Modeling and Analysis of Confluence Attack by Hardware Trojan in NoC

Sachin Bagga[1], Ruchika Gupta[1,2], and John Jose[2]

[1] Chandigarh University, Mohali, Punjab, India
sachin8510@gmail.com
[2] Indian Institute of Technology Guwahati, Assam, India
gupt009@rnd.iitg.ac.in
johnjose@iitg.ac.in

Abstract. Over the years, System-on-Chip (SoC) designs have evolved extensively sophisticated in order to fulfill the need of increasing complexity of running applications driven by the advancement of VLSI technology. The next generation multiprocessor system on chip (MPSoC) integrates hundreds of processing elements on a single chip, expected to achieve high performance, low latency, and low power consumption. Tiled Chip Multicore Processors (TCMP) with Network-on-Chip (NoC) have become a foundation for the computation critical embedded and real time systems. Faster time-to-market restraint and business competition compelled manufacturers to look for the prospects of manufacturing SoCs integrated with various third party Intellectual Property (3PIP). Usage of 3PIP gave rise to exploit the underlying interconnect while adding some unwanted malicious circuit known as Hardware Trojan (HT), making NoC vulnerable to get attacked. A tiniest manipulation of any communication attribute by HT can degrade the overall behaviour of the system significantly while impacting NoC performance metrics. In this paper, we present one of such study considering HT malign behaviour of manipulating the output port of each incoming flit from specific ports once genuine route computation takes place and re-directs all to one port causing disruption in tile communication.The proposed HT is intermittent in nature and activates for few cycles. We study the behaviour of proposed confluence attack and analyse its impact over network level. The empirical evaluation exhibits the misbehaviour of the packet communication in terms of wrong usage of VCs and extra hop distance overhead. To validate the proposed work, various performance metrics like Buffer utilization, Virtual Channel Utilization, number of flits processed, link utilization and the like are analysed. We also show that such HTs are difficult to detect due to marginal increase at malicious port traffic leaving no trace of malicious conduct.

Keywords: Network-On-Chips, Hardware Trojans, Confluence Attack

1 Introduction

In the last decade, momentous expansion in consumer electronics gadgets has resulted in the emergence of powerful Tiled Chip Multicore Processor (TCMP)

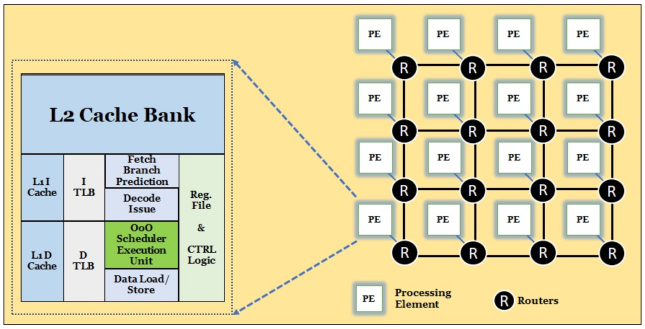

Fig. 1. 4×4 mesh NoC based SoC

with Network on Chip (NoC) as the underlying communication framework [1]. To reduce the overall system design cost, the use of third party intellectual property (IP) blocks are becoming a common practice. Such inexpensive and potentially insecure IPs put forth crucial data security challenges [2] [3]. Due to the exposure of complete data travel information and the IP connectivity, NoC becomes a natural choice for the adversaries to exploit the vulnerability and mount the attacks. One such exploit is the HT attack, the addition of unwanted circuitry into the IP design that can result in creating malicious activity including Denial of Service (DoS), information leakage, and data-stealing attacks [4] [5].

