

A Sensor-less NBTI mitigation methodology for NoC architectures

Davide Zoni and William Fornaciari

Politecnico di Milano – Dipartimento di Elettronica e Informazione

Via Ponzio 34/5, 20133 Milano, Italy

{zoni, fornacia}@elet.polimi.it

Abstract—CMOS technology improvement allows to increase the number of cores integrated on a single chip and makes Network-on-Chips (NoCs) a key component from the performance and reliability standpoints. Unfortunately, continuous scaling of CMOS technology poses severe concerns regarding failure mechanisms such as NBTI and stress-migration, that are crucial in achieving acceptable component lifetime. Process variation complicates the scenario, decreasing device lifetime and performance predictability during chip fabrication. This paper presents a novel sensor-less methodology to reduce the NBTI degradation in the on-chip network virtual channel buffers, considering process variation effects as well.

Experimental validation is obtained using a cycle accurate simulator considering both real and synthetic traffic patterns. We compare our methodology to the best sensor-wise approach used as reference golden model. The proposed sensor-less strategy achieves results within 25% to the optimal sensor-wise methodology while this gap is reduced around 10% decreasing the number of virtual channels per input port. Moreover, our proposal can mitigate NBTI impact both in short and long run, since we recover both the most degraded VC (short run) as well as all the other VCs (long term).

Index Terms—Multi-core; Network-on-Chip; Reliability

I. INTRODUCTION

Continuous technology scaling leads to an exponential increase in processor performance enabling also true System-on-Chip (SoC) integration. To cope with power/performance trade-off [1], the accepted solution to integrate multiple cores in a single chip poses new communication challenges, and Network-on-Chip (NoC) [2] emerges as a viable and effective design paradigm to manage performance and reliability requirements. However, the actual integration density is limited by the reliability of the device, which has become a requirement of paramount importance in current and future multi-core designs. Physical failure mechanisms, i.e., Negative Biased Temperature Instability (NBTI) or stress-migration, can seriously limit the device lifetime. In particular, the introduction of nitrided oxides, the rising of operating temperatures and the increase in gate oxide fields, caused by technology scaling, highlight NBTI as one of the worst threat to sub-nanometer CMOS technology.

Considering CMOS circuits, NBTI occurs in PMOS transistors when $V_{gs} = -V_{dd}$, namely when the PMOS is active. Such *stress condition*, has been faced by increasing the threshold voltage (V_{th}) producing, as a side-effect, a degradation of the driving current. This can lead to performance degradation or, in the worst scenario, to the stuck of the transistor which cannot longer switch its logic state. It has been shown that NBTI can increase $|V_{th}|$ by as much as 50mV for devices operating at 1.2V or below [3], while the circuit performance degradation may reach 20% in 10 years [4]. On the other side, when a logic "1" is applied to the gate, i.e. $V_{gs} = 0$, NBTI stress is removed. This latter condition, called *recovery state*, induces a progressive, yet partial, recovery of the device threshold voltage.

At the same time, the difficulty in controlling sub-wavelength lithography and channel doping when technology scales down

manifests as *Process variability* (PV), producing unexpected power/performance variations, which are becoming a major fabrication challenge for the upcoming technology scales. In particular a latency degradation up to 40% and a leakage power variation on buffers of about 90% due to process variation has been observed [5]. Finally, the integration of huge amount of transistors in a single chip operating at higher frequencies increases the power densities with the risk of creating thermal hot spots decreasing the reliability, dramatically [6]. Although many works address the NBTI and PV considering either device, circuit or system levels, they are primarily focused on the computational logic of a multi-core chip. From the best of our knowledge, a comprehensive framework to manage NBTI issues in the on-chip networks considering the PV impact has not been proposed yet.

A. Novel contributions

This paper provides a sensor-less methodology to mitigate NBTI degradation in the on-chip network virtual channel (VC) buffers of multi-core architectures, also considering process variation issues. Moreover, our developed simulation flow allows for the extraction of accurate performance and NBTI estimates. In summary, the paper encompasses the following two aspects:

- *Network-on-Chip NBTI estimation framework* – The proposed approach has been integrated in a multi-core simulator that can simulate NoC communication subsystems considering both synthetic traffic as well as real benchmarks. We provide an NBTI sensor library coupled with the simulation environment starting from a flexible analytical NBTI model from the available literature [7]. It is worth to notice that the NBTI sensor library is used to assess the validity of our methodology that works without any NBTI information.
- *Reliability enhancement* – A sensor-less NBTI mitigation methodology for router's buffers is proposed, providing significant NBTI improvement against NBTI-less methodologies, while behaves slightly worsen than the optimal sensor-wise methodology that we use as golden model. Moreover, our sensor-less approach provides NBTI reduction thanks to the exploitation for each router of the information on coming traffic from neighbor routers. The proposed strategy reduces the NBTI impact by reducing the buffers' stress periods.

B. Paper structure

The paper is organized as follows. Section II reports an overview of the state-of-the-art on NBTI related methodologies, for both computational and communication logic. Section III describes the proposed sensor-less NBTI strategy as well as an *oracle* policy used as reference model that uses sensor. Experimental results are then reported and discussed in Section IV. Conclusions are drawn in Section V.

