Securing a Wireless Network-on-Chip Against Jamming-Based Denial-of-Service and Eavesdropping Attacks

Abhishek Vashist, Andrew Keats, Sai Manoj Pudukotai Dinakarrao, *Member, IEEE*, and Amlan Ganguly, *Member, IEEE*

***Abstract*— Wireless networks-on-chips (NoCs) (WiNoCs) have emerged as a possible solution to the nonscalable multihop data transmission paths in traditional wired NoC architectures. Using low-power transceivers in NoC switches, novel WiNoC architectures have been shown to achieve higher energy efficiency with improved peak bandwidth and reduced on-chip data transfer latency. However, using wireless interconnects for intrachip data transfer over an unguided medium introduces additional security vulnerabilities in on-chip communication arising from either external attackers or internal hardware Trojans. In this article, we propose a mechanism to make the wireless communication in a WiNoC secure against persistent jamming-based denial-of-service (DoS) attacks and eavesdropping (ED) from both external and internal attackers. Persistent jamming attacks on the on-chip wireless medium will cause interference in data transfer over the duration of the attack resulting in errors in contiguous bits, known as burst errors. Therefore, we use a burst-error correction code to monitor the rate of burst errors received over the wireless medium and deploy a machine-learning (ML) classifier to detect the persistent jamming attack and distinguish it from random burst errors. In the event of a persistent jamming attack, alternate routing strategies are proposed to avoid the DoS attack over the wireless medium, so that a secure data transfer can be sustained even in the presence of persistent jamming. In the event of an external ED attack, we deploy a low-latency and lightweight data scrambling method to secure communication over the wireless channel. In the case of an internal ED, we propose a mechanism to identify the attacker and prevent the attack. We evaluate the proposed techniques on a WiNoC in the presence of DoS and ED attacks from both internal and external attackers. On an average, 99.87% of the attack on DoS detection was achieved with the chosen ML classifier. A bandwidth degradation of <3% is experienced in the event of both DoS and ED internal attacks. The wireless interconnects are disabled in the presence of a persistent external jamming DoS attack for security, therefore eliminating the advantages of the wireless interconnections making the performance of the WiNoC comparable with that of a wired NoC. Although scrambling overheads are incurred in the presence of an external ED attack, the overheads are minimized by adopting simple XOR-based encoding and decoding.**

***Index Terms*— Denial-of-service (DoS), eavesdropping (ED), jamming, machine learning (ML), network-on-chip (NoC), on-chip security, wireless interconnect.**

I. Introduction

With the advent of a multicore or a many-core paradigm toward enhanced performance, traditional bus-based interconnect mechanisms were found to be nonscalable from a design perspective. This led to the adoption of the network-on-chip (NoC) paradigm for interconnecting tens to hundreds of cores on a single die. Regular NoC architectures such as mesh or torus-based architectures have shown a reduced design complexity and provided benefits such as ease of replication, ease of verification, and reduction in time-to-market. However, such regular architectures resulted in nonscalable performance with an increase in the number of cores due to long multihop paths over wired links [1]. Along with other emerging interconnect technologies such as silicon photonics or through-silicon-vias (TSVs) for 3-D NoCs, wireless interconnects were envisioned to enable scalable communication fabrics in multicore chips [2]. Though emerging interconnects such as silicon photonics and 3-D TSVs provide high bandwidth communication, the design and overhead costs, and the concerns of reliability limit their adoptability [3]. In contrast, advancements in low-power millimeter-wave wireless transceivers, efficient on-die miniature antennas, and smart designs of hybrid architectures with wired as well as single-hop wireless links resulted in lower packet latency and energy consumption in on-chip communication and facilitated investigation of wireless NoCs (WiNoCs) in emerging many-core systems [4] as well as in multichip computing systems.

A case study is performed to compare a wired NoC and a WiNoC for a 64-core system with four wireless interfaces (WIs) overlaid on a mesh in a 65-nm technology node over a die of 20 mm $\times$ 20 mm. The size of each packet is set to 2 Kb. More details on the experimental setup is presented in Section VI-A. One can observe from Fig. 1 that the WiNoC improves the bandwidth per core and the energy per packet by 15% and 39%. This demonstrates the potential benefits of

Manuscript received January 31, 2019; revised June 2, 2019; accepted June 26, 2019. The work of A. Ganguly was supported in part by the U.S. National Science Foundation (NSF) CAREER under Grant CNS-1553264. *(Corresponding author: Sai Manoj Pudukotai Dinakarrao.)*

A. Vashist, A. Keats, and A. Ganguly are with the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: av8911@rit.edu; axk7655@rit.edu; amlan.ganguly@rit.edu).

S. M. Pudukotai Dinakarrao is with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA (e-mail: spudukot@gmu.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2019.2928960

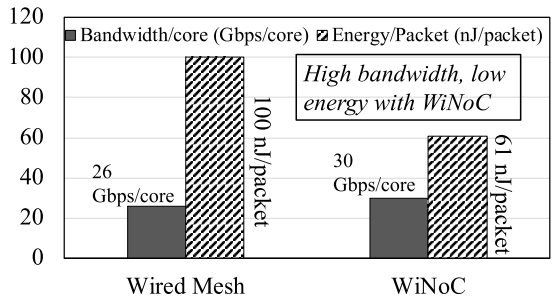


Fig. 1. Comparison of wireless versus wired NoCs in terms of bandwidth and energy consumption.

adopting WiNoCs for interconnecting multicore or many-core processors.

Although extensive research has been carried out toward improving performance and energy dissipation in WiNoCs, relatively little attention has been given to the information integrity and security or privacy aspects of WiNoCs. While the security of traditional wired NoCs against various kinds of attacks such as hardware Trojans (HT) and eavesdropping (ED) has resulted in appropriate defense mechanisms, the additional threats that unguided wireless interconnects can engender have not received the necessary attention. Wireless interconnects in WiNoCs are vulnerable to attacks, similar to those encountered in other wireless networks such as sensor networks or mobile networks. Furthermore, conventional defenses against persistent jamming attacks such as frequency or channel hopping [5] are not applicable in a WiNoC, as the WIs have access to a single shared channel and extremely limited resources. This calls for an embedded defense mechanism for current and emerging WiNoC-based multicore systems.

Many different security attacks such as Denial-of-Service (DoS), ED, and spoofing are possible in a WiNoC, where the communication happens over a shared wireless medium, with each attack requiring its own detection and defense mechanism. In this article, we focus on a persistent DoS attack that jams the wireless medium as well as the ED attacks, as these are some of the most prominent and low-complexity yet powerful attacks on wireless communication systems. We consider an external attacker who produces a high-energy electromagnetic (EM) radiation that causes interference in the wireless medium used by the WiNoC. Moreover, it is also possible that an 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 medium access control (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 [5], those techniques are not directly applicable to the WiNoC scenario due to a specific architecture and MAC constraints in WiNoCs. Similarly, for the ED attack, we assume an external receiver can be tuned to the wireless channel used for the WiNoC resulting in information leakage or that an internal HT passes packets received over the wireless channel downstream to a malicious node even when the packet is not addressed to the particular receiver. Such an attack can lead to leakage of sensitive information either directly through an external eavesdropper or through an internal malicious node.

