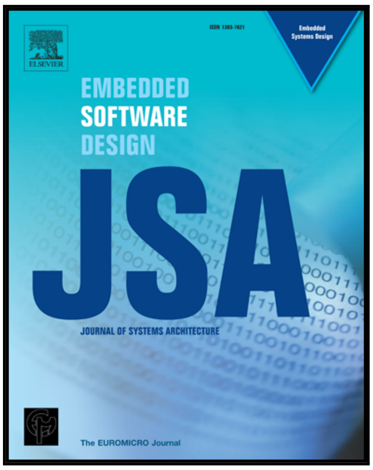
Accepted Manuscript



Effectiveness of HT-assisted Sinkhole and Blackhole Denial of Service Attacks Targeting Mesh Networks-on-chip

Li Zhang, Xiaohang Wang, Yingtao Jiang, Mei Yang, Terrence Mak, Amit Kumar Singh

PII: S1383-7621(18)30067-5
DOI: https://doi.org/10.1016/j.sysarc.2018.07.005
Reference: SYSARC 1512

To appear in: *Journal of Systems Architecture*

Received date: 5 March 2018
Revised date: 12 July 2018
Accepted date: 25 July 2018

Please cite this article as: Li Zhang, Xiaohang Wang, Yingtao Jiang, Mei Yang, Terrence Mak, Amit Kumar Singh, Effectiveness of HT-assisted Sinkhole and Blackhole Denial of Service Attacks Targeting Mesh Networks-on-chip, *Journal of Systems Architecture* (2018), doi: https://doi.org/10.1016/j.sysarc.2018.07.005

Effectiveness of HT-assisted Sinkhole and Blackhole Denial of Service Attacks Targeting Mesh Networks-on-chip[☆]

Li Zhang[a], Xiaohang Wang[a,*], Yingtao Jiang[b], Mei Yang[b], Terrence Mak[c], Amit Kumar Singh[d]

[a]*South China University of Technology, China*
[b]*University of Nevada, Las Vegas, USA*
[c]*University of Southampton, UK*
[d]*University of Essex, UK*

Abstract

There are ample opportunities at both design and manufacturing phases to meddle in a many-core chip system, especially its underlining communication fabric, known as the networks-on-chip (NoC), through the inclusion of malicious hardware Trojans (HT). In this paper, we focus on studying two specific HT-assisted Denial-of-Service (DoS) attacks, namely the sinkhole and blackhole attacks, that directly target the NoC of a many-core chip. As of the blackhole attacks, those intermediate routers with inserted HTs can stop forwarding data packets/flits towards the packets' destination; instead, packets are either dropped from the network or diverted to some other malicious nodes. Sinkhole attacks, which exhibit similar attack effects as blackhole attacks, can occur when the NoC supports adaptive routing. In this case, a malicious node actively solicits packets from its neighbor nodes by pretending to have sufficient free buffer slots. Effects and efficiencies of both sinkhole and blackhole DoS attacks are modeled and quantified in this paper, and a few factors that influence attack effects are found to be critical. Through fine-tuning of these parameters, both attacks are shown to cause more damages to the NoC, measured as over 30% increase in packet loss rate. Even with current detection and defense methods in place, the packet loss rate is still remarkably high, suggesting the need of new and more effective detection and defense methods against the enhanced blackhole and sinkhole attacks as described in the paper.

Keywords: networks-on-chip, hardware Trojan, denial-of-service attack

1. Introduction

Hardware Trojans (HT) can pose a serious threat to many-core chips, as they might cause the chips to malfunction, or leak sensitive information. HTs can be inserted by embedding a malicious circuit during the design or manufacturing phase of a chip [1]. In the literature [2, 3, 4], a few HT designs were proposed and they could be used to launch denial-of-service (DoS) attacks against the networks-on-chip (NoC) component of a many-core chip, and the HT-enabled DoS attacks can cause serious damages to NoC, including dropping of packets, jamming of certain network node(s), leaking sensitive information, or modification of functionalities, *etc.* [1].

In this paper, we consider two HT-assisted Denial-of-Service (DoS) attacks, namely sinkhole and blackhole attacks targeting NoC in a many-core chip. Both attacks can cause great damage to the chips and the users of the chips. For a balckhole attack, it can cause chips to malfunction and leak the sensitive information, while a sinkhole attack can aggregate the traffic and intercept the packets. In addition, the two attacks are easy to be realized by an HT that has extremely low area and power costs, and can be hard to detect.

To enable a blackhole attack, HTs are inserted into the routers such that packets are not forwarded to their intended destination; instead, the packets are either dropped out from the network or forwarded to other malicious nodes. Suppose node 1 in the example shown in Fig. 1 has a packet that needs to be sent to node 9, and the packet is routed through a malicious node, node 6. Upon receiving the packet, node 6 actually sends the packet to a malicious node, node 5 in this example. Node 9, the intended recipient of the packet, will not be able to receive a single packet from node 1. It should be noted that as there is only one malicious interceptor, and as so, there will be no deadlock in the network caused by the attack.

[☆] This research program is supported by the Natural Science Foundation of China No. 61376024 and 61306024, Natural Science Foundation of Guangdong Province 2015A030313743 and 2018A030313166, Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase), and the Science and Technology Research Grant of Guangdong Province No. 2016A010101011 and 2017A050501003, Pearl River S&T Nova Program of Guangzhou No. 201806010038, and Tip-top Scientific and Technical Innovative Youth Talents of Guangdong special support program (No. 2014TQ01X590).

[*] Corresponding author

Email addresses: `201721045909@mail.scut.edu.cn` (Li Zhang), `xiaohangwang@scut.edu.cn` (Xiaohang Wang), `yingtao.jiang@unlv.edu` (Yingtao Jiang), `mei.yang@unlv.edu` (Mei Yang), `tmak@ecs.soton.ac.uk` (Terrence Mak), `a.k.singh@essex.ac.uk` (Amit Kumar Singh)

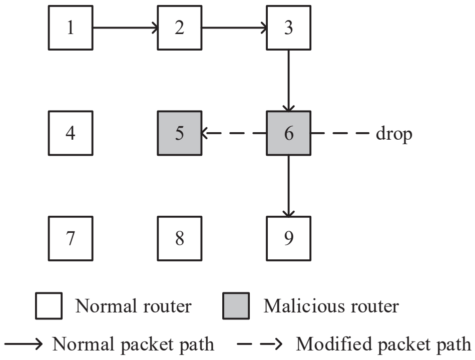


Figure 1: An example illustrating a blackhole attack.

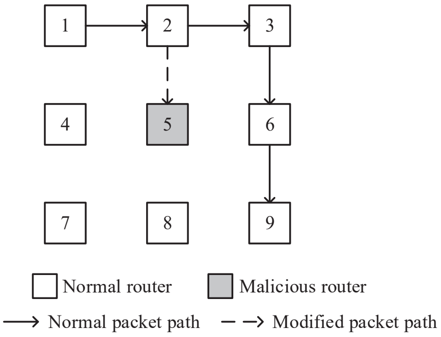


Figure 2: An example illustrating a sinkhole attack.

From the example shown in Fig. 1, one can see that in a blackhole attack, the malicious nodes passively drop packets or reroute packets to unintended recipients. This is quite different from a sinkhole attack, where a malicious node actively solicits packets from its neighbor nodes by pretending to have sufficient free input buffer slots with adaptive routing. To illustrate this type of attack, let us assume that node 1 in Fig. 2 needs to send a packet to node 9. When a packet reaches node 2, it has to make a decision regarding which of the two downstream routers, nodes 3 and 5, shall be forwarded to. This decision is largely determined by node 2's knowledge about how many input buffer slots that each of nodes 3 and 5 has. Node 5, a malicious router in this example, has purposely notified node 2 that it has more empty slots in its input buffer than node 3. As a result, a packet passing through node 2 will more likely be routed to node 5 than to node 3. This malicious node can then either drop any packet coming to it or forward a packet to some other nodes for more harm. Either way, node 9 will not be able to receive some or all the packets that were designated to it.

