SecureNoC

(Team Colonel Panic): Xiaoming Guo, Sijia He, Amlan Nayak, Jay Zhang

November 13, 2015

1 Problem Statement

Networks on Chip (NoCs) have been steadily investigated from reliability, efficiency, and performance perspectives.However, little effort has been directed towards the security of complex on-chip networks. These networks are prone to attacks similar to those perpetrated against large scale networks such as datacenters and the internet. If a single core within an NoC is compromised, it can be used as a vehicle to contaminate other cores on the network, or even the entire network itself.

2 Importance

Datacenter security is of utmost importance given the explosive growth of big-data centric applications. The amount of data that a single user generates on a diurnal rhythm is staggering and thus high performance computing with large numbers of integrated cores are now a necessity. Since a vast portion of the data being generated needs to remain private, secure handling of data by data center applications and hardware is paramount. Though mechanisms exist to thwart attacks on a large scale network of servers, little attention has been paid to the security of on-die NoCs.

3 Solution

Two important security vulnerabilities of NoCs are Denial of Service (DoS) attacks and Extraction of Secret Information attacks. Now, in order to ensure secure communication between cores on an NoC, we propose a novel scheme. We augment each router within the network with a traffic monitoring unit and a mechanism to block the communication between local port and router. These modules carry out the following objectives :

- Monitor injection rate in order to detect DoS attacks and halt an attack in progress
- Support secure exchange of priority packets between different routers.

4 Progress

- We have implemented a threshold based DoS attack detection mechanism both in Booksim and in SystemVerilog. The system can detect an abnormally high flit injection rate from any node in a given epoch. Once this abnormal behavior is detected, the system will temporally stop accepting more packets from the compromised node for two epochs. This will allow the "suspiciously compromised" core to re-schedule the packet injection process. After the stalling period, if the core keeps injecting with high rate, it will be shut down permanently, as shown in Figure 1; otherwise, we allow the core to send packets normally, which is illustrated in Figure 2. This mechanism will help eliminating false positive.

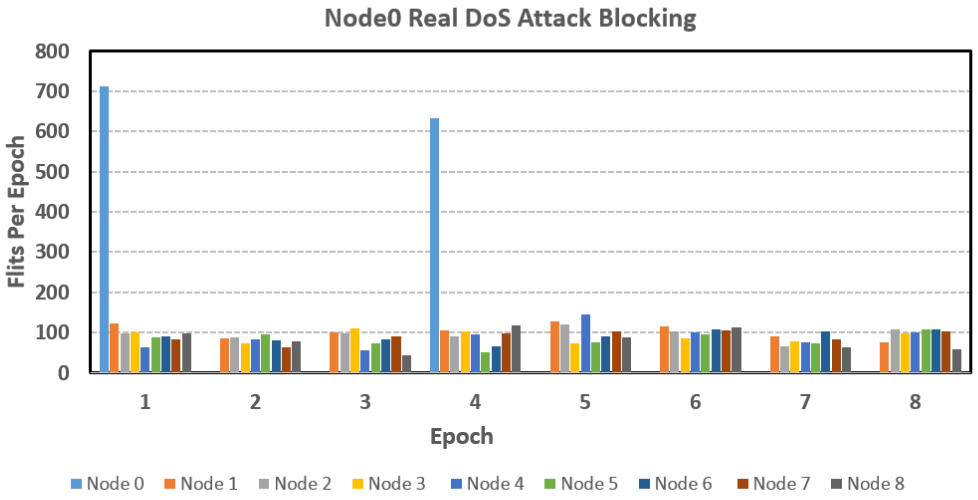

Figure 1: Read attack

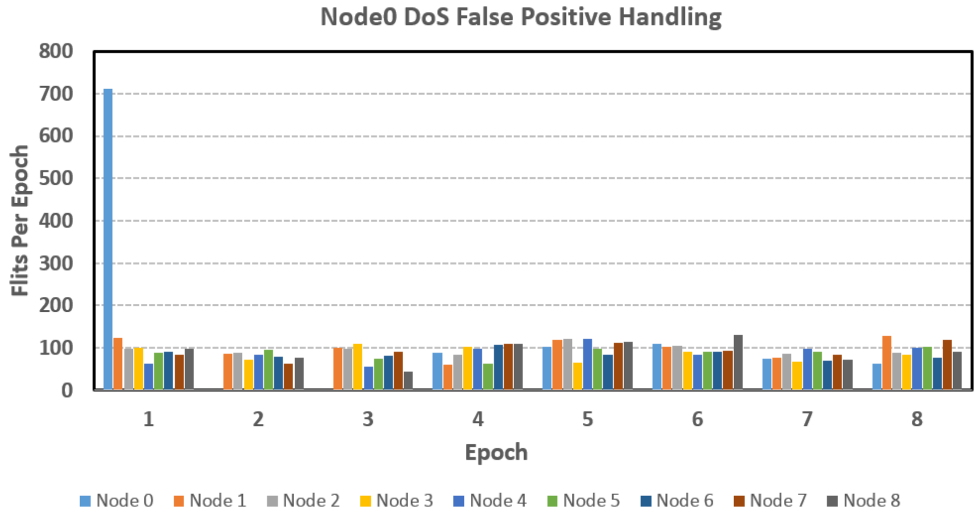

Figure 2: False positive handling

- We justify the threshold based mechanism from the observation that the average packet latency will shoot up once the injection rate for a given node goes beyond a certain threshold value. And this threshold value shows little dependency on the injection rate of other nodes (as long as they are working normally), number of VCs, etc. This discovery is shown in Figure 3, where the threshold is around 0.3 for all different configurations.