In this article, we propose a mechanism to detect and recover from persistent jamming-based DoS attacks that can disable the wireless interconnections in the WiNoC. We present and evaluate the design of a detection unit that monitors the number of interference-generated errors in the received data and employs a machine-learning (ML) classifier to distinguish between random errors and those due to an attack and a defense unit (DU) that aids the WiNoC in recovering from an attack based on whether the attacker is internal or external. We propose to equip every wireless transceiver in the WiNoC with the proposed DU. In addition to the DoS attack, we propose to equip each wireless transceiver with a mechanism to prevent information leakage through ED. To achieve this, each wireless transceiver will encode sensitive data using a secret code to minimize the overheads as well as maintain a high throughput. We use a thorough simulation framework using tools at various levels of abstraction to evaluate the WiNoC with the proposed DoS and ED detection and defense mechanisms.

II. Related Work

Considerable research has been done on developing techniques for securing conventional NoCs and NoC-based multicore processors [6]–[9]. However, these security measures are confined to wired NoCs and not applicable to wireless interconnections. Very little attention has been paid to this important problem of securing on-chip wireless communication, although it has been identified as an important challenge to be overcome to make WiNoCs a reality [4]. In [10], a small-world graph-based WiNoC architecture was proposed to mitigate DoS attacks. On the other hand, hash-based authentication to prevent ED has been proposed in [11]. In [12], a secure WiNoC architecture has been proposed that can protect against DoS, ED, and spoofing but engages the Operating System to block DoS attacks in a WiNoC with a contention-free channel access, which is the type of WiNoC considered in this article. In addition, there exist techniques such as signature-based attack detection [13] and event-based attack detection, which are primarily carried out in software, as the hardware to support such techniques will incur overheads. Though such techniques can detect anomalies, they are hampered by its large latencies and processing overheads, which might not suit a multicore NoC. Furthermore, threshold-based attack detection [14] can be seen as another viable approach for attack detection with low complexity. However, if one utilizes the recoverable error rate as a threshold to distinguish burst errors and jamming-induced errors, the chances of false negatives could be high, as unrecoverable burst errors need not always be caused by jamming. In this article, detecting and defending against jamming as well as ED attacks in WiNoCs have been addressed in the NoC itself.

Similarly, ED poses another threat in securing the communication information. Encryption has been widely proposed in order to secure the communication information. Some of the encryption techniques such as asymmetric key encryption, though efficient, pose large overheads, especially in the case of on-chip communications due to the utilization of hash tables and computational complexities. This necessitates the adoption of symmetric key encryption techniques such as Advanced Encryption Standard (AES) [15]. Though AES has a proven robustness against side channels in the networking domain, adapting it for WiNoC communications adds large processing overheads and, thus, is not feasible. In contrast, this work performs encoding to scramble or mask the data to protect against external ED and an embedded functional block to detect and prevent internal ED, leading to lower overheads with sophisticated detection schemes for different kinds of attacks.

On the other hand, although ML has been used in the context of NoC systems for congestion-aware routing [16], it is not used for securing NoC, especially against DoS attacks due to resource constraints. However, there exist works on detecting DoS attacks on cloud or IoT systems. We review some of them and outline the differences here. In [17], a decision-tree (DT)-based algorithm is devised for detecting DoS attacks in cloud environment. Furthermore, it is combined with signature detection techniques for improving efficiency. Similar works using radial basis function (RBF) neural networks (RBFNNs) [18] and artificial NNs (ANNs) [19] are proposed, and [20] presents a comparison of different ML algorithms when detecting distributed DoS (DDoS) attacks in cloud and IoT devices. The work in [21] employs 23 features to detect the DDoS attacks using different ML classifiers. Despite having the similar objective of detecting DoS/DDoS attacks, the constraints, protocols, and traffic flow are different for miniature NoC systems.

Thus, the main differences and challenges compared with existing works using ML for security against DoS/ED attacks can be outlined as follows: in the existing works, the detection is carried out in a cloud or resource-ample environment, where complex computations can be afforded. However, on an NoC-like miniature system that is considered in this article, the overhead and processing resources are limited and play a pivotal role. As such, a direct adoption is inefficient and leads to large overhead and performance penalties.

III. System Description

Here, we describe the architecture of the WiNoC, which is the platform for which our security systems are designed. This adopted WiNoC architecture is sufficiently generic and adopts elements from various designs over the past few years.

A. Wireless NoC System Architecture

In the adopted WiNoC architecture, each core in the multicore chip is connected to an NoC switch through a network interface (NI). The switches are then connected by wired links forming a mesh topology. We adopt a mesh architecture for the wired NoC topology due to its low complexity, ease of verification, and ease of manufacturing due to uniformity in link lengths. However, other topologies such as torus or small-world can be chosen if required by the system design constraints. In addition to the wired links, a few NoC switches are equipped with an additional port connected to a wireless transceiver to access to the mm-wave channel, thus forming a hybrid WiNoC architecture. These switches are referred to as WIs.

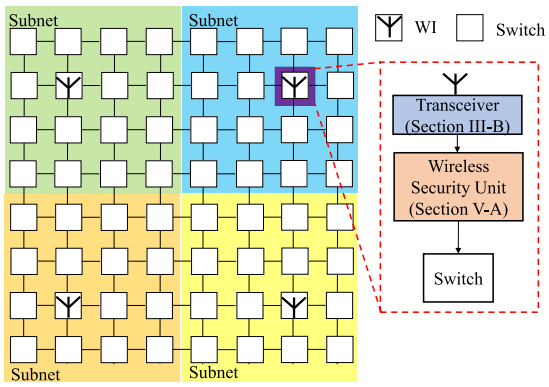


Fig. 2. System architecture of the proposed secure WiNoC.

Based on several previous works such as [22], we partition the mesh into multiple subnets to deploy the WIs among the NoC switches, as shown in Fig. 2. A central switch in each subnet is a WI to facilitate access to the wireless medium. The selection of subnet size (or the number of WIs) offers a tradeoff between the performance of the WiNoC and the area overhead of the WIs, which can be designed with system-level simulations. The underlying cores are not shown for the purpose of brevity. In this article, we propose to equip the WIs with a wireless security unit (WSU), which can detect and protect against persistent jamming and ED attacks from both internal and external attackers. The WSU is embedded in the WIs so that it can process the data and detect the attack before the data pass downstream to other NoC switches. More details on WSU are presented in Section V-A.