Effects of the above described sinkhole and blackhole DoS attacks depend on a number of factors, including the number of HTs and their distributions in the NoC, traffic characteristics of the applications, and a few system parameters. In this paper, we intend to study and model how these factors and parameters can be explored to enhance the effectiveness of attacks. We will also examine how well the current detection and defense methods respond to the enhanced blackhole and sinkhole attacks described in the paper.

The contribution of the paper is two-fold. First, we propose the HT-assisted blackhole and sinkhole attacks targeting NoC, which can be launched by an HT that is found hard to be detected. Design of the HT is thus described and the attack flow is presented. The stealthiness of the HT is also analyzed and guaranteed. Second, we explore the factors that correlate to the attack effects of sinkhole and blackhole attacks. A method to maximize the attack effects is presented, and correspondingly, an HT insertion methodology is proposed for maximized attack effects in different situations.

The rest of the paper is organized as follows. Section 2 reviews the related works. Realization of blackhole/sinkhole attacks is described in Section 3, followed by a detailed study on various parameters that can contribute to the attack effects in Section 4. Section 5 introduces the detection and defense methods that can be employed to thwart the sinkhole and blackhole DoS attacks. Section 6 reports the results and assesses the effectiveness of the two DoS attacks with or without detection/defense employed. Finally, Section 7 concludes the paper.

2. Related Work

2.1. Hardware Trojans

There are many possible channels that can get an HT into a chip. For instance, when a many-core chip design house employs a foundry to manufacture the chip, a compromised staff member of the foundry can secretly insert HTs to the chip layout before it is fabricated. Once the infected chips after manufacturing gets to end users' systems, a hacker can gain the access to the chip by activating the HTs in the chip. Designing and defending against HTs in a chip has been a hot research topic [2, 3, 4, 1, 5, 6]. Various HTs and countermeasures have been proposed [2, 7, 8, 9, 10, 11, 12]. A typical HT is made of the trigger, the Trojan circuit and the payload [1]. Hiding in the many-core chips, HTs often hibernate most of the time and wake up for specific signals or events [13]. Once a specific signal or an event is present, the trigger of an HT first activates, and then the payload circuit launches the attack. Such operational model makes an HT difficult to be detected during the design phase through computer simulations or by off-line testing without explicit knowledge of the specific signals or events that trigger the attacks [14]. Hardware Trojans may engage in different actions, including modifying the functionality or specification of the hardware, leaking sensitive information, or launching denial of service attacks [3, 13, 15, 16].

HTs can be categorized into combinational Trojans and sequential Trojans according to their triggering methods [5]. A combinational Trojan is activated by a set of specific signals, while a sequential Trojan requires a sequence of specific events to trigger its payload. Based on the triggering condition, hardware Trojans can then be classified as logic-based, sensor-based and always-on Trojans [6]. In a logic-based Trojan, a specific binary pattern, say 00110101, in the payload of a data packet is reserved to activate the Trojan. A sensor-based Trojan, on the other hand, will be fired up by reaching certain temperature and power levels as determined by the on-chip sensors. An always-on Trojan, as its name suggests, is up running all the time, and it does not need a trigger.

Current countermeasures against HT attacks can be classified into three categories: HT prevention, HT detection and HT defense [1].

HT prevention is a practice that takes place during the chip design stage to prevent the insertion of HTs in the first place [17, 18].

HT detection relies on various approaches to determine the existence of HTs, and locate them if they do exist. In [19], a sustained vector methodology, where vectors are repeated multiple times at the inputs of both the genuine and the Trojan circuits, was proposed to help detect a Trojan that hides in a chip. Another study provided a proof-of-concept demonstration of the potential benefit of using logical implications for the detection of combinational hardware Trojans [20]. In [21], the authors concerned on the hardware Trojan detection in the network interfaces of networks-on-chip using the state obfuscation.

HT defense is a process that wipes out the HTs entirely, or at least, reduces the attack effects of the HTs. In [22], a method that attempts to detect the presence of Trojans by continuous monitoring and testing of the chip; if a core is found infected with an HT, this infected core will no longer be used and all its computing tasks will be switched over to some other core(s). Another method, path security (P-Sec) validation technique, was proposed to protect compromised networks-on-chip architectures from fault injection side channel attacks [23]. Traffic isolation, a method to reduce the latency incurred by partitioning, also can be used to protect against DoS and bandwidth attacks because of the static time allocation to different domains [24, 25].

2.2. HT-enabled DoS attacks on NoC

HT-assisted DoS attacks [26, 16, 27, 28, 29] can directly target the NoC of a many-core chip, as malfunctioning of NoC can cause the entire chip to be disconnected and disintegrated, even though each single core might still be fully functional. In [26], a bandwidth denial attack that increases the network latency by rejecting the resource request was described, and a detection method referred as RLAN (Runtime Latency Auditor for NoCs) was suggested. In a simple term, RLAN detects the HT when network latency is found abnormal. In [27], a DoS attack in wireless NoCs was launched by reducing normal nodes bandwidth and thus causing widespread bandwidth loss. The authors also proposed a DoS resilient wireless architecture to defend against such an attack, and they also suggested countermeasures that can alleviate the effect of the DoS attack with defense methods at both physical and data routing levels. In [28], various flooding-based DoS attacks were evaluated and the robustness of mesh-based NoC architectures under these attacks was examined. In [29], a target-activated sequential payload (TASP) HT in support of a new type of DoS attack was proposed. To circumvent the threat of HTs, the author proposed a heuristic threat detection model to classify faults and discover HTs within compromised links.

2.3. Blackhole and sinkhole DoS attacks

Blackhole and sinkhole attacks are two of the most severe attacks known for sensor networks [30, 31, 32, 33, 34, 35]. When a blackhole attack is launched, a malicious node captures data from its neighboring nodes and stops forwarding the data packets to their original destinations [33]. Such a malicious node is called as blackhole node and the region that encompasses such a node is known as the region of blackhole. To identify and mitigate the malicious nodes, in [33], route request packets are flooded across the network to create a reliable path between the source and the destination. In a sinkhole attack, the traffic is directed to the hostile node and then many attacks like selective and blackhole can be empowered by a sinkhole attack [35]. A sinkhole attack is shown to be detectable using the Delphi (Delay per Hop Indicator) technique as proposed in [35].

Although there have a lot of studies on blackhole and sinkhole attacks in the context of sensor networks, few consider these attacks in NoC and the design of hardware Trojans needed to launch and sustain such attacks. In [26], a DoS attack targeting the NoC was envisioned, bearing a great deal of similarity to such an attack ever seen in sensor networks. As a matter fact, the attacks described in [26] actually can be easily detected by comparing latencies of similar packets.

In the next sections, we shall exploit how to ensure the invisibility of the attack and assess attack effects. In addition, we shall show that how to maximize the attack effects by tuning a few critical parameters, such as number and distribution of HTs, and a few more.

3. HT Designs for Sinkhole/Blackhole Attacks

In this section, we provide an HT design that enables the sinkhole and blackhole DoS attacks. In Section 3.1, we analyze the configuration that an HT needs and provide details regarding the process of launching an attack. Section 3.2 provides the detailed design of the HT module, and the low degree of detectability of the designed HT is shown in Section 3.3.

