Exploiting Space Diversity and Dynamic Voltage Frequency Scaling in Multiplane Network-on-Chips

Exploiting Space Diversity and Dynamic Voltage Frequency Scaling in Multiplane Network-on-Chips

Andrea Bianco, Paolo Giaccone, Mario Roberto Casu, Nanfang Li
Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Italy
Email: {firstname.lastname}@polito.it

Abstract—**Network-on-Chips (NoCs) have been proposed as a scalable solution to interconnect multiple components on a silicon chip. In this paper, we approach NoCs power optimization through Dynamic Voltage and Frequency Scaling (DVFS) under the hypothesis that two NoC *planes* are available, each with a different voltage supply and clock frequency. We show the high potential benefit of applying DVFS independently in each plane. We propose three strategies that allocate the traffic in the two planes to minimize power consumption. We evaluate them through a comparison with an ideal traffic allocation policy based on a linear programming technique. We show that load balancing in the two planes is not always the best policy. Indeed, in an unbalanced traffic scenario, concentrating the high-load flows in one plane and the remaining low-load flows in the other plane, is more power efficient.**

I. Introduction

Network-on-Chips (NoCs) offer a scalable alternative to standard busses for the interconnection of Processing Elements (PEs) in a large-scale System-on-Chip (SoC). Current SoC designs implement aggressive power minimization techniques to stay within a limited power budget. Minimizing the power consumption of each and every SoC component is mandatory, be it a PE, a memory block, or the NoC supporting the traffic between them. Dynamic Voltage and Frequency Scaling (DVFS) is a very effective technique for power optimization. In a previous paper [1] we exploit DVFS and different routing policies to reduce power consumption in single-plane NoCs. In this paper we consider the multiplane NoC architecture proposed in [2] and combine it with DVFS to boost NoC power saving without focussing on routing policies. We assume to have two parallel, independent NoC *planes*, as shown in Fig. 1. Each PE is connected to two routers, one per plane. Each plane is supplied by a different voltage and clock frequency, to exploit DVFS separately and independently.

In a DVFS setting, clock frequency and supply voltage are jointly reduced when the bit activity is low. This method exploits the dependency of a CMOS gate's dynamic power consumption on the square of supply voltage, rather than its linear dependence on clock frequency:

$$P \propto fV^2 \tag{1}$$

In our NoC-based communication framework, clock frequency f is chosen in the range $[f_{\min}, f_{\max}]$ according to the required average number of bit transitions (from 0 to 1 and vice-versa) per clock period, i.e. the average load. We define $\rho \in [0,1]$ to be such average value and consequently $f = \rho f_{\max}$. Notice that through this definition, clock frequency and bit rate become synonymous. The supply voltage V is chosen in range $[V_{\min}, V_{\max}]$. We define α as the voltage reduction factor in such a way that $V = V_{\max}/\alpha$. Due to the voltage lower bound $V_{\min}$, α is also upper bounded by $\alpha_{\max} = V_{\max}/V_{\min}$: $\alpha \in [1, \alpha_{\max}]$. We can now reformulate (1) as follows:

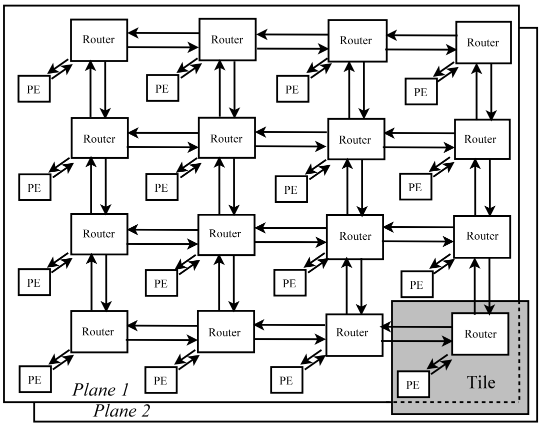

Fig. 1. A two planes NoC architecture. The interconnection network among the routers in the second plane is the same as in the first plane. Each processing element (PE) is connected to two routers, one for each plane.

$$P \propto fV^2 = \rho f_{\max}(V_{\max}/\alpha)^2 \tag{2}$$

Supply voltage and clock frequency are interrelated in a CMOS digital circuit. Given a voltage V, the maximum frequency the circuit can run at is a monotonically increasing function of V: a function which also depends on technology and circuit parameters. When the bit rate decreases, the bit duration, i.e. the clock period in our formulation, increases and the voltage can be decreased. Approximately, when decreasing the voltage by α, the bit duration can be increased by the same factor. Thus α can be interpreted as the bit expansion factor: $\alpha = 1/\rho$.

