

Power-aware Manhattan routing on chip multiprocessors

Anne Benoit, Rami Melhem, Paul Renaud-Goud, Yves Robert

▶ **To cite this version:**

Anne Benoit, Rami Melhem, Paul Renaud-Goud, Yves Robert. Power-aware Manhattan routing on chip multiprocessors. 2011. inria-00629804v2

HAL Id: inria-00629804

https://inria.hal.science/inria-00629804v2

Preprint submitted on 6 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Power-aware Manhattan routing on chip multiprocessors

Anne Benoit — Rami Melhem — Paul Renaud-Goud — Yves Robert

N° 7752

October 2011

Distributed and High Performance Computing



ISSN 0249-6399 ISRN INRIA/RR--7752--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE - RHÔNE-ALPES

Power-aware Manhattan routing on chip multiprocessors

Anne Benoit , Rami Melhem , Paul Renaud-Goud , Yves Robert

Theme : Distributed and High Performance Computing
Équipe-Projet GRAAL

Rapport de recherche n° 7752 — October 2011 — 26 pages

Abstract: We investigate the routing of communications in chip multiprocessors (CMPs). The goal is to find a valid routing in the sense that the amount of data routed between two neighboring cores does not exceed the maximum link bandwidth while the power dissipated by communications is minimized. Our position is at the system level: we assume that several applications, described as task graphs, are executed on a CMP, and each task is already mapped to a core. Therefore, we consider a set of communications that have to be routed between the cores of the CMP. We consider a classical model, where the power consumed by a communication link is the sum of a static part and a dynamic part, with the dynamic part depending on the frequency of the link. This frequency is scalable and it is proportional to the throughput of the link. The most natural and widely used algorithm to handle all these communications is XY routing: for each communication, data is first forwarded horizontally, and then vertically, from source to destination. However, if it is allowed to use all Manhattan paths between the source and the destination, the consumed power can be reduced dramatically. Moreover, some solutions may be found while none existed with the XY routing. In this paper, we compare XY routing and Manhattan routing, both from a theoretical and from a practical point of view. We consider two variants of Manhattan routing: in single-path routing, only one path can be used for each communication, while multi-paths routing allows to split a communication between different routes. We establish the NP-completeness of the problem of finding a Manhattan routing that minimizes the dissipated power, we exhibit the minimum upper bound of the ratio power consumed by an XY routing over power consumed by a Manhattan routing, and finally we perform simulations to assess the performance of Manhattan routing heuristics that we designed.

Key-words: routing; chip multiprocessor; energy; power; Manhattan; single-path; multi-paths; complexity.

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Routage de Manhattan minimisant la puissance dissipée sur des processeurs multi-cœurs

Résumé : Nous nous intéressons au routage des communications dans un processeur multi-cœur (CMP). Le but est de trouver un routage valide, c'est-à-dire un routage dans lequel la quantité de données routée entre deux cœurs voisins ne dépasse pas la bande passante maximale, et tel que la puissance dissipée dans les communications est minimale. Nous nous positionnons au niveau système : nous supposons que des applications, sous forme de graphes de tâches, s'exécutent sur le CMP, chaque tâche étant déjà assignée à un cœur. Nous avons donc un ensemble de communications à router entre les cœurs. Nous utilisons un modèle classique, dans lequel la puissance dissipée par un lien de communication est la somme d'une partie statique et d'une partie dynamique, cette dernière dépendant de la fréquence du lien. Cette fréquence est ajustable et proportionnelle à la bande passante.

La politique la plus utilisée est le routage XY : chaque communication est envoyée horizontalement, puis verticalement. Cependant si nous nous autorisons à utiliser les chemins de Manhattan entre la source et la destination, la puissance dissipée peut être considérablement réduite. De plus, il est parfois possible de trouver une solution, alors qu'il n'en existait pas avec un routage XY. Dans ce papier, nous comparons le routage XY et le routage via des chemins de Manhattan, aussi bien d'un point de vue théorique que d'un point de vue pratique. Nous considérons deux variantes du routage par chemins de Manhattan : dans un routage à chemin unique, un seul chemin peut être utilisé pour chaque communication, tandis que le routage à chemin multiples nous permet d'éclater une communication et de lui faire emprunter plusieurs routes. Nous établissons la NP-complétude du problème consistant à trouver un routage Manhattan qui minimise la puissance dissipée, exhibons la borne supérieure minimale du ratio entre la puissance dissipée par un routage XY et celle dissipée par un routage Manhattan, et pour terminer, nous effectuons des simulations pour étudier les performances de nos heuristiques de routage Manhattan.

Mots-clés : routage ; processeur multi-cœur ; énergie ; puissance ; Manhattan ; chemin unique ; chemins multiples ; complexité.