Fig. 1 shows internal architecture of a 4x4 mesh NoC based TCMP in which a tile comprises of a processing element (PE), private L1 caches, a share of L2 cache bank, and other relevant logic units required to carry out the communication. In TCMP, NoC packets are generated when a cache miss encounters and the required data needs to be brought from the remote tile. NoC packets travel in the form of smaller flow control units known as flit across the network. Cache miss request packets consists of single head flit with mandatory attributes for routing like source address and destination address while cache miss reply packets are multi-flit packets containing a head flit followed by multiple body flit and a tail flit carrying the data/payload. Wormhole switching is followed by the packets such that all the body flits and tail flit of a head flit automatically follows the same path as that of the head flit. NoC router has input buffers, route computation unit, virtual channel allocator, switch allocator, and a crossbar. Route computation is done at every intermediate router for forwarding the packet to its destination based on the underlying routing technique. The mere presence of an unwanted circuit behaving maliciously is enough to bring down the system performance drastically. In this paper, we propose one of such malign HTs sitting at NoC router altering the computed output port every time to one output port irrespective of incoming direction calling a confluence attack. We make the following contributions in the paper:

(i) We identify a suitable location to place a HT such that there is no amendment in the packet header information.
(ii) We propose the feasibility of HT over one of the nodes assuming HT is intermittent and gets activated with a probability 'p'.

(iii) Modelling the proposed confluence attack HT.
(iv) Implementation and analysis of the confluence attack induced by HT over NoC level parameters.

2 Related Work

HT enabling packet misrouting to cause denial of service, delay of service, and injection suppression is modelled and based upon the impact shielding technique is proposed to decrease the HT impact [3]. Various performance metrics like effective average deflected packet latency, effective average packet latency, and throughput are discussed. Corresponding to the mitigation techniques proposed, analysis in terms of hardware and timing overhead are also discussed. Another HT residing inside the route computation unit resulting in misrouting of the flits creating the impacts like deadlock, decrease in packet injection, delay of service, and denial of service is modelled while further a dynamic shielding technique is proposed to isolate HT infected IP [4]. Validation by performing analysis of the latency of packets is also. performed. A potential threat model that alters the NoC packet and leads to creation of dead flit in a router buffers is also discussed in which impact analysis of dead flit with two variants: one modifies the head flit to body flit and another modification from a body flit to head flit is studied. The impact analysis in terms of variation of the instruction per cycle, average buffer occupancy, and cache miss penalty is also gathered [6].

There can be a possibility of HT mounted on the input buffers of NoC routers while changing the destination address field of chosen NoC packets. Such possibility is proposed and modelled while the impact of HT at network, cache, and core level is captured [7]. In this work HT significantly impacts the L1 cache and is capable of bringing an application to a complete halt. The analysis in terms of assumption of Re-transmission of the impacted packet is also done while the assumptions of re-transmission is later considered to be unacceptable because of high latency overhead. Importance of electromagnetic(EM) radiation analysis for the purpose of hardware security is highlighted [8]. A novel hardware security solution is proposed which is based on the various analysis related to EM. Another side-channel-aware detection technique using test generation approach working on the principle of Multiple Excitation of Rare Switching is also proposed [9]. The proposed work significantly increase the sensitivity of HT, thus helps in easy detection of HT using side channel analysis techniques.

Adaptive Routing technique having non-interference characteristics is proposed in the literature to secure NoC from the timing attacks [10]. The work prevents the information leakage with 2-20% improved routing performance with power penalty of 1.84%.

3 Architecture Details for Baseline TCMP

A tile broadly includes Processor, L1 Instruction Cache, L1 Data Cache, L2 Cache, and a Network Adapter. L2 cache is uniformly shared among all the tiles and is accessible in a sequential manner. Crossbar, Switch Allocator, Virtual Channel Allocator, Routing Unit, and Input port buffers are some of the major components of the modern router [7]. The components of a router are given as follows:

(i) Buffering of incoming flits (BW)
(ii) Route Computation (RC)
(iii) Virtual Channel Allocation (VA)
(iv) Switching packets from input port to output port (SA)
(v) Switch Traversal (ST)
(vi) Link Traversal (LT)
(vii) Management of power including link scaling

Proposed work uses wormhole routing in which the routing is performed with the following characteristics:

(i) Route computation is performed only once per packet.
(ii) Virtual channel allocation is done once per packet and is embedded in the head flit of the packet.
(iii) Head flit acts as a parent node while the body flit and tail flit inheriting the head flit information following the same path.