By setting the latter equivalence in (2), we get the rule of thumb that the dynamic power is a cubic function of the average load: $P_{\text{DVFS}} = \rho^3 f_{\max} V_{\max}^2$ when DVFS is fully exploited[1]. On the contrary, when no voltage and frequency scaling is adopted ($\alpha = 1$), $P_{\text{NoDVFS}} = \rho f_{\max} V_{\max}^2$ and power scales linearly with ρ. By comparing P_{DVFS} and P_{NoDVFS}, the

[1] We have substituted symbol $\propto$ with $=$ assuming a suitable normalization.

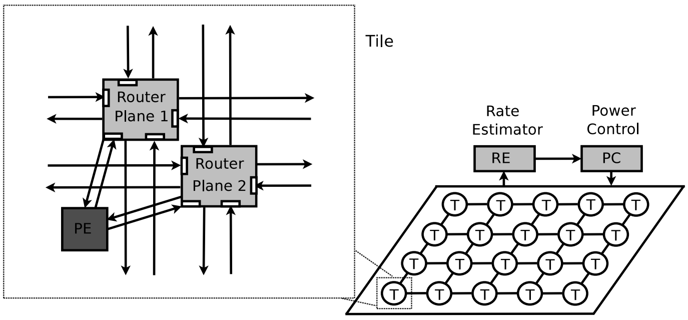

Fig. 2. The physical implementation of a two planes NoC and one tile architecture

potential power gain due to DVFS when the load is low is clear.

The motivation for exploiting multiplane NoCs together with DVFS is that the DVFS effectiveness on single plane NoCs is limited by the "bottleneck" link on chip. Indeed, consider a single plane NoC with each link loaded by some amount of traffic, which depends on the applications running on the PEs and on the routing policy. Assume that the whole chip is supplied by a single voltage[2]. The maximum loaded link, the "bottleneck", limits DVFS effectiveness, since the maximum allowed bit expansion factor α is constrained by the load on such link. In a two planes NoC with each plane working at a different voltage, we show that through a proper traffic allocation algorithm it is possible to minimize the impact of bottleneck links and save more power by concentrating most of the bottleneck flows on a single plane.

The rest of the paper is organized as follows. Sec. II presents the system model. Sec. III proposes different traffic allocation policies while Sec. IV evaluates their performance in terms of power saving. We conclude the paper in Sec. V.

II. Multi-plane NoC model

We consider a mesh with N nodes, one of the most common topologies for NoCs [5] due to its simplicity and low power consumption [6]. Two identical planes are considered, as shown in Figs. 1 and 2. The network is partitioned into "tiles", and each tile corresponds to one logical PE and two physical routers, one per plane. We assume all the PEs potentially associated with the same router as one logical PE since all the generated/received flows from/by the PEs are routed by the same router. We consider input queuing switch without virtual channels. FIFO buffers of limited size are available in PEs and routers, and wormhole routing is adopted to save buffer space. Each data packet is split into smaller units, called "flits" that are individually routed across the NoC without interleaving. Deterministic *XY routing* is used since deadlock free without virtual channels [7]: data is first routed in the X direction, until reaching the X coordinate of the destination, and then routed in the Y direction. Each flow is transferred across a single plane, to avoid the extra-complexity to route the flow in two planes, possibly with different frequency and voltage pairs. Our approach doubles network resources, but compensates this cost with a high power saving, as shown in Sec. IV.

A. Power Model

All links in a plane are supplied with a unique voltage and frequency pair, similar to a Single Voltage/Single Frequency (SVSF) chip [8], but two planes can work at different voltages chosen in range $[V_{\min}, V_{\max}]$ and at different frequencies in range $[f_{\min}, f_{\max}]$. The extreme values for frequency and voltage depend on the adopted technology and chip design. We focus only on the minimization of dynamic power due to the data transferred between routers, neglecting leakage power consumed when routers are idle. The power model at the network level is hop based; such model was proposed and validated in the literature [9], [10]. When transmitting continuously at rate r bit/s along a path of h hops, from (1) and from the equivalence between bit rate and clock frequency, the power consumption is proportional to rhV^2. To be admissible, the flow cannot overload the links along its path. This implies that $r \leq f_{\max}$ and the normalized load is defined as $\rho = r/f_{\max}$. To fully exploit DVFS and to avoid any throughput degradation, the bit duration can be expanded (at most) by a factor of $\alpha = 1/\rho = f_{\max}/r$, which has been previously defined as the expansion factor; thus, the voltage can be decreased by α, as we already noted. Using (2), the total power for transmitting a flow with rate r across h hops is

$$P = rh\left(\frac{V_{\max}}{\alpha}\right)^2 \propto \frac{hr}{\alpha^2}. \tag{3}$$

Given the minimum allowed voltage $V_{\min}$ in the considered technology, it must be $\alpha \leq \alpha_{\max}$ where $\alpha_{\max} = V_{\max}/V_{\min}$ corresponds to the maximum expansion factor. Note that $\alpha_{\max}$ is often around 2 and never more than 3 [11].