II. RELATED WORKS

Reliability issues related to electronic devices have been extensively investigated during last decades. However, most of the proposals focus on NBTI mitigation considering PV effects on processor core architectures, while only a few target on-chip networks. This section provides an overview of the state-of-the-art related to NBTI mitigation design methodologies considering PV issues.

Liang et al. [8] exploited variable latency techniques in register file and functional unit design, to mitigate the effects due to process variation. A linear programming based approach has been proposed in [9], to optimally drive the inputs to individual gates in order to prevent static NBTI fatigue. The work addresses the NBTI stress due to long stand-by periods. Li et al. exploits idle time in functional unit inside a processor core to recover NBTI degradation with a negligible performance loss and area overhead [10].

The NBTI impact on SRAM cells has been investigated in [11], where it is shown that read stability degrades due to NBTI, accounting for the negative PV impact. In this track, several studies have been proposed on PV in NoC design. Li et al. observes how process variation can greatly influence on-chip network design choices [12]. To this extent, Ogras et al. studied the multiple voltage-island design technique applied to on-chip networks to face the PV issues [13].

Xin Fu et al. present a comprehensive approach to mitigate NBTI degradation considering NoC architectures [14]. This work presents several approaches to effectively manage NBTI impact, focusing on different micro-architectural components, namely buffers and arbiters. However, [14] does not exploit the information on actual NoC traffic between each router pair, thus the solution loses the possibility to aggressively reduce NBTI degradation as well as the exploitation of NBTI/performance trade-off.

The tightly coupling between NoC performance and virtual channel buffer structure motivated the work iDEAL [15], a framework to design energy and area-efficient links that are capable of data transmission as well as data storage when required. The goal is a reduction in the router buffer size by controlling the repeaters along the links to adaptively work as link buffers during NoC congestion.

Despite the depth of each specific investigation, the lack of current literature resides mainly in the missing of a comprehensive methodology and the related supporting NoC analysis and design framework, capable to concurrently take into account several figures of merit such as NBTI, PV while exploring cost/performance/reliability trades-off in a way so flexible to enable considering alternative micro-architectural solutions for the on-chip network.

III. PROPOSED ESTIMATION FLOW

This section presents the proposed methodology through two different steps. First, Section III-A discusses the baseline NoC considered as well as the *oracle* approach, that we use to assess the validity of our methodology. The round-robin sensor-less methodology rationale and the logic modifications to the baseline NoC are detailed in Section III-B.

The simulation flow is underpinned by *GEM5* performance simulator that is able to perform multi-core cycle accurate simulations considering an accurate NoC model. Moreover, we validate our work considering both synthetic traffic and traffic from real benchmark scenarios, using an out-of-order CPU model for the latter case. We rely on the *GEM5 syscall emulation* approach that mimics bare-metal execution and can simulate the underlying hardware with precise processor models and collect statistics at micro-architecture level. We choose this simulation approach, since *full-system* simulation mode, that *GEM5* can also offer, requires an OS to support application

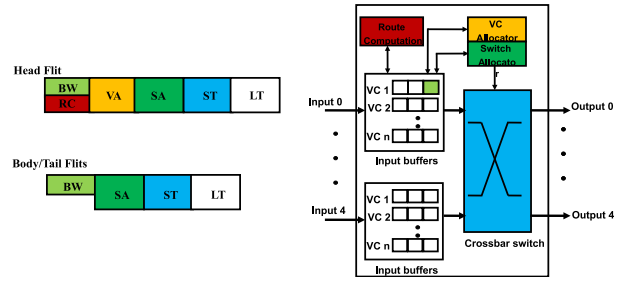


Fig. 1. Pipeline steps for head and body/tail flits for the considered baseline router micro-architecture.

execution, introducing undesired overhead and making it harder to obtain results clarifying the benefits of our methodology.

A. Baseline NoC model and the sensor-wise oracle approach

This section details the baseline NoC as well as the sensor-wise optimal approach to recover the most degraded virtual channel, i.e. *oracle*. We start from a baseline 3-stage pipelined router, depicted in Figure 1, and implemented in the Garnet [16] available in GEM5. A packet is considered split in multiple atomic transmission units, called flits. The first flit of each packet is the header flit. A body flit represents an intermediate flit of the original packet, while the tail flit is unique for each packet and represents the final flit of the packet itself. Figure 1 pinpoints the pipeline stages for an header and a body/tail flit. When a flit enters in the baseline router from one of the input ports, if it is an header flit, it has to pass through five pipeline stages. First, it is stored in the virtual channel (VC) buffer that has been reserved by the upstream router, through a buffer write stage (BW). Route Computation (RC) stage is performed in the same clock cycle only if the flit is an header one, which determines the output port for this new packet. The virtual channel allocation (VA) represents the next pipeline stage, that reserves to the new packet an available virtual channel in the downstream router, from the selected output port. If the VC allocation is successful, the flit competes for a crossbar switch path to its output port during the switch allocation (SA) stage. Finally, if the flit wins the switch allocation, the following steps are switch traversal (ST) and link traversal (LT), that account for the delay to traverse the upstream router crossbar and upstream-downstream router links, respectively. Tail or body flits require to traverse fewer pipeline stages since they exploit some resources and information already reserved to the packet by the header flit, i.e. VC and RC.

