SCRIPT: A Multi-Objective Routing Framework for Securing Chiplet Systems Against Distributed DoS Attacks

Ebadollah Taheri
Colorado State University
Fort Collins, USA
ebad.taheri@colostate.edu

Pooya Aghanoury
University of California, Los Angeles
Los Angeles, USA
aghanoury@g.ucla.edu

Sudeep Pasricha
Colorado State University
Fort Collins, USA
sudeep@colostate.edu

Mahdi Nikdast
Colorado State University
Fort Collins, USA
mahdi.nikdast@colostate.edu

Nader Sehatbakhsh
University of California, Los Angeles
Los Angeles, USA
nsehat@ucla.edu

ABSTRACT

Heterogeneous 2.5D integration enables seamless integration of chiplets, hence reducing design time and costs. Concerns arise when dealing with untrustworthy chiplets, emphasizing the need for dependable Network-on-Interposer (NoI). This paper introduces SCRIPT, a secure routing framework to mitigate Distributed Denial-of-Service (DDoS) attacks in chiplet systems. SCRIPT obscures predictable paths exploited by attackers, disrupting orchestrated attacks. SCRIPT considers chiplet trust and criticality and employs a multi-objective optimization technique to enhance NoI performance and reliability. Evaluations show that SCRIPT enhances NoI security by at least 64% against DDoS attacks.

CCS CONCEPTS

• **Hardware → Network on chip**; • **Security and privacy → Hardware attacks and countermeasures**.

ACM Reference Format:
Ebadollah Taheri, Pooya Aghanoury, Sudeep Pasricha, Mahdi Nikdast, and Nader Sehatbakhsh. 2024. SCRIPT: A Multi-Objective Routing Framework for Securing Chiplet Systems Against Distributed DoS Attacks. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3649476.3658763

1 INTRODUCTION

The chiplet system architecture is becoming a cost-effective solution for building large-scale systems. Using off-the-shelf chiplets for modular integration significantly reduces design time and costs, as these foundational building blocks can be procured from third-party entities. However, as the employment of 2.5D chiplet systems grows further, *unique and new security challenges arise* [3, 13, 15, 18, 19]. For example, when handling chiplets from potentially untrusted sources, the security of Network-on-Interposer (NoI) communication is at risk. The vulnerability arises as malicious chiplets may focus traffic on the NoI or Vertical Links (VL), jeopardizing the interposer's service availability. This situation leads to Denial-of-Service (DoS) attacks.

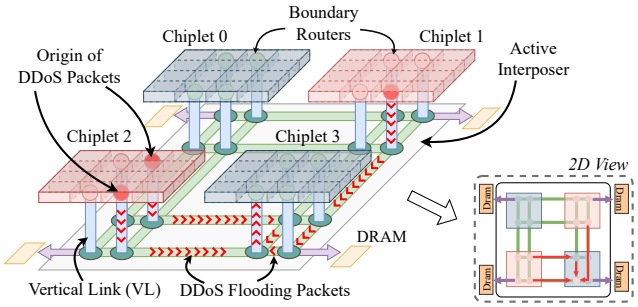

Figure 1: High-level overview of a 2.5D network with 4 chiplets, subdivided into 16 IPs. Each chiplet contains four boundary routers interface with the interposer. Chiplet 1 & 2 are malicious and transmit DDoS flood packets to Chiplet 3.

Detecting and preventing DoS attacks originating from single or multiple sources (IPs) within a chip has been addressed in several prior works [4, 20], but the challenges *differ* when various *chiplets* coordinate to execute a Distributed Denial-of-Service (DDoS) attack [5] at the NoI level. Moreover, the traditional methods of avoiding DoS attacks through routing obfuscation at the IP level may inadvertently degrade chiplet-based network performance.

The intrinsic modularity of chiplet systems renders them more susceptible to DDoS attacks. Notably, the NoI is shared among multiple chiplets, a departure from traditional monolithic chips. The availability of the NoI is pivotal, as the entire system relies on its functionality for inter-chiplet communication. Furthermore, VLs represent a costly and inherently critical network path. As a result, attackers consider them particularly valuable. For instance, an attack scenario is illustrated in Fig. 1 where multiple (two in this example) untrusted chiplets maliciously transmit packets to a destination chiplet, seeking to concentrate the traffic load on a specific VL in an attempt to induce a DDoS condition.

This paper presents a novel routing-based framework, called SCRIPT, developed to mitigate DDoS attacks in 2.5D chiplet systems by intentionally obscuring the predictable paths typically exploited by attackers. In contrast to conventional routing techniques

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3658763

vulnerable to targeted congestion, our proposed solution incorporates elements of obfuscation and randomization, thereby disrupting the predictability underlying orchestrated attacks. Within this new framework, the network dynamically generates randomized paths, confounding potential adversaries and enhancing the network resilience against coordinated traffic designed to compromise NoI service availability. Furthermore, our framework considers the trust levels of chiplets as a measure for quantifying the threat. It introduces a multi-objective optimization to improve the network's reliability against DDoS attacks, while concurrently minimizing performance degradation. The main contributions of this paper are:

- We propose obfuscated routing on the NoI to ensure system security, even in the presence of malicious chiplets.
- We mitigate DDoS attacks on VLs originating from one or multiple malicious chiplets in a distributed manner.
- We present multi-objective optimization for source-destination average-distance and load distribution on VLs to avoid performance penalty due to DDoS attacks while achieving security.

