A Security Aware Routing Approach for NoC-based MPSoCs

Ramon Fernandes[1], César Marcon[2], Rodrigo Cataldo[3], Jarbas Silveira[4], Georg Sigl[5], Johanna Sepúlveda[6]
[1,2,3]Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
[5,6]Institute for Security in Information Technology, Technical University of Munich (TUM)
[5]Fraunhofer Research Institution for Applied and Integrated Security (AISEC)
[4]LESC-DETI, Federal University of Ceará (UFC)
Email: {ramon.fernandes[1],rodrigo.cataldo[3]}@acad.pucrs.br, cesar.marcon@pucrs.br[2],
jarbas@lesc.ufc.br[4], sigl@tum.de[5], johanna.sepulveda@tum.de[6]

***Abstract*—Malicious applications target Multi-Processors System-on-Chip (MPSoCs) to capture sensitive information or disrupt normal operation; therefore, security is now a design requirement for MPSoC design. Network-on-Chip (NoC) is a key communication structure to aid in the overall MPSoC protection. Firewall-based NoC protection allows data exchange monitoring and controlling according to the MPSoC security policy. Secure NoCs enable to detect and prevent a broad range of software-based attacks. However, complex security policies may turn firewalls costly. This paper proposes a protection technique based on the NoC routing algorithm. By manipulating the routing of packets, security zones can be built. Our routing algorithm prioritizes communication among paths deemed secure while guaranteeing deadlock freedom. We evaluate the scalability of the proposed technique using synthetic and real application scenarios, as well as the security of the proposed technique.**

***Keywords*—MPSoCs; NoCs; routing algorithm; security;**

I. Introduction

The next generation of Multi-Processors System-on-Chip (MPSoCs) will integrate hundreds of Intellectual Property (IP) modules, such as processors, memories and other application specific components into a single chip. MPSoCs promise to achieve high performance and flexibility [1]. The high communication parallelism of several applications targeting MPSoC architectures has turned the Network-on-Chip (NoC) as the more suitable interconnection structure [2][3], providing a reliable and scalable communication infrastructure [4].

Data is exchanged in the NoC as packets, which are transmitted by a set of routers and links. The router is the underlying communication fabric of the system, switching packets from its inputs ports to the output ports. The links perform the interconnection between routers, and the routing algorithm defines the selection of the output port of the router. The final configuration of the NoC should satisfy the performance and cost requirements of the system while preventing undesirable behaviors, such as deadlocks. IPs are linked to the routers through the Network Interface (NI), responsible for packing and unpacking data sent or received through the NoC. Each IP module has a unique address in the system used by other IPs for message exchanging.

Since MPSoCs have become a trend in the industry as a major design solution for the increasing demands of mobile platforms, such as the Internet of Things (IoT) [5], the concern for security has also gained increasing relevance in MPSoC design. Such devices may perform critical tasks, as well as store and process sensitive and private information [6].

Attacks at MPSoC aim to extract sensitive data, modify the system behavior or denial the system operation (*Denial-of-Service*, *DoS*) [7]. Software attacks account for 80% of security incidents in embedded systems, often preceded by abnormal communication events [8]. Software attacking techniques are getting more complex and effective. The resource sharing in MPSoC is widely exploited; thus, the computation and communication of sensitive data must be isolated [7]. However, to enhance the performance of the MPSoC, the designer usually spreads the applications on the computation resources, forcing the sensitive traffic through the NoC. Aiming to protect the MPSoC, security services can be implemented in the computation structure by using encryption techniques to avoid plain data transmission among IP modules [9]. Security may also be integrated into the communication structure, employing firewalls to monitor the traffic, detect abnormal communication behavior in the system and isolate sensitive traffic [10].