An HT can be inserted into the pipeline of a credit-based virtual channel router [36], as shown in Fig. 3. An HT has two parts: (1) the main part, named the main HT, which is used to configure and perform the blackhole

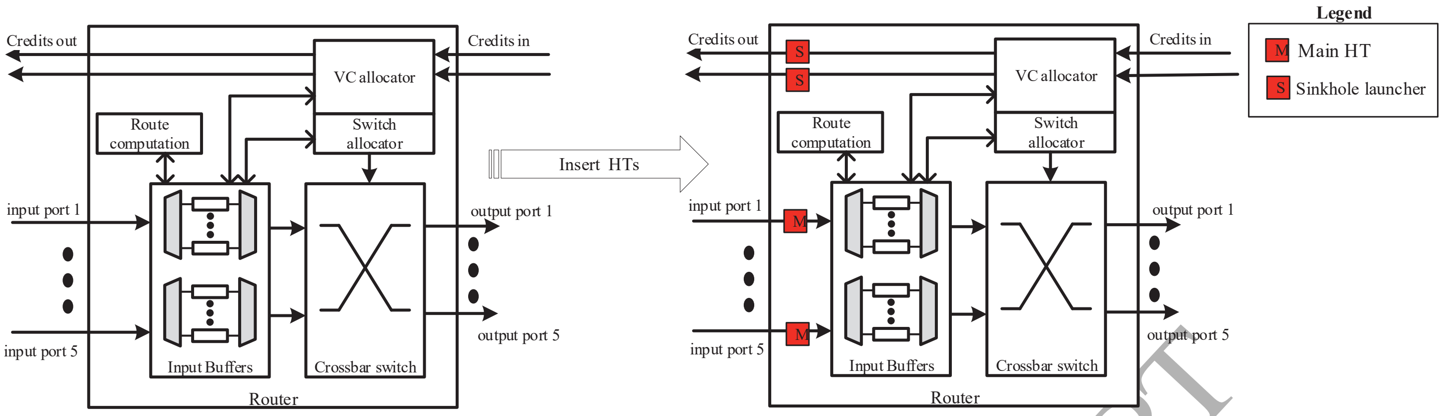


Figure 3: The hardware Trojan's insertion in a credit-based virtual channel router

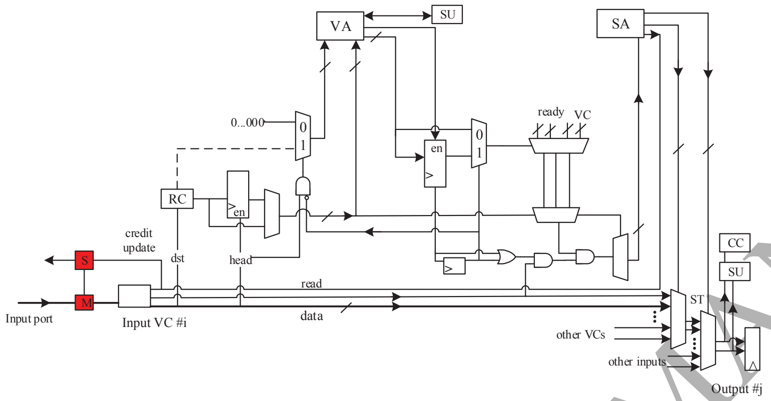


Figure 4: The hardware Trojan's insertion in a single-cycle VC-based router at circuit level

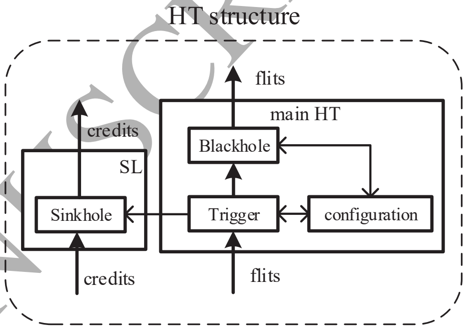


Figure 5: The structure of the proposed HT

attack, denoted as "M" in Fig. 3, and (2) the other part, named sinkhole launcher, that is used to perform sinkhole attack, denoted as "S" in Fig. 3. Each input port need an HT, so there are totally five main HTs and five sinkhole launchers, and each sinkhole launcher is synchronized with its pairing main HT through a control signal. For more details at circuit level, the HT's insertion of one port in a single-cycle VC-based router [37] is shown in Fig. 4.

Shown in Fig. 5, the HT includes a main HT and a sinkhole launcher. The main HT consists of three modules, the trigger, the configuration, and the blackhole function modules. The sinkhole launcher only needs a sinkhole function module. The specific design of hardware Trojan is shown in Fig. 7. Both blackhole and sinkhole attacks are supported by the proposed HT circuit. Based on the trigger mode it adopts, the HT can operate on (a) always on, (b) destination-triggered, or (c) command-triggered modes. An always-on HT is active all the time, while a destination-triggered HT is activated only when the destination of a packet matches the victim ID configured by the hacker. A command-triggered HT is activated by a command generated by the hacker program.

The workflow of an HT is shown as in Fig. 6. Before a hacker launches attack, it first sends a configuration packet to the HT's router and the HT's configuration information is then saved in a set of registers. Once configured, the HTs can check what type of attack shall be launched, and whether or not the attack shall be launched at the time when they engage in packet transmissions.

3.1. *Configuring the HT*

Once the hacker wishes to launch an attack, he/she activates the HTs for a short period of time by running a program that manages to send the configuration packets. The HTs can quickly return to be inactive to reduce the chance to be detected.

First, the hacker needs to locate the malicious node in order to send the configuration packets to its HTs. When the cores communicate through Message Passing Interface (MPI) [38], the hacker can easily gain a specific node's ID. Alternatively, the reverse engineering can be performed to infer the ID of a network node by parsing the cache address of the bank ID [39]. With knowledge of the node ID, topology and routing, the hacker shall be able to locate the HTs.

Second, a hacker or a malicious program can be sneak in as a customer that has the privilege to run on a many-core server compromised by HTs. In this case, the hacker can send configuration packets by either MPI calls (specifying the target node), or access a special memory address

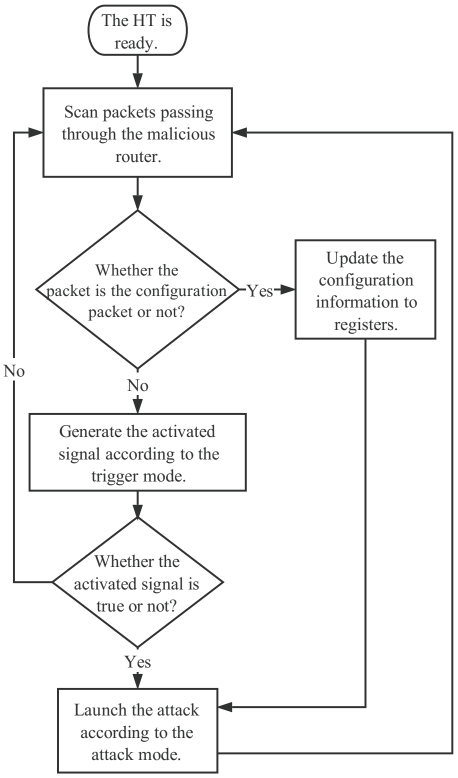

The HT is ready.
Scan packets passing through the malicious router.
Whether the packet is the configuration packet or not?
Yes
Update the configuration information to registers.
No
No
Generate the activated signal according to the trigger mode.
Whether the activated signal is true or not?
Yes
Launch the attack according to the attack mode.

Figure 6: The workflow of the hardware Trojan's configuration and activation

(by reverse engineering [39], the address can be translated to a cache bank ID, corresponding to the target node ID). The malicious program can also multicast (making multiple MPI calls or accessing memory addresses corresponding to multiple cache banks) configuration packets to setup the HTs.

A configuration packet has the following fields:

- Config cmd. Any packet that contains a bit pattern of 00110101 in its cmd field is an HT configuration packet.

- Trigger mode. The trigger mode field specifies the method regarding how the HT is activated, and there are three trigger modes: always-on, command-triggered, and destination-triggered.

- Attack mode. The attack mode field defines the attack type the HT shall launch; in this study, three attack types are supported, namely the blackhole attack, sinkhole attack, or both.

- Activation signal (flag). When this field is asserted, the HT is activated. Otherwise, the HT is supposed to be inactive.

- Victim ID. The victim ID specifies the packet destination address that shall trigger the HT, when HT is in the destination-triggered mode.

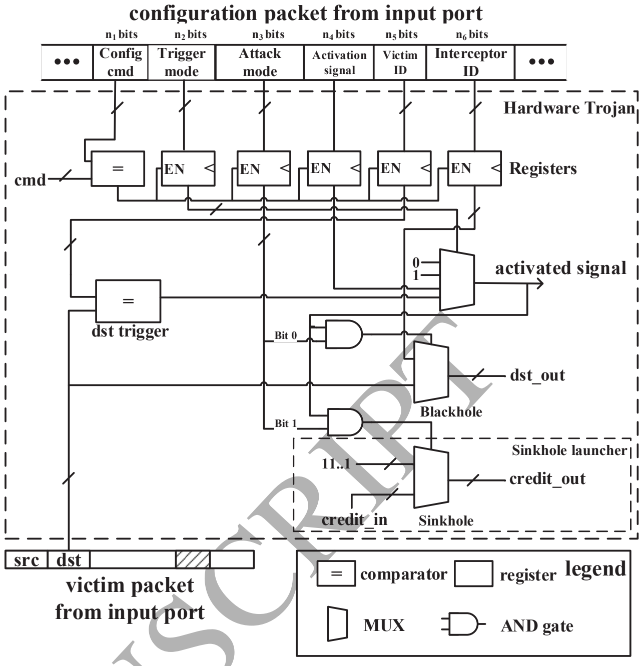


Figure 7: The specific design of a hardware Trojan

- Interceptor ID. When the HT launches the blackhole attack, every data packet passing through the HT will be diverted to the malicious node that has an address specified in this field.