The organization of this paper is structured as follows. We delve deeper into the background and related work on NoC's security in Section 2. Following that, in Section 3, we discuss our threat model. In Section 4, we present our proposed performance- and security-aware interposer. Section 4 also involves an exploration of the design space for performance and security. Our simulation results are showcased in Section 5, and finally, in Section 6, we draw conclusions and highlight the significance of our contributions.

2 BACKGROUND AND RELATED WORK

A significant challenge in chiplet systems lies in designing the NoI to provide reliable and low-latency inter-chiplet communication [22]. While routing algorithms have been proposed to achieve low hardware costs [14, 24] and high flexibility in VL selection [23], they often neglect the security challenges inherent in chiplet systems. Previous work on security has primarily focused on designing routing algorithms for conventional Networks-on-Chip (NoCs) [8, 16] but has not specifically addressed the unique requirements of NoIs in chiplet systems. This lack of specificity renders them inefficient when applied to NoI.

A major threat to NoCs is DoS attacks where the aim is to block access to network services by overwhelming the system. These attacks involve malicious IPs flooding the NoC with packets, impacting the availability of on-chip resources [5]. For instance, memory-intensive applications can lead to heavy traffic on routers connected to memory controllers, and if a malicious IP targets the same node, it results in significant degradation of NoC performance. The flooding can cause congestion in routers, leading to severe delays in responses. Various works, including one with Hardware Trojans embedded in routers, highlighted the performance degradation caused by flooding the network with additional packets [7].

Furthermore, DoS attacks can be carried out through packet corruption [9, 12] and traffic flow manipulation [11]. Regardless of the attack type, these efforts either explore methods of employing such attacks, their implications, or ways of *detecting* attacks. [8] proposed a zone-based routing algorithm to separate secure regions from insecure ones, enhancing defense against DoS and timing attacks. However, the zone-based routing approach, despite its complexity, constrains resource utilization and the flexibility of the network particularly when dealing with dynamic workloads.

Related work primarily focuses on DoS attacks. However, in the case of DDoS attacks originating from multiple chiplets, the detection mechanisms may prove ineffective without profiling application behavior as a priori [5], particularly when dealing with small packet injection rates that go undetected. For instance, in a recent study [20], the minimum flit injection rate assumed for attacks is 0.3. In contrast, we will demonstrate that a DDoS can occur in a chiplet system with an injection rate 10× smaller than that.

3 THREAT MODEL AND ASSUMPTIONS

Our threat model assumes the following: similar to Fig. 1, we assume a system with multiple chiplets connected to routers within an NoI. Each chiplet is composed of multiple IPs/cores (e.g., sixteen in Fig. 1), each with its own router that is part of an NoC, whereas only a few (e.g., four in Fig. 1) of these routers serve as *boundary routers* which connect to the interposer's routers (NoI) via VL.

We assume that there can be multiple malicious chiplets (e.g., two in Fig. 1). Furthermore, our model assumes the NoI is *trusted*. It is also assumed the attacker of a compromised chiplet has full control over its routers, with the ability to send as many packets as desired. However, the attacker *does not* possess full knowledge of the underlying router architecture on the NoI. Furthermore, we assume that various chiplets in the system have different *criticality* and *trust levels*. Specifically, we assume some chiplets are critical hence their availability should be protected. Further, we define trust level as whether or not a chiplet can be trusted. For example, from NoI's perspective, chiplets from vendor A are trusted while chiplets from vendor B are not (i.e., due to supply chain issues).

In a DoS scenario, a malicious chiplet can inundate a critical chiplet with a high volume of traffic, jeopardizing its connectivity to the NoI. For a DDoS attack, *multiple* malicious chiplets collaborate to coordinate an attack. In both cases, the goal is to concentrate a load on a specific part of the NoI or a VL associated with a critical chiplet. Detecting DDoS attacks poses significant challenges as a low injection rate from multiple chiplets can collectively generate a high volume of traffic, complicating the identification of malicious activity. Additionally, the collaboration of two or more malicious chiplets can cause congestion by reciprocally forwarding traffic, leading to congestion on a specific path of the NoI.

4 SCRIPT: PROPOSED PERFORMANCE- AND SECURITY-AWARE ROUTING FRAMEWORK

4.1 Overview

The SCRIPT framework, shown in Fig. 2, introduces a comprehensive approach to enhance the security and performance of 2.5D chiplet systems. This framework includes two key components: (a) Design-time optimization, and (b) Secure and performance-preserved routing. The design-time optimization is a pivotal aspect of SCRIPT, where the trust level and criticality of individual chiplets are taken into account. This optimization process aims to strike a balance between enhancing security in the runtime routing and ensuring that it does not introduce substantial performance overhead. Additionally, performance objectives are taken into account during