B. Wireless Interconnections

We propose the use of on-chip embedded miniature antennas operating in the 60-GHz mm-waveband unlicensed by Federal Communications Commission (FCC), which can establish direct communication channels between the WIs. We intend that the chosen antenna to be compact as well as nondirectional, so that they can communicate with other WIs in all directions in the WiNoC. We adopt the 60-GHz zig-zag antenna with these characteristics from [22].

To ensure high bandwidth and energy efficiency, we adopt a transceiver design where low-power design considerations are considered [23], [24]. Noncoherent on-off keying (OOK) modulation is chosen, as it allows relatively simple and low-power circuit implementation without the need for power-hungry carrier recovery and high-frequency synchronization circuitry. Each WI is a combined transceiver with a single antenna enabling half-duplex communication. Parallel data from an NoC switch are serialized using a parallel-in

serial-out (PISO) register before transmission and vice versa after the reception, where they are received into a serial-in parallel-out (SIPO) buffer. The PISO buffer receives data from the output virtual channel (VC) of the transmitting WI, while the SIPO sends the received data to the input VCs of the receiving WI, as shown in Fig. 3.

To avoid nonscalable central arbitrations and power-hungry synchronization across the chip and facilitate contention-free wireless channel access, we adopt a distributed wireless token-passing MAC mechanism to grant the access of the shared wireless channel to only the WI possessing the token. Each WI can only occupy the token for a predetermined maximum time that is optimized based on system-level simulations.

We use a forwarding-table-based routing algorithm over precomputed shortest paths along a minimum spanning tree (MST) determined by Dijkstra's algorithm. Consequently, deadlock is avoided by transferring flits along the extracted shortest path routing tree. The routing decisions are made locally based on the forwarding table for determining the next hop and is done only for the header flit, reducing computing requirements and maintaining global routing information.

IV. Security Attack Model on WiNoCs

Here, we discuss the attacks and their manifestations on the adopted WiNoC considered in this article. Several security and privacy attacks have emerged in recent times on multicore processors [25]. In this article, we consider persistent jamming-based DoS attacks and ED (arising internally as well as externally) on the wireless interconnections of a WiNoC. In the presence of 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 over a relatively long period of time, 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 multiple consecutive flits of a packet. On the other hand, burst errors in both wired and WiNoC links can happen as a random event as well. Burst errors can also be a result of power source fluctuations, ground bounce, or crosstalk [26]. However, the burst errors due to random events such as crosstalk will be relatively short-lived, typically, a single clock cycle, due to the data transition pattern in that cycle. On the other hand, burst errors resulting from persistent jamming could be sustained for a longer duration, as a short DoS attack is not an effective attack.

A few burst errors caused by a short-lived DoS can be corrected/detected by a burst-error correction/detection (BEC/BED) code depending on its correction capability. In the absence of such a BEC mechanism, a request for retransmission can be sent in the case of erroneous flits from the upper layers of the NoC protocol stack. Therefore, to be truly effective as an attack, the jamming has to last for a relatively long duration to cause enough flits to be in error such that the existing BEC mechanism either cannot correct it or retransmission requests are prohibitively expensive due to a potentially large number of requests. Therefore, we need a mechanism to detect jamming-based DoS attacks and distinguish it from a random burst error. In this article, we consider attacks either from a single external attacker or a single internal HT-based attacker that affect one or more WIs in the WiNoC. The jamming signal can be caused by an external source equipped with an RF transmitter tuned to the spectral band used in the WiNoC. Another likely scenario is when a particular WI already existing in the WiNoC is affected by an HT that forces the WI to ignore the contention-free MAC mechanism and continue to inject traffic from the transmitter of the WI. This constitutes an internal attack. We do not assume that an additional WI is placed as a Trojan in the chip as that would be relatively easy to detect. Rather, one of the existing WIs is infected by the Trojan and ignore the MAC rules and create jamming even when it is not supposed to transmit over the shared wireless medium. The potential sites of this HT are shown in Fig. 3.

In a similar manner, we consider that the ED attack can arise either from an external or internal attacker. In both cases, we assume that the attacker is passive, thus hard to detect and can receive any information communicated between different nodes in the WiNoC by tuning into the unguided and unprotected wireless channel. For the external eavesdropper, we consider a passive external receiver tuned to the band used in the WiNoC with enough sensitivity capable of receiving the data transmitted over the wireless channel. In the case of internal eavesdropper, the passive attacker receives data that are not addressed for it and route it downstream to a malicious agent.

V. Secure Wireless NoC

Now, we discuss the proposed mechanism of securing the adopted WiNoC against the DoS and ED attacks, as discussed before. To enable the proposed secure WiNoC, each WI is equipped with a WSU to sustain the functionality of the interconnection fabric even under attack.

A. System Architecture of Proposed WSU

The proposed WSU shown in Fig. 3 has two main components: the DoS security block and the ED security block. The DoS security block consists of a linear feedback shift register (LFSR) called MAC-LFSR, a burst-error control unit (BEU), an attack detection unit (ADU), and a DU. In the normal mode of operation, the data flits are received at the SIPO buffer of an NoC switch equipped with a WI. Upon reception of flits at the receiver's SIPO buffer, flits are sent to the BEU. The BEU then detects a burst error and sends its output to the ADU. The BEU employs the BEC proposed in [26] to detect burst errors. The corrected flits after (BEU) are sent to the input VCs of the NoC switch to be routed downstream in parallel to the error-related information (as discussed in Section V-B) being sent to the ML Classifier. This removes the DoS detection mechanism from the critical path of the data transfer. The ADU further comprises of an intelligent unit, which uses an ML classifier, and an attacker detection unit. The ML classifier is responsible for detecting if the system is under attack based on the input it receives