4 Threat Model

In this study, we propose a malicious implant called HT that resides in the NoC router. Out of the various pipeline stages in router architecture viz. BW, RC, VA, SA, ST, and LT, the HT is activated during RC: Route Computation stage as shown in the Fig. 2.

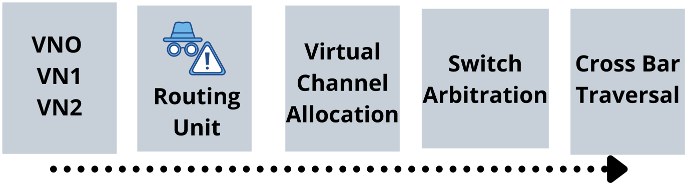

Fig. 2. HT embedded inside the RC unit resulting in flit misrouting to wrong output port

If an HT flicker happens, malicious behaviour is triggered by changing the legitimate output port always to the south direction and thus all the flits confluence to one output port. This HT behaviour leads to undesired service delays and service denial scenario. Cache miss requests, replies, evicted cache blocks, and coherence messages are all carried in NoC packets. An exploited NoC router

with the suggested HT can misroute these packets causing latency-critical applications to function poorly at the application level. Such HTs can be added to a NoC IP at any point during IC life cycle, including the specification, design, fabrication, and manufacturing phases [3].
In this paper, we assume that the suggested HT is deployed in NoC IP during the pre-silicon stage, either by an attacker with access to the system design or through an untrustworthy 3PIP. In the proposed work HT is randomly activated with a probability of 0.1 i.e. 10% probability along with the following two conditions existing simultaneously:

(i) Router-ID 5 must come in routing path of the flit.
(ii) Incoming port of the flit must be either East or West.

Since proposed HT is internally triggered and behaves intermittently malicious, targeting only few flits, thus catching such HT with verification/code, Electronic design automation (EDA) tools will be difficult.

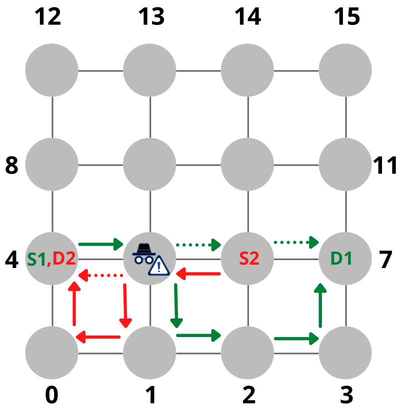

Fig. 3. HT Modelling for deflection of flits coming from East and West direction

Fig. 3 shows an instance of HT residing at Router-ID 5 and behaving maliciously. Any source-destination pair passing through HT node 5 gets deflected from the genuine route and confluence to a single direction (south). Example shows source-destination pairs 4-7 and 7-4 in which confluence attack is captured due to the fact of a packet coming from east and west input port respectively. However, the 13-1 and 1-13 source-destination pairs even if the confluence attack hits would not get impacted for the obvious reason of being in the south direction. By the virtue of the source-destination travelling combination with XY routing, there is no effect on flits coming from the north input port or south input port. As a result, all the flits trapped by Router-Id 5 during activation as a HT will have to cover the extra distance before reaching the final destination. Flit can only reach the target if a particular node usually acts and the real output port is obtained. The effect of the HT can be seen in terms of Delay of Service as the flits are misrouted hence delayed and can only be reached to the destination by traversing extra hops. This leads to delay of service attack and end up utilizing unavoidably more resources for packet transmission and reception. Fig. 4 shows the instance when HT is triggered during the wormhole routing. Misrouting of the head flit during route computation by HT results in misrouting of all the subsequent flits of that packet. Depending upon the injection rate value with the same probability (0.1), the number of flits deflected may increase or decrease.