Each HT in malicious routers has a set of registers corresponding to the above packet fields. At any malicious router in the NoC, all the incoming data packets will be scanned. If a packet is determined as a Config cmd packet, the configuration registers in the HT are updated based on the payload of the Config cmd packet received.

3.2. Launching the attack

Table 1: The optional modes and their relevant parameters

Operation options		Relevant parameters
Trigger mode	Always on	Null
	Destination	Packet's destination and victim ID
	Command	Activation signal
Attack mode	Blackhole	Packet's destination and interceptor ID
	Sinkhole	Router's credit

Once an HT has been configured, it can be activated by the trigger module according to its triggering mode. The operational modes and their relevant settings are shown in Table 1. Once the HT is activated, the function module is responsible for sustaining the blackhole attack, sinkhole attack or both. In a blackhole attack, the destination of the packet is replaced with the address of a preset malicious node or an invalid address. In a sinkhole attack, the routing resource (e.g., free buffer slot number in the input ports) is modified to claim a higher value. By doing so, the malicious router can attract the data packets from its neighboring routers to be routed to itself.

The procedure of launching the attack in the HT is as follows:

step 1: The hacker sends a configuration packet to a malicious node.

step 2: The HT in the malicious node receives the configuration packet and updates its configuration information, after which the HT starts operating.

step 3: The trigger module of the HT selects its trigger mode based on the data stored in the trigger mode register.

step 4: The activated signal is generated from the trigger module according to the select trigger modes: (a) for the always on mode, the activated signal is asserted all the time; (b) for the command-triggered mode, the activated signal is set to true when the activation signal register holds the activation signal; and (c) for the destination-triggered mode, the activated signal is set to be true when the destination field of each packet matches that saved in the victim ID register.

step 5: Once the activated signal is found to be true, the function module of the HT starts to launch the attack.

step 6: Function module of the HT selects its attack mode (blackhole or sinkhole attacks, or both) from the attack mode register: (a) for the blackhole mode, the interceptor's ID replaces the original destination's address as the output; and (b) for the sinkhole mode, instead of using the true credit, the output is falsely given a higher value.

3.3. Stealthiness analysis

The proposed HT is quite hard to be detected due to two reasons:

First, the HT has a low silicon footprint and consumes low energy to operate. The blackhole and sinkhole attacks can be launched by the same HT circuit. Implemented using a 45nm TSMC technology, every single HT has an area of $17.46\mu m^2$, and consumes just $2.21\mu W$ power (from the synthesis result generated by the Synopsys Design Compiler^{TM}). That is, all the five HTs in one router actually adds an area of $87.30\mu m^2$ and consumes additional $11.05\mu W$ of power. For comparison, a router of modest size with four VCs and 4-flit depth FIFO has a total area of $130538\mu m^2$ and consumes $56mW$ power. One can see that the HTs' power and area overheads are extremely low, only 0.07% and 0.02% of an NoC router, respectively. As far as the delay is concerned, the HT-inserted router circuit sees a delay increase of less than $0.06ns$, which is 6% of a single-cycle router running at a frequency of $1GHz$. The offline testing methods or side channel fingerprint method [40] for NoC may find it extremely difficult to detect a HT of such a small size.

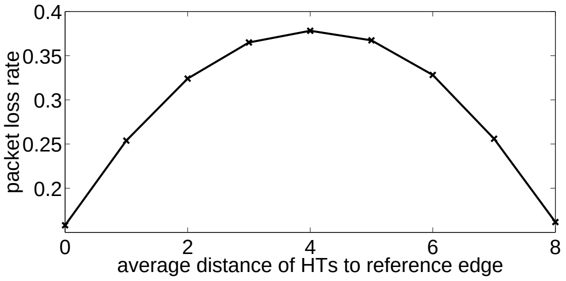


Figure 8: The impact of average distance of HTs to the reference edge on the blackhole attack effect, measured as packet loss rate.

Second, the HT is conditionally triggered. On the one hand, the HT is controlled by the hacker program. Without explicit knowledge of the format of the configuration packet, it is hard to activate and detect the HT. Furthermore, the HT is triggered only when the destination of the packet is the victim and the activation signal is asserted, which makes the similar package detection method or similar destination detection method [26] simply not work.

4. Evaluation of the DoS Attack Effects

In this section, the attack effects of the proposed DoS attacks are evaluated. We measure the attack effect through packet loss rate, defined as the ratio of the number of packets that cannot reach their destinations over the total number of packets transmitted. Attack effects due to HT distribution, and application traffic characteristics are determined in this section as well.

4.1. Factors relevant to the effects of DoS attacks

In this section, we evaluate the attack effects against different parameters and determine the conditions that can maximize the attack effects. Fig. 8 shows how effects of blackhole attack vary with the distribution of HTs, which is measured by the average distance of HTs to the reference edge (selected to be the chip bottom edge in this paper) and the average pairwise distance of the HTs in a 9 × 9 mesh NoC. One can see when the average distance of HTs to the reference edge is 4, the attack effect reaches its peak. Generally speaking, we use two measures, the average distance of HTs to the reference edge (x_3) and the average pairwise distance of the HTs (x_4), to characterize the HT distribution. Besides the distribution of the HTs, attack effects of the DoS are found to be also related to the following parameters.

- Number of HTs (x_1).
- Average path length of the packets (x_2). This is the number of routers in the path from a given source to a destination.

- Number of victims (x_5).
- Victim distribution. We use two variables, the average distance of victims to the reference edge (x_6) and the average pairwise distance of the victims (x_7), to characterize the victim distribution.
- Number of packets (x_8). Number of packet is the total number of packets that are injected into the network in a cycle.
- Number of hotspots (x_9). In the NoC, some nodes that experience much more traffic than other nodes are referred as hotspots. In this work, a node receives more than 30% packets per cycle on average is defined to be a hotspot.
- Hotspot distribution. We use two variables, the average distance of hotspots to the reference edge (x_{10}) and the average pairwise distance of the hotspots (x_{11}), to characterize the hotspot distribution.

Among all the measures and parameters listed above, x_1, x_3, x_4, x_5, x_6, x_7 are the ones that can be set by the attacker, while all the others are considered as the system parameters of the NoC, and they are out of the attacker's reach. In addition, parameters x_1, x_2, x_3, x_4 are found to have higher impact than the others in defining the attack effectiveness.

4.2. Modeling the attack effects

The blackhole and sinkhole attacks have significant implications on system performance in terms of network latency, packet loss, and power consumption. As the direct result of an attack is to stop the flow of data packets, the attack effect is directly measured by packet loss rate, y,

$$y = \frac{n}{N} \tag{1}$$

where n is the number of lost packets caused by the attacks, and N is the total number of packets in NoC.

The relationship between the packet loss rate and the input parameters can be modeled as follows,

$$y = \sum_{i=1}^{11} \sum_{j=0}^{p} a_{ij} \times x_i^j \tag{2}$$

where a_{ij}'s are the regression coefficients of the polynomial regression model of p-th order, which can be computed by the maximum likelihood method [41].

4.3. Maximizing the attack effects

Following the attack effect model described above, hackers can maximize the attack effects by solving an optimization problem with an upper limit set by the number of HTs. This problem is formally described as follows:

Given a total of N HTs, find the best HT distribution to maximize the attack effect modeled in Eq. 2. That is,

$$\max\ y \tag{3}$$

subject to

$$x_1 \leq N \tag{4}$$

where N is the upper limit of HT number.

This problem can be solved through an exhaustive search of the values of x_1, x_3, x_4 to find the best value for y.