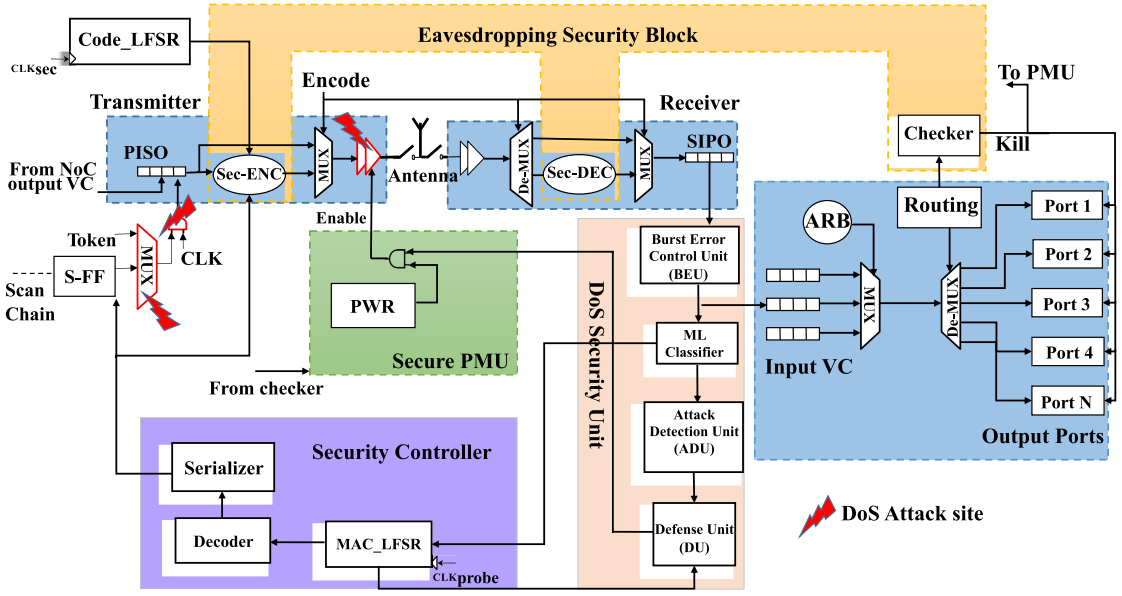


Fig. 3. WSU architecture in a WI.

from the BEU. More details of the ML classifier is presented in Section V-B. 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 the ML classifier and determines if the attack is internal or external as discussed in Section V-B2. Based on the kind of attack, the corresponding security measure is chosen.

The ED Security Block consists of an LFSR called code-LFSR, security encoder (Sec-ENC), and security decoder (Sec-DEC), as shown in Fig. 3. The code-LFSR generates codewords pseudorandomly that are used to encode the data flits using parallel bitwise XOR gates in the Sec-ENC. The Sec-DEC in the receiver uses the same code to XOR the received flits to recover the original data. To protect against brute-force ED, the code-LFSR is clocked periodically to generate a new code. This will protect against a suspected external ED. To protect from internal ED, each packet that accesses the wireless interconnect will have the address of the intermediate WIs embedded in a header field in addition to its final destination. A rule-based checker in the WI compares the address of the target WI of the received packet headers with that of the WI to verify if it is a legitimate packet or if it is being eavesdropped. The operations of these components are elaborated in Section V-B.

B. DoS Attack Detection and Defense

In order to detect a DoS attack, we deploy an ML classifier in this work. First, we present the details of the ML classifier and modeling of the DoS attack, followed by DoS attack detection and activated defense in the event of DoS attack detection.

1) 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). In the proposed WiNoC, 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 to evaluate the robustness and the efficiency of attack detection in the proposed system. The different ML classifiers considered here are a multilayer perceptron (MLP), a support-vector machine (SVM), k-nearest neighbors (KNN), a DT, and J48 classifiers. The rationale for experimenting with different classifiers are: 1) there exist no unique classifier that has "perfect" yield; 2) different classifiers have different resource requirements and performance (accuracy and latency); and 3) the chosen classifiers represent different branches of ML, thus representing a wide spectrum of ML classifiers. The ML classifiers in the ADU uses an offline learning with runtime inference to alleviate the complexity and processing overheads and facilitate faster inference (attack detection). The ML classifiers output a flag signal when an attack is detected. The ML classifier does not send any data to the switch buffers. This prevents an ML classifier from creating any DoS attack. In addition, we also assume that the detector unit along with all other security blocks is designed, verified, and tested in a secure environment, similar to secure integrated circuit design, thereby preventing any HT insertion in the security blocks.

In order to train the ML classifier, the attacks mentioned in Section IV are deployed on a WiNoC (shown in Fig. 2) with no security mechanisms deployed. A cycle-accurate NoC simulator was modeled to operate in one of the three modes: normal, random burst errors, and DoS 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 [24]. The second mode (random burst errors) is modeled with higher BERs as the burst errors are contiguous bits of flits. BERs of 10^{-5} are used in this case [26]. Finally, under the DoS attack, a high BER of 0.5 is assumed, as for identically

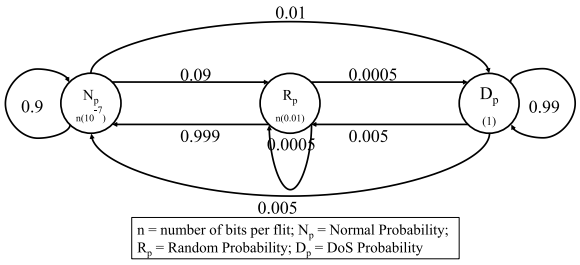


Fig. 4. Markov Chain to control system operating state.

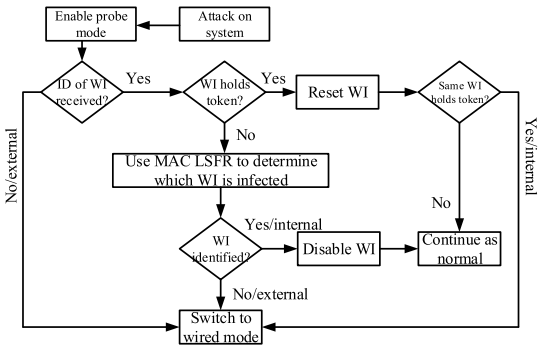


Fig. 5. Algorithm design for reaction to DoS jamming attack.

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 the data bits are represented as the presence or absence of transmission. Therefore, a persistent 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 BER information, 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. 4. The manifestation of the DoS attack is considered to result in the same kind of burst errors for both the internal and external attackers. The probability of staying in the attack mode, when already under attack is considered high, as a persistent jamming attack is effective only when it is sustained for a long duration. The probability of staying in a random burst-error mode when already in it, is modeled as low as random burst errors are short-lived phenomena. The probability of transition into a normal mode from a random burst-error mode is therefore high. The specific probability values can be altered to model any particular scenario. These observed data (number of errors, flits transmitted, and flits received) along with the operating mode as encountered in each WI are used to train the ML classifier at that WI. As the duration of the individual states is determined by the Markov chain randomly, each specific instance of the states has varying duration, resulting in a diverse training data set.

For the inference, i.e., attack detection, the ML classifiers are fed runtime information, 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 the system. Training of ML classifiers is performed with 100 000 cycles of data.