The baseline NoC supports bidirectional communication between each pair of routers via two links for each communication directions: a network link from source to destination allowing packet transmission and a destination to source control link to send back control flow information. The baseline router, allocates packets to available VCs in a round-robin fashion, disregarding NBTI degradation on buffers.

We provide some useful definitions concerning NBTI that are valid for the rest of the paper, to enhance the readability of our methodology. Since the NBTI has two phases *stress* and *recovery*, we consider a buffer in a stress phase every time it is storing some flit or when it is in an idle state from the NoC point of view. An idle state means that the buffer has no valid flit stored in it, while from the NBTI point of view it must be considered stressed, since there is an input configuration vector on its inputs even if it is meaningless. A buffer is in the recovery phase if it is switched-off. We do not consider any specific recovery technique here, since the final goal of this work is to provide a proof of the NBTI stress period reduction, while the

use of a particular recovery technique, i.e. power gating, input control vector, is left as future work. Since the duty-cycle definition for NBTI purposes is quite different from the traditional duty-cycle definition, we defined the *NBTI-duty-cycle* as:

$$\text{NBTI-duty-cycle} := \frac{\text{stress-cycles}}{\text{stress-cycles} + \text{recovery-cycles}} * 100 \quad (1)$$

where *stress-cycles* and *recovery-cycles* are the NBTI stress period and the NBTI recovery period respectively.

NBTI is a time dependant mechanism and it is known to be exponentially dependant on temperature, supply voltage and *NBTI-duty-cycle*, as well explained by the generally accepted long-term model reported as closed-form Equation 2 [17]:

$$|\Delta V_{th}| \approx \left(\frac{\sqrt{K_v^2 \cdot T_{clk} \cdot \alpha}}{1 - \beta_t^{1/2n}} \right)^{2n} \quad (2)$$

where K_v is a parameter dependent on supply voltage and operating temperature, T_{clk} is clock period, α is the stress probability of PMOS devices, i.e. the *NBTI-duty-cycle*, β_t is dependent on temperature, while n is generally set to 1/6 for hydrogen molecule diffusion model [18]. To this extent our methodology tries to recover the NBTI impact reducing the *NBTI-duty-cycle* that has been shown to be a key factor to control the NBTI dynamic.

In such a scenario the best approach that we can cast should use the less degraded buffer to store packets, while the most degraded ones must be *recovered*, where the actual NBTI degradation for each virtual channel buffer is supposed to be known, i.e. by NBTI sensors [19]. In addition, for each cycle we can maintain all idle virtual channel buffers recovered except the less degraded one, since a single flit can be sent for each cycle at most, between each upstream downstream routers pair. Moreover, the only virtual channel buffer left idle, should be aggressively recovered if no new packets from the upstream router address the specific input port of the downstream router. This could be done since the VA stage happens in the upstream router and arbitrates for virtual channels on the input port of the downstream one. In this way, the upstream router knows how many incoming packets want to go to a specific downstream router and has not a VC allocated in the downstream router yet, thus it knows if more active VCs are needed to support the network load. We call this policy *oracle* that represents the best possible way to recover the most degraded virtual channel.

B. Sensor-less round-robin approach

The *oracle* policy represents the best possible solution to face the NBTI impact on the most degraded VC buffer, since it uses information from the NBTI sensors attached to each buffer. However, it introduces a significant overhead mainly due to the need for an NBTI sensor for each buffer, as well as other communication lines to exchange information between the upstream and downstream router: the NBTI degradation is computed in the downstream router, while the actual VA stage happens in the upstream one. To this extent we use the *oracle* as reference and we propose an aggressive round-robin approach *rr_aggr* to recover NBTI degradation on the most degraded VC without use any NBTI sensor. The main idea of this approach is to periodically recover all idle virtual channels for each input port of the downstream router in a round robin fashion, since we do not have information on the most degraded VC buffer. This approach can reduce NBTI impact both in the short term as well as in the long run. All the virtual channel buffers will be recovered following a round-robin fashion improving the long run NBTI degradation. Moreover

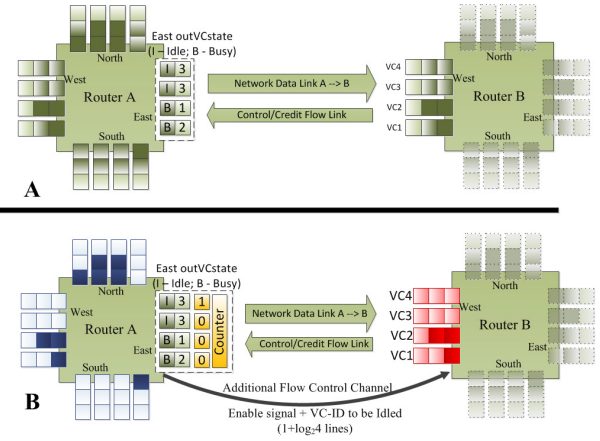


Fig. 2. Logic blocks for the baseline (A) and sensor-less NBTI-aware (B) micro-architectures.

even the most degraded virtual channel buffer is recovered, even if it is not known, thus also the short term NBTI degradation is reduced.