B. Traffic Model

We assume that the traffic flows among the N PEs are known. Depending on the actual application, such flows can be either known in advance, or estimated on-line by the rate estimator shown in Fig. 2. The average traffic flow from node i to j is denoted by r_{ij}, measured in [bit/s]. All the links have maximum capacity μ [bit/s], which is achievable only for maximum frequency $f_{\max}$ and maximum voltage $V_{\max}$.

Definition 1: Let $\Lambda = [\lambda_{ij}]$ be the $N \times N$ traffic matrix, in which λ_{ij} is the normalized traffic rate from node i to j, defined as $\lambda_{ij} = r_{ij}/\mu$ with $\lambda_{ij} \in [0, 1]$.

Definition 2: Given a routing policy $\mathcal{R}$ and traffic matrix Λ, $\gamma_\Lambda^\mathcal{R}$ is defined as the bottleneck load, i.e. the maximum offered load, normalized to ρ, among all the edges in the topology.

Definition 3: Λ is said to be *admissible* according to the routing policy $\mathcal{R}$ iff $\gamma_\Lambda^\mathcal{R} \leq 1$.

Since in-sequence delivery of messages belonging to same flows is crucial to avoid complex re-ordering functionality and to reduce the memory requirements, we do not consider splittable flows in our work, i.e, λ_{ij} is routed along one single

[2] We do not consider the case of each link working at an independent voltage as in [3], which is complex to implement, or other sophisticated architectures such as voltage islands [4].

path on a single plane. Even though, under this assumption it is not possible to get the minimum power that a splittable policy would achieve, as we show later, it is still possible to save considerable power. We will compare our traffic allocation policies with an ideal one that allow splittable flows and allocates flows by solving an optimization problem. The power obtained by such benchmark strategy can be then taken as the lower bound.

III. Traffic Allocation for Two-Planes NoC

The objective of our power control is to allocate the traffic for the two-planes NoC, to minimize the total power consumption while satisfying the traffic demands and the link capacity constraints. In the following section we describe a toy-scenario in which it is possible to highlight the potential power gain due to different traffic allocation algorithms.

A. A toy scenario

To fully exploit multiplane NoC for power saving, one option could be to load balance the traffic among the two NoCs. Contrary to common belief, we show that this policy is not always optimal.

As an example scenario, consider two NoC planes, supplied with voltages $V_1 = V_{\max}/\alpha_1$ and $V_2 = V_{\max}/\alpha_2$. All the links on plane i (with $i = 1, 2$) run at a maximum frequency $f_{\max}$ when $\alpha_i = 1$. Assume to have only $k+1$ traffic flows among routers that are adjacent in the topology. Thus, all the flows do not share any link. The first flow is at rate $f_{\max}$ bit/s and is denoted as "max-rate flow". All the other flows are at rate $\rho f_{\max}$ bit/s, with some small $\rho \in (0, 1)$, and are denoted as "low-rate flows". To emphasize the possible power gains due to DVFS, we assume $\alpha_{\max} = 3$, coherently with [11].

Now consider the following traffic allocation schemes:

- Route all the $k+1$ flows in the first plane. In this case, DVFS cannot be exploited because of the max-rate flow on the bottleneck link. Hence, $\alpha_1 = 1$ and, thanks to (3), the overall power consumption is

$$\begin{aligned} P^{(1)} &= \frac{f_{\max}}{\alpha_1^2} V_{\max}^2 + \frac{k\rho f_{\max}}{\alpha_1^2} V_{\max}^2 \\ &= (1 + k\rho) P_0 \end{aligned} \tag{4}$$

  having defined $P_0 = f_{\max} V_{\max}^2$.
- Balance the traffic among the two planes, assuming that flows can be split across the two planes. Thus the power is an optimistic lower bound of the actual value achievable with any load balancing scheme that do not allow flow splitting. The bottleneck link is transferring $0.5 f_{\max}$ bit/s per plane and the DVFS can be fully exploited by setting $\alpha_1 = \alpha_2 = 2$. According to (3), the overall power consumption is

$$\begin{aligned} P^{(2)} &= \frac{0.5 + k\rho/2}{\alpha_1^2} P_0 + \frac{0.5 + k\rho/2}{\alpha_2^2} P_0 \\ &= (1 + k\rho) \frac{P_0}{4} \end{aligned} \tag{5}$$