Security zones at NoCs are used to wrap IPs and protect the sensitive flows from attackers. Isolation of traffic avoids that attackers capture data-dependent traffic characteristics of critical tasks, thus revealing sensitive data [7]. Firewalls are used in [8] to create security zones. However, complex security policies may turn firewalls costly [11]. This work proposes for the first time the utilization of NoC routing algorithm for implementing security zones. Considering the security configuration of applications mapped in the MPSoC, the routing algorithm is able of establishing safe communication paths. We show that the routing algorithm can be an efficient and scalable alternative for promoting the data protection in the NoC. The novelties of this work are:

- Implementation of a routing algorithm technique, based on *Region-based Routing* algorithm (*RBR*) method [12], which is aware of the security requirements of the system, defined by security zones;
- The exploration of the proposed routing technique regarding scalability and secure communication paths, consid-

978-1-5090-2736-1/16/$31.00 ©2016 IEEE

ering the *Segment-based Routing* (*SBR*) deadlock prevention method [13].

This papers organization is as follows: Section II presents related work on NoC-based MPSoC security. Section III discusses the considered threat model. Section IV presents the proposed secure routing mechanisms and algorithms. Section V shows the evaluation criteria. Section VI analyzes the results. Finally, Section VII concludes our work.

II. Related Work

Previous works have shown that the integration of security services at NoCs may aid in the overall MPSoC protection. Firewall-based protection is the most common strategy to implement NoC-based MPSoC security. Such hardware structures are integrated into the NI and NoC components to filter data according to the security policy of the system. According to the updating capabilities of the security tables that store the policy, firewalls can be classified as static or dynamic. Static firewalls are used in [6], while reconfigurable firewalls, able to update the security policy, are used in [8][14][15].

In [6][14][15][16], firewalls are integrated at the initiator and destination NIs. Those firewalls check different information embedded into the packet. Source and privileges are checked in [6][14][16]. Address ranges and read/write operation requested by IPs are checked in [15][16]. Firewalls can be integrated into the NoC, as for instance, the work of [8] that proposes 3D-ACeNoC, integrating firewalls at the NIs and between consecutive routers. By restricting the NoC traffic, the firewalls wrap IPs inside a security zone. Components inside the same security zone are trusted and considered secure among them. The concept of security zones is adopted in this paper.

Communication scheduling for security is explored in the three-dimensional NoC of [16]. The vertical interconnection based on *Through-Silicon-Vias* (*TSVs*) greatly suffer from coupling effects, turning this technology vulnerable to integrity and operational problems. Malicious traffic flows could promote electro-migration effects that may alter packets in adjacent TSVs. Thus, the packets are scheduled in the different TSVs to avoid sensitive data interference.

Despite the good results regarding protection of the sensitive content, none of the approaches mentioned above attempt to deal with system protection at NoC routing level, which is the approach of this work that applies routing strategies to implement the security zones in the MPSoC.

III. Threat Model

The MPSoCs considered in this work are composed of a set of IPs interconnected by a shared NoC. Parallel applications are split into smaller pieces of code, called tasks, and mapped into different and several IPs to enhance system performance. For sensitive applications, such strategy forces the communication of sensitive data through the NoC. The sensitive communication between a pair of IPs, that execute the critical application, is called sensitive path. Fig. 1 shows an MPSoC

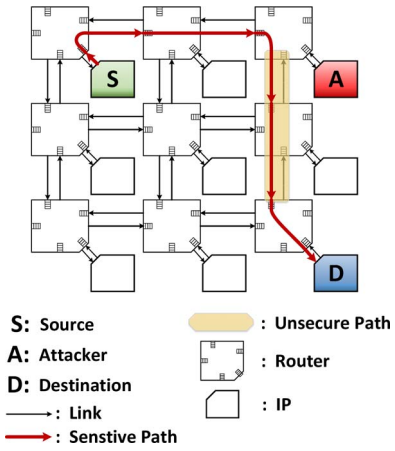

Fig. 1. A source process S communicates with a destination D over an insecure path due to a DoS or Timing Attack from an attacker A to D.

that includes 9 IPs, highlighting a sensitive communication between the IPs *S* (source) and *D* (destination).

The attacker performs a software attack infecting an IP through a malicious code. We assume that the NoC is secure. The malicious program is loaded into the MPSoC and executed by an IP. It can be done by downloading malwares directly from the Internet to the chip or by modifying the external memory used by the MPSoC to store applications. Depending on the attack type (extraction of sensitive data), it may be desirable to retrieve data from the MPSoC, which requires that the attacker infects an IP with the right to use the peripherals. An infected IP *A* (attacker) is shown in Fig. 1.