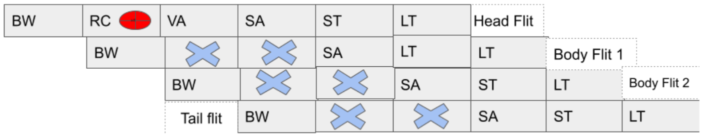

Fig. 4. HT triggered for Wormhole routing during Route computation

4.1 Performance Metrics for Impact Analysis

To conduct the empirical evaluation and to observe the HT impact following network and router related performance metrics are selected:

(i) **Virtual Channel Availability:** To acquire the statistics for the traffic of a certain router, we calculate the number of VCs utilised dynamically within a stipulated cycle period.

(ii) **Injection Rate:** It is represented as a number of packets injected per node per cycle and its value varies from 0 to 1.

(iii) **Router Load:** It indicates the router participation in the processing of total network traffic, more number of flits passing through the router indicates more router load.

(iv) **Flits processed by router:** It represents the total count of flits transmitted through a certain router.

(v) **Buffer Utilization:** When packets or flits can't be forwarded to output links right away, they're stored in buffers. Buffer utilisation measures how full a buffer is. Back-pressure and overload of existing connections can be detected using them.

(vi) **Internal Link Utilization(ILU):** All outbound links to nearby routers and the network adapter at the local port are monitored for link utilisation. To determine the available bandwidth, link utilisation monitors are employed during resource allocation. Internal links are one-way and form mesh connections. It connects the routers to create a particular topology. It gives insights into the total amount of bandwidth used for transmission.

5 Experimental Setup and Results

We use an event driven simulator gem5 to model and implement the 16 tiles 4x4 mesh NoC [11]. Inside gem5, Garnet 2.0 [12] is an interconnection network model simulated using the parameters shown in Table 1. Gem5's ruby memory system model provides the topology and routing infrastructure for Garnet 2.0. The implementation of the micro-architectural system for an on-chip network router is validated with the Garnet 2.0 module [13]. The simulation runs for the injection rate 0.1, and 0.2. To make the HT difficult to detect by common metrics like power and energy consumption, the deployment of malicious activity is modelled only in a single router.

Table 1. Simulation Parameter

Parameter	Description
Network	Garnet 2.0
CPU Count	16
Topology	Mesh - XY
Mesh Rows	4
Sim Cycles	5000
HT Probability	0.1
Injection Rate	0.1, 0.2
VC per vnet	2
Traffic	Synthetic Uniform Random
Execution Status	Baseline(B), Hardware Trojan(T)

In Traffic pattern Synthetic Uniform Random, all CPUs have an equal chance of being randomly selected as a source or destination node, and flits begin travelling accordingly for the selected source-destination pair, resulting in an unbiased examination for the proposed work. In the proposed work, simulation is executed under the two conditions:

(i) **Baseline Case (B):** With no HT activation (ideal case)
(ii) **Hardware Trojan Case (T):** HT is activated

To study the impact of HT, percentage change is calculated for the HT case with respect to the baseline or idle condition as shown in equation (1) below

$$Percentage\_Change = \frac{T\_Value - B\_Value}{B\_Value} * 100 \tag{1}$$

5.1 Effect of HT on Virtual Channel Utilization

A virtual channel (VC) is a distinct queue in the router that allows numerous VCs to share the physical wires (physical link) between two routers. Head-of-line blocking can be minimized by associating numerous distinct queues with each input port. On a cycle-by-cycle basis, virtual channels arbitrate physical link bandwidth. Every VC has its control buffer, which contains the following values: Packet Length (PL), Status (S), Virtual Channel Identifier (VCID), and Output Port (OP). When a flit arrives at a router, the input port demultiplexer extracts the VCID from the incoming flit's common prefix and stores it in the appropriate VC. Fig. 5 shows the analysis related to used virtual channels corresponding to