Algorithm 1: 2P-Balance

```
Input: Traffic matrix $\Lambda$
Output: $\Omega_1$, $\Omega_2$, $\alpha_1$, $\alpha_2$
$\Omega = \Omega_1 = \{(i,j), \forall i,j\}$, $\Omega_2 = \emptyset$
$\mathbb{S} = \mathtt{BF}(\Omega_1) \cap \Omega$
while $\mathbb{S} \neq \emptyset$ do
    $(i,j) = \arg\max_{(i',j') \in \mathbb{S}} \{\lambda_{i'j'}\}$
    if $\mathtt{BL}(\Omega_1 \setminus \{(i,j)\}) \geq \mathtt{BL}(\Omega_2 \cup \{(i,j)\})$ then
        $\Omega_2 = \Omega_2 \cup \{(i,j)\}$, $\Omega_1 = \Omega_1 \setminus \{(i,j)\}$
    end if
    $\Omega = \Omega \setminus \{(i,j)\}$, $\mathbb{S} = \mathtt{BF}(\Omega_1) \cap \Omega$
end while
$\alpha_1 = \mu/\mathtt{BL}(\Omega_1)$, $\alpha_2 = \mu/\mathtt{BL}(\Omega_2)$
```

- Concentrate the max-rate flow on the first plane and allocate all the other small-rate flows on the other plane. Hence, $\alpha_1 = 1$ and $\alpha_2 = \min\{1/\rho, \alpha_{\max}\}$, thus the overall power consumption becomes:

$$\begin{aligned} P^{(3)} &= \frac{1}{\alpha_1^2} P_0 + \frac{k\rho}{\alpha_2^2} P_0 \\ &= P_0 + k\rho \frac{P_0}{(\min\{1/\rho, \alpha_{\max}\})^2} \\ &= \left(1 + k \max\left\{\rho^3, \frac{\rho}{\alpha_{\max}^2}\right\}\right) P_0 \end{aligned} \tag{6}$$

Note that, in a mesh topology with N nodes, the number of links is in the order of N^2. Hence, in our toy scenario k grows as fast as N^2. Therefore, for large enough N, (4)-(6) can be approximated by:

$$\begin{aligned} P^{(1)} &\approx k\rho P_0 \\ P^{(2)} &\approx k\rho P_0/4 \\ P^{(3)} &\approx k \max\{\rho^3, \rho/\alpha_{\max}^2\} P_0 \end{aligned}$$

From the results above, by comparing $P^{(2)}$ and $P^{(3)}$ with $P^{(1)}$, it is clear the power reduction due to the DVFS. Instead, when comparing $P^{(2)}$ with $P^{(3)}$, the third policy is better than load-balancing when $\rho < 0.5$ and $\alpha_{\max} > 2$, even if the load-balancing is allowing flow splitting. In other words, contrary to some common belief, *to fully exploit DVFS and minimize power, load balancing across the planes is not always the optimal strategy*. Intuitively, it is better to "concentrate" all the high-rate flows in one plane, for which the voltage is kept at maximum, and route all the small-rate flows in the other plane, that runs at a lower voltage and fully exploits DVFS.

B. Traffic Allocation Algorithms

Inspired by the above toy scenario, we consider three algorithms to allocate flows according to different criteria. As discussed in Sec. II, no flow splitting is allowed.

The first algorithm, denoted as 2P-Balance, balances the traffic flows between the two planes. Then we choose the most convenient frequency and voltage for each plane according to the resulting bottleneck load. The corresponding pseudo code is Algorithm 1. Let Ω be the set of all the flows, identified by a couple (i, j) for the source PE i and the destination

PE j. Let Ω_1 and Ω_2 be the set of the flows that have been allocated to plane 1 and 2, respectively. They are the outputs of the allocation scheme, together with the corresponding expansion factors α_1 and α_2. The input for the algorithm is the normalized traffic matrix Λ. The algorithm starts considering all the flows in first plane ($\Omega_1 = \Omega$, $\Omega_2 = \emptyset$), which is called the *Master Plane (MP)*. Incrementally, the algorithm considers all the flows contributing to the bottleneck link in MP and evaluates the load of the new bottleneck link if each flow was moved in the second plane, named as the *Slave Plane (SP)*. In the pseudo code, function BL returns the load corresponding to the bottleneck link, whereas BF returns the set of all the flows contributing to the bottleneck links, either on MP or SP depending on whether the argument is Ω_1 or Ω_2. Every time a bottleneck flow has been considered for being moved to SP, it is removed from Ω, to avoid further consideration in the following iterations of the algorithm. The expansion factors α_i for each plane are computed as $\alpha_i = \mu/\mathtt{BL}(\Omega_i)$, since the bottleneck load is the only one affecting the DVFS. $\mathbb{S}$ is the set for the flows contributing to the bottleneck link.