To increase the efficiency of the attack it is desirable that the infected IP be located inside the sensitive path. The attacker must know the MPSoC mapping strategy and the NoC routing algorithm. This latter requirement is not mandatory for some attacks of *Denial-of-Service*, whose goal is to disrupt the system operation by overloading the resources.

The attacks considered in this work are the timing and *DoS* attacks, previously described by [7][17][18][10]. Timing attacks use the communication collision between the sensitive traffic and the attacker A request in order to reveal a secret. Data dependences of critical applications are reflected in the traffic pattern [18]. DoS attacks are performed by flooding the NoC resources that are used by the sensitive path with useless communications.

Both attacks exploit the collisions and interferences of the sensitive traffic. Security zones arise as an alternative for isolating sensitive traffic inside protected environments. IPs that belong to the security zone are considered secure. Selecting a path that avoids the router linked to the malicious *A* core and the possible malicious paths aids to mitigate the attack. In this paper, we propose a deadlock-free routing algorithm that maximizes the encapsulation of the sensitive path inside a security zone. Such strategy aids to decrease the attacks.

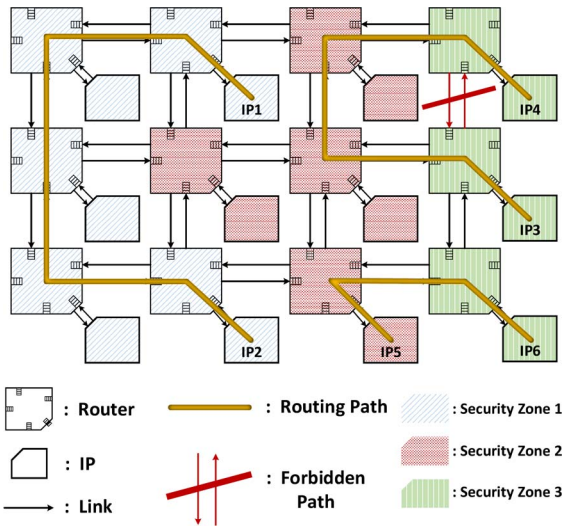

Fig. 2. The three communication scenarios with security zones: ***FIZ*** (IP1 to IP2); ***PIZ*** (IP3 to IP4); and ***IZ*** (IP5 to IP6).

IV. Security Aware Routing

This work proposes *SBR Security Zone Awareness* (*SBR-SZA*), an alternative segment computation heuristic that is aware of the security characteristics of the system. The secure routing approach is based on two concepts: (i) *Security Zones*; and (ii) *Routing Algorithm*. These concepts and their utilization are described in the following two Subsections.

A. Security Zones

A security zone SZ is a physical space (continuous or disrupted) that wraps the IPs that execute critical applications. IPs that belong to the security zone are considered trusted among them [11]. The task mapping of critical applications inside the MPSoC defines the shape of the security zone.

A set of IP cores (IP) that executes a critical application defines a security zone SZ, such that the elements $p_i, p_j \in IP$ are considered secure and trusted. A transaction from p_i to p_j, where $i, j \in [0, N-1)$ with N representing the total amount of IP blocks in the system, is called sensitive and must be performed inside SZ. However, typical NoC routing algorithms may force the route of the sensitive path outside the SZ. Three communication scenarios are shown in Fig. 2:

- ***Full intra-zone communication (FIZ)***: *S* and *D* are in the same SZ. The sensitive path is **completely** inside the SZ, e.g., the path from *IP1* to *IP2*;
- ***Partial intra-zone communication (PIZ)***: *S* and *D* are in the same SZ. However, the sensitive path is **partially** inside the SZ. ***PIZ*** takes place in disrupted security zones and for irregular SZ shapes, when typical routing algorithms force out the sensitive traffic, e.g., the path from *IP3* to *IP4*;
- ***Inter-zone communication (IZ)***: *S* and *D* are in different SZ, e.g., the path from *IP5* to *IP6*.

FIZ communication is the most secure situation, as sensitive communications are contained in secure elements of the system. Whenever possible, traffic flows should adhere to this model. ***PIZ*** occurs when the security zone is fragmented, or the routing algorithm forbids a communication path, usually to avoid a route that would lead to a routing deadlock. Lastly, ***IZ*** should occur by communications among distinct applications in the system.