4.4. HT insertion methodology

A malicious agent can try to maximize the attack effect by solving Eqs. 3 and 4. After the best HT distribution is found, the malicious agent may manage to insert the HTs as so determined to the SoC chip before it gets manufactured.

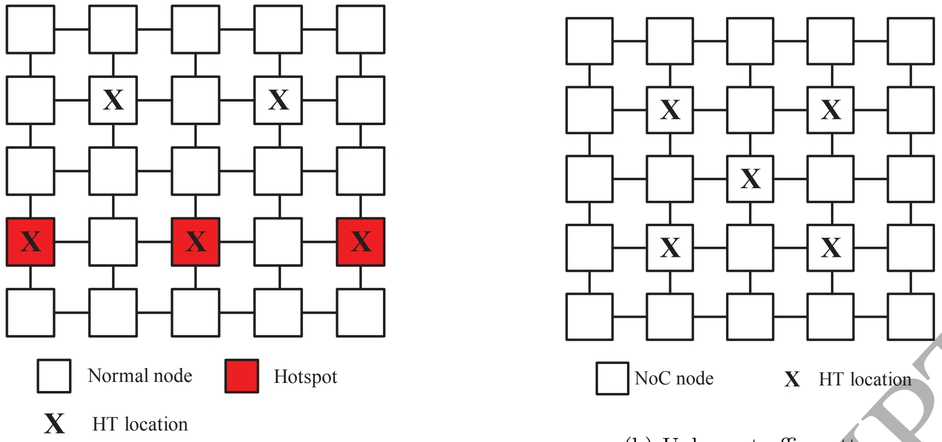
The HT insertion methodology mainly considers two distinct cases: (a) known traffic pattern, and (b) unknown traffic pattern.

- Known traffic pattern. If the malicious agent knows the traffic pattern and the hotspots' locations (for example, from profiling or logging of previous versions of chips), he/she first inserts a subset of HTs close to these hotspots. Then the remaining HTs will be modified so that the HTs' locations match the best HT distribution. For example, Fig. 9(a) shows a 5 × 5 mesh NoC whose traffic pattern is known and it has three hotspots. Assume the malicious agent can insert at most five HTs into the NoC and we have the freedom to select their locations. First we calculate the best HT distribution with $x_3 = 1.8$ and $x_4 = 2.2$. Next we place the three HTs in the locations of the hotspots and the remaining HTs far from the hotspots as shown in Fig. 9(a), which satisfies the exact values of x_3 and x_4 as specified. By setting these parameters, its is guaranteed that the attack effect is maximized.
- Unknown traffic pattern. If the malicious agent doesn't have a priori knowledge of the traffic pattern, we will have to assume a uniform distribution for the traffic. Correspondingly, the number of hotspots is set to be zero. In this case, the HTs are uniformly placed across the chip. For example, Fig. 9(b) shows the same NoC as Fig. 9(a), but in this case, the malicious agent doesn't know the traffic pattern. The best HT distribution is found to be $x_3 = 2$ and $x_4 = 2$. As a result, the malicious agent would place the five HTs uniformly, as shown in Fig. 9(b).

5. Detection and Defense

5.1. The detection method

To detect the HTs, the global manager injects special detection request packets into the network nodes periodically. The detection method is shown as Fig. 10. The



(a) Know traffic pattern

(b) Unknow traffic pattern

Figure 9: The scenario of site selection

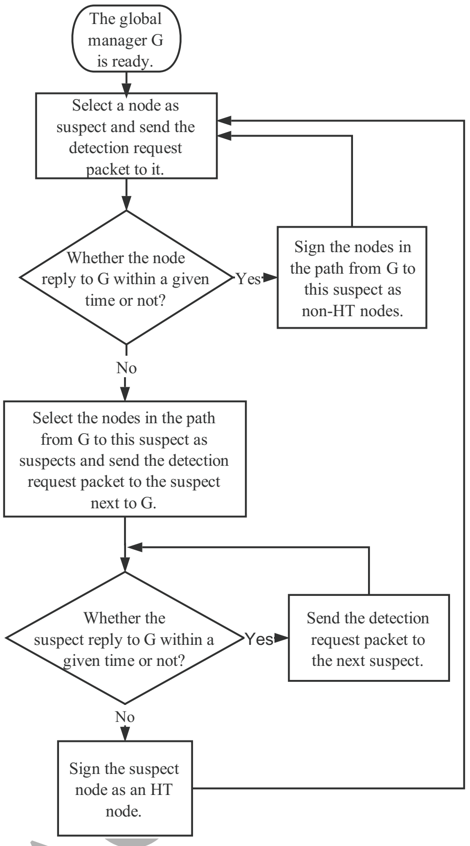

The global manager G is ready.
Select a node as suspect and send the detection request packet to it.
Whether the node reply to G within a given time or not?
Yes
Sign the nodes in the path from G to this suspect as non-HT nodes.
No
Select the nodes in the path from G to this suspect as suspects and send the detection request packet to the suspect next to G.
Whether the suspect reply to G within a given time or not?
Yes
Send the detection request packet to the next suspect.
No
Sign the suspect node as an HT node.

Figure 10: The workflow of the detection method

global manager G first randomly selects a node, and then sends detection request packets to it. If this node replies to G within a given time, all the nodes on the path are believed to be HT-free. Otherwise, if the reply time of this select node exceeds a threshold, or there is no reply from the node at all, the nodes on the path from the node to G are further checked to find the exact location of the HT-infected router. This simple method can only detect the HTs with always on triggered mode, but it will fail to detect conditionally triggered HTs. The HTs are only active for a certain amount of time, typically short, and when the HTs are inactive, the global manager will not be able to find any anomalies, and this detect method fails to find the HTs.

5.2. *The defense method*

Once the HT routers have been detected, a detour routing method can be applied, where packets will be routed along the paths that do not include the routers infected with HTs. In this case, each router needs to keep a local record of all its neighboring nodes regarding whether they are legitimate or malicious. Once a malicious node is detected, the global manager G broadcasts this information to all the clean routers, the ones that are not infected by HTs. At the same time, the routers whose downstream node is identified as a malicious one save the locations of the malicious nodes into their own local storage. Once a packet is to be forwarded by a router, this router will have to look at its record to see if the downstream router is a malicious one or not. If not, the packet will be routed as it should be. Otherwise, it finds another output channel and checks again. The west-first turn model [42] can be applied to avoid any routing deadlock.

6. Experimental Evaluation

In this section, we first run experiments to (i) determine the appropriate regression model presented at Section 4.2 for the attack effects, (ii) determine the parameters that impact the effectiveness of attacks, (iii) evaluate the attack effects and (iv) evaluate the defense methods.

6.1. *Experimental setup*

Experiments are performed using POPNET, an event-driven NoC simulator. Table 2 shows the specific simulator configuration. In the follow experiments, we select a 9×9

Table 2: Configuration used in the simulation

NoC size	9×9
Number of routers	81
NoC flit size	72-bit
NoC latency	router 1 cycle, link 1 cycle
NoC input buffer depth	4
NoC VC number	4
Routing algorithm	XY routing, adaptive routing, west-first turn based bypass routing

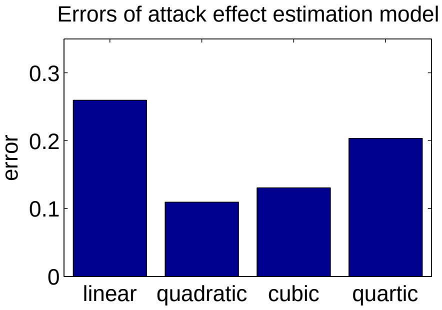


Figure 11: Errors of the attack effect estimation models

2D mesh as the underline NoC topology with a total of 81 routers.

In particular, the simulations with different network sizes, 8×8, 9×9, 16×16, are also performed, and their results are very similar to what is obtained in a network of smaller size, in terms of attack model, attack effects, and defense effects. Here we pick a median-sized network (of size 9×9) to demonstrate the results. We adopt three routing algorithms in our experiments. XY routing is employed when we evaluate the effects of blackhole attack in Sections 6.3 and 6.4, while adaptive routing is used to evaluate the effects of sinkhole attacks. When we evaluate the defense effect of the proposed defense method in Section 6.5, bypass routing based on west-first turn is employed.