Whereas 2P-BALANCE tends to distribute the flows among the two planes, the second algorithm we propose, denoted as 2P-MINI, concentrates the flows with higher load into *MP* while the flows with lower load in *SP*. The pseudo code of 2P-MINI is described in Algorithm 2.

The key difference for this algorithm compared to 2P-BALANCE is that we force the bottleneck load in SP to be low (i.e., $\mathtt{BL}(\Omega_2) \le 1/\alpha_{\max}$), to guarantee the SP running at the minimum possible frequency $f_{\min}$ and exploit fully DVFS in at least one of the two planes. On the contrary, 2P-BALANCE tends to equalize the bottleneck load among the two planes MP and SP (i.e., $\mathtt{BL}(\Omega_1) \approx \mathtt{BL}(\Omega_2)$). In the pseudo code, the additional loop in 2P-MINI (lines 10-16) considers the flows in MP that have not been considered in the first loop (lines 3-9). Since the first loop considers only flows contributing to the bottleneck link, it is still possible to move additional flows from MP to SP. This allows to load the SP as long as not violating the bottleneck load condition and to better exploit DVFS.

Algorithm 2: 2P-MINI

```
Input: Traffic matrix $\Lambda$
Output: $\Omega_1$, $\Omega_2$, $\alpha_1$, $\alpha_2$
 1: $\Omega = \Omega_1 = \{(i,j), \forall i,j\}$, $\Omega_2 = \emptyset$
 2: $\mathbb{S} = \mathtt{BF}(\Omega_1) \cap \Omega$
 3: while $\mathbb{S} \neq \emptyset$ do
 4:    $(i,j) = \arg\max_{(i',j')\in\mathbb{S}}\{\lambda_{i'j'}\}$
 5:    if $\mathtt{BL}(\Omega_2 \cup \{(i,j)\}) \le 1/\alpha_{\max}$ then
 6:       $\Omega_2 = \Omega_2 \cup \{(i,j)\}$, $\Omega_1 = \Omega_1 \setminus \{(i,j)\}$
 7:    end if
 8:    $\Omega = \Omega \setminus \{(i,j)\}$, $\mathbb{S} = \mathtt{BF}(\Omega_1) \cap \Omega$
 9: end while
10: while $\Omega \neq \emptyset$ do
11:    $(i,j) = \arg\max_{(i',j')\in\Omega}\{\lambda_{i'j'}\}$
12:    if $\mathtt{BL}(\Omega_2 \cup \{(i,j)\}) \le 1/\alpha_{\max}$ then
13:       $\Omega_2 = \Omega_2 \cup \{(i,j)\}$, $\Omega_1 = \Omega_1 \setminus \{(i,j)\}$
14:    end if
15:    $\Omega = \Omega \setminus \{(i,j)\}$
16: end while
17: $\alpha_1 = \mu/\mathtt{BL}(\Omega_1)$, $\alpha_2 = \mu/\mathtt{BL}(\Omega_2)$
```

The last algorithm we proposed is 2P-4PHASE and it is an extension of 2P-MINI. Indeed, after some preliminary tests, we noticed that 2P-MINI allocates too many flows in MP, and they can dominate the total power cost. To improve the power reduction, we propose 2P-4PHASE, that computes explicitly the power cost for each flow according to the formula in (3), to better choose which flows to move from MP to SP after running 2P-MINI. As the name suggests, 2P-4PHASE consists of 4 phases:

P1) Move a flow in $\mathtt{BF}(\Omega_1)$ to SP if $\mathtt{BL}(\Omega_2) \le 1/\alpha_{\max}$.

P2) Move a flow in Ω_1 to SP if both $\mathtt{BL}(\Omega_1)$ and $\mathtt{BL}(\Omega_2)$ do not change.

P3) Move a flow in $\mathtt{BF}(\Omega_1)$ to SP that increases the value of $\mathtt{BL}(\Omega_2)$ and the SP power cost, but the power increase is lower than the power decrease in MP.

P4) Move a flow in Ω_1 but not in $\mathtt{BF}(\Omega_1)$ to SP, that increases the value of $\mathtt{BL}(\Omega_2)$ and the SP power cost, but the power increase is lower than the power decrease in MP.

The first two phases are exactly the same as 2P-MINI and the flow movements in these phases always reduce the total power cost. The last two phases require to compute the power changes for both planes, to decide whether to move a flow or not. At the end, 2P-4PHASE guarantees that it is not possible to move a single flow to reduce the total power cost. For the lack of space, we omit the formal description of this algorithm.