2) DoS Attack Detection Unit: In this section, we discuss the logic block that is designed to distinguish an external attacker from an internal one in the proposed secure WiNoC, ensuring different defense mechanisms are activated. The detector takes as an input the signal from the ML classifier that detects the occurrence of a jamming-based DoS attack. On the detection of an attack, the ADU activates the probe mode, in which all the WIs operate according to the token-based MAC mechanism controlled by the MAC-LFSR. The MAC-LFSR is enabled when the ML classifiers of any of the WIs detect an attack. They send this single-bit signal to the MAC-LFSR. We consider the MAC-LFSR to be located in a secure part of the chip and it is reasonable to assume that it is not affected by the wireless jamming attack model assumed here. The MAC-LFSR then grants access to the wireless medium to each WI in a pseudorandom pattern. A probe-clock (CLK_{probe}) triggers the MAC-LFSR to generate the encoded GRANT signal that is decoded to create a one-hot signal that is sent over a pipelined link to the transmitters of all the WIs. A parallel-load shift register is used to serialize this one-hot signal. The token register in each WI is converted into a scan Flip-Flop. At each transmitter, this signal is ANDed with the power supply routed from a secure power management unit (PMU) [27], which is not vulnerable to the wireless attacks, to regulate the power supply to the transmitter. Thus, only one transmitter transmits data flits over the WI in one instance.

The very first signal 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. The probe mode is then terminated and the decision is sent to the defense block for an appropriate action. However, if, in this case, there are 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 the 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. The identity of this WI is then passed to the defense block. The data packet that each WI sends in the probe mode can be preprogrammed pseudorandom data that can be distinguished from random bits when the WIs receive this packet. It could also be same or different for each sending WI. Alternatively, these can be generated by a local LFSR and compacted into signatures at the receiving WIs to match with a known signature. Therefore, the algorithm declares the WI that is enabled by the MAC-LFSR in which case correct data packet or signature is received as the internal attacker. Fig. 5 shows the ADU process.

3) Defense for Security Against DoS Attack: The ADU passes the address of the WI that is determined to be the attacker to the DU. In the case when the attacker is an external

agent, the address is an all-zero string. If the address received indicates an external attacker, the DU sends a signal to the secure PMU to shut down all the WIs and also update the routing tables of the WIs such that the wireless links are not used for data routing. These updates of the routing tables can be done without hardware overhead as these alternative values can be precomputed for each WI for the alternative shortest path routing when the WIs are not available and stored in the operating system. Therefore, in this case, all the WIs are disabled and data are routed by the wired links, eliminating the advantage of the wireless interconnections. In order to benefit from the wireless interconnection, the probe mode is periodically activated by the ADU to check if the attack has stopped. In this case, the use of the WIs can be resumed by using the PMU and by updating the routing tables.

If the address passed on to the DU indicates the address of an internal attacker, the DU sends a signal to disable only the power supply to the indicated WI and updates the routing table of its NoC switch to not use the WI. In this way, only the Trojan-infected WI is disabled and the rest of the WIs continue to use the wireless medium. Unlike the previous case, as the attacker is an internal HT, the associated WI may never be safe to use again, and, therefore, will be permanently disabled using the PMU and quarantined. The core or cores attached to the infected WI will continue to route their packets over wired links using the NoC switch, as the HT does not influence the wired part of the NoC in the threat model that we have considered in this work.

C. Defense Against Eavesdropping

We discuss protection against passive ED as it is relatively easy to launch and very difficult to detect. We discuss our protection strategy against both an internal and an external passive ED attack from a single attacker.

1) Protection From External Eavesdropper: For a single-agent external passive ED, it is complex and nearly impossible to detect with the available resources and capability of the WiNoC. This is because there may not be any change in the behavior of the overall system during such an attack. Furthermore, we assume that the WiNoC communications will have enough power and/or the ED attacker is sensitive enough to pick up the transmission and decipher the information. This is in line with real-world ED attack scenarios. The attacker needs to be equipped with a wireless receiver tuned to the wireless channel used in the WiNoC and has basic depacketization functions that are extremely simple and lead to low overhead in NoCs, and, therefore, easy to instantiate.

In order to address this threat, we propose to deploy a simple XOR-based data scrambling approach. The header flits are not encoded to enable routing as in traditional networks. The rest of the flits, which are the body flits, are XORed with a codeword from each WI and transmitted over the wireless channel. We propose to use the same length of the codeword as that of each flit with parallel bitwise XOR gates to reduce the delay in communication. Therefore, the bandwidth is not affected, as the number of bits transmitted for a flit does not change. At the receiver, the same code will be used to XOR the received flit to receive the uncoded data back. In general, unless an eavesdropper has the same code, it cannot decode the received flit. However, with enough time, an eavesdropper can determine the used code with brute force trials. Therefore, such schemes continuously change the code used by each transmitter. In order to change the code periodically, we generate the codes from an LFSR in each WI. The LFSR can be of the same length as that of the flit size (in number of bits). If a higher degree of pseudorandomness is desired, then a larger LFSR can also be used. In this article, we consider the LFSR to be of the same size as that of the flit. We refer to these LFSRs as code-LFSRs. The enable signals for these code-LFSRs to cycle through and generate a new code can be routed from the Security Controller through the serializer in the normal mode of operation (not in the probe mode when a jamming attack has been detected by the ADU). The special all-"1" code can be used to signal all the code-LFSRs to change the codes they are creating to the next pseudorandom code in all the transmitters. All transmitters have the same code-LFSR that is shared with the receivers collocated with the transmitter. This code is used by the security encoder (Sec-ENC) and the security decoder (Sec-DEC) in each WI, as shown in Fig. 3. Therefore, the code used by all the WIs is the same at all times. Each transmitter does not need to have a unique code-LFSR as this mechanism is for protection from an external eavesdropper and not an internal one.

2) Protection From Internal Eavesdropper: We model the internal ED as follows: we assume that one of the WIs is an internal eavesdropper. The attack model is such that this WI is either always or intermittently processing data packets transmitted over the on-chip wireless medium, which are not meant for it. Therefore, the attacker WI can receive and leak (to the outside) data that are not meant for it. This can be achieved with an HT that is embedded at the wireless input port, which does not allow the port to drop a packet that is not addressed for the particular WI. As this is an internal attacker, we propose a mechanism to detect such an attacker and to protect the WiNoC once such an attack is detected.

In order to detect the internal eavesdropper, power consumption-based detectors could be deployed. However, deploying such power measurement units incurs additional silicon footprints as well as computational overheads even in the absence of attacks. Therefore, we propose equipping the input port of each WI with a low-complexity rule-based checker. Moreover, as a WI transmits a packet over the wireless medium, it will embed the address(es) of the recipient WIs, which may then pass the packet further downstream to the final destinations. The rule-checker will match the WI address(es) of the header with the local address of the receiving WI. If there is no match, the WI should not pass this header to any downstream port and kill the packet to avoid packet duplication in the WiNoC. However, if this WI sends this packet to any outgoing port including the local port to the core, the checker raises a flag and this triggers an action in the secure PMU. The PMU then powers down the particular WI to prevent it from ED further.

