in Fig. 3(d). After detecting a potential external jamming attack as described in subsection 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.

3.3.1. 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 Linear Feedback Shift Registers (LFSRs) according to the preferred polynomial pair and XOR their output to finally generate the desired Gold sequence. Fig. 4 shows the LFSR configuration to generate a 32 bit gold code. Moreover, to generate a different PN sequence for each of the transmitter, we choose different seed values for each of the transmitters.

3.3.2. ACDMA communication mechanism under attack

During any persistent jamming attack, all the WIs in the multichip system change the MAC to ACDMA mode as discussed in section 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 Fig. 3. 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 memory 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 [29]. The AMAC steps are also depicted in Fig. 2. The transmitter and receiver architecture will be discussed in the next section.

3.4. Physical Layer

To combat the persistent jamming, physical layer implements the components required for both NMAC and AMAC protocols along with WIs. We propose the use of on-chip miniature zig-zag antennas operating in the unlicensed 60 GHz mm-wave band to establish direct communication channels between the WIs. Such antenna provides a bandwidth of 16GHz for both intra and inter-chip communications [24]. We adopt the transceiver design from [31] [30].

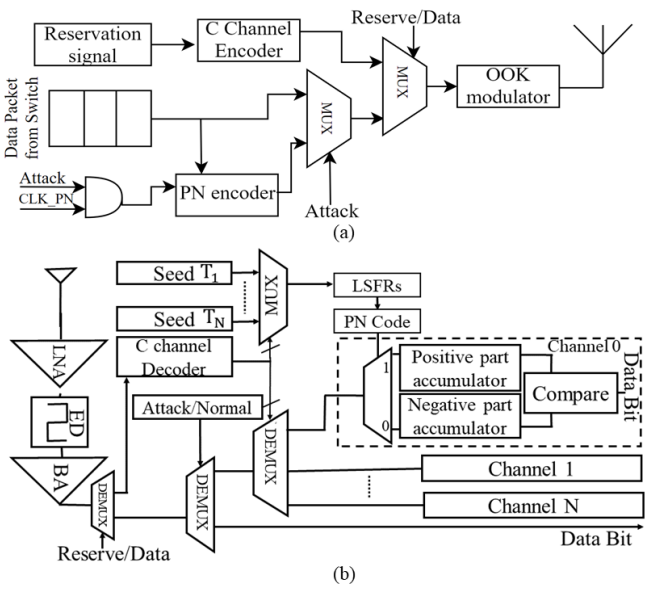

Figure 5: (a) Transmitter (b) Receiver block diagram

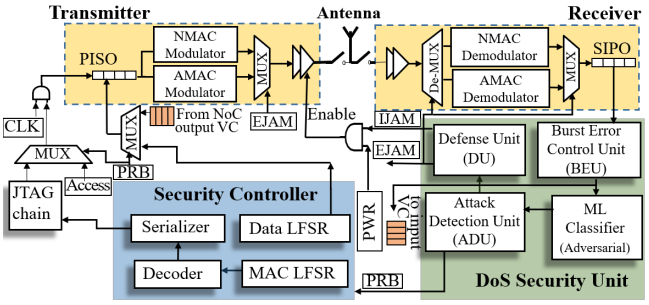

Figure 6: Proposed security framework.

Non-coherent On-Off Keying (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 [24]. 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 Fig. 5(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 Fig. 5(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. Attack Model and Detection

In this section, we present the attack model, our proposed detection and defense mechanism that ensures robust communication under external and internal jamming attack scenario using ML and AML approach.

4.1. Attack Model

In this work, as aformentioned, we consider persistent jamming-based DoS attacks 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 [6]. 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, we consider upto a single HT per chip 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 [1].

We employ ML techniques 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* [9, 25]. 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 [16, 21]. In similar manner, in this work, we introduce an adversarial attacker 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 [10]. Once the reverse engineered model is built, then, the attacker tries to estimate the model and introduce the per-

turbations by incrementally increasing the noise to the input features that are sensitive similar to [18] to evade detection or to induce false alarms. In this work, we utilize Fast Gradient Sign Method (FGSM) attack [9] 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.