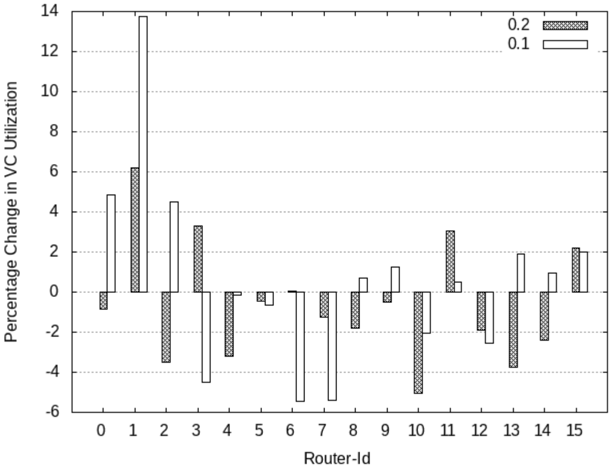

Fig. 5. Percentage change in average virtual channel utilization for complete mesh at 0.2 and 0.1 injection rate

each router of mesh for (B) and (T) at injection rate 0.1 and 0.2. The percentage change is calculated for (B) and (T) of each router at injection rates 0.1 and 0.2. During analysis, it is figured out that there is a positive percentage increase of up to 5, 13, and 4 percent in the usage of virtual channels for Router-ID 0, 1, and 2. The effect of HT is clearly visible on an increase in virtual channel utilization and it will result in earlier network saturation and more number of flits dropping will be there.

5.2 Effect of HT on Flits Processing and Flits Deflection Count

When a message is sent over the network, it is first divided into a data packet, which will then be divided into fixed-length flits, or flow control units. To compute the next outgoing port, the route computation unit extracts the destination ID from the head flit, and the output port is changed correspondingly. As a result, after a head flit's routing is complete, the output port stores the next outgoing port information for all following flits of that packet. Table 2 shows flit processed by Router-ID 5 during cycle times, when HT is triggered at injection rates 0.2 and 0.1. As HT is deflecting the flits coming from East and West port, for injection rate 0.2 in total 365 flits are misrouted by Router-ID 5, out of which 197 were entering from East port and 168 were entering from West port, for injection rate 0.1 in a total of 180 flits are misrouted by Router-ID 5, out of which 86 were entering from East port and 94 were entering from West port. The deflected flits result in increasing the network traffic in that particular link, there is no effect on the flits coming from North or South port. Fig. 6 shows the analysis corresponding to the flits processed by each router of mesh for (B) and (T) at injection rates 0.1 and 0.2. The percentage change is calculated for (B) and (T) of flits processed at injection rates 0.1 and 0.2. During the analysis, it is figured out that there is a positive percentage increase in flits processed of up to 5, 30, 5, and 5 percent for routers 0, 1, 2, and 4. The increase in the number of flits processed by a certain number of routers gives clear indications of the presence of HT, as the number of flits injected, VCs and injection rate were the same still certain routers processed more number of flits.

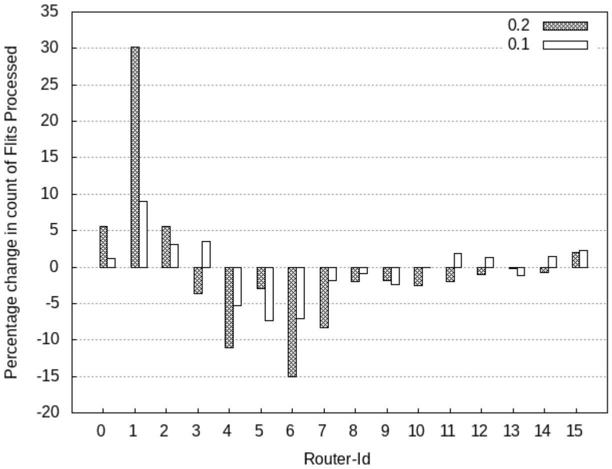

Fig. 6. Change in number of flits processed by each router of mesh for (B) and (T) for injection rate 0.2 and 0.1

Table 2. Misrouting of flits by HT(Router-Id 5) when got triggered

Direction	**Injection Rate 0.2**		**Injection Rate 0.1**	
	B	**T (Deflected)**	**B**	**T (Deflected)**
East	230	197	88	86
West	170	168	92	94
Total	**400**	**365**	**180**	**180**

5.3 Effect of HT on Link Utilization

If two candidate ports have the same number of available VCs, link utilisation is considered for the selection process. Then, within the current monitoring period, the port with the lowest link utilisation is chosen. In the uncommon event that both ports have the same link utilization, the first port is used.