This section details the logic modifications implemented in the baseline on-chip network to exploit the sensor-less NBTI. For the sake of clarity and conciseness, despite the proposed methodology is fairly general and applicable to the whole NoC, we restrict the presentation of details to a simple while representative example.

The baseline on-chip network and the modified NBTI aware NoC are proposed in Figure 2A and 2B respectively, considering two routers – *Router A* and *Router B* – and a single communication direction (from A to B), thus *Router A* is the upstream router, while *Router B* is the downstream one. We consider a four input/output port router model (North, South, West, East) with four virtual channel buffers for each input unit. At each time each buffer can store one or more flits of a packet, or can be idle from the on-chip network point of view, while each channel can be stressed or recovery from the NBTI point of view. The methodology tries to reduce the *NBTI-duty-cycle* that is one of the most important factors steering NBTI degradation. We focus on the west input port of *Router B* (red buffers in Figure 2B) and the corresponding east output port of *Router A*. It is worth to notice that the discussion is valid for each other input and output port pair in the NoC. Figure 2B reports the *output VC*

Algorithm 1: Pre-VA stage for Router A east output port.

```

Input: out_vc_state, active_candidate_vc
Output: enable, active_vc

1 enable ← 0;
2 active_candidate = get_vc_candidate();
3 if not is_new_traffic_east_outputport() then
4   enable ← 0;
5   active_vc ← active_candidate;
6   return;
7 offset_vc ← active_candidate;
8 foreach iter ∈ (1..num_vcs) do
9   if is_idle(offset_vc) or is_recovery(offset_vc) then
10    set_idle(offset_vc);
11    enable ← 1;
12    active_vc ← offset_vc;
13    return;
14 offset_vc++;
15 if offset_vc ≥ num_vc then
16   offset_vc ← 0;

```

TABLE I
EXPERIMENTAL SETUP: PROCESSOR AND ROUTER
MICRO-ARCHITECTURES, AND TECHNOLOGY PARAMETERS.

Processor core	1GHz, out-of-order Alpha core
Int-ALU	4 integer ALU functional units
Int-Mult/Div	4 integer multiply/divide functional units
FP-Mult/Div	4 floating-point multiply/divide functional units
L1 cache	64kB 2-way set assoc. split I/D. 2 cycles latency
L2 cache	512kB per bank, 8-way associative
Coherence Prot.	MOESI token (for real traffic)
Router	3-stage wormhole switched with 32b link width 2/4 virtual channels for each virtual network 2/6 virtual networks (Garnet network [16]) - data virtual network used for real traffic simulations - instruction virtual network used for synthetic traffic simulations
Topology	2D-mesh, based on Tileria iMesh network [20] for link width and NoC frequency (@1GHz)

state, i.e. *outVCstate*, for the east output port in Router A instead of the input buffers to enhance the presentation of the methodology. In the baseline router the *output VC state* contains useful information that track the state of the corresponding virtual channel buffers in the west input port of Router B, i.e. the state of the corresponding VC and the available flit slots. The yellow blocks in the east *output VC state* represent the additional logic to support the NBTI sensor-less methodology. In particular we model an additional counter buffer to the east output port of router A, i.e. *counter* in Figure 2B, while for each *output VC state* of the same output port a single bit is added. We use the additional bit associated to the *output VC state* as the *active_candidate* marker. One hot *active_candidate* encoding only is possible, since a single *output VC state* can be the *active_candidate*. The Algorithm 1 details the selection of the active virtual channel for each VA stage iteration, if a new VC is necessary. Since we aggressively recover all the Router B west port idle VCs in a round robin fashion and at each cycle at most one new virtual channel could be required, to store flit from a new packet, the *active_candidate* bit signals which is the first virtual channel that must be considered for use if a new virtual channel is needed. If no new packets require VA stage on the east output port Router A de-asserts the *enable* signal, thus the downstream router can recover all idle VCs on its west input port, as detailed in lines 2-5. Otherwise Router A scans all *out VC states* starting from the *active_candidate* VC, selecting the first idle VC. When an idle VC is found it updates and sends to Router B the *active_vc* lines and asserts the *enable* signal (see lines 7-15 in Algorithm 1). Thus, Router B recovers all idle VCs on its west input port, but the *active_vc*. It is worth to notice that Algorithm 1 may terminate without finding an idle VC, since all the VC could be already active and busy due to high traffic load. The *active_candidate* bit is set for VC0 at system start up and remains unchanged until the counter buffer saturates. When the counter buffer saturates VC1 becomes the new *active_candidate* VC. The counter buffer of the east output port in Router A is incremented every time a VA stage happens in Router A. The necessary logic for Algorithm 1 is inserted in the VA stage of the baseline pipeline, but execute before the baseline VA stage to provide a single idle VC (*active_vc*), if needed for new packets, to the subsequent baseline VA stage.