The location of the checker is critical to the reduction of the overheads and the delay in detecting such an ED. As the location of the checker should be downstream from the logic block that is supposed to flush out a packet not meant for the WI, we propose to implement this checker after the input arbiter of the WI switch. In this way, an eavesdropped packet can be detected if it is not flushed out of the input buffers and progresses to the next step of routing. As during routing the destination address of the header flit will be parsed anyway, it can also be used in parallel to check for ED. This will minimize the additional overhead of this checking. Moreover, in this way, we do not delay the routing of the header of all legitimate packets due to this checking. If the result is positive (detected ED), then the flag is simply sent to the PMU that will prevent further reception of packets at that WI. In addition, the flag is also sent to all the output ports of the WIs to flush out the current packet when routing is completed to prevent information leakage of that packet. This ensures quick reaction on the detection of an internal ED. The flushing of the input or output buffers is achieved by activating the reset on the buffers without the need for any additional circuitry.

However, there are some exceptions to the proposed internal eavesdropper detection and defense. For instance, when the packets are broadcast to all the WIs or the eavesdropper WI happens to be one of the addressees in the packet header, the proposed defense mechanism will fail and will have to rely on mechanisms at higher layers of the system such as the application layer.

VI. Experimental Evaluation

In this section, we present the evaluations of the proposed secure WiNoC and the simulation tools used to evaluate it.

A. Simulation Setup

Simulation of wireless interconnection requires a combination of multiple simulation tools. We use ASIC design flows with a Synopsys Design Compiler using 65-nm chip multiprocessor standard cell libraries (https://mycmp.fr/) to model the digital parts of the WiNoC such as NoC switches and the WSU. The BEU encoder and decoder are implemented as two pipelined stages in the WIs to accommodate their delay [26], thereby maintaining the pipelined communication of the WiNoC. Each switch has three pipeline stages implementing backpressure flow control [28]. We consider each input and output port of a switch including those with the wireless transceivers to have eight VCs with a buffer depth of 4 flits for all the architectures considered in this work. We consider a packet size of 64 flits with a flit size of 32 bits in our experiments. A uniform random traffic distribution is assumed with a self-similar temporal behavior at a maximum injection load of 1 flit/core/cycle to evaluate the NoCs under worst case traffic. All the digital components are driven by a 2.5-GHz clock and 1-V power supply. The delay and energy dissipation on the wireline links is obtained through Cadence simulations considering the specific lengths of each link based on the NoC topology assuming a chip of 20 mm $\times$ 20 mm. The adopted wireless transceiver circuits consume 2.075 pJ/bit at 16 Gb/s in 65-nm technology [23], [24].

The adopted antenna has a 3-dB bandwidth of 16 GHz [22]. The characteristics of the transceivers, routers, and wired links are annotated into a system-level cycle-accurate simulator to evaluate the performance of the WiNoC in the presence of DoS attacks and the proposed defense mechanism. The simulator monitors the progression of flits on a cycle-by-cycle basis accounting for all flits that move or are stalled. We evaluate the proposed system in terms of average packet latency, peak bandwidth per core, and average packet energy. Average packet latency is defined as the number of cycles required for a packet to reach its final destination after being injected on an average. Peak bandwidth per core is defined as the number of bits received per core of the WiNoC per second with full injection load. Average packet energy is the average energy dissipated by a packet to be transferred to the final destination over the WiNoC fabric through switches, wired links, and wireless links.

B. DoS Attack-Detection Performance by ML Classifier

Table I presents the accuracy and robustness [in terms of precision, recall, and the area-under-curve (AUC) metrics] of different ML classifiers when deployed to detect the DoS attacks. The higher the value of accuracy and robustness metrics, the better the performance will be.

One can observe from Table I that among different classifiers, KNN achieves a high attack detection accuracy of nearly 99.87%, which is higher than the other techniques. We anticipate this behavior, as no assumptions are made regarding the data during the training phase of the KNN. For the KNN, a Euclidean distance function is employed with $k = 1$ due to its lower complexity. Therefore, the KNN is deployed in this ADU for attack detection. Though SVM displayed 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. For the neural network (MLP), a single hidden layer with 20 nodes is used. It can be argued that the hyperparameters of the ML classifiers can be tuned to improve the performance; however, optimizing the ML classifiers is not the focus of this article. To compare the ML classifiers with a heuristic method, we consider a threshold-based approach. As shown in Table I, the threshold-based mechanism is not as accurate as the chosen ML (KNN) approach. Despite having low latency, the threshold-based approach has higher area and power consumption due to the involved floating-point computations and comparisons, as shown in Table II. In this threshold-based approach, two thresholds are necessary to separate among the attack mode, the burst-error mode, and the normal mode. The thresholds are computed based on the same data that were used to train the ML algorithms. The threshold between the attack mode and the 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 the burst

TABLE I
ATTACK DETECTION PERFORMANCE OF ML CLASSIFIERS

ML classifier	Accuracy (%)	Recall	F-score	AUC
MLP	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 II
OVERHEAD ANALYSIS FOR DIFFERENT ML CLASSIFIERS

Classifier	Area (μm^2)	Power (μW)	Timing (ns)
MLP	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

TABLE III
AREA AND POWER OVERHEAD FOR MAC-LFSR, SINGLE-SCAN FLIP-FLOP

Metric	MAC LFSR	Decoder	Scan FF	ML Detector	BEU	Code-LFSR	Sec-ENC/DEC	Address Checker	Wireless Tx-Rx
Area (μm^2)	41.08	37.96	15.08	105.28	4357.5	326.4	26	219.49	200000
Power (mW)	0.0594	0.0147	0.0247	0.027	0.0047	0.48	0.0361	0.859	36
Delay (ns)	0.16	0.09	0.07	0.56	0.80	0.16	0.07	0.24	0.0625

TABLE IV
SYSTEM PERFORMANCE UNDER ATTACKS AND COMPARISON

	Wired mesh	WiNoC with 4 WIs	WiNoC external DoS	WiNoC internal DoS	WiNoC external ED	WiNoC internal ED
Bandwidth/core (Gbps)	26.4	30.4	26.4	29.5	26.5	29.5
Latency (Cycles)	395.96	286.80	396.00	319.00	299	319.00
Packet Energy (nJ)	100	61	101	78	63.06	78

mode and the normal mode. For all the employed classifiers, the inputs (flits received, flits at error, and flit error ratio) and output classes (normal, random error, and DoS error) are same.

In addition to performance benefits, ML classifiers also incur silicon and resource overheads. Table II presents the incurred overhead in terms of area, power, and delay of the deployed ML classifiers. The characteristics of the various classifiers that are obtained from post-synthesis register transfer logic models are synthesized using 65-nm standard cell libraries, as mentioned earlier. As the KNN classifier has the highest accuracy and lowest area and power consumption, we adopt the KNN classifier for the evaluation of the overall system. Although the delay of the KNN classifier is not optimal, 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 Fig. 3. One can question the impact of false negatives, i.e., DoS is not detected despite its presence. This scenario can lead to a DoS attack. However, for any classification technique, false negatives/positives are inevitable. However, the deployed classifier has shown robustness against such scenarios (0.13% for employed KNN, smaller than others), which indicates a high detection capability and low probability of misclassification.