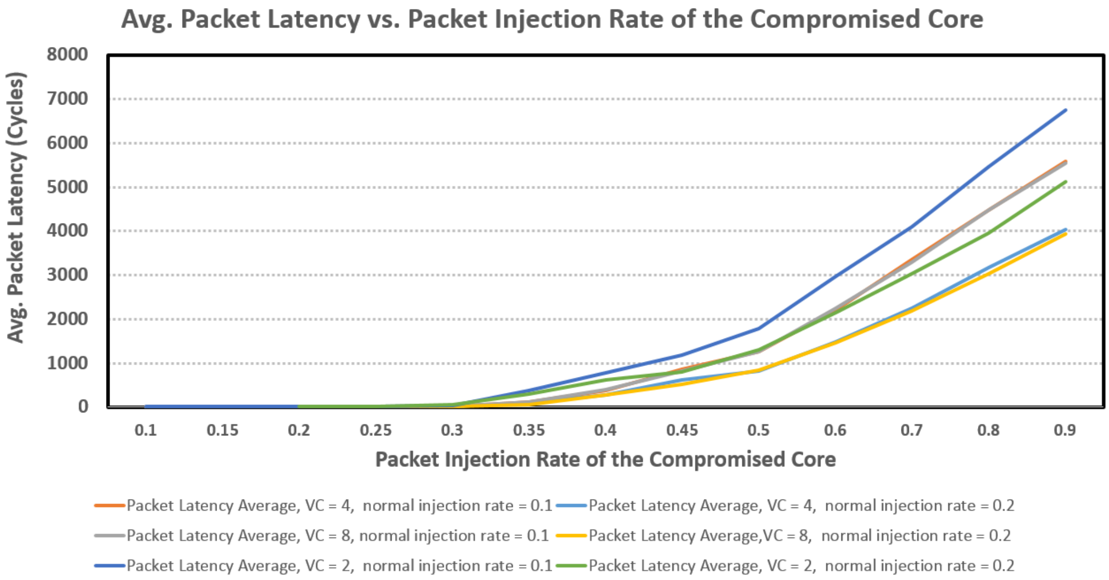

Figure 3: Packet latency vs. injection rate

- We augmented the RTL model from Prof. Dally's group to incorporate our DoS attack detection mechanism. We synthesized our implementation and compared the data with the baseline. The results are given in Table 1.

Table 1: Baseline vs. DoS Detection

	clock period (min)	**area**
Baseline	2.9ns	1896527.2
DoS Detection	2.9ns	1912835.5
Overhead	0	0.86%

- We came up with **S**ecure**P**acket**E**xch**A**nge**R** (**SPEAR**), a mechanism that can support the secure exchange of priority packets between different routers. It can be used to exchange keys for encryption, private information, etc. Figure 4 shows how this mechanism works. Secure packets with a high privilege level (indicated by a bit during packet injection) are assigned strictly to a single VC. VC0 is reserved for such packets. Once a privileged packet is inserted, the source router sends a `block_vcs` message to the destination router along the deterministic XY path. Every router in the path blocks all its VCs except for reserved VC0 once it receives the message. In addition, it cuts off the connection between local core and the router, effectively cutting off other cores from snooping the incoming secured packets. The destination router responds with an `ack` message which allows the source to start sending the secured flits. Once the tail flit has been delivered, all VCs are re-enabled in each router along the path. Further, the connection between local core and router is re-established.

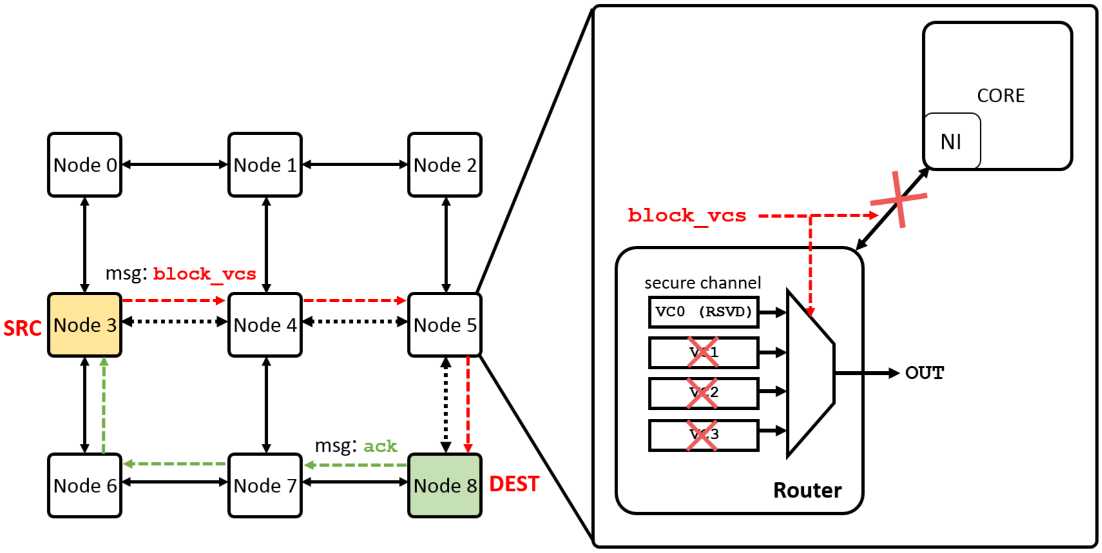

Figure 4: **S**ecure **P**acket **E**xch**A**nge**R** (**SPEAR**)

5 Issues/Showstoppers

1. We are a little bit behind the schedule for the implementation of the secure exchange of priority packets. (We have already started implementation in BookSim, but there are some bugs that we need to look into.)

2. We plan to block the local ports for all the routers on the path before sending the priority packet, is it enough to only block the local core? Or for each router on the path, all the VCs, except the ones for priority packet.