6.2. Evaluation of the errors of the model

We compare the errors of the polynomial regression models with different orders in Eq. 2. The error of the model is defined as,

$$\varepsilon_a = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{5}$$

where m is number of experiments, y_i and $\hat{y}_i$ are the packet loss rates obtained from the simulation and calculated by the model in Eq. 2, respectively. From Fig. 11, one can see that the quadratic regression model gives the lowest level of error, and its correlation coefficient is about 0.86. In the following, we will use this quadratic regression model for the estimation of attack effects.

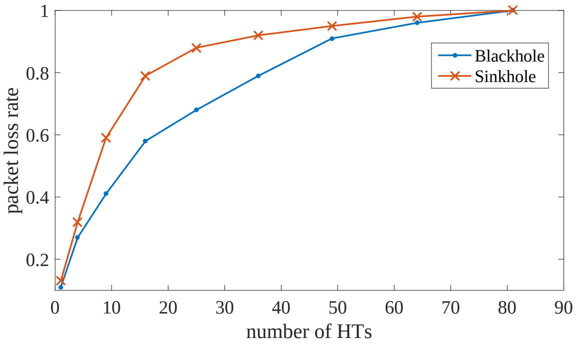


Figure 12: The number of HTs vs. packet loss rate for blackhole and sinkhole attacks

6.3. Analysis of the factors impacting attack effects

6.3.1. Number of HTs

In this set of experiments, we set the average distance of HTs to reference edge (x_3) to be 3, the average pairwise distance of HTs (x_4) to be 2, and x_2 to be 6. Fig. 12 shows the relationship between the number of HTs (x_1) and the packet loss rate for blackhole and sinhole attacks. For blackhole attack, one can see from Fig. 12 that when the number of HTs is less than 16, the packet loss rate increases rapidly with the increase of the number of HTs. When the number of HTs is over 16, the increase of packet loss rate slows down significantly and gradually approaches to 1. When the number of HTs reaches 81, i.e., all the 81 router nodes have their own HT, the packet loss rate reaches 100% as predicted. The sinkhole attack performs the similar result as blackhole attack and it's attack effect is better.

6.3.2. HT distribution

We set x_1 to be 5, x_2 to be 6, and x_4 to be 2. Fig. 13 shows the relationship between the average distance of HTs (with respect to the reference edge (x_3)) and the packet loss rate. When x_3 is less than 4, the packet loss rate increases with the increase of x_3, and reaches the peak when x_3 is equal to 4. When x_3 is greater than 4, the packet loss rate starts to decrease. The reason is that when x_3 is very small (e.g., 1 or 2) or very large, (e.g., 7 or 8), HTs are placed close to the edges of network; when x_3 is around 4, HTs are near the center of network. As the network traffic shows a uniform traffic pattern, the traffic volume around the center nodes are higher than that of nodes far from the center. As a result, the HTs located at or near the center nodes tend to cause more severe damages to the system. The sinkhole attack performs the similar result as blackhole attack and it's attack effect is better.

Next we set x_3 to be 3 and vary x_4. Fig. 14 shows the relationship between the average pairwise distance of HTs (x_4) and the packet loss rate. One can see that when x_4 is less than 3, the packet loss rate increases gradually and reaches the peak when $x_4 = 3$; when x_4 is greater than

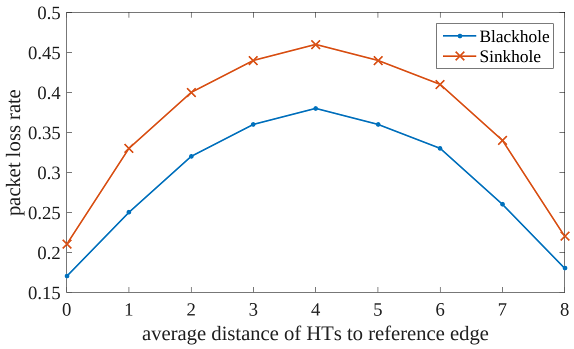


Figure 13: The average distance of HT to the reference edge vs. packet loss rate for blackhole and sinkhole attacks

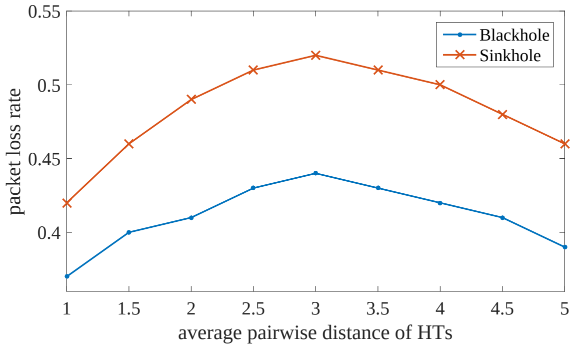


Figure 14: The average pairwise distance of HTs vs. packet loss rate for blackhole and sinkhole attacks

3, the packet loss rate declines. The reason is that when x_4 is small, the HTs are in close proximity. When HTs are scattered and are far away from each other, they may be close to the chip corners. In both cases, fewer packets travel through the HTs, and thus, fewer packets get dropped from the system. The sinkhole attack performs the similar result as blackhole attack and it's attack effect is better.

6.3.3. *Path length of the packets*

In this set of experiments, we set x_1 to be 5, x_3 to be 3, and x_4 to be 2. Fig. 15 shows the relationship between the average path length of packets (x_2) and the packet loss rate. From the Fig. 15, as the average path length increases, the packet loss rate increases in a nearly linear fashion. The reason is that, when the path length is long, there is a high probability that a packet may pass through a node infected with HT. The sinkhole attack performs the similar result as blackhole attack and it's attack effect is better.

6.4. *Evaluating the attack effects after optimization*

We compare the attack effects of the optimized HT distribution by solving the optimization problem in Eqs. 3 and 4 (*e.g., with optimization*) and that of a random HT distribution (*i.e., without optimization*). In this set of experiments, the number of HTs varies from 1 to 6. The average path length of the packet is set to be 2, 4, 8, respectively. Fig. 16(a)-(c) show the blackhole attack effects with and without optimization. The results are normalized to the case that the optimization is applied. One can see that the optimized HT distribution improves the attack effect by 30% compared to a random HT distribution, on average. From Fig. 16(a), when the average path length is 2 and the number of HTs is 1, the packet loss rate of the optimized HT distribution is 24% higher than that of the random HT distribution. From Fig. 16(c), when the average path length is 8 and the number of HTs is 3, the packet loss rate of the optimized HT distribution is 36% higher than that of the random HT distribution.

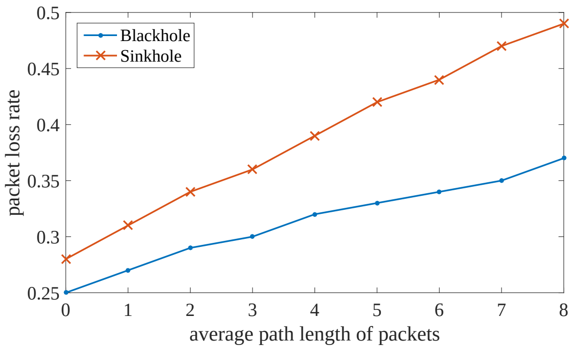


Figure 15: The average path length of packets vs. packet loss rate for blackhole and sinkhole attacks

Fig. 17(a)-(c) compare the sinkhole attack effect with and without optimization. One can see that the optimized HT distribution improves the attack effect by 34% compared to the random HT distribution, on average. From Fig. 17(a), when the average path length is 2 and the number of HTs is 4, the packet loss rate of the optimized HT distribution is 25% higher than that of the random HT distribution. From Fig. 17(c), when the average path length is 8 and the number of HTs is 1, the packet loss rate of the optimized HT distribution is 41% higher than that of the random HT distribution.

6.5. *Evaluating the defense methods*

In this set of experiments, the number of chosen suspects varies from 2 to 10. The results are normalized to that of the case without applying any defense. Fig. 18(a) shows the packet loss rate caused by blackhole attack without and with defense measures deployed. One can see that packet loss rate in a system with defense employed is reduced by 42% on average. Even with more nodes being selected as the suspects and checked, the packet loss rate is still quite high, at 52% of the original. This is because when the HT changes its attack target, the detection method is no longer considered effective.