To compare the performance, we also consider an ideal Minimum Power control that allows flow splitting across the two planes, which by all means is an impractical strategy, yet the one that results in the minimum possible power. Therefore we take it as the lower bound to which we compare our traffic allocation algorithms. The corresponding problem is not linear and is formalized as follows:

$$\min_{1\le\alpha\le\alpha_{\max},\, f_{ij}^{ml}\ge 0} \quad \sum_{i,j,m,l\in\mathbb{V}} f_{ij}^{ml}\frac{1}{\alpha_1^2} + \sum_{i,j,m,l\in\mathbb{V}} g_{ij}^{ml}\frac{1}{\alpha_2^2} \tag{7}$$

subject to:

$$\sum_{i,j\in\mathbb{V}} f_{ij}^{ml}\alpha_1 \le \mu \qquad \forall m,l \in \mathbb{V} \tag{8}$$

$$\sum_{i,j\in\mathbb{V}} g_{ij}^{ml}\alpha_2 \le \mu \qquad \forall m,l \in \mathbb{V} \tag{9}$$

$$\sum_{m\in\mathbb{V}} f_{ij}^{mk} - \sum_{m\in\mathbb{V}} f_{ij}^{km} = \begin{cases} \lambda_{ij}^{MP}, & k=i \\ -\lambda_{ij}^{MP}, & k=j \\ 0, & \text{otherwise} \end{cases} \qquad \forall i,j,k \in \mathbb{V} \tag{10}$$

$$\sum_{m\in\mathbb{V}} g_{ij}^{mk} - \sum_{m\in\mathbb{V}} g_{ij}^{km} = \begin{cases} \lambda_{ij}^{SP}, & k=i \\ -\lambda_{ij}^{SP}, & k=j \\ 0, & \text{otherwise} \end{cases} \qquad \forall i,j,k \in \mathbb{V} \tag{11}$$

$$\lambda_{ij}^{MP} + \lambda_{ij}^{SP} = \lambda_{ij} \qquad \forall i,j \in \mathbb{V} \tag{12}$$

where $f_{ij}^{ml} \geq 0$ is the amount of traffic, from source node i to destination node j, sent on link $m \to l$ in MP; similarly, g_{ij}^{ml} refers to SP. $\mathbb{V}$ is the set of the nodes. Eqs. (8)-(9) model the maximum bit expansion compatible with the bottleneck load in each plane, in order to guarantee the maximum throughput. Eqs. (10)-(11) are the classical flow conservation constraints. Finally, Eq. (12) guarantees to serve all the traffic in one or both the two planes.

IV. Performance Evaluation

We developed a flow level NoC simulator to evaluate the whole NoC transmission power cost. We simulated a two-planes mesh network of size 5×5 (i.e., with $N = 25$ nodes/tiles). We generated the traffic matrix Λ according to the following scenarios:

- **Uniform:** All nodes send the same amount of traffic to any other node, i.e. λ_{ij} is constant for any pair of nodes.
- **Normal:** Λ is obtained as the summation of N permutation matrices[3]. By construction $\sum_k \lambda_{ik} = \sum_k \lambda_{kj}$ for any i, j, i.e. all nodes are both source and destination of the same aggregate amount of traffic but the traffic is not uniformly distributed among each node pair.
- **Tornado:** A node in column x of the mesh sends traffic to the node in the same row and in column $(x+2) \mod 5$, i.e., two hops to the right (with wrapping).
- **Hot-spot:** Each node sends traffic with probability 0.6 to an hot-spot node located in the center of the topology, and with probability 0.4 uniformly to any other node. This traffic was proposed in [12].

To coherently compare the different scenarios under admissible traffic, for each scenario we compute the most loaded link in the case that all the flows are allocated to a single plane and are routed according to XY routing. Then, we re-normalize the load to such value, using a parameter $\rho \in [0, 1]$, denoted as the *normalized load*. More formally, given a traffic matrix Λ, the offered traffic matrix $\Lambda' = [\lambda'_{ij}]$ for the simulation is computed as:

$$\lambda'_{ij} = \rho \frac{\mu}{\gamma_\Lambda^{XY}} \lambda_{ij} \qquad (13)$$

where γ_Λ^{XY} is the bottleneck load when all the traffic is routed according to XY routing on a single plane. This definition implies that when $\rho = 1$, a naive XY routing without DVFS will saturate at least one link. For a fair comparison, values of $\rho > 1$ will not be considered since the traffic is not sustainable on a single plane.