Moreover, we introduce an additional control flow link from Router A to Router B that is used by the Router A to signal to Router B the *active_vc* for each VA stage. This additional control flow channel drive the *enable* signal also to invalidate the *active_vc* identifier in case of aggressive recovery, since no new virtual channel is needed for the current VA stage. The additional control flow link has $1 + \log_2 v$ lines only, where v is the number of virtual channels driving the *active_vc* information and one more line is for the *enable* signal.

Last, we assessed an area overhead below 3% for our methodology, with respect to the baseline router. In particular we extract link area

information for a 45nm technology node using *Orion2.0* [21]. We obtained an area overhead for the *up_down* link of 1.9% with respect to a single 64bits data link. Moreover, we synthesized Algorithm 1 inside every upstream router output port using NetMaker¹, that is a library of synthesizable NoC components coupled with an opensource standard cell 45nm library. The final design has been synthesized using Cadence Encounter compiler providing an area overhead of the additional logic below 0.5% of the considered router.

IV. RESULTS

This section details the results of the proposed methodology to mitigate the NBTI impact, and also considering process variation issues as well. Experimental setup is discussed in detail in Section IV-A, while results for both synthetic and real traffic patterns are given in Section IV-B and Section IV-C respectively.

A. Experimental setup

We consider a multi-core processor as reference architecture, where some architecture parameters have been chosen to resemble a commercially available processor similar to the Tileria multi-core family with on-chip network [20]. To this extent, we considered a 2D-mesh topology with 3-stages pipelined routers inspired from [22]. The multi-core architecture is composed of tiles. Each tile in the 2D-mesh is composed of an out-of-order processor based on Alpha-21264 ISA, private L1 cache and shared distributed L2 cache banks, and a memory controller. For synthetic traffic estimation, a traffic generator is used instead of the processor core, while we extract results from the instruction virtual network. Table I summarizes the main architectural setup of the simulated multi-core architecture.

NBTI is a time dependent mechanism and it is known to be exponentially dependent on temperature *NBTI-duty-cycle* and supply voltage. In this work we face NBTI through *NBTI-duty-cycle* reduction, as described in Section III, while we marginally consider temperature profiles as worst case scenario. In particular we employ the adapted simulation flow from [23], as described in Section III, for temperature extraction [23]. We simulate both 4 and 16 multi-core architectures with both synthetic and real traffic patterns. For 16 cores architectures we observed a maximum operating temperature close to 343K; we then consider a worst-case temperature of 340K for all of our simulations.

In addition, we do consider process variation as a true driver of variability in scaled technologies. Process variation is a combination of random effects and systematic effects [24], and can impact both die-to-die and within-die parameters. In this work we are mainly interested in within-die process variation, and assume for simplicity the impact of die-to-die variation to be constant in the same chip [14]. Process variability manifests itself as a divergence of sensible design parameters (e.g., initial threshold voltage V_{th}) from their nominal values. For these reasons, we assume the same initial V_{th} for each VC, and equal to the highest V_{th} , namely that of the most degraded PMOS transistor in the buffer. We then associate a PMOS transistor to each virtual channel buffer of each router; each modeled PMOS transistor has its own starting V_{th} , that has been extracted from a Gaussian distribution with absolute average value of 0.180 Volt for 45nm technology, and a standard deviation equal to 0.005 [25].

It is worth considering the fact that real NBTI sensors have been shown to be feasible in VLSI systems [11], [19], while they also exhibit a small area footprint. To this extent sensor-wise methodology should also be an interesting direction to be investigated since the

¹http://www-dyn.cl.cam.ac.uk/~rdm34/wiki/index.php?title=Main_Page

small area overhead required and the focused recovery that we can obtain on the most degraded virtual channel buffer. However in this paper we use the sensor-wise approach as the *oracle* solution that is used to assess the sensor-less round-robin proposal.

B. Synthetic results

This section reports the results obtained considering uniform traffic patterns, on both 4-core and 16-core 2D-mesh architectures, with 2 and 4 virtual channels per input port and considering varying injection rates of 0.1, 0.2 and 0.3 flit/cycle/vnet. To collect representative statistics, we simulated each scenario for 30×10^6 cycles, that are 30ms at 1GHz, while at the end of the simulation we sample the *NBTI-duty-cycle* values for each virtual channel. It is worth noticing that the NoC reached its steady state after 6×10^6 and 9×10^6 cycles for 4-core and 16-core scenarios, respectively. Each result is sampled from the upper left-most router on its east input port due to symmetry in the topology.