Contents

1 Introduction **4**

2 Related Work **5**

3 Framework **6**
3.1 Platform and power consumption model 6
3.2 Communications 6
3.3 Routing rules 7
3.4 Problem definition 8
3.5 Comparison of routing rules 9

4 Theoretical results **9**
4.1 Manhattan vs XY 9
4.2 NP-completeness 16

5 Heuristics **17**
5.1 Simple greedy (**SG**) 17
5.2 Improved greedy (**IG**) 17
5.3 Two-bend (**TB**) 18
5.4 XY improver (**XYI**) 18
5.5 Path remover (**PR**) 19

6 Simulations **19**
6.1 Sensitivity to the number of communications 20
6.1.1 Small communications 20
6.1.2 Mixed communications 20
6.1.3 Big communications 20
6.2 Sensitivity to the size of communications 20
6.2.1 Few communications 20
6.2.2 Some communications 21
6.2.3 Numerous communications 21
6.3 Sensitivity to the average length of communications 21
6.3.1 Numerous small communications 21
6.3.2 Some mid-weighted communications 23
6.3.3 Few big communications 23
6.4 Summary of simulations 23

7 Conclusion **24**

1 Introduction

Advances in technology enabled the integration of large numbers of processor cores into a single chip multiprocessor (CMP) and this trend is expected to continue in the future [2]. This integration creates the need for high bandwidth on-chip communication. It also increases the power consumption of a CMP and necessitates the use of clever management technique to reduce power consumption and mitigate its effect on chip temperature and reliability. A significant fraction of the CMP power is consumed in the on-chip interconnection [14, 6] and many schemes has been devised to reduce and manage this power.

In this paper, we consider CMPs with mesh interconnections and we investigate the reduction of the power consumed for on-chip communication through power-aware routing. Specifically, we consider the following problem: given a set of inter-node communications on the CMP, each with some bandwidth requirement expressed in bytes per second, find the best routes for these communications so that the total power consumed on all the communication links is minimized. Here we target the problem at the system level rather than at the application level: there are several parallel applications executing on the CMP, and each of them has been mapped onto a set of nodes, resulting in one or several communications between CMP nodes. From a system's point of view, a communication between two nodes is characterized by its requested bandwidth (in terms of bytes per second) irrespective of the application that generates the communication.

Each communication is routed from source to destination along a given path using either source routing or table-based routing. The total power consumed for the communication consists of a static part (mostly resulting from leakage) and a dynamic part (which depends on the number of bytes transmitted). An effective technique for managing the power consumption of interconnection networks is based on scaling the frequency and voltage of the communication links to match the traffic traversing those links [17]. Specifically, assume that routing the communications is such that the total traffic on a link L_ℓ resulting from all communication is D_ℓ bytes per second. Hence, to satisfy the requests and minimize power consumption, link L_ℓ must operate at a frequency f_ℓ that matches or exceeds D_ℓ/W, where W is the width of the communication link in bytes. This translates into $f_\ell = D_\ell/W$ if we have a model with continuous frequencies, or into $f_\ell = f_{\min} \geq D_\ell/W$ if frequencies are discrete, where $f_{\min}$ is the lowest frequency matching the constraint. The dynamic power dissipated by link L_ℓ is proportional to the α^{th} power of f_ℓ, where α is between 2 and 3. The total dynamic power dissipated by the communications is the sum over all links.

The most natural and widely used algorithm to handle communications in 2-dimensional meshes is XY-routing: for each communication, data is first forwarded horizontally, and then vertically, from source to destination. However, many alternate routing paths can be used in meshes. In fact, all Manhattan paths from the source to the destination are natural candidates to route the message. This freedom in routing can help dramatically reduce power consumption, when the static part of the power consumption can be neglected. For example, if there are two equal-volume communications from the same source to the same destination, the first can be routed along an XY path and the second along a YX path, thus reducing the constraint on each link by half, and thereby reducing the power consumed on that link by a factor of 2^α; this reduces the total

dynamic power consumption by $2^{\alpha-1}$. However, the number of links used is doubled in this case, and the static power consumption is doubled too. In the general case, given a set of communications, our goal is to determine one or several routing paths for each communication, so that the total power consumption is minimized. This requires that our heuristics achieve good trade-offs between static and dynamic power consumption. Note that we consider only shortest path (Manhattan) routing and we assume that a deadlock avoidance technique is used (such as resource ordering [5] or escape channels [3]).

The rest of the paper is organized as follows. In Section 2 we survey related work in the domain of routing in CMPs. Then in Section 3, we expose the framework in which our results take place. The theoretical results (worst case analysis and NP-completeness) are presented in Section 4. Finally we describe the heuristics in Section 5, and show their performance in Section 6. We conclude in Section 7.

2 Related Work

Routing algorithms for on-chip networks can be oblivious to the application traffic [16] or can dynamically adapt to that traffic [4]. If, however, the characteristics of the traffic are statically known, then routing algorithms can take advantage of that knowledge to optimize the performance of the interconnection network. For on-chip routing, there have been many proposals to design traffic-aware routes with the goal of maximizing the communication bandwidth and/or minimizing its delay [13, 8].

When power consumption of the network was recognized as a major component of the total power consumption in CMPs, many techniques have been investigated to manage the power on the links and switches of the interconnection network. Dynamic Voltage and Frequency Scaling (DVFS) and turning off unused links are among the most efficient techniques that can take advantage of the variation in traffic to reduce power [17, 1, 10]. Static knowledge of the traffic patterns obtained by compiler analysis was also used to optimize the frequency/voltage scaling of the individual interconnection links in the network [11]. Recent research proposes the adaptive use of back-gate biasing for managing the dynamic power of on-chip interconnect [9] and the dynamic redistribution of the power between the on-chip cores and routers to adapt to the variation in the computation and communication demands of applications [12].

In [18], an off-line link speed assignment algorithm was presented for energy efficient on-chip networks with voltage scalable links. Given the task graph of a periodic real-time application, genetic algorithms are used to first assign the tasks to processors and then to assign appropriate communication speeds to the communication links with the goal of reducing power consumption. In this paper, we isolate the routing problem and provide theoretical results