C. Secure WiNoC Performance in Presence of DoS Attacks

Here, we evaluate the performance of the proposed WiNoC in the presence of DoS attacks from internal and external attackers. We consider a WiNoC with 64 cores in a die of 20 mm $\times$ 20 mm interconnected with a wired mesh and overlaid with four WIs at the central node of each subnet of 16 cores. The WiNoC with embedded security is also compared with an equivalent 64-core wired mesh in terms of performance. The characteristics of the individual blocks in the WIs of the secure WiNoC are shown in Table III and used in the simulation platform for the system-level evaluations. From Table IV, it is clear that the WiNoC outperforms the wired mesh in terms of peak bandwidth, latency, and packet energy due to the low-power wireless shortcuts between distant cores, which reduce the average path length and also uses a low-power wireless medium for communication. This performance can change depending on the number of subnets and WIs deployed on the WiNoC [22]; however, that study or optimization is not within the scope of this article. The security measures developed in this article will be effective irrespective of the number of WIs for the assumed attack model.

Next, it can be seen that in the presence of an external DoS attack, the performance of the WiNoC is similar to that of the wired mesh. This is expected, as on detecting an external attacker, the WSU deactivates all the WIs leaving wired links as the only medium of communication for the purpose of security. On the other hand, when the attacker is an internal agent, only the infected WI is disabled, retaining the advantage due to the presence of the rest of the WIs. Thus, in the case of an internal DoS attack, $<3\%$ degradation in communication bandwidth compared with WiNoC without any attack is achieved. That is why the ADU is an important design element to distinguish an internal attacker from an external attacker.

D. Security Against Eavesdropping

In this section, we evaluate the performance of the WiNoC in the presence of ED attacks. For external ED, we adopt the XOR-based data scrambling approach. The overheads of the additional code-LFSR and Sec-ENC/DEC are shown in Table III. Owing to the parallel XOR gates scrambling all the bits of the body flits in parallel, the delay of the encoder or decoder is very low, minimally affecting the packet latency. The delay of the code-LFSR is not in the path of the data; therefore, it does not affect the packet latency. Because of the adopted lightweight scrambling approach, the impact of a threat of external ED is negligible on the performance of the secure WiNoC.

In the case of internal ED where the rule checker is able to detect the attack, it will disable the infected WI, and therefore, that WI will neither be able to send nor receive packets over the wireless interconnections. Moreover, the checker will add additional overhead albeit really marginal, as shown in Table III. Therefore, the performance will degrade compared with the system with no attack, as shown in Table IV. Owing to the disabling of the infected WI, the overall performance

of this system is similar to the case of a detected internal DoS attack as, in that case, the system disables the attacking WI as well.

E. System Overhead Analysis

As noted in Fig. 3, each WI is equipped with the BEU, an ML classifier, an ADU, DUs, a code-LFSR, a sec-ENC/DEC, and the ED checker. The largest blocks as shown in Table III are the BEU, the KNN classifier, and the code-LFSR. The adopted KNN classifier occupies an area of 105.3 μm^2. The BEU [26] occupies an area of 4357.5 μm^2. The code-LFSR occupies 326.4 μm^2 in each WI. The area of the ADU, DU, and Sec-ENC/DEC blocks is negligible. Therefore, the total area overhead for each WI is 0.005 mm^2. The area of each wireless transceiver is 0.2 mm^2, making this overhead 2.5% per wireless transceiver in the system. The area of the single MAC-LFSR and its decoder is 41 and 38 μm^2. As can be seen, the area overhead incurred by embedding proposed secure mechanism is small when compared with the die size of 400 mm^2 considered in this work.

VII. Conclusion

In this article, we present a detection and defense mechanism for WiNoCs against jamming-based DoS attacks and ED originating from either an internal HT or an external attacker. We use (BEU) codes to estimate the number of burst errors in received packets over the wireless interconnects. This is then used in an ML classifier to distinguish DoS attacks from random transient burst errors. In the event of attack detection by the ML classifier, a logic block will analyze the attacker as either internal or external and enact a defense mechanism accordingly. Using this proposed mechanism, data transfer over the WiNoC is sustained even in the presence of DoS jamming. In the presence of an external DoS attacker, the wireless interconnects are disabled and data are routed using the wired NoC only. However, in the presence of an internal DoS attack, the performance is still better than a wired NoC due to the quarantine procedure of the infected WI enabled by our proposed method. Similarly, our scheme is able to detect and isolate an internal eavesdropper sustaining a better bandwidth and energy efficiency than the wired NoC. High performance and low packet energy is sustained in the WiNoC even in the presence of an external eavesdropper due to the use of the lightweight data scrambling method using parallel XOR gate-based encoding and decoding.

References

[1] U. Y. Ogras and R. Marculescu, "'It's a small world after all': NoC performance optimization via long-range link insertion," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 7, pp. 693–706, Jul. 2006.

[2] L. P. Carloni, P. Pande, and Y. Xie, "Networks-on-chip in emerging interconnect paradigms: Advantages and challenges," in *Proc. ACM/IEEE Int. Symp. Netw.-on-Chip*, May 2009, pp. 93–102.

[3] M. P. D. Sai, H. Yu, Y. Shang, C. S. Tan, and S. K. Lim, "Reliable 3-D clock-tree synthesis considering nonlinear capacitive TSV model with electrical–thermal–mechanical coupling," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 11, pp. 1734–1747, Nov. 2013.

[4] A. Ganguly *et al.*, "Intra-chip wireless interconnect: The road ahead," in *Proc. Int. Workshop Netw. Chip Architectures*, Oct. 2017, p. 3.

[5] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks," in *Proc. Wireless Netw. Secur.*, 2007, pp. 103–135.

[6] J. Sepúlveda, D. Flórez, M. Soeken, J. Diguet, and G. Gogniat, "Dynamic NoC buffer allocation for MPSoC timing side channel attack Protection," in *Proc. IEEE 7th Latin Amer. Symp. Circuits Syst.*, Feb./Mar. 2016, pp. 91–94.

[7] S. Manoj P. D. *et al.*, "A scalable network-on-chip microprocessor with 2.5D integrated memory and accelerator," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 6, pp. 1432–1443, Jun. 2017.

[8] P. Cotret, J. Crenne, G. Gogniat, J. Diguet, L. Gaspar, and G. Duc, "Distributed security for communications and memories in a multiprocessor architecture," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops Phd Forum*, May 2011, pp. 326–329.

[9] C. H. Gebotys and R. J. Gebotys, "A framework for security on NoC technologies," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Feb. 2003, pp. 113–117.