Results for 4- and 16-cores scenarios are reported in Table II considering 2 virtual channels per input port, while the same 4- and 16- architectures results using 4 virtual channels per input port are reported in Table II(b). Table II(a) and II(b) share the same format, where the first column reports the identifier of the simulated scenario. The second column of each table contains the *NBTI-duty-cycle* we achieve on the most degraded VC considering the *oracle* approach, that is used as reference to assess the validity of our sensor-less round-robin methodology. The *NBTI-duty-cycle* for the most degraded VC that we obtain using the round-robin, i.e. *rr* policy is reported in the third column of each table. This behaves as the *rr_aggr* policy while it does not check on incoming traffic maintaining always an idle VC. Moreover, the multicolumn labeled as *aggr_rr* details for each of its sub columns the *NBTI-duty-cycle* obtained on each VC of the considered architecture. In particular, for each scenario the * symbol identifies the most degraded VC for the *rr_aggr* multicolumn to enable direct comparison against *rr* and *oracle* policies. The most degraded VC changes from scenario to scenario since initial PMOS values are randomly selected to mimic process variation impact. We provide an in-depth view of the recovery obtained using this policy since this is our sensor-less policy. The last column of each table reports the *NBTI-duty-cycle* gap for the most degraded VC between the *oracle* and the *aggressive round-robin* (*rr_aggr*) policies. This allows us to show how a sensor-less round-robin approach can behave similarly with respect to the optimal recovery offered through the use of an NBTI sensor, i.e. using the *oracle* policy. We can draw four relevant conclusions from the reported data. First, the *oracle* policy always provide the lowest *NBTI-duty-cycle*, since it can directly recover the most degraded virtual channel. At the same time, the *rr_aggr* always performs better than the *rr* policy since both share the same behavior, but *rr_aggr* can recover even the single idle virtual channel if no new traffic is present in the corresponding upstream router for the input port. This last observation supports the aggressive recovery for highly reduced degradation. For example the second row in Table II(a) reports an *NBTI-duty-cycle* of 74.7% for the most degraded VC, i.e. VC1 considering the *rr* policy. However considering the *rr_aggr* policy we obtain a value of 39.3% for the same VC, saving around 34% more *NBTI-duty-cycle*.

The second key observation is related to the scalability of the proposed approach: *rr_aggr* scales well against different 2D-mesh topologies, since the *NBTI-duty-cycle* gap between the *rr_aggr* and the *oracle* policies on the most degraded VC, i.e. column *Gap* in Table II(a) shows small increments between 4- and 16-cores. As an example, in the first and fourth rows of Table II(a) the *Gap* is

TABLE II
NBTI-DUTY-CYCLE (%) FOR THE MOST DEGRADED VC FOR USING *oracle* AND *rr* POLICIES, WHILE ALL VCS FOR *rr_aggr* POLICY. RESULTS FOR 4-CORES AND 16-CORES ARCHITECTURES WITH VARIABLE INJECTION RATE AND 2,4 VCS.

Scenario (2 VCs)	oracle	rr	rr_aggr.		Gap (rr_aggr - oracle)
			VC0	VC1	
4core-inj0.10	10.4%	63.5%	23.9%	23.8% *	23.8 - 10.4 = 13.4%
4core-inj0.20	26.5%	74.7%	39.3% *	39.3%	39.3 - 26.5 = 12.8%
4core-inj0.30	46.7%	84.8%	56.5%	56.2% *	56.2 - 46.7 = 9.5%
16core-inj0.10	20.1%	71.8%	33.5%	33.5% *	33.5 - 20.1 = 13.4%
16core-inj0.20	51.5%	88.2%	61.8% *	61.6%	61.8 - 51.5 = 10.3%
16core-inj0.30	65.3%	99.1%	73% *	73%	73 - 65.3 = 7.7%

(a)

Scenario (4 VCs)	oracle	rr	rr_aggr.			Gap (rr_aggr - oracle)
			VC0	VC2	VC3	
4core-inj0.10	0.1%	44.1%	11.9%	12.1%	12.5% *	11.7 - 0.1 = 11.6%
4core-inj0.20	1.2%	56.8%	19.8%	19.9%	19.8%	20.2 - 1.2 = 19.0%
4core-inj0.30	4.0%	64.7%	27.6% *	27.8%	27.8%	27.6 - 4.0 = 23.6%
16core-inj0.10	0.9%	60.4%	17.6%	17.6%	17.5% *	17.3 - 0.9 = 16.4%
16core-inj0.20	7.9%	77.8%	31.6%	31.4%	31.6% *	31.6 - 7.9 = 23.7%
16core-inj0.30	19.5%	81.7%	46.2%	46.2%	46.2%	46.2 - 19.5 = 26.7%

(b)

even equal to 13.4% for both the configurations. On the other hand the *NBTI-duty-cycle* differences between *rr_aggr* and *oracle* policies diverges increasing the VCs number per input port. The *oracle* policy can always aggressively recover the most degraded VC that is known, while it can exploit the increased VCs number to move traffic to other VCs. On the contrary, the *rr_aggr* is not aware of the most degraded VC, thus it must recover in a round robin fashion on a greater number of VCs. However, this aspect is not critical since real architectures usually provide up to 4 VCs.