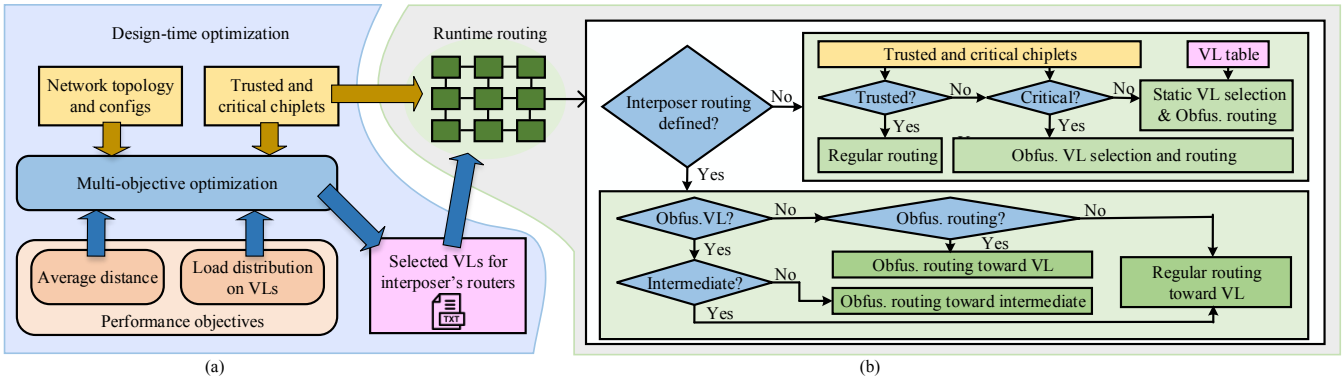

Figure 2: SCRIPT framework: (a) Design-time optimization takes into account the trust level and criticality of chiplets, along with performance objectives, to propose a list of Vertical Links (VLs) for use in the runtime routing. (b) In routing, a VL is selected, and the routing mode (obfuscated or regular) is determined to deliver the packet to the specified VL.

this phase. The outcome is the generation of an optimized set of VLs, which serves as the regular VL selection for the subsequent runtime routing of the network. The runtime routing process in SCRIPT involves the dynamic selection of VLs based on the optimized list generated during design time and a random VL selection for obfuscation. Furthermore, it offers the flexibility to choose between obfuscated and regular routing. The obfuscated routing mode adds a layer of security by intentionally introducing complexity into the routing path, contributing to the framework's overall resilience against potential security threats.

4.2 SCRIPT Routing Process

In chiplet systems, the routing algorithm utilizes boundary routers as intermediate routers to deliver packets from one chiplet to another. A boundary router is specifically defined as a router on a chiplet that is directly connected to the NoI through a VL. The routing process for sending a packet from a source chiplet to a destination chiplet involves five distinct steps: (1) the packet is routed from the source router to a boundary router on the source chiplet; (2) vertical routing occurs as the packet moves from the boundary router on the source chiplet to the NoI; (3) the packet is routed across the NoI to the VL connected to the destination chiplet; (4) vertical routing occurs from the NoI to the boundary router on the destination chiplet, and (5) the packet is routed from the boundary router to its ultimate destination within the destination chiplet. Our primary focus is on steps 3 and 4, where attacker chiplets can potentially concentrate traffic on a specific path on the NoI or a VL to initiate an attack. Note that attack mitigation within a chiplet (i.e., IP-level DoS attacks) is beyond the scope of this paper, as there is substantial existing research on this topic.

In step 3, concerning intra-interposer routing, we aim to implement obfuscated routing for untrusted packets. A comprehensive explanation of our obfuscated routing approach is provided in Section 4.5. For the VL selection employed in step 4 of the routing process, we enhance network security against DDoS by improving VL selection, as detailed in Section 4.3. Nevertheless, this security enhancement can have a significant impact on network performance. Therefore, we introduce a multi-objective optimization strategy, discussed in Section 4.4. Please note that the first VL selection on the source chiplet (i.e., step 2) is beyond the scope of this paper and we consider the source VL-selection strategy used in prior work (e.g., see DeFT [23]) in our experiments (see Section 5).

4.3 Enhancing Security in VLs

To enhance the security of 2.5D chiplet systems, it is crucial to address the risk of attacks on VLs that connect critical chiplets to NoI. VLs (i.e., microbumps) are limited resources compared to metal links on chips, making them susceptible to DDoS attacks due to their scarcity and relatively smaller bandwidth. The combination of their limited availability and bandwidth amplifies their vulnerability to congestion, which malicious chiplets can exploit—i.e., to deliberately congest a specific path or VL, triggering a DDoS attack. Therefore, one of the key objectives in SCRIPT is to distribute the load across various VLs of chiplets to prevent congestion.

Note that we assume the NoI is trusted, thus we maintain control over the routing of the NoI. Hence, the main feature of our routing algorithm is the strategic NoI routing. The selection of VLs (on the destination chiplet) as part of the NoI routing process is crucial in this context. An attacker might attempt to send multiple packets from different sources to a victim chiplet, inducing congestion on a specific VL at the victim chiplet's input. Current routing algorithms for 2.5D chiplet systems often employ a static VL selection [14, 23, 24], making them susceptible to attacks where the attacker sends packets to the destinations using the same VL, leading to congestion. For instance, an attacker could strategically distribute packets from multiple sources of untrusted chiplets to multiple destinations near a specific VL, causing congestion on that VL. To counter this, we randomly select destination VLs for packets originating from an untrusted source chiplet and destined for a critical chiplet. However, it is important to note that this random selection has a notable impact on network performance.