We evaluated the power consumption under XY routing on a single plane, with or without DVFS technique, referred as *XY DVFS* and *NoDVFS* respectively in the figures. Then we compare the power cost with the proposed traffic allocation algorithms when two planes are considered, namely 2P-Balance, 2P-Mini and 2P-4Phase, respectively.

In general, the results suggest that with DVFS, the NoCs power scales cubicly as the traffic load decreases, when comparing NO DVFS with XY DVFS for a single plane. This result is not surprising for a single hardware component (i.e, CPU, router) with DVFS, but it is quite interesting for the whole chip transmission power cost. Indeed, it can be proved that:

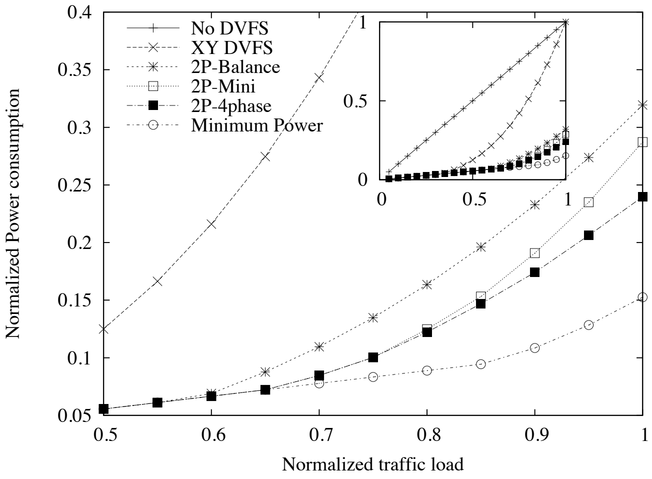

Fig. 3. Power consumption of 5×5, double plane, mesh network under normal traffic pattern and different loads.

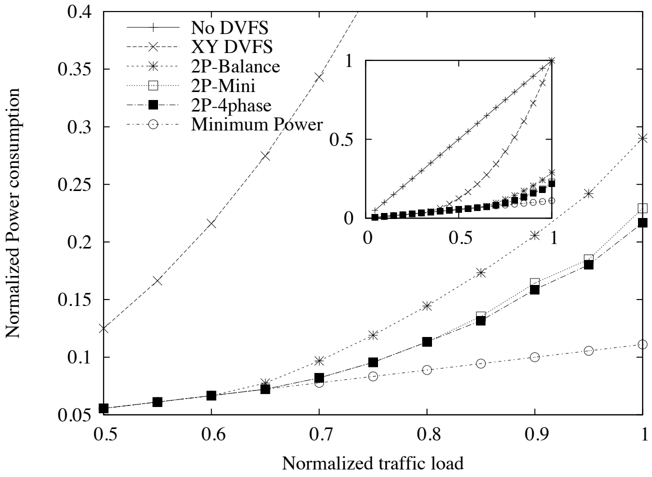

Fig. 4. Power consumption of 5×5, double plane, mesh network under hot-spot traffic pattern and different loads

Property 1: Assume ideal DVFS with unbounded $\alpha_{\max}$ (i.e., $V_{\min} = 0$). Given a routing policy $\mathcal{R}$ and a sustainable traffic, then the overall power cost of the NoC is a cubic function of the normalized load ρ: $P_{tot}(\rho) = P_{tot}^{\max} \rho^3$, for $0 < \rho \leq 1$.
The proof is omitted due to the lack of space.

Comparing the results for a single plane and for two planes, it is obvious that the power is lower for the two planes, independently of the proposed algorithms, since we exploit an additional plane and double the switching resources. According to Property 1, by halving the load ρ thanks to the two planes, we would expect a power proportional to $(\rho/2)^3$ for each plane, which implies $\rho^3/4$ for the two planes. Thus, a perfect load balancing (with flow splitting) would allow

[3] A permutation matrix is a binary square matrix in which exactly one element is equal to 1 for each row and for each column

to achieve a power reduction of factor 4. Interestingly, the observed gain can be larger than 4 using some of our proposed algorithms. Note that the lower bound provided by MINIMUM POWER shows the maximum range of power gain due to a two planes NoC, which can reach also a factor of 6 to 9 with respect to a single plane. Especially in Fig. 4, under unbalanced traffic the minimum achievable power is almost approaching the maximum achievable gain $\alpha^2_{\max} = 9$. These promising results show the great potential of multiplane NoCs to reduce the power, and our proposed algorithms are devised to exploit it.