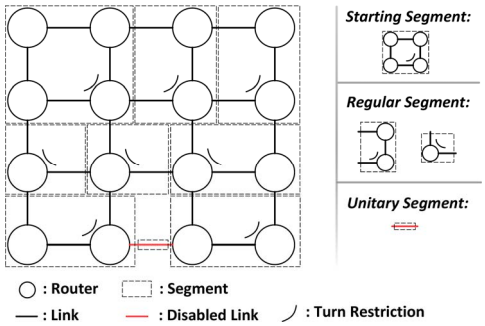

Fig. 3. Segments and turn restrictions computed by the *SBR* algorithm in a 4x4 2D Mesh NoC (based on [13]).

B. Routing Algorithm

Searching for secure paths on MPSoCs demands greater flexibility [11]. As shown in Fig. 2, establishing secure paths inside the SZ requires in some cases non-minimal paths, e.g., the path from *IP1* to *IP2*. It heavily depends on the shape of the security zone.

Several routing algorithms have been studied before in the areas of high performance and fault-tolerant MPSoCs. *Segment-based Routing* (*SBR*) [13] and *Region-based Routing* (*RBR*) [12] have been used in conjunction to efficiently find non-minimal paths. *SBR* is responsible for deadlock prevention while *RBR* computes the routing entries.

SBR is composed of two phases: (i) segment computation; and (ii) placement of routing restrictions. At segment computation, *SBR* partitions the NoC into segments comprising routers and links. As shown in Fig. 3, each segment is characterized by a turn restriction that avoids routing deadlocks. *SBR* classifies segments into three types: (i) *starting*, which starts and ends at the same router forming a loop; (ii) *regular*, which starts at a link, contains at least one router, and ends on another link; and (iii) *unitary*, which contains a single link that does not allow traffic. *SBR* aims to create segments that minimize the number of elements per segment in order to reduce the occurrence of *unitary* segments.

When placing routing restrictions, *SBR* defines that each segment can contain a localized turn restriction that best suits the routing of elements in that segment. Globally, *SBR* guarantees deadlock freedom and connectivity among all elements in the network; i.e., the turn restrictions still allow communication among all source-destination pairs.

RBR takes the turn restrictions, computed by *SBR*, to find paths between all origins and destinations in the NoC. As a result, the routing entries for each router are generated. The main advantage of *RBR* is that a single routing entry

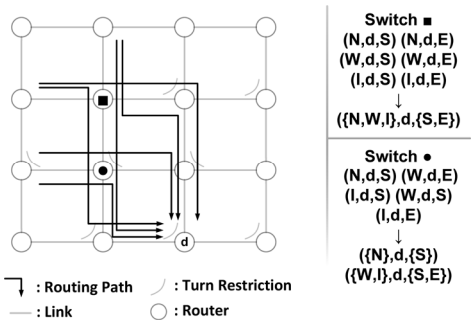

Fig. 4. Paths computed with *RBR* algorithm to a destination d from two different source switches (based on [12]).

can represent a path to more than one destination, which can reduce the size of routing tables significantly.

RBR computation occurs in three steps. The first step is the *routing computation* from each NoC router to every other router. The path-selection is performed according to the designer goals and system requirements. The found paths are stored in the routing table of the router. They are represented as a 3-tuple (N, d, S), where N is the packet input port, d the destination address, and S is the output port.

The second step is the *region computation*, where multiple entries are joined based on the input and output port values. Fig. 4 shows the paths computed by *RBR* algorithm to a destination ***d*** from two different source routers. The entries (N, d, S), (W, d, S) and (I, d, S) of router ■, that have the same set of output ports for the same destination, can be grouped. As a result, a single routing entry $(\{N, W, I\}, d, S)$ is stored. Analogously, the 3-tuples with the E output port can be packet into $(\{N, W, I\}, d, E)$. Further packing can be done within the grouping results, leading to a single entry $(\{N, W, I\}, d, \{S, E\})$. This result represents an adaptive routing, as more than one output port exists to reach the same destination.

The third step, *region merge*, merges overlapping regions in order to reduce the amount of routing entries.