Fig. 3 assesses the average distance overhead introduced by this approach. Here, we assume a four-chiplet system under uniform traffic, where each chiplet is a 4×4 mesh. As depicted, the average distance increases with the growing number of untrusted chiplets. A larger average distance necessitates packets to traverse a

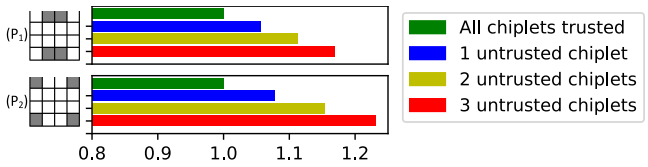

Figure 3: Normalized average distance overhead with two different patterns of chiplet-connected VLs (P_1 and P_2).

greater number of routers, leading to increased latency and energy consumption in low-traffic scenarios. Also, it raises the risk of congestion. Consequently, we opt for dynamic random destination VL selection exclusively for packets originating from untrusted chiplet sources and destined to critical chiplets. We term these packets "VL obfuscated packets (VOP)" due to their routing in an obfuscated manner on the NoI, in addition to their random VL selection.

Given that the random VL selection for the VOP packet is contingent on the trust level of the source chiplet, we define T_c as the trust factor of chiplet c, with a trust factor of 0 denoting that the chiplet is completely untrusted, and a trust factor of 1 indicating full trust in the chiplet. In our framework, we utilize this trust factor to prioritize performance enhancement for trusted chiplets, while focusing on improving routing to enhance security against DDoS attacks for untrusted chiplets. Similarly, we define Cr_c as the criticality level of a chiplet as a victim chiplet of DDoS attacks. Therefore, we label a packet as an VOP packet given that the packet generated from source s and forwarded to destination d with probability:

$$Ov_d^s = \begin{cases} T_{c_s}, & \text{if } Cr_{c_d} = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here, T_{c_s} is the trust factor of the source chiplet and Cr_{c_d} is the criticality level of the destination chiplet.

To alleviate the impact on system performance under SCRIPT's secure VL selection, we conduct design-time optimization for static VL selection (predefined VLs selected during design time), which is employed by trusted packets. These optimizations are geared towards enhancing performance while taking into account the considerations of our secure VL selection methodology.

4.4 Multi-Objective Optimization

As previously mentioned, in addition to the dynamic random VL selection for VOP packets, we employ a static VL selection for regular packets. An optimized set of VLs is pre-stored in routers for runtime usage (cost is analyzed in Section 5.4). Note that static VL selection is only used for regular packets, and not for VOP packets that are from untrusted sources. This section delves into the optimization of the selected VL set, with an emphasis on enhancing both the security and performance aspects of the network.

While random online selection is suitable for obfuscation purposes, it is inefficient for performance due to the necessity for the NoI's router to search among available VLs and make real-time selections. To address this, we opt for offline selection for regular packets, a method commonly employed in state-of-the-art routing algorithms [14, 23, 24], as described below.

Let NC represent the total number of chiplets, N_c represent the number of nodes in chiplet c, and NV_c represent the number of VLs in chiplet c. We also define, L_{d_v} as the load on vertical link v connected to destination chiplet d. We calculate L_v based on the load generated by trusted chiplets. Note that the load generated by untrusted chiplets is already distributed among VLs since an online random selection is used. Therefore, the load on vertical link v based on the trust level of chiplets is:

$$L_{d_v} = \sum_{c=1}^{NC} \sum_{i=1}^{N_c} \sum_{j=1}^{N_d} W_{(i,j)} \times (1 - Ov_j^i) \times U_{v_{(i,j)}}. \tag{2}$$

In this equation, $W_{(i,j)}$ denotes the weight of communication between node i and j. Ov_j^i, which we introduced in (1), is based on the trust factor of the source chiplet and the criticality factor of the destination chiplet. We use this term to exclude the VOP packets from the load, as we have already distributed the load coming from untrusted chiplets on VLs. The variable $U_{v_{(i,j)}}$ signifies whether communication between node i and node j utilizes vertical link v or not. It is defined as:

$$U_{v_{(i,j)}} = \begin{cases} 1, & \text{if } v \text{ is used to send a packet from } i \text{ to } j \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

We use utilization variance of L_{d_v} in our design-time optimization to mitigate potential congestion on s [17, 21]. This objective enhances potential congestion for the network's performance and also secures the network against DDoS attacks. We define utilization variance on the VLs of chiplet d as:

$$UtVar_d = \frac{1}{NV_d} \sum_{v=1}^{NV_d} (L_v - \mu_d)^2, \tag{4}$$

where μ_d denotes the average load on all VLs of chiplet d:

$$\mu_d = \frac{1}{NV_d} \sum_{v=1}^{NV_d} L_v. \tag{5}$$

Now that we have utilization variance for every chiplet, we want to calculate to total utilization variance on VLs as the objective in our optimization. Thus:

$$UtObj = \sum_{d=1}^{NC} UtVar_d. \tag{6}$$

The number of hop counts between the source and destination significantly impacts the network performance. Large hop counts create more traffic and worsen congestion. It also increases the latency of each packet and results in higher energy consumption during packet transmission. To address this issue, we incorporate *weighted distance* as one of our performance objectives during optimization. Weighted distance is defined as the sum of source-destination distances weighted by the average communication frequency (i.e., communication weight) between the source and destination. We define the weighted distance objective as:

$$DisObj = \sum_{c=1}^{NC} \sum_{i=1}^{N_c} \sum_{j=1}^{N_d} W_{(i,j)} d_{(i,j)} \times IC_{(i,j)} \times (1 - Ov_j^i), \tag{7}$$

where $d_{(i,j)}$ is the source-destination hop count between node i and j. We use $IC_{(i,j)}$ to exclude source-destination pairs on the same chiplet since they are not affected by our VL selection optimization. Therefore, $IC_{(i,j)} = 0$ indicates that routers i and j are on the same chiplet, while $IC_{(i,j)} = 1$ indicates that routers i and j are