The third most critical aspect to notice is represented by the validity of our round-robin approach that can always recover the most degraded VC within a 15% than the *oracle* policy, but without additional overhead induced by the sensor. This recovery gap raises to 26.7% if we consider the 16-cores with 4 VCs scenario as reported in last row of Table II(b). However this is a quite boundary case since NoCs usually provide 1 to 4 VCs per input port.

Last, the proposed round-robin methodology can optimize both the short term NBTI mitigation as well as the long run degradation, since we consider a stable network after 30×10^6 simulated cycles. This is shown for each scenario in Table II(a) and II(b) where the *rr_aggr* provides the same fair *NBTI-duty-cycle* for each VC.

C. Real traffic scenarios results

This section reports results obtained for 4- and 16-cores 2D-mesh topologies considering 2 and 4 VCs and a simulation time of 30×10^6 cycles. For each simulated scenario we randomly pick up a set of benchmarks, i.e. one for each core of the simulated architecture, from SPLASH2 and WCET benchmark suites.

For each combination of core number and virtual channel number we simulated 8 different random iterations, for a grand total of 32 simulated scenarios, while only the 3 most representative results for each scenario configuration are reported in Table III using *rr_aggr* policy due to space limits.

Table III(a) reports results for 4- and 16-cores considering 4 VCs while Table III(b) shows values for 4- and 16-cores considering 2 VCs per input port. It is worth to notice that the reported scenarios are selected from the whole simulation set to show the behavior of our approach against different network demands/loads.

There are three main observations. First, we observe a greater variability in network load against different configurations, that is mainly due to the selected application mix. In particular we report a network load of 0.5% per VC in the first line of Table III(b) while

TABLE III
NBTI-DUTY-CYCLE (%) FOR ALL VCS USING THE AGGRESSIVE POLICY
(RR_AGGR), CONSIDERING 4- AND 16-CORES ARCHITECTURES WITH
VARIABLE BENCHMARK MIXES.

Scenario (4 VCs)	rr_aggr			
	VC0	VC1	VC2	VC3
4core-real	6.1%	6.0%	6.2%	6.3%
4core-real	19.8%	18.1%	11.9%	12.5%
4core-real	25.4%	25.7%	19.7%	18.8%
16core-real	4.3%	6.4%	3.2%	8.2%
16core-real	14.8%	13.3%	14.0%	13.4%
16core-real	14.9.1%	15.6%	15.0%	14.9%

(a)

Scenario (2 VCs)	rr_aggr	
	VC0	VC1
4core-real	0.5%	0.5%
4core-real	12.1%	11.8%
4core-real	26.1%	26.2%
16core-real	10.1%	9.8%
16core-real	20.3%	20.6%
16core-real	37.9%	37.6%

(b)

the fourth line in the same table shows a load above 37% for each VC. Second, *NBTI-duty-cycle* reduction seems independent from the multi-core size and from the available virtual channels but it seems to be related to the traffic generated by the applications.

Last, we observe that *rr_aggr* policy reduces the *NBTI-duty-cycle* slightly unfair in some cases. For example we report the worst case gap in the third line in Table III(a), where the *NBTI-duty-cycle* of VC0 is 25.4% while VC3 is below 19%. This means that we cannot optimally balance the *NBTI-duty-cycle* if VC0 is the actual most degraded virtual channel. Although this aspect is coupled to benchmark characteristic, i.e. traffic bursts, and is hardly controllable, the discussed results provide the higher gap we found across all 32 simulations. Moreover, this unbalanced recovery issue is more evident with the increase in VC number, thus it is upper-bounded by the real number of virtual channels that is often not greater than 4.

V. CONCLUSIONS

In this paper, we presented a sensor-less methodology to minimize the impact of NBTI degradation in the most degraded virtual channel buffer of a NoC router acting on the *NBTI-duty-cycle* reduction. The validation have been carried out using an in-house developed simulation framework capable to jointly provide performance and reliability estimates. Furthermore, an accurate NBTI library has been integrated in the simulation flow to mimic NBTI sensors based on the models presented in [7], and to provide NBTI degradation cycle accurate statistics. The NBTI model has been used to assess the obtained results considering an NBTI sensor-wise policy as reference model. The round-robin methodology (*rr_aggr*) has been tested considering both synthetic and real traffic patterns scenarios, against the sensor-wise *oracle* NBTI mitigation approach, that is the best possible strategy to recover the most degraded VC. The benefits obtained with the proposed methodology are very promising, since *rr_aggr* provides *NBTI-duty-cycle* always within 25% of the *oracle* policy, while in the best case it reaches only 9% of divergence. Moreover, *rr_aggr* achieves both a short term as well as long run NBTI mitigation, since in the short term the *NBTI-duty-cycle* is greatly reduced on the most degraded VC even if unknown, while the same reduction is performed also against all the other VCs for a long term NBTI mitigation.

ACKNOWLEDGMENTS

This research work is supported by European Community Seventh Framework Programme (FP7/2007-2013), under agreements no. 247816 (2PARMA project www.2parma.eu).