(i) **Average Link Utilization** This parameter takes into consideration all the links like External(IN & OUT) which are bidirectional, and Internal that are unidirectional. The link utilization is calculated as per equations (2) and (3). Activity mentioned in equation (2) is a count of how many times a particular link is utilized. Time delta is calculated as the difference of curCycle() and start cycle(). curCycle() gives the current simulation cycle time and start cycle() is the starting simulation time.

$$AverageLinkUtilization+ = Activity/TimeDelta \tag{2}$$

$$TimeDelta = curCycle() - StartCycle() \tag{3}$$

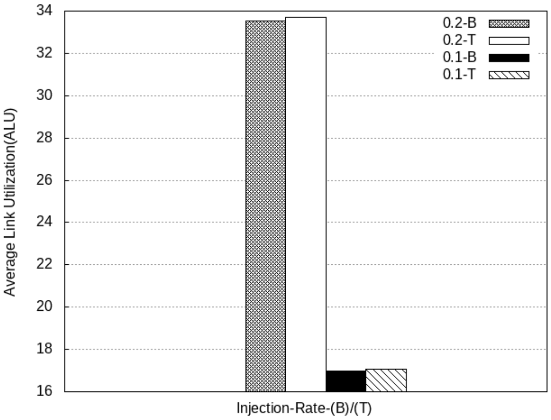

Fig. 7. Average link utilization for total number of flits injected

Fig. 7 briefs about the effect of HT on average link utilization by the flits for the same source and destination in the case of (B) and (T). Overall there is a 47 per cent increase in the average link utilization for the (T) case in comparison to the (B).

(ii) **Internal Link Utilization**
Internal link utilization is the count of activities in a particular link between two routers. The link utilization is calculated as per Equation 4.

$$IntLinkUtilization+ = activity; \tag{4}$$

Fig. 8 briefs about the effect of HT on average internal link utilization by the flits for the same source and destination in case of (B) and (T). There is a 0.84 per cent increase in the internal link utilization for the (T) case in comparison to the (B) as HT is deflecting more number of flits in one particular direction.

5.4 Effect of HT on router load in terms of count of activation of each router of mesh

In the proposed work a counter is deployed inside the Input Unit of each router that keeps on increasing every time a router is activated. Fig. 9 shows the analysis