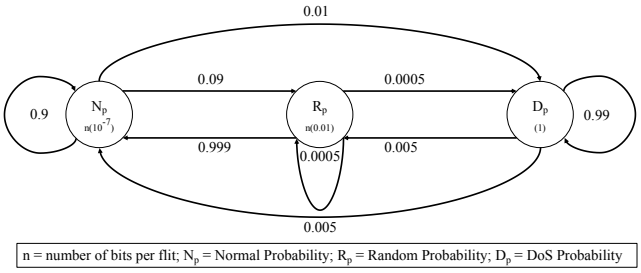

Figure 7: Markov Chain to generate training and test data

4.2. Attack Detection Methodology

To detect a persistent jamming attack we re-use the JTAG test infrastructure for probing the wireless interconnect. The architecture of the proposed security framework is shown in Fig. 6. When the Probe (PRB) signal is asserted to the security controller from the Attacker Detection Unit (ADU), 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 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, we equip the receivers of WIs with Wireless Security Unit (WSU) that will enable detection of persistent jamming from both internal HTs as well as external attacker. In the next subsection, we briefly discuss the architecture of the WSU.

4.2.1. 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 [6] 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.

4.2.2. Machine Learning for Attack Detection

As aforementioned, the considered attacks in this work 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. We experimented with multiple ML classifiers such as multi-layer perceptron (MLP), support vector machine (SVM), k-nearest neighbors (KNN), Decision tree (DT), and J48 to evaluate the robustness and efficiency of attack detection. For the MLP, we considered a single hidden layer with 10 nodes is utilized, with two neurons in the output layer. We utilize a polynomial kernel based SVM in this work, as it considers the combination of the input features as well as input features for classification. Similarly, we experimented with k-nearest neighbors with k=1 and 3 in this work. In addition, we consider two variants of decision classifiers namely DT and J48, where J48 is an optimized version of DT with reduced search space [22].

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 Bit Error Rate (BER) of 10^{-10} or less [30]. 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 [6]. 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.

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 Fig. 7. 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.

4.2.3. Strengthening Attack Detection with Adversarial Learning

In order to craft the adversarial perturbations, we consider a functionally reverse engineered ML classifier 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 [9] is given by

$$x^{adv} = x + \epsilon sign(\nabla_x L(\theta, x, y)) \quad (1)$$

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 [23]. 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.

4.2.4. 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 Fig. 6. 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.

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,

Table 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 [27], 16Gbps, 0.66 pJ/bit
Memory links	128Gbps, 6.5pj/bit [11]
Technology	65nm, 1V supply, 1GHz clock

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 Fig. 3(d) over the common reservation channel by setting the EJAM and reservation flag on the transmitter side. The reservation channel like the other ACDMA channels is resilient to jamming. Moreover, 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. 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.

6. Results and Analysis

Here, we evaluate the performance of the proposed unified test and security of WiNiP under different attack scenarios. 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.

6.1. Simulation Setup

Simulation of wireless interconnection requires a combination of multiple simulation tools. We use ASIC design flows with 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 [6]. The characteristics of the antenna and the transceivers are simulated in High-Frequency Structural Simulator (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 1.

We evaluate the proposed system 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 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 [4]. In order to map these traffic patterns to the MCMC environment, we consider multiple threads of the same application kernel running on the MCMC system where each processing core executes a single thread and the memory stacks are shared among threads. Before discussing the results for application specif traffic we first evaluate the performance of the ML classifiers.

6.2. Performance of the Attack Detection

Table 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 [13], we consider a similar threshold-based approach. For the neural network (MLP) a single hidden layer with 10 nodes is utilized. One can observe from Table 2, among different classifiers, KNN achieves high attack detection accuracy of nearly 99.87%, higher than other techniques. We anticipate this behavior, as no assumptions are made regarding the data during the training phase of KNN. We have experimented with $k = 1, 3$ for KNN and have observed a similar performance, hence considered $k = 1$ in this work 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 2, 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.

Table 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
Thresh	94.55	0.92	0.92	0.95

Table 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

6.3. Detection Accuracy with Adversarial Attacks

We also evaluate 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 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 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 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 6. 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. 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.

Table 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
Thresh	24262.63	22515.2	0.07

6.4. Performance Evaluation Under Persistent Jamming

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

6.4.1. 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. We consider three scenarios for our performance evaluation under internal attack. First, we consider MCMC system with one CWI for the entire (4 chip) system (1-CWI/sys). Second, we consider an MCMC system having one CWI per chip (1-CWI/chip). We compared these scenarios 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. The baseline system represents WiNiP operating in jamming-free condition. As we considered two WIs per chip, a system having more than one CWIs in a chip indicate a complete system failure and JTAG chain can be used for multichip communication with huge latency penalty It can be observed from Fig. 8 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.

6.4.2. 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 Fig. 9. It can be observed from the figure that, using a higher CL increases latency and energy consumption while providing higher security.