Algorithm 1 SCRIPT virtual channel selection

```
if Reset then
    Round_Robin ← 0
if current is the first router on NoI then
    if Packet is ROP then
        Virtual channel ← 0
    else
        if Round_Robin is TRUE then
            Virtual channel ← 1
            Round_Robin ← FALSE
        else
            Virtual channel ← 0
            Round_Robin ← TRUE
else if current is the intermediate and packet is ROP then
    Virtual channel ← 1
```

Algorithm 2 SCRIPT routing on interposer

```
if packet arrived at destination vertical link then
    Route vertically to up
else if packet is ROP then
    if the intermediate is not visited then
        Route West-first toward intermediate
    else
        Route East-first toward the vertical link
else if packet is regular then
    if Virtual channel is 0 then
        Route West-first toward the vertical link
    else
        Route East-first toward the vertical link
```

on different chiplets. Similar to the utilization variance objective, we use the term $(1 - Ov_j^i)$ to exclude VOP packets because they are not affected by static VL optimization. We use $DisObj$ and $UtObj$ in our optimization of static VL selection. We employ a simulated-annealing-based multi-objective optimization algorithm [1] for the optimization. More details of our optimization and solution selection process are presented in Section 5.2.

4.5 Obfuscated Deadlock-Free Routing

In addition to dynamic random VL selection, we employ obfuscated routing on the NoI for "route obfuscated packets (ROP)" with probability $OR_d^s = T_{c_s}$. To enhance unpredictability and deter potential attacks, we randomly select an intermediate router for ROP packets. However, applying obfuscated routing to all packets from untrusted chiplets is not performance-efficient. Therefore, we implement obfuscated routing for untrusted packets if their path traverses a *critical region* of the NoI. We define the critical region of the NoI as the area where routers are directly connected to a critical chiplet.

We implement our obfuscated routing using the virtual channel (VC) concept [6, 23] but without any extra VCs compared to VCs already used for deadlock prevention in chiplet systems. VCs are employed to separate traffic flows in virtual networks, preventing cyclic dependencies, and avoiding deadlock. Employing the deadlock-freedom solution in the DeFT routing algorithm [23], packets are switched to the second VC across the NoI. This implies that VC switching is permitted when a packet enters from the source chiplet into the NoI and should occur before the packet exits the NoI. We leverage this opportunity to develop an obfuscated routing algorithm for the NoI. Packets for which we intend to obfuscate (i.e., ROP packets) are initially routed to an intermediate router using the first VC and then directed to the VL using the second VC. To guarantee livelock freedom, a 1-bit flag is employed in the header flit. This flag marks packets that have already been routed to an intermediate router, ensuring that the packet will be routed minimally to the VL after visiting the intermediate router.

For regular packets, we utilize a round-robin approach at the first router on the NoI, evenly distributing the load across VCs. As a result, 50% of the packets undergo switching upon entering the NoI, while the remaining 50% undergo switching upon exiting the NoI. However, for ROP packets at the first node in the NoI, the first VC is employed, and the packet switches to the second VC upon reaching the intermediate router. The details of the VC-selection process are illustrated in Algorithm 1. Note that following deadlock freedom rules in DeFT, the packets are initially routed in the first VC from a source chiplet.

Algorithm 2 also outlines our routing approach on the NoI. We utilize West-first routing (adaptive for routing to the east direction) in the first VC and East-first routing (adaptive for routing to the west direction) in the second VC. West-first routing is introduced in prior work [10], and East-first routing is created by exchanging west and east directions in West-first routing. The reason for employing partially adaptive routing in two opposite directions is that each routing strategy provides a better load distribution in a specific direction. Choosing two opposite directions allows us to achieve a higher load distribution. Moreover, these partially adaptive routing strategies not only contribute to traffic distribution but also enhance security since it becomes less predictable for potential attackers to determine the exact path for a routing scenario.

5 EVALUATION AND SIMULATION RESULTS

5.1 Simulation Setup

We extend the capabilities of the Noxim simulator [2] to accommodate chiplet systems. The simulation framework is augmented to facilitate a comparative analysis with DeFT [23] and MTR [24] routing strategies. Although DeFT and MTR are not specifically proposed for security purposes, we compare with these well-known routing approaches as there is currently no other work proposing a routing algorithm to enhance the security of chiplet systems. For the simulation, we adopt the same configuration parameters as employed in DeFT and MTR for a consistent evaluation. The chiplet system consists of four chiplets, each with a 4×4 mesh network configuration. Similarly, the NoI is modeled as a 4×4 mesh network. The interconnection between the chiplets and the NoI involves four routers per chiplet, designated as boundary routers, which are linked to the NoI. The arrangement of boundary routers is the same as the pattern used in DeFT and MTR, represented also as P_1 in Fig. 3. Each router is equipped with buffers at each port with space for 4 flits, 2 virtual channels, and a 128-bit link and flit width. This configuration ensures a comparable performance evaluation of SCRIPT to DeFT and MTR routing algorithms. We also employ partially adaptive routing (West-first and East-first) on NoI for DeFT and MTR to have a fair comparison.