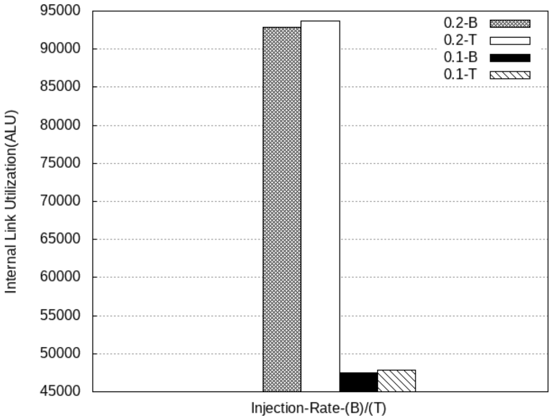

Fig. 8. Internal link utilization for total number of flits injected

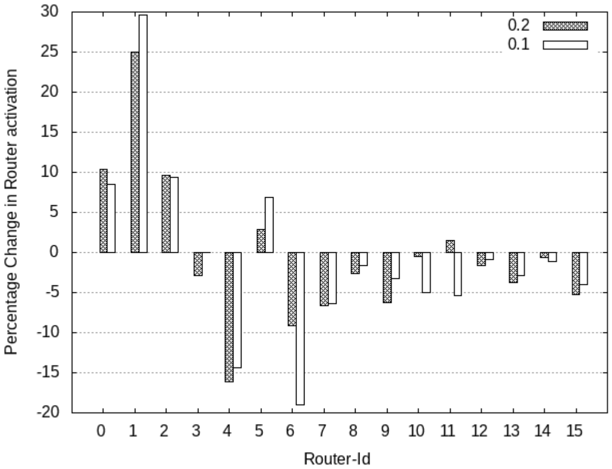

Fig. 9. Change in router load of for (B) and (T) at injection rate 0.2 and 0.1

corresponding to each router activation of mesh for (B) and (T) at injection rates 0.1 and 0.2. During the analysis, it is figured out that there is a positive percentage increase in router activation of up to 10, 29, 9, and 6 percent for Router-ID 0, 1, 2, and 5. Also from the given figure, it can be seen that for the baseline case the flits processed is similar for Router-ID (0, 3, 12, and 15),(1, 2, 4, 7, 8, 11, 13, and 14) , and (5, 6, 9, and 10) as analysis is done for Uniform random traffic and equal chances are given to every router for flits processing. But when the HT is active this symmetrical relation is disrupted, which gives signals of the presence of some malicious activity in the chip.

5.5 Effect of HT on variation in buffer reads for each router of mesh

When packets or flits can't be forwarded to output links right away, they're stored in buffers. On both the input and output ports, flits can be buffered. When the switch's allocation rate is higher than the channel's, output buffering occurs. Proposed work uses the wormhole routing having a provision that a packet must not be completely received for flit transmission and the subsequent router does not need to have buffer space available for the entire packet, which results in minimal buffer needs, and reduces the delay. But HT affects the buffer requirements up to a great extent and its effect is exclusively studied in the proposed work.

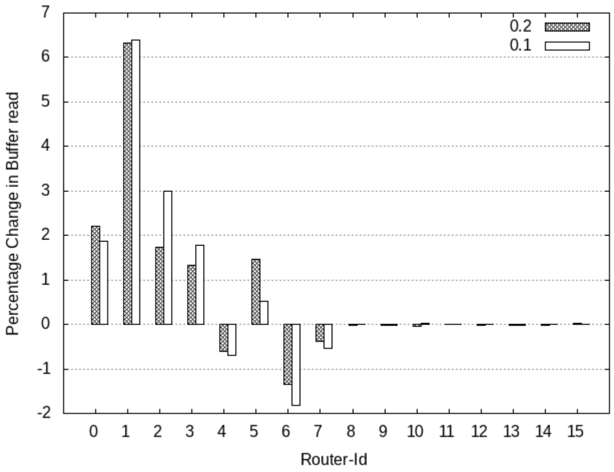

Fig. 10. Change in Buffer Utilization for (B) and (T) at 0.2 and 0.1 injection rate

Fig. 10 shows the analysis related to buffer reads corresponding to each router of mesh for (B) and (T) at injection rates 0.1 and 0.2. There is a positive percentage increase in the buffer read for Router-ID 0, 1 , 2 , 3, and 5 with value 1.88, 6.39, 3.00, 1.7, and 0.52 at injection rate 0.1 respectively and with respect to injection rate 0.2 change is 20.21, 6.32, 1.73, 1.32, and 1.46. The excessive usage of a buffer can result in earlier saturation and will result in the delay of service.

5.6 Effect of HT on Hop count, Network Latency, and Queuing Latency

Proposed modelled HT will redirect the flits in the wrong output port that will further result in covering extra hops for the same source and destination, which further effect the network latency. Table 3 shows the analysis done for the flits passing through Router-ID 5 in terms of Hop count, network latency and queuing latency. There is approximately a 7 per cent increase in the sum of hops, a 2 per cent increase in network latency and a 5.16 per cent decrease for the flits passed through Router-ID 5 as a result of HT activation.