RBR defines regions in the NoC by grouping entries based on their destination. A single entry can represent routing entries that have the same set of input and output ports to distinct destinations.

Regarding *the computational cost* of *SBR*, the segment computation has a cost of $O(m)$, while placing routing restrictions has a cost of $O(s)$, where m is the number of links in the NoC and s the number of segments. Moreover, for *RBR*, the cost of checking all sources to every possible destination is $O(n^2)$, where n represents the amount of routers in the NoC. It becomes important to use an efficient pathfinding algorithm to alleviate these costs, such as *Dijkstra*'s shortest path or *A**.

C. Security Zone Awareness (SBR-SZA)

SBR capabilities can be used to compute the segments and turn restrictions based on the security zones of the MPSoC. Fig. 5 shows two cases of segment computation for the same six routers of the MPSoC. Depending on the NoC segments

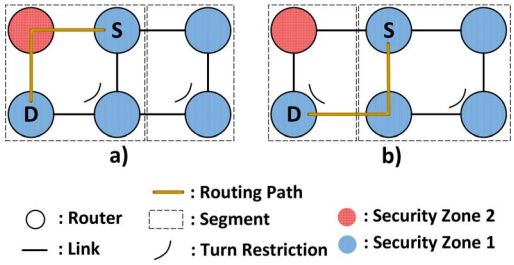

Fig. 5. Two *SBR* segment computations: a) the path among S and D goes through an insecure element due to a routing restriction; b) a set of restrictions in the segments enables a secure path between S and D.

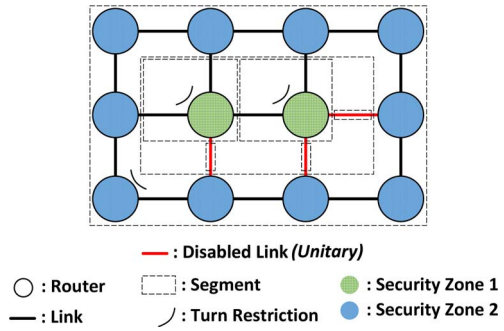

Fig. 6. Segmentation example with *SBR-SZA*. While ***PIZ*** scenarios do not occur, the segmentation results in *unitary* segments.

computed by *SBR*, the communication path between *S* and *D* can be either ***PIZ***, as shown in Fig. 5a; or ***FIZ***, as shown in Fig. 5b.

SBR-SZA uses the *SBR* algorithm in order to compute NoC routes that favor the occurrence of ***FIZ*** scenarios. The segments are tailored to a security zone so that *SBR-SZA* creates the smallest possible segments that contain elements from the same security zone.

While *SBR-SZA* should favor the occurrence of ***FIZ*** scenarios, a performance impact is a possibility since the segmentation can lead to a greater occurrence of *unitary* segments, as Fig. 6 illustrates.

D. Modeling of Security Zones

We model the MPSoC as a graph. Vertexes correspond to routers and their associated IPs, while edges to links between routers. Each vertex belongs to a security zone, and each edge has a positive weight that is relative to the pathfinding iteration.

When computing the path from a source router to a destination, the weight of edges adjusts to favor paths to other routers from the same security zone as the source. In Fig. 7, there are three paths from *S* to *D*. The topmost path traverses an insecure element in the context of *S*, so it has a higher edge cost than edges that lead to a router of the same security zone of *S*. The middle path traverses two secure routers, reaching *D*. Meanwhile the bottommost path has a routing restriction that forbids traffic from *S* to *D*.

Determining the edge cost is about the system requirements. It could represent the cost to protect a package while traversing

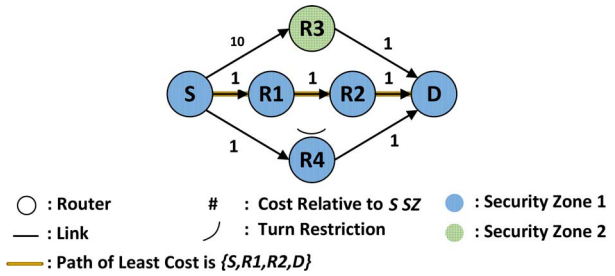

Fig. 7. Modeling of the paths from *S* to *D*. Traversing a different security zone leads to a higher path cost.

to an insecure zone by a cryptographic module, or the cost of passing a firewall. The spectrum of possibilities to determine the edge cost is too wide and not a focus of study in this paper. Meanwhile, it suffices to say that our model defines the cost of traversing an insecure element in such a way that it is always preferable to route inside a secure zone.