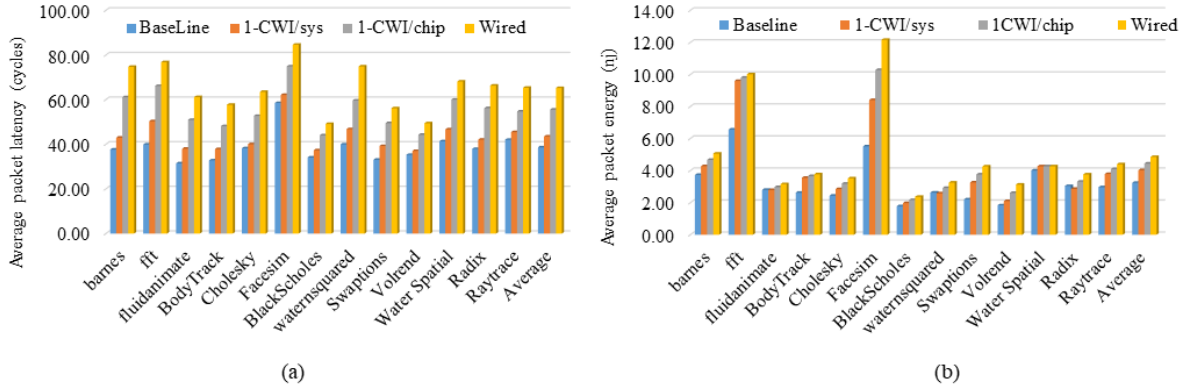

Figure 8: Performance evaluation (a) Latency (b) Energy under internal jamming attack for different systems.

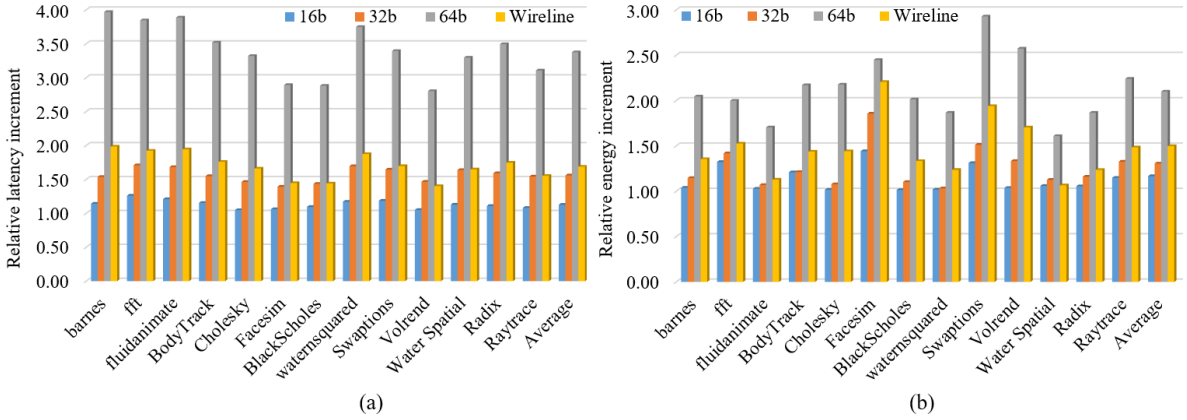

Figure 9: Performance evaluation (a) Latency (b) Energy under external jamming attack for different PN code length.

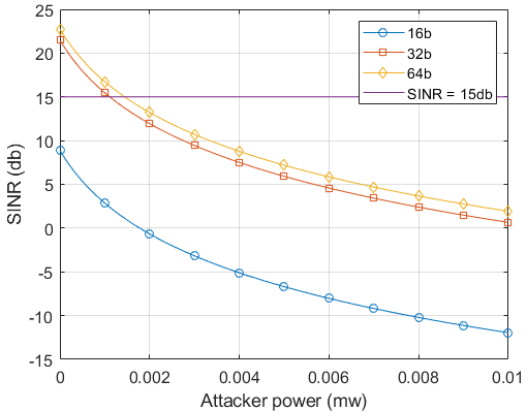

Figure 10: SINR value for different PN code length.

6.5. Optimum CL Selection

In AMAC mode communication, the system performance and communication security are heavily depended on PN code length. Fig. 9 shows the effect of PN CL on system latency and energy. In this subsection we analyze 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 power to decrease the Signal to Noise plus Interference Ratio (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 [24] 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 [27]. We consider one valid communicating WI pair and model other and attacker transmission as noise in the receiver side. Fig.10 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 Fig.9. 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 Fig.9. From Fig. 9 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.