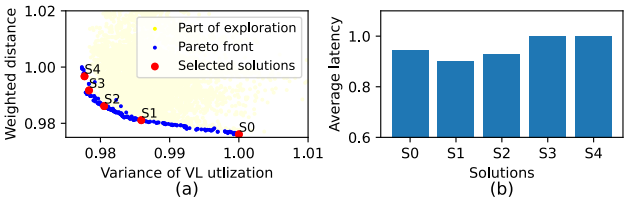

Figure 4: Multi-objective optimization and solution selection process. (a) Pareto front under weighted distance and utilization variance objectives. Five solutions on the Pareto front are selected for evaluation. (b) Latency results of the simulation under the selected solutions.

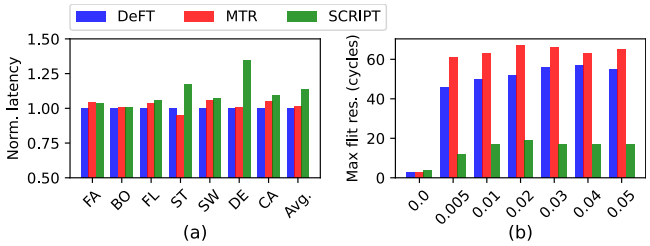

Figure 5: (a) Normalized average latency under real application scenarios without attacks. The X-axis shows the initial two letters of each application. (b) Maximum average flit residency in a router among all routers of NoI under different attack scenarios (results are averaged among all applications).

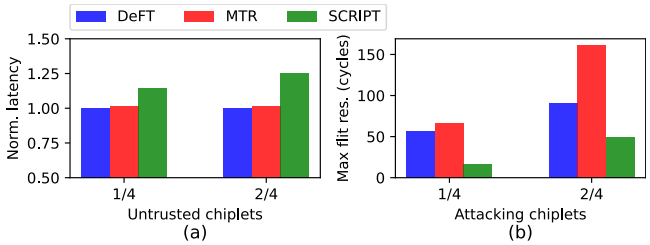

Figure 6: Impact of an increased number of untrusted and attacker chiplets on (a) average latency, and (b) flit residency.

We incorporate real application traffic by utilizing a set of widely recognized PARSEC applications: *facesim, bodytrack, fluidanimate, streamcluster, swaptions, dedup, and canneal.* To emulate these benchmarks, we employ Gem5 in full-system simulation mode. The simulated system architecture comprises 64 x86 cores, each equipped with a private L1 cache. Additionally, the system incorporates four L2 cache banks to manage shared resources efficiently. This configuration reflects a realistic representation of a multi-core processing environment, allowing us to evaluate the performance and behavior of the proposed enhancements under conditions that closely resemble practical scenarios.

5.2 Multi-Objective Optimization Results

In our design-space exploration, we implemented the Simulated Annealing-based Multi-Objective Optimization Algorithm (AMOSA) [1] to optimize the performance of our NoI. As detailed in Section 4.4, our optimization considers two primary objectives related to performance, with additional considerations for the trust and criticality levels of chiplets. The Pareto front generated by AMOSA, shown in Fig. 4.a, showcases various solutions, each representing a trade-off between optimized weighted distance and utilization variance on VLs. The input to this optimization is the average traffic over all the PARSEC applications. From this Pareto front, we selected several solutions for further evaluation and simulated them using our enhanced Noxim simulator. Fig. 4.b illustrates the average network latency comparison for these selected solutions.

Solution 1, in particular, has been selected due to its minimized average latency. It is worth noting that the optimal solution can vary depending on the nature of the traffic. In scenarios with low congestion, solutions emphasizing smaller average distances are more efficient. Conversely, in highly congested situations, a solution that minimizes utilization variance proves beneficial.

It is important to highlight that our optimization approach does not tailor the traffic to each application individually; rather, we adopt an average over all the traffic types as our optimization input, making a pessimistic assumption. However, considering the traffic of each application in the design-time optimization could lead to further improvements, but might not be realistic.

Our primary contribution lies in the flexibility of our proposed framework, which is not tied to any specific multi-objective optimization method. Our framework can be implemented with any multi-objective optimization technique without loss of generality. We opted for AMOSA due to its simplicity and efficiency. While it is acknowledged that simulated annealing-based search may be slower compared to other optimization techniques, the optimization process in our case is conducted offline, and we simplify objectives using our formalization, mitigating concerns about speed. Indeed, in our evaluation, the optimization was completed in less than an hour. In our implementation of AMOSA, we utilized specific parameters to optimize the exploration of solution space. We initialized the temperature at a relatively high value of $1.0e5$ degrees, allowing for a wide exploration range in the early stages of the algorithm. This high starting temperature encourages acceptance of uphill moves, which helps in escaping local optima. To ensure convergence towards an optimal solution, we set the stopping temperature to a significantly low value of $1.0e-5$ degrees. This stringent criterion enables the algorithm to focus on exploiting promising regions as it approaches convergence. Moreover, we incorporated an annealing schedule with a cooling factor (alpha) of 0.95. This factor determines the rate at which the temperature decreases, balancing between exploration and exploitation throughout the iterations. By gradually reducing the temperature, we control the balance between exploration and exploitation, promoting deeper exploration initially while fine-tuning towards optimal solutions later in the process. To manage computational resources efficiently, we conducted a fixed number of iterations set at 100, ensuring a reasonable trade-off between solution quality and computational cost.