Fig. 18(b) shows the packet loss rates caused by sinkhole attack in both cases without and with defense. One can see that the packet loss rate of the case that employs defense is reduced by 39% on average. Even with more

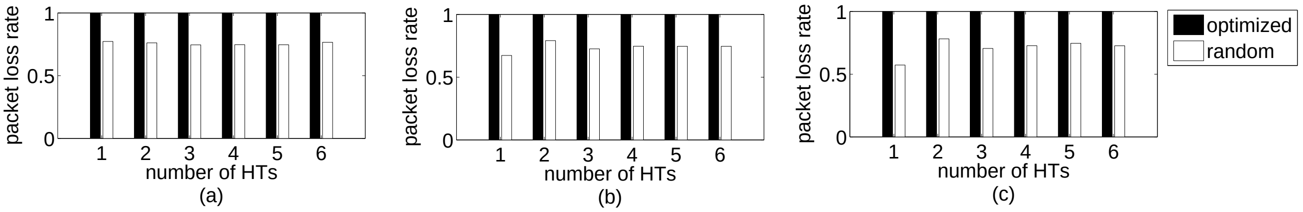


Figure 16: The packet loss rate comparison between the optimal HT distribution and random HT distribution for blackhole attack when average packet path length is (a) 2, (b) 4, (c) 8.

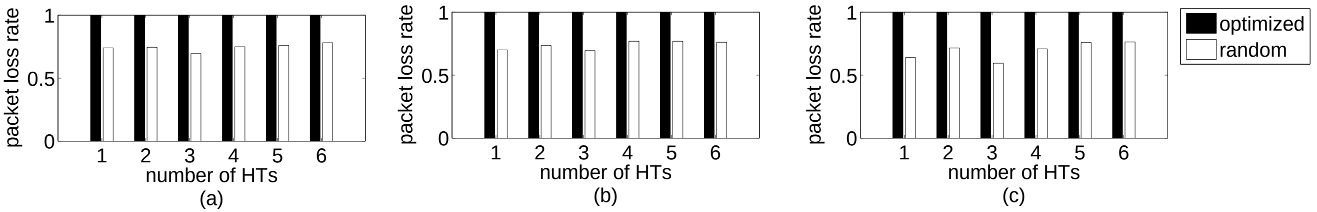


Figure 17: The packet loss rate comparison between the optimal HT distribution and random HT distribution for sinkhole attack when average packet path length is (a) 2, (b) 4, (c) 8.

nodes being selected as the suspects, the packet loss rate remains high, at 57% of the original. In a simple word, the effectiveness of the current detection and defense against the enhanced blackhole and sinkhole attacks, as described in Section 5, tends to be quite limited.

As alluded before, the configuration module can help change the trigger conditions. When the attack target and attack interval are changed, the detection method makes little effect because they fail to trigger the HTs.

7. Conclusion

Blackhole/sinkhole DoS attacks targeting the NoC systems of many-core chips can cause severe packet losses and/or divert traffic to malicious nodes other than their intended designations. In this paper, the effects of the attacks as measured by packet loss rate, were quantitatively modeled by considering several critical parameters, including number of HTs and their distribution in NoC. Through fine-tuning of these parameters, both attacks are shown to cause more damages to NoC, with the packet loss rate jumped by more than 30%. Even with detection and defense methods in place, the packet loss rate can still reach 52%. This research indicates a strong need to develop more effective countermeasures to thwart these enhanced attacks.

References

[1] H. Li, Q. Liu, J. Zhang, A survey of hardware Trojan threat and defense, Integration, the VLSI Journal 55 (2016) 426–437.

[2] D. M. Shila, V. Venugopal, Design, implementation and security analysis of hardware Trojan threats in FPGA, in: Proc. IEEE Int'l Conf. Communications, 2014, pp. 719–724.

[3] R. S. Chakraborty, S. Narasimhan, S. Bhunia, Hardware Trojan: Threats and emerging solutions, in: Proc. IEEE Int'l High Level Design Validation and Test Workshop, 2009, pp. 166–171.

[4] Y. Jin, N. Kupp, Y. Makris, Experiences in hardware Trojan design and implementation, in: Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust, 2009, pp. 50–57.

[5] S. Bhunia, M. S. Hsiao, M. Banga, S. Narasimhan, Hardware Trojan attacks: threat analysis and countermeasures, IEEE J. Proc. IEEE 102 (8) (2014) 1229–1247.

[6] M. Tehranipoor, F. Koushanfar, A survey of hardware Trojan taxonomy and detection, IEEE Design Test of Computers 27 (1) (2010) 10–25.

[7] Y. Jin, Y. Makris, Hardware Trojan detection using path delay fingerprint, in: Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust, 2008, pp. 51–57.

[8] X. Wang, H. Salmani, M. Tehranipoor, J. Plusquellic, Hardware trojan detection and isolation using current integration and localized current analysis, in: Proc. IEEE Int'l Symp. Defect and Fault Tolerance VLSI Systems, 2008, pp. 87–95.

[9] M. R. Kakoee, V. Bertacco, L. Benini, A distributed and topology-agnostic approach for on-line NoC testing, in: Proc. ACM/IEEE Int'l Symp. Networks-on-Chip, 2011, pp. 113–120.

[10] S. Bhasin, F. Regazzoni, A survey on hardware trojan detection techniques, in: Proc. IEEE Int'l Symp. Circuits and Systems, 2015, pp. 2021–2024.

[11] B. Cha, S. K. Gupta, A resizing method to minimize effects of hardware Trojans, in: Proc. IEEE Symp. Asian Test, 2014, pp. 192–199.

[12] M. Banga, M. S. Hsiao, A region based approach for the identification of hardware Trojans, in: Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust, 2008, pp. 40–47.

[13] J. H. G. S. N Jacob, D Merli, Hardware Trojans: current challenges and approaches, IET Computers and Digital Techniques 8 (6) (2014) 264–273.

[14] S. M. J. M.Hicks, M.Finnicum, Overcoming anuntrusted computing base: detecting and removing malicious hardware automatically, in: Proc. IEEE Symp. Security and Privacy, 2010, pp. 159–172.

[15] C. Reinbrecht, A. Susin, L. Bossuet, G. Sigl, J. Sepuveda, Side channel attack on NoC-based MPSoCs are practical: NoC Prime+Probe attack, in: Proc. Symp. Integrated Circuits and Systems Design(SBCCI), 2016, pp. 1–6.

[16] A. Malekpour, R. Ragel, A. Ignjatovic, S. Parameswaran, DoS-Guard: Protecting pipelined MPSoCs against hardware Trojan based DoS attacks, in: Proc. IEEE Int'l Conf. Application-specific Systems, Architectures and Processors(ASAP), 2017, pp. 45–52.

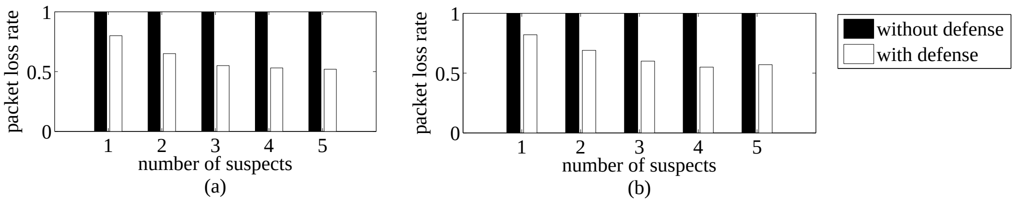


Figure 18: Comparison of the packet loss rate with and without defense methods, for the (a) blackhole attack, and (b) sinkhole attack.

[17] Q. Yu, J. Dofe, Z. Zhang, Exploiting hardware obfuscation methods to prevent and detect hardware Trojans, in: Proc. IEEE Int'l Midwest Symposium on Circuits and Systems (MWSCAS), 2017, pp. 819–822.

[18] K. Xiao, D. Forte, M. Tehranipoor, A Novel Built-In Self-Authentication Technique to Prevent Inserting Hardware Trojansg, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 33 (12) (2014) 1778–1791.