Regarding the traffic allocation algorithms, the results obtained with 2P-BALANCE and 2P-MINI show that load concentration is better than load balancing, and the more unbalanced traffic is, the larger performance gap between the two algorithms. Indeed, since the traffic cannot be split for both policies, 2P-BALANCE does not balance the traffic between the two planes perfectly. The plane with a higher bottleneck link could accommodate a larger number of minor flows, increasing the total power cost. Instead, 2P-MINI is able to allocate unsplittable flows more efficiently, as long as there are high load and low load links, since 2P-MINI can distribute them into different planes and save more power, as also shown in the toy scenario from Sec. III. Coherently, in Fig. 4, 2P-MINI saves a factor 4.4 in the power for $\rho = 1$, compared to the case without DVFS, better than the perfect load balancing, for which the gain would be 4. This is because hot-spot traffic pattern is very unbalanced and the link load among the hot-spot node is very high whereas the link load "far away" from the hot-spot node is much lower. In general any hot-spot scenario (which is quite realistic) tends to highlight the beneficial effects of load concentration to save power.

Algorithm 2P-4PHASE is devised to exploit the space diversity and load concentration more efficiently. Indeed, Figs. 3 and 4 show a power reduction factor of 4.2 and 4.7 respectively, for $\rho = 1$, better than the other two proposed algorithms.

Not reported simulation results show that uniforma and tornado traffic are not suitable for load concentration since all link loads are exactly the same for both patterns. Even worse, the minimum achievable power obtained from MINIMUM POWER is at most 4 times lower than the single plane case. This suggests that no algorithm is able to further exploit the two planes and reduce power by a factor larger than 4.

As a summary, simulation results show that given a two planes NoC architecture with each plane running its own clock frequency for all the transmission links, load concentration is better than load balancing when the traffic pattern is unbalanced. Our algorithm 2P-4PHASE appears to be the best one to efficiently exploit the load concentration for power saving.

V. CONCLUSIONS

This paper considers a two-planes NoC architecture, in which each plane exploits Dynamic Voltage and Frequency Scaling (DVFS) independently of the other plane. We show how to leverage the spatial diversity provided by the two planes to reduce more power than the one achieved by a naive load-balancing scheme. The main idea is to concentrate the high-traffic flows in one plane and the low-traffic flows in the other plane; in this way, at least one plane can run at a reduced voltage and frequency to better exploit the beneficial effects of DVFS. We propose three traffic allocation algorithms, trading performance and complexity, and investigate the corresponding power consumptions under different traffic matrices. We compare their performance with respect to the single-plane architecture and to the optimal allocation.

REFERENCES

[1] A. Bianco, P. Giaccone, and N. Li, "Exploiting dynamic voltage and frequency scaling in networks on chip," in *IEEE 13th International Conference on High Performance Switching and Routing (HPSR)*, June 2012.

[2] S. Noh, V.-D. Ngo, H. Jao, and H.-W. Choi, "Multiplane virtual channel router for network-on-chip design," in *First International Conference on Communications and Electronics (ICCE'06)*, Oct. 2006, pp. 348–351.

[3] L. Shang, L.-S. Peh, and N. K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *9th International Symposium on High-Performance Computer Architecture*, Washington, DC, USA, 2003.

[4] U. Y. Ogras, R. Marculescu, P. Choudhary, and D. Marculescu, "Voltage-frequency island partitioning for gals-based networks-on-chip," in *Design Automation Conference*, 2007.

[5] F. G. Moraes, N. Calazans, A. Mello, L. Moller, and L. Ost, "Hermes: an infrastructure for low area overhead packet-switching networks on chip," *Integration*, vol. 38, pp. 69–93, 2004.

[6] M. Mirza-Aghatabar, S. Koohi, S. Hessabi, and M. Pedram, "An empirical investigation of mesh and torus noc topologies under different routing algorithms and traffic models," in *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools*, Washington, DC, USA, Aug. 2007.

[7] F. G. Moraes, N. L. V. Calazans, A. V. de Mello, and L. C. Ost, "Evaluation of routing algorithms on mesh based NoCs," Faculdade de informatica pucrs - Brazil, Tech. Rep., 2004.

[8] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, "An 80-tile sub-100-w teraflops processor in 65-nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, Jan. 2008.

[9] J. Hu and R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures," *IEEE Transactions on computer-aided design of intergrated circuits and systems*, vol. 24, no. 4, pp. 551–562, 2005.

[10] S. Bhat, "Energy models for network on chip components," Ph.D. dissertation, Technische Universiteit Eindhoven, 2005.

[11] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 11, pp. 1239–1252, 2005.

[12] G. F. Pfister and V. A. Norton, "Interconnection networks for high-performance parallel computers," in *Interconnection networks for high-performance parallel computers*, Los Alamitos, CA, USA, 1994.