V. Evaluation

In this work, we are mainly concerned with two aspects of our proposed secure routing approach: (i) the scalability of the *RBR* routing tables; and (ii) to minimize ***PIZ*** routing scenarios. For these purposes, we compute some *SBR* segmentation configurations by varying the initial router (seed) used for segment computation. This approach results in different turn restriction placement (shown in Fig. 5) along the NoC and therefore different routing tables for each configuration and ***FIZ/PIZ*** distributions.

We consider two synthetic scenarios and another based on a real application benchmark mapped to an MPSoC. The first synthetic scenario (*Synth1*) consists of a 6x4 NoC with four security zones; the second synthetic scenario (*Synth2*) has a 10x6 NoC with ten security zones. Both scenarios only contain continuous security zones, and aim to reproduce the situations depicted in Fig. 2, Fig. 5 and Fig. 6.

The real application consists of the *NASA Numerical Aerodynamic Simulation* (*NAS*) Benchmark [19]; a 13x13 NoC contains the mapping of five applications used by the *NAS* benchmark using a *Simulated Annealing* mapping algorithm. Each application contains 32 tasks, and we consider that each task fully occupies an IP core in the system. Tasks from an application characterize the security zones, totaling five zones for the *NAS* scenario.

Additionally for the *NAS* scenario, we also estimate the latency impact of using security zones with *SBR* and *SBR-SZA*. Assuming that packets which are traversing an insecure zone require encryption, we evaluate for the six block ciphers techniques in [20] as the protection mechanism.

VI. Results

Due to the modeling of security zones with non-minimal distance routing paths, it is expected that additional *RBR* routing entries are necessary to accommodate routes within security zones. Fig. 8 illustrates the average distribution of

TABLE I
Summary of Obtained Results

		Scenario		
	Algorithm	**NAS**	**Synth1**	**Synth2**
Increase in Routing Entries per Router	**SBR**	16.50%	9.42%	15.57%
	SBR-SZA	16.56%	4.52%	14.30%
Coefficient of Variation in Routing Entries per SBR Seed	**SBR**	1.01%	2.47%	2.42%
	SBR-SZA	1.02%	5.37%	2.83%
Minimum PIZ Scenarios	**SBR**	230	0	0
	SBR-SZA	198	0	0
Coefficient of Variation in PIZ Scenarios per SBR Seed	**SBR**	18.21%	117.95%	89.95%
	SBR-SZA	18.63%	115.14%	76.22%

routing entries per router in each scenario. Considering a *Base-NoC* with no security zones, there is an average increase of 16.50% and 16.56% of routing entries per router for the *NAS* scenario with *SBR* and *SBR-SZA*, respectively. Scenarios *Synth1* and *Synth2* also incur in an overhead for the required routing entries to accommodate security zones, as Table I shows.

Regarding the ***PIZ*** and ***FIZ*** communication scenarios, Fig. 9 illustrates the distribution of ***PIZ*** communications for each scenario per *SBR* seed. ***FIZ*** and ***PIZ*** scenarios are inversely proportional. A configuration with zero ***PIZ*** scenarios means all communication occurs as ***FIZ***.

As Fig. 9 and Table I show, there is no ideal secure configuration for the *NAS* scenario for any seed used in *SBR*; however, ***SBR-SZA*** yields more secure communication paths than *SBR*. Both *Synth1* and *Synth2* can be configured in a way that guarantees only ***FIZ*** routing with either *SBR* or *SBR-SZA*.

In both evaluations changing the seed used by *SBR* or *SBR-SZA* yields different results, as expected. While the routing tables present a small coefficient of variation when changing the seed, the occurrence of ***PIZ*** and ***FIZ*** scenarios can vary greatly due to the segmentation process. The proposed *SBR-SZA* model can also result in smaller routing tables and increase ***FIZ*** routing scenarios in some situations, although its benefits depend on the shape of the security zones.