[19] H. M. S. Banga M, A novel sustained vector technique for the detection of hardware Trojans, in: Proc. IEEE Int'l Conf. VLSI Design, 2009, pp. 327–332.

[20] N. Cornell, K. Nepal, Combinational hardware Trojan detection using logic implications, in: Proc. IEEE Int'l Midwest Symposium on Circuits and Systems (MWSCAS), 2017, pp. 571–574.

[21] J. Frey, Q. Yu, Exploiting state obfuscation to detect hardware trojans in NoC network interfaces, in: Proc. Int'l Midwest Symp. Circuits and Systems, 2015, pp. 1–4.

[22] A. Malekpour, R. Ragel, A. Ignjatovic, S. Parameswaran, TrojanGuard: Simple and effective hardware Trojan mitigation techniques for Pipelined MPSoCss, in: Proc. Design Automation Conference (DAC), 2017, pp. 1–6.

[23] T. Boraten, A. K. Kodi, Packet security with path sensitization for NoCs, in: Proc. Design, Automation Test Europe Conf. Exhibition, 2016, pp. 1136–1139.

[24] H. M. G. Wassel, Y. Gao, J. K. Oberg, T. Huffmire, R. Kastner, F. T. Chong, T. Sherwood, SurfNoC: A low latency and provably non-interfering approach to secure networks-on-chip, in: Proc. Symp. Int'l Symp. Computer Architecture(ISCA), 2013, pp. 296–310.

[25] A. Psarras, J. Lee, I. Seitanidis, C. Nicopoulos, G. Dimitrakopoulos, PhaseNoC: Versatile network traffic isolation through TDM-Scheduled virtual channels, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 35 (5) (2016) 844–857.

[26] R. JS, D. M. Ancajas, K. Chakraborty, S. Roy, Runtime detection of a bandwidth denial attack from a rogue network-on-chip, in: Proc. Int'l Symp. Networks-on-Chip, 2015, pp. 8:1–8:8.

[27] A. Ganguly, M. Y. Ahmed, A. Vidapalapati, A denial-of-service resilient wireless NoC architecture, in: Proc. the Great Lakes Symp. VLSI, 2012, pp. 259–262.

[28] D. Fang, H. Li, J. Han, X. Zeng, Robustness analysis of mesh-based network-on-chip architecture under flooding-based denial of service attacks, in: Pro. IEEE Int'l Conf. Networking, Architecture and Storage, 2013, pp. 178–186.

[29] T. Boraten, A. K. Kodi, Mitigation of denial of service attack with hardware Trojans in NoC architectures, in: Proc. IEEE Int'l Symp. Parallel and Distributed Processing, 2016, pp. 1091–1100.

[30] C. Karlof, D. Wagner, Secure routing in wireless sensor networks: attacks and countermeasures, Ad Hoc Networks 1 (2) (2003) 293–315.

[31] D. R. Raymond, S. F. Midkiff, Denial-of-service in wireless sensor networks: Attacks and defenses, IEEE Pervasive Computing 7 (1) (2008) 74–81.

[32] A.-S. K. Pathan, H.-W. Lee, C. S. Hong, Security in wireless sensor networks: issues and challenges, in: Proc. Int'l Conf. Advanced Communication Technology, Vol. 2, 2006, pp. 6–pp.

[33] H. Kaur, A. Singh, Identification and mitigation of black hole attack in wireless sensor networks, in: Proc. Int'l Conf. Micro-Electronics and Telecommunication Engineering (ICMETE), 2016, pp. 616–619.

[34] M. U. Farooq, X. Wang, R. Yasrab, S. Qaisar, Energy preserving detection model for collaborative black hole attacks in wireless sensor networks, in: Proc. Int'l Conf. Mobile Ad-Hoc and Sensor Networks (MSN), 2016, pp. 395–399.

[35] M. Kaur, A. Singh, Detection and mitigation of sinkhole attack in wireless sensor network, in: Proc. Int'l Conf. Micro-Electronics and Telecommunication Engineering (ICMETE), 2016, pp. 217–221.

[36] L.-S. P. Natalie Enright Jerger, Tushar Krishna, On-chip networks, second edition, Synthesis Lectures on Computer Architecture, 2017.

[37] S. I. Dimitrakopoulos G, Psarras A, Microarchitecture of network-on-chip routers, Springer, 2015.

[38] S. H.-L. M Snir, S Otto, J. Dongarra, MPI–the complete reference: the MPI core, MIT press, 1998.

[39] M. Kayaalp, N. Abu-Ghazaleh, D. Ponomarev, A. Jaleel, A high-resolution side-channel attack on last-level cache, in: Proc. Design Automation Conference (DAC), 2016, pp. 72:1–72:6.

[40] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, Trojan detection using IC fingerprinting, in: Proc. Symp. Security and Privacy(SP), 2007, pp. 296–310.

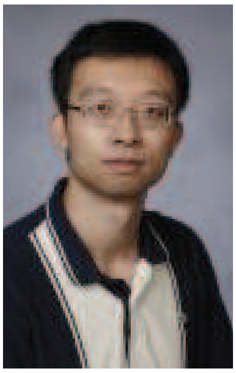
[41] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, R. Tibshirani, The elements of statistical learning, Springer, 2009.

[42] C. J. Glass, L. M. Ni, The Turn model for adaptive routing, in: Proc. Ann. Int'l Symp. Computer Architecture, 1992, pp. 278–287.

Biography

Li Zhang received the bachelor's degree in software engineering from South China University of Technology, Guangzhou, China. He is working toward the master's degree in the school of software engineering, South China University of Technology. His research interests include hardware security, and NoC-based systems.

Xiaohang Wang received the B.Eng. and Ph.D degree in communication and electronic engineering from Zhejiang University, in 2006 and 2011. He is currently an associate professor at South China University of Technology. He was the receipt of PDP 2015 and VLSI-SoC 2014 Best Paper Awards. His research interests include many-core architecture, power efficient architectures, optimal control, and NoC-based scystems.



Yingtao Jiang joined the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas in Aug. 2001, upon obtaining his Ph.D degree in Computer Science from the University of Texas at Dallas. He has been a full professor since July 2013 at the same university, and now assumes the role of the Department Chair. His research interests include algorithms, computer architectures, VLSI, networking, nano-technologies, etc.

Mei Yang received her Ph. D. in Computer Science from the University of Texas at Dallas in Aug. 2003. She has been a full professor in the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas since 2016. Her research interests include computer architectures, networking, and embedded systems.

Terrence Mak is an Associate Professor at Electronics and Computer Science, University of Southampton. Supported by the Royal Society, he was a Visiting Scientist at Massachusetts Institute of Technology during 2010, and also, affiliated with the Chinese Academy of Sciences as a Visiting Professor since 2013. Previously, He worked with Turing Award holder Prof. Ivan Sutherland, at Sun Lab in California and has awarded Croucher Foundation scholar. His newly proposed approaches, using runtime optimisation and adaptation, strengthened network reliability, reduced power dissipations and significantly improved overall on-chip communication performances. Throughout a spectrum of novel methodologies, including regulating traffic dynamics using networks-on-chip, enabling unprecedented MTBF and to provide better on-chip efficiencies, and proposed a novel garbage collections methods, “defragmentation”, together led to three prestigious best paper awards at DATE 2011, IEEE/ACM VLSI-SoC 2014 and IEEE PDP 2015, respectively. More recently, his newly published journal based on 3D adaptation and deadlock-free routing has awarded the prestigious 2015 IET Computers & Digital Techniques Premium Award. He has published more than 100 papers in both conferences and journals and jointly published 4 books.

Amit Kumar Singh received the B.Tech. degree in Electronics Engineering from Indian Institute of Technology (Indian School of Mines), Dhanbad, India, in 2006, and the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University(NTU), Singapore, in 2013. He was with HCL Technologies, India for year and half before starting his PhD at NTU, Singapore, in 2008. He worked as a post-doctoral researcher at National University of Singapore (NUS) from 2012 to 2014 and at University of York, UK from 2014 to 2016. Currently, he is working as senior research fellow at University of Southampton, UK. His current research interests include system level design-time and run-time optimizations of 2D and 3D multi-core systems with focus on performance, energy, temperature, and reliability. He has published over 45 papers in the above areas in leading international journals/conferences.