5.3 Security and Latency Analysis

We assess the average latency under real application scenarios without any attacks, as depicted in Fig.5.a. For this evaluation, we assume one out of the four chiplets is untrusted, while another is critical. SCRIPT introduces a minor latency overhead by employing obfuscated routing and VL selection for certain packets, enhancing security against DDoS attacks. This overhead is slightly higher in high-load applications such as *streamcluster*, *dedup*, and *canneal*. On average, SCRIPT exhibits a 14% increase in the average latency.

To evaluate the reliability of SCRIPT under DDoS attacks, we consider the maximum average flit residency in a router across all NoI's routers under various attack scenarios, shown in Fig.5.b. This assessment is conducted across all real applications. Flit residency refers to the average time a flit remains in a router, experiencing blocking. A prolonged waiting time results in a DoS, as packets cannot be delivered in real time. In this experiment, we investigate the impact of four malicious nodes attacking the critical chiplet with varying rates of malicious flit injection (flits/cycle/node). Even with low injection rates from malicious nodes, the effects are significant on DeFT and MTR, due to the distributed nature of the attack. This highlights the potential benefits of SCRIPT as a security-aware routing solution. Traditional attack detection approaches may struggle to identify attacks with low injection rates. For instance, in [20], the minimum assumed attack injection rate is 0.3. However, we show that under an injection rate of, for example, 10× smaller (i.e., 0.03 in Fig. 5.b), a flit residency of 56 cycles is imposed with an insecure routing like DeFT. Therefore, our SCRIPT framework can be used in conjunction with an attack detection mechanism to handle undetected DDoS attacks. SCRIPT effectively handles such attacks and reduces flit residency by at least 64% in the given attack scenarios.

We further assess the performance under an increased number of untrusted chiplets, as illustrated in Fig. 6 (1/4 means one chiplet out of four chiplets). Assuming half of the chiplets are untrusted, SCRIPT incurs a latency overhead of at most 26%, while concurrently reducing flit residency by at least 46%.

5.4 Area and Power Analysis

We implement the SCRIPT routing using Cadence Genus under a 15-nm technology node, considering a 128-bit flit width, 1V voltage, 2GHz frequency, and a 5-port router configuration. The results of the area and power analysis are summarized in Table 1, revealing negligible overhead in both aspects. Compared to DeFT, the area and power overhead of SCRIPT are 1.5% and 0.8%, respectively.

Compared to MTR and DeFT, our approach needs only four VL addresses per chiplet, facilitating a round-robin (RR) selection for obfuscated VLs without significant impact. Moreover, to optimize the selection of random intermediates and minimize overhead, we adopt an RR approach. Eight random routers are statically selected and stored in the router, facilitating an RR intermediate selection process. This strategy ensures the robustness of SCRIPT while keeping area and power overhead at a minimal level. Similarly, we used RR with the precision of 3 bits (8 levels) for implementing the probability of obfuscation (see Equ. 1). Also, SCRIPT uses 4 bits of header flit to store the address of an intermediate destination, which is negligible compared to 128-bit flit width.

Table 1: Area and power: SCRIPT vs. DeFT and MTR

Routing	DeFT	MTR	SCRIPT
Router area (μm^2)	9635.7	9496.4	9782.7
Router power (mW)	9.47	9.44	9.48

6 CONCLUSION

This paper introduced SCRIPT, a routing framework tailored to improve reliability and performance against DDoS attacks in 2.5D chiplet systems. By using obfuscation routing techniques on the NoI, SCRIPT addressed security challenges associated with modular integration. Considering chiplet trust levels and implementing a multi-objective optimization, SCRIPT enhanced network resilience against both single-source DoS attacks and DDoS scenarios. Our new framework enhances the security of NoI by at least 64% under DDoS attacks, while minimizing performance degradation.

ACKNOWLEDGMENTS

This work has been supported, in part, by NSF grants CNS-2046226, CNS-2211301, CNS-2303115, and CNS-2312089. The findings are those of the authors and do not necessarily reflect the views of NSF.