Considering the *NAS* scenario, Fig. 10 illustrates the variation in latency when compared to a NoC without security zones. The choice of the encryption technique can have a significant impact on the communication latency, with **HIGHT** encryption having $< 1.0\%$ performance penalty, while **AES-128** can more than double the average communication latency. As shown in Fig. 10, *SBR-SZA* also incurs in greater communication latency than standard *SBR*, which is expected as *SBR-SZA* can generate more *Unitary* segments, decreasing link availability and creating longer communication paths.

VII. Conclusion

This work proposes a routing technique aware of security requirements in the system. As information security and availability become a new design requirement in MPSoCs, the exploration of different system elements for security purposes, such as the routing algorithm, presents alternative approaches to aid in the overall MPSoC security.

The proposed technique is sufficient to address the identified threat model with a small to moderate overhead in *RBR* routing

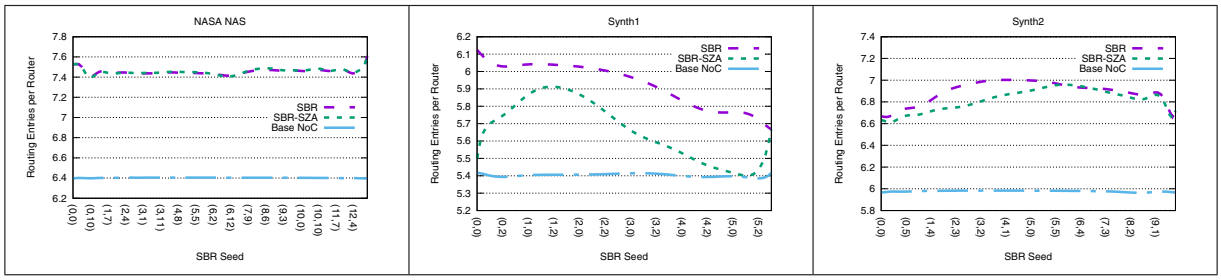

Fig. 8. Average distribution of routing entries per router per *SBR* Seed, using *SBR* and *SBR-SZA*, compared to a Base-NoC without security zones.

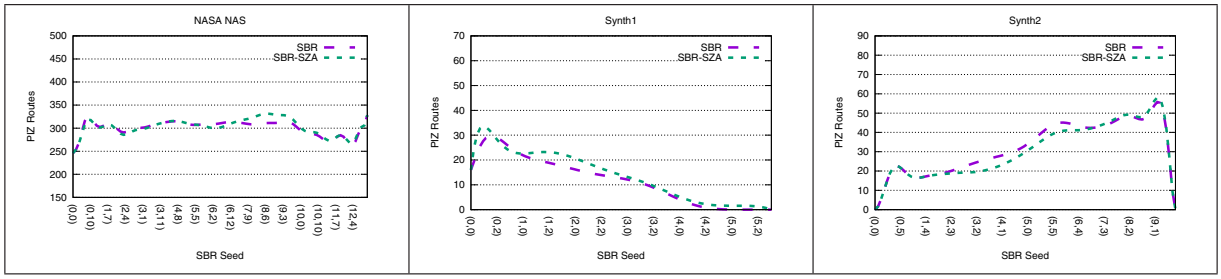

Fig. 9. Average distribution of ***PIZ*** routes per *SBR* Seed, using *SBR* and *SBR-SZA*. When zero ***PIZ*** routes occur, it characterizes a configuration with only ***FIZ*** communication.

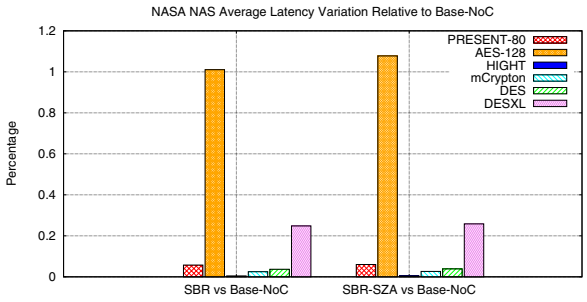

Fig. 10. Average latency variation for *NAS* scenario with different data protection mechanisms.

tables. *SBR-SZA* can also benefit certain security scenarios while providing configurations with routing contained to a security zone, albeit at a slightly larger application performance impact. Further studies regarding different applications can also yield a broader understanding of the proposed model.