REFERENCES

[1] S. Borkar, "Thousand core chips: a technology perspective," in *Annual ACM IEEE Design Automation Conference*, 2007.

[2] A. Banerjee, R. Mullins, and S. Moore, "A Power and Energy Exploration of Network-on-Chip Architectures," in *NOCS '07*. IEEE Computer Society, 2007, pp. 163–172.

[3] L. Peters, "Nbti: A growing threat to device reliability," 2004.

[4] S. Nassif, K. Bernstein, D. Frank, A. Gattiker, W. Haensch, B. Ji, E. Nowak, D. Pearson, and N. Rohrer, "High performance cmos variability in the 65nm regime and beyond," in *IEDM '07*, pp. 569–571.

[5] C. Nicopoulos, S. Srinivasan, A. Yanamandra, D. Park, V. Narayanan, C. Das, and M. Irwin, "On the effects of process variation in network-on-chip architectures," *Dependable and Secure Computing, IEEE Transactions on*, vol. 7, no. 3, pp. 240–254, 2010.

[6] C. J. M. Lasance, "Thermally driven reliability issues in microelectronic systems: status-quo and challenges," *Microelectronics Reliability*, vol. 43, no. 12, pp. 1969–1974, 2003.

[7] W. Wang, V. Reddy, A. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, "Compact modeling and simulation of circuit reliability for 65-nm cmos technology," *Device and Materials Reliability, IEEE Transactions on*, vol. 7, no. 4, pp. 509–517, 2007.

[8] X. Liang and D. Brooks, "Mitigating the impact of process variations on processor register files and execution units," in *MICRO 39*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 504–514.

[9] D. Bild, G. Bok, and R. Dick, "Minimization of nbtI performance degradation using internal node control," in *Design, Automation Test in Europe Conference Exhibition, DATE*, 2009, pp. 148–153.

[10] L. Li, Y. Zhang, J. Yang, and J. Zhao, "Proactive nbtI mitigation for busy functional units in out-of-order microprocessors," in *DATE*, 2010, pp. 411–416.

[11] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "Impact of nbtI on sram read stability and design for reliability," in *In International Symposium on Quality Electronic Design*, 2006, pp. 27–29.

[12] B. Li, L.-S. Peh, and P. Patra, "Impact of process and temperature variations on network-on-chip design exploration," in *NoCS Second ACM/IEEE International Symposium on*, 2008, pp. 117–126.

[13] U. Ogras, R. Marculescu, and D. Marculescu, "Variation-adaptive feedback control for networks-on-chip with multiple clock domains," in *DAC 45th ACM/IEEE*, 2008, pp. 614–619.

[14] X. Fu, T. Li, and J. Fortes, "Architecting reliable multi-core network-on-chip for small scale processing technology," in *Dependable Systems and Networks (DSN), IEEE/IFIP Conference on*, 2010, pp. 111–120.

[15] A. Kodi, A. Sarathy, A. Louri, and J. Wang, "Adaptive inter-router links for low-power, area-efficient and reliable network-on-chip (noc) architectures," in *ASP-DAC*, 2009.

[16] N. Agarwal, T. Krishna, L.-S. Peh, and N. Jha, "Garnet: A detailed on-chip network model inside a full-system simulator," in *ISPASS '09*, 2009, pp. 33–42.

[17] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the nbtI effect for reliable design," in *Custom Integrated Circuits Conference, CICC '06. IEEE*, pp. 189–192.

[18] A. Krishnan, C. Chancellor, S. Chakravarthi, P. Nicollian, V. Reddy, A. Varghese, R. Khamankar, and S. Krishnan, "Material dependence of hydrogen diffusion: implications for nbtI degradation," in *Electron Devices Meeting, 2005. IEEE International*, pp. 4 pp. –691.

[19] P. Singh, E. Karl, D. Sylvester, and D. Blaauw, "Dynamic nbtI management using a 45 nm multi-degradation sensor," *Circuits and Systems I, IEEE Transactions on*, vol. 58, no. 9, pp. 2026–2037, 2011.

[20] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. Brown, and A. Agarwal, "On-chip interconnection architecture of the tile processor," *Micro, IEEE*, vol. 27, no. 5, pp. 15–31, 2007.

[21] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, "Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in *DATE '09*, 2009, pp. 423–428.

[22] L.-S. Peh and W. Dally, "A delay model and speculative architecture for pipelined routers," in *HPDA 7*, 2001, pp. 255–266.

[23] D. Zoni, S. Corbetta, and W. Fornaciari, "Hands: Heterogeneous architectures and networks-on-chip design and simulation," in *IEEE ISLPED'12 International Symposium on Low Power Electronics and Design, Redondo Beach, California, USA*, aug. 2012.

[24] M. Alam, K. Kang, B. Paul, and K. Roy, "Reliability- and process-variation aware design of vlsi circuits," in *Physical and Failure Analysis of Integrated Circuits. IPFA 2007*, pp. 17–25.

[25] K. Agarwal and S. Nassif, "Characterizing process variation in nanometer cmos," in *44th ACM/IEEE Design Automation Conference*, June 2007, pp. 396–399.