[10] A. Ganguly, M. Y. Ahmed, and A. Vidapalapati, "A denial-of-service resilient wireless NoC architecture," in *Proc. Great Lakes Symp. VLSI*, May 2012, pp. 259–262.

[11] F. Pereñíguez and J. L. Abellán, "Secure communications in wireless network-on-chips," in *Proc. 2nd Int. Workshop Adv. Interconnect Solutions Technol. Emerg. Comput. Syst.*, Jan. 2017, pp. 27–32.

[12] B. Lebiednik, S. Abadal, H. Kwon, and T. Krishna, "Architecting a secure wireless network-on-chip," in *Proc. IEEE/ACM Int. Symp. Netw.-on-Chip (NOCS)*, Oct. 2018, pp. 1–8.

[13] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in *Proc. Annu. Comput. Secur. Appl. Conf.*, Dec. 2007, pp. 421–430.

[14] M. A. Faizal, M. M. Zaki, S. Shahrin, Y. Robiah, S. S. Rahayu, and B. Nazrulazhar, "Threshold verification technique for network intrusion detection system," *Int. J. Comput. Sci. Inf. Secur.*, vol. 2, no. 1, pp. 1–8, Jun. 2009.

[15] W. Zhao, Y. Ha, and M. Alioto, " AES architectures for minimum-energy operation and silicon demonstration in 65nm with lowest energy per encryption," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 2349–2352.

[16] E. Kakoulli, V. Soteriou, and T. Theocharides, "Intelligent hotspot prediction for network-on-chip-based multicore systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 3, pp. 418–431, Mar. 2012.

[17] M. Zekri, S. E. Kafhali, N. Aboutabit, and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," in *Proc. Int. Conf. Cloud Comput. Technol. Appl. (CloudTech)*, Oct. 2017, pp. 1–7.

[18] R. Karimazad and A. Faraahi, "An anomaly-based method for DDoS attacks detection using RBF neural networks," in *Proc. Int. Conf. Netw. Electron. Eng.*, Sep. 2011, pp. 44–48.

[19] P. A. R. Kumar and S. Selvakumar, "Distributed denial of service attack detection using an ensemble of neural classifier," *Comput. Commun.*, vol. 34, no. 11, pp. 1328–1341, Jul. 2011.

[20] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 29–35. doi: 10.1109/SPW.2018.00013.

[21] M. Suresh and R. Anitha, "Evaluating machine learning algorithms for detecting DDoS attacks," in *Proc. Int. Conf. Netw. Secur. Appl.*, 2011, pp. 441–452.

[22] S. Deb *et al.*, "Design of an Energy-Efficient CMOS-Compatible NoC Architecture with Millimeter-Wave Wireless Interconnects," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2382–2396, Dec. 2013.

[23] 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 Trans. Microw. Theory Techn.*, vol. 62, no. 10, pp. 2357–2369, Oct. 2014.

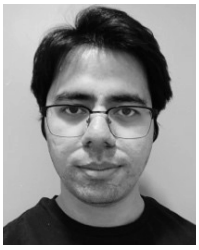
[24] X. Yu, H. Rashtian, S. Mirabbasi, P. P. Pande, and D. Heo, "An 18.7-Gb/s 60-GHz OOK demodulator in 65-nm CMOS for wireless network-on-chip," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 5, pp. 799–806, Mar. 2015.

[25] E. M. Rudd, A. Rozsa, M. Günther, and T. E. Boult, "A survey of stealth malware attacks, mitigation measures, and steps toward autonomous open world solutions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1145–1172, 2nd Quart., 2017.

[26] B. Fu and P. Ampadu, "Burst error detection hybrid ARQ with crosstalk-delay reduction for reliable on-chip interconnects," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2009, pp. 440–448.

[27] R. JayashankaraShridevi, C. Rajamanikkam, K. Chakraborty, and S. Roy, "Catching the flu: Emerging threats from a third party power management unit," in *Proc. ACM/EDAC/IEEE Design Autom. Conf.*, Jun. 2016, Art. no. 86.

[28] 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 Trans. Comput.*, vol. 54, no. 8, pp. 1025–1040, Aug. 2005.



Abhishek Vashist received the B.Tech. degree from the ABES Engineering College, Ghaziabad, India, in 2014, and the M.S degree in electrical engineering from the Rochester Institute of Technology, Rochester, NY, USA in 2017, where he is currently working toward the Ph.D. degree at the Department of Computer Engineering.

His current research interest includes design and test of on-chip interconnection networks for multi-core processors and machine learning-based design of localization systems for autonomous vehicles using 60-GHz wireless sensors.

Andrew Keats is currently working toward the B.S. degree in computer engineering at the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA.

His current research interest includes data security for on-chip wireless interconnection architectures.

Sai Manoj Pudukotai Dinakarrao (S'13–M'15) received the master's degree in information technology from the International Institute of Information Technology Bangalore (IIITB), Bengaluru, India, in 2012, and the Ph.D. degree in electrical and electronics engineering from Nanyang Technological University, Singapore, in 2015.

He was an Assistant Professor at George Mason University (GMU), Fairfax, VA, USA, where he served as a Research Assistant Professor and a Postdoctoral Research Fellow. He was a Postdoctoral Research Scientist at the System-on-Chip Group, Institute of Computer Technology, Vienna University of Technology (TU Wien), Vienna, Austria. He is currently an Assistant Professor at GMU. His current research interests include on-chip hardware security, neuromorphic computing, adversarial machine learning, self-aware SoC design, image processing and time-series analysis, emerging memory devices and heterogeneous integration techniques.

Dr. Pudukotai Dinakarrao is nominated for the Best Paper Award in Design Automation and Test in Europe (DATE) 2018. He won Xilinx Open Hardware Contest in 2017 (student category). He was a recipient of the A. Richard Newton Young Research Fellow Award in Design Automation Conference in 2013.

Amlan Ganguly (M'11) received the B.Tech. degree from IIT Kharagpur, Kharagpur, India, and the M.S. and Ph.D. degrees from Washington State University, Pullman, WA, USA, in 2005, 2008, and 2010, respectively.

He is currently an Associate Professor at the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA. His current research interests include robust and scalable intrachip and interchip interconnection architectures and novel datacenter networks with emerging technologies such as wireless interconnects. His research is funded by the U.S. National Science Foundation and Toyota Material Handling North America.

Dr. Ganguly is a member of the Technical Program Committee of several conferences such as International Green and Sustainable Computing (IGSC) and International Network-on-Chip Symposium (NOCS). He was a recipient of the U.S. NSF Faculty Early CAREER Development Award in 2015. He is an Associate Editor for the *Elsevier Journal of Sustainable Computing Systems (SUSCOM)* and the *MDPI Journal of Low Power Electronics and Applications (JLPEA)*.