REFERENCES

[1] Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. 2008. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation* 12, 3 (2008), 269–283.
[2] Vincenzo Catania et al. 2016. Cycle-accurate network on chip simulation with noxim. *ACM TOMACS* 27, 1 (2016), 1–25.
[3] Gino A Chacon, Charles Williams, Johann Knechtel, Ozgur Sinanoglu, and Paul V Gratz. 2022. Hardware Trojan Threats to Cache Coherence in Modern 2.5 D Chiplet Systems. *IEEE Computer Architecture Letters* 21, 2 (2022), 133–136.
[4] Subodha Charles, Yangdi Lyu, and Prabhat Mishra. 2019. Real-time detection and localization of DoS attacks in NoC based SoCs. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*.
[5] Subodha Charles, Yangdi Lyu, and Prabhat Mishra. 2020. Real-time detection and localization of distributed DoS attacks in NoC-based SoCs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 12 (2020), 4510–4523.
[6] Masoumeh Ebrahimi and Masoud Daneshtalab. 2017. EbDa: A new theory on design and verification of deadlock-free interconnection networks. In *the 44th Annual International Symposium on Computer Architecture*. 703–715.
[7] Dabin Fang, Huikai Li, Jun Han, and Xiaoyang Zeng. 2013. Robustness analysis of mesh-based network-on-chip architecture under flooding-based denial of service attacks. In *IEEE Eighth International Conference on Networking, Architecture and Storage*. IEEE, 178–186.
[8] Ramon Fernandes, César Marcon, Rodrigo Cataldo, Jarbas Silveira, Georg Sigl, and Johanna Sepúlveda. 2016. A security aware routing approach for NoC-based MPSoCs. In *29th Symposium on Integrated Circuits and Systems Design (SBCCI)*.
[9] Jonathan Frey and Qiaoyan Yu. 2017. A hardened network-on-chip design using runtime hardware Trojan mitigation methods. *Integration* 56 (2017), 15–31.
[10] Christopher J Glass et al. 1992. The turn model for adaptive routing. *ACM SIGARCH Computer Architecture News* 20, 2 (1992), 278–287.
[11] Rajesh JS, Dean Michael Ancajas, Koushik Chakraborty, and Sanghamitra Roy. 2015. Runtime detection of a bandwidth denial attack from a rogue network-on-chip. In *the 9th International Symposium on Networks-on-Chip*. 1–8.
[12] Manoj Kumar Jyv, Ayas Kanta Swain, Sudeendra Kumar, Sauvagya Ranjan Sahoo, and Kamalakanta Mahapatra. 2018. Run time mitigation of performance degradation hardware trojan attacks in network on chip. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 738–743.
[13] M Shafkat M Khan, Chengjie Xi, Aslam A Khan, M Tanjidur Rahman, Mark M Tehranipoor, and Navid Asadizanjani. 2022. Secure interposer-based heterogeneous integration. *IEEE Design & Test* 39, 6 (2022), 156–164.
[14] Pritam Majumder, Sungkeun Kim, Jiayi Huang, Ki Hwan Yum, and Eun Jung Kim. 2020. Remote control: A simple deadlock avoidance scheme for modular systems-on-chip. *IEEE Trans. Comput.* 70, 11 (2020), 1928–1941.
[15] Mohammed Nabeel, Mohammed Ashraf, Satwik Patnaik, Vassos Soteriou, Ozgur Sinanoglu, and Johann Knechtel. 2020. 2.5D root of trust: Secure system-level integration of untrusted chiplets. *IEEE Trans. Comput.* (2020).

[16] Ahmad Patooghy, Mahdi Hasanzadeh, Amin Sarihi, Mostafa Abdelrehim, and Abdel-Hameed A Badawy. 2023. Securing Network-on-chips Against Fault-injection and Crypto-analysis Attacks via Stochastic Anonymous Routing. *ACM Journal on Emerging Technologies in Computing Systems* 19, 3 (2023), 1–21.

[17] Sirui Qi, Yingheng Li, Sudeep Pasricha, and Ryan Gary Kim. 2023. MOELA: A Multi-Objective Evolutionary/Learning Design Space Exploration Framework for 3D Heterogeneous Manycore Platforms. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1–6.

[18] Yousef Safari, Pooya Aghanoury, Subramanian S Iyer, and Nader Sehatbakhsh. 2023. Secure and Scalable Key Management for Waferscale Heterogeneous Integration. In *73rd Electronic Components and Technology Conference*. 2125–2130.

[19] Yousef Safari, Pooya Aghanoury, Subramanian S Iyer, Nader Sehatbakhsh, and Boris Vaisband. 2023. Hybrid obfuscation of chiplet-based systems. In *60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[20] Mitali Sinha, Setu Gupta, Sidhartha Sankar Rout, and Sujay Deb. 2021. Sniffer: A machine learning approach for DoS attack localization in NoC-based SoCs. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* (2021).

[21] Ebadollah Taheri, Ryan G Kim, and Mahdi Nikdast. 2021. AdEle: An adaptive congestion-and-energy-aware elevator selection for partially connected 3D NoCs. In *58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 67–72.

[22] Ebadollah Taheri, Mohammad Amin Mahdian, Sudeep Pasricha, and Mahdi Nikdast. 2023. TRINE: A Tree-Based Silicon Photonic Interposer Network for Energy-Efficient 2.5 D Machine Learning Acceleration. In *Proceedings of the 16th International Workshop on Network on Chip Architectures*. 15–20.

[23] Ebadollah Taheri, Sudeep Pasricha, and Mahdi Nikdast. 2022. DeFT: A deadlock-free and fault-tolerant routing algorithm for 2.5 d chiplet networks. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*.

[24] Jieming Yin, Zhifeng Lin, Onur Kayiran, Matthew Poremba, Muhammad Shoaib Bin Altaf, Natalie Enright Jerger, and Gabriel H Loh. 2018. Modular routing design for chiplet-based systems. In *ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 726–738.