6.6. Overall Area Overhead

In the previous sections we have observed the area overheads incurred by the ML classifiers. Here, we summarize the overall overhead of other components of the WSU. 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 0.17 mm^2 [31][30], making

Table 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	0.17	23	0.0625

the overhead only 3.9% of the transceivers. Table 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 [6].

7. Conclusions

In this work, we propose secure mm-wave wireless interconnection architecture for MCMC systems. While wireless interconnects can improve the performance of the off-chip communication in MCMC system through energy efficient single hop links, they are also vulnerable to various security threats like jamming-based DoS attack. 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 paper, 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.

References

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

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

[3] Ahmed, M.M., et al., 2018. A one-to-many traffic aware wireless network-in-package for multi-chip computing platforms, in: IEEE SOCC.

[4] Badr, M., Jerger, N.E., 2014. Synfull: Synthetic traffic models capturing cache coherent behaviour, in: ACM SIGARCH Computer Architecture News, IEEE Press. pp. 109–120.

[5] Chang, K., Deb, S., Ganguly, A., Yu, X., Sah, S.P., Pande, P.P., Belzer, B., Heo, D., 2012. Performance evaluation and design trade-offs for wireless network-on-chip architectures. J. Emerg. Technol. Comput. Syst. 8, 23:1–23:25.

[6] Fu, B., Ampadu, P., 2009. Burst error detection hybrid ARQ with crosstalk-delay reduction for reliable on-chip interconnects, in: IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems.

[7] Ganguly, A., Ahmed, M.Y., Vidapalapati, A., 2012. A denial-of-service resilient wireless NoC architecture, in: GLSVLSI.

[8] Ganguly, A., Chang, K., Deb, S., Pande, P.P., Belzer, B., Teuscher, C., 2011. Scalable hybrid wireless network-on-chip architectures for multicore systems. IEEE Trans. on Computers 60, 1485–1502.

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

[10] Khasawneh, K., et al., 2017. RHMD: Evasion-resilient hardware malware detectors, in: IEEE/ACM Int. Symp. on Microarchitecture.

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

[12] Laha, S., Kaya, S., Matolak, D.W., Rayess, W., DiTomaso, D., Kodi, A., 2014. A new frontier in ultralow power wireless links: Network-on-chip and chip-to-chip interconnects. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34, 186–198.

[13] Lebiednik, B., Abadal, S., Kwon, H., Krishna, T., 2018. Architecting a secure wireless network-on-chip, in: IEEE International Symposium on Networks-on-Chip (NOCS).

[14] Lebiednik, B., Abadal, S., Kwon, H., Krishna, T., 2018. Spoofing prevention via rf power profiling in wireless network-on-chip, in: Proceedings of the International Workshop on Advanced Interconnect Solutions and Technologies for Emerging Computing Systems, pp. 1–4.

[15] Lepak, K., et al., 2017. The next generation amd enterprise server product architecture. IEEE Hot Chips 29.

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

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

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

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

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

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

[22] Quinlan, J.R., 1994. C4.5: Programs for machine learning. Machine Learning 16, 235–240.

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

[24] Shamim, M.S., Mansoor, N., Narde, R.S., Kothandapani, V., Ganguly, A., Venkataraman, J., 2017. A wireless interconnection framework for seamless inter and intra-chip communication in multichip systems. IEEE Transactions on Computers 66, 389–402.

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

[26] Vashist, A., Keats, A., D, S.M.P., Ganguly, A., 2019. Securing a wireless network-on-chip against jamming based denial-of-service and eavesdropping attacks. IEEE Transactions on Very Large Scale Integration Systems (TVLSI) .

[27] Vijayakumaran, V., et al., 2014. CDMA enabled wireless network-on-chip. ACM JETC 10, 28.

[28] Wang, X., Ahonen, T., Nurmi, J., 2007. Applying cdma technique to network-on-chip. IEEE TVLSI 15, 1091–1100.

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

[30] Yu, X., Rashtian, H., Mirabbasi, S., Pande, P.P., Heo, D., 2015. An 18.7-Gb/s 60-GHz OOK demodulator in 65-nm CMOS for wireless network-on-chip. IEEE Trans. on Circuits and Systems I 62, 799–806.

[31] Yu, X., Sah, S.P., Rashtian, H., Mirabbasi, S., Pande, P.P., Heo, D., 2014. A 1.2-pJ/bit 16-Gb/s 60-GHz OOK transmitter in 65-nm CMOS for wireless network-on-chip. IEEE Trans. on Microwave Theory and Tech. 62, 2357–2369.