Table 3. Analysis for the flits passing through Router-ID 5 in terms of Hop count, network latency and queuing latency for (B) and (T) at injection rates 0.1 and 0.2

Parameter	**Injection Rate 0.2**		**Injection Rate 0.1**	
	B	**T**	**B**	**T**
Count_Flits	400	365	180	180
Hops_Count_Sum	1160	1246	572	75
Hops_Count_Avg	2.9	342	3.17	759
Network_latency_Sum	7434	7584	2445	2995
Network_latency_Avg	18.63	20.84	13.58	16.64
Queuing_latency_Sum	864	815	360	360
Queuing_latency_Avg.	2.16	2.23	2	2

From the above analysis, it is clear that out of the complete chip our region of interest (ROI) has been confined to the Router-ID 0, 1, 2, 3, and 5. These routers have shown anomalies in the performance metrics related to network statistics, flits processing, router load, buffer reads. But based upon the above analysis it will be very difficult to commit exactly which is the malicious Router-ID in the given region. The nature of the HT like activation for a very limited time and deflection of flits only in particular ports hides the malicious node very well. HT showing unusual variation in power consumption and energy can easily lead to detection, but the proposed HT modelled is internally triggered and intermittently kept it hidden.

6 Conclusion and Future Work

In the proposed work, the usual 3-stage pipelined input buffered router is implemented. HT is embedded inside the route computation and triggered with the probability "p". Suggested HT contrives a novel confluence attack by altering the genuine output port always into a fixed specific output port. The HT notably influenced the cores surrounding the HT router node in a 16 tiles 4x4 mesh NoC. The HT impact has clearly shown the potential of HT attack to impair the system performance by keeping the resources busy unnecessarily. In the proposed study, various types of analyses are performed that are helpful to detect the presence of malicious activity in the network. Since HT is activated for a smaller duration and act maliciously for specific directions only, it becomes extremely difficult to trace the exact location of the HT in the network, however, the potential suspicious region can be filtered out. Localization of the HT infected Router-ID responsible for misrouting of the packets while creating confluence attack holds the promising scope as the future work.

References

1. S. Charles, M. Logan, and P. Mishra, "Lightweight anonymous routing in noc based socs," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 334–337, IEEE, (2020).
2. S. Charles, Y. Lyu, and P. Mishra, "Real-time detection and localization of dos attacks in noc based socs," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1160–1165, IEEE, (2019).
3. R. Manju, A. Das, J. Jose, and P. Mishra, "Sectar: Secure noc using trojan aware routing," in *14th International Symposium on Networks-on-Chip (NOCS)*, pp. 1–8, IEEE, (2020).
4. M. Rajan, A. Das, J. Jose, and P. Mishra, "Trojan aware network-on-chip routing," *Network-on-Chip Security and Privacy*, p. 277, (2021).
5. A. Das, S. Babu, J. Jose, S. Jose, and M. Palesi, "Critical packet prioritisation by slack-aware re-routing in on-chip networks," in *Twelfth International Symposium on Networks-on-Chip (NOCS)*, pp. 1–8, IEEE, (2018).
6. M. H. Khan, R. Gupta, J. Jose, and S. Nandi, "Dead flit attack on noc by hardware trojan and its impact analysis," in *Proceedings of the 14th International Workshop on Network on Chip Architectures*, pp. 10–15, (2021).
7. V. J. Kulkarni, R. Manju, R. Gupta, J. Jose, and S. Nandi, "Packet header attack by hardware trojan in noc based tcmp and its impact analysis," in *15th International Symposium on Networks-on-Chip (NOCS)*, pp. 21–28, IEEE, (2021).
8. J. He, X. Guo, M. Tehranipoor, A. Vassilev, and Y. Jin, "Em side channels in hardware security: Attacks and defenses," *IEEE Design Test*, pp. 1–1, (2021).
9. Y. Huang, S. Bhunia, and P. Mishra, "Scalable test generation for trojan detection using side channel analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2746–2760, (2018).
10. T. H. Boraten and A. K. Kodi, "Securing nocs against timing attacks with non-interference based adaptive routing," in *Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pp. 1–8, IEEE, 2018.
11. https://www.gem5.org/documentation/learning_gem5/introduction/ Last accessed 26 January 2022
12. https://www.gem5.org/documentation/general_docs/ruby/garnet2/ Last accessed 26 January 2022
13. N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, *et al.*, "The gem5 simulator," *ACM SIGARCH computer architecture news*, vol. 39, no. 2, pp. 1–7, (2011).