REFERENCES

[1] ITRS. *International Technology Roadmap for Semiconductors*. 2015. URL: http://www.itrs.net/reports.html (visited on 11/29/2015).

[2] M. Sgroi et al. "Addressing the System-on-a-chip Interconnect Woes Through Communication-based Design". In: Design Automatic Conference (DAC). 2001, pp. 667–672.

[3] Terry Tao Ye et al. "Packetization and Routing Analysis of On-chip Multiprocessor Networks". In: *J. Syst. Archit.* 50.2-3 (Feb. 2004), pp. 81–104.

[4] L. Benini et al. "Networks on chips: a new SoC paradigm". In: *Computer* 35.1 (Jan. 2002), pp. 70–78.

[5] P. Greenhalgh. "big.LITTLE Processing with ARM Cortex-A15 and Cortex-A7". In: (2011).

[6] L. Fiorin et al. "Secure Memory Accesses on Networks-on-Chip". In: *IEEE Transactions on Computers* 57.9 (Sept. 2008), pp. 1216–1229.

[7] M. J. Sepúlveda et al. "NoC-Based Protection for SoC Time-Driven Attacks". In: *IEEE Embedded Systems Letters* 7.1 (Mar. 2015), pp. 7–10.

[8] J. Sepulveda et al. "Elastic security zones for NoC-based 3D-MPSoCs". In: *Electronics, Circuits and Systems (ICECS)*. Dec. 2014, pp. 506–509.

[9] D. M. Ancajas et al. "Fort-NoCs: Mitigating the threat of a compromised NoC". In: *Design Automation Conference (DAC)*. June 2014, pp. 1–6.

[10] MJ Sepúlveda et al. "An Hybrid Switching Approach for NoC-based Systems to avoid Denial-of-Service SoC Attacks". In: *16th Iberchip Wksp (IWS 2010)* (2010), pp. 23–25.

[11] J. Sepulveda et al. "Reconfigurable security architecture for disrupted protection zones in NoC-based MPSoCs". In: *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2015 10th International Symposium on*. June 2015, pp. 1–8.

[12] J. Flich et al. "Region-Based Routing: An Efficient Routing Mechanism to Tackle Unreliable Hardware in Network on Chips". In: *Networks-on-Chip (NOCS)*. May 2007, pp. 183–194.

[13] A. Mejia et al. "Segment-based routing: an efficient fault-tolerant routing algorithm for meshes and tori". In: *Parallel and Distributed Processing Symposium (IPDPS)*. Apr. 2006, pp. 105–115.

[14] Ramon Fernandes et al. "A Non-intrusive and Reconfigurable Access Control to Secure NoCs". In: IEEE International Conference on Electronics, Circuits and Systems (ICECS). Nov. 2015, pp. 316–319.

[15] M. D. Grammatikakis et al. "Security Effectiveness and a Hardware Firewall for MPSoCs". In: *High Performance Computing and Communications (HPCC)*. Aug. 2014, pp. 1032–1039.

[16] J. Sepulveda et al. "3D-LeukoNoC: A dynamic NoC protection". In: *ReConFigurable Computing and FPGAs (ReConFig)*. Dec. 2014, pp. 1–6.

[17] Yao Wang et al. "Efficient Timing Channel Protection for On-Chip Networks". In: *IEEE/ACM Sixth International Symposium on Networks-on-Chip*. 2012, pp. 142–151.

[18] J. Sepúlveda et al. "Dynamic NoC Buffer Allocation for MPSoCs Timing Side Channel Attack Protection". In: *Latin American Symposium on Circuits and Systems (LASCAS)* (2016), p. 4.

[19] NASA. *NAS Parallel Benchmarks*. 2016. URL: http://www.nas.nasa.gov/publications/npb.html (visited on 03/31/2016).

[20] A. Bogdanov et al. "PRESENT: An Ultra-Lightweight Block Cipher". In: *Cryptographic Hardware and Embedded Systems (CHES)* (2007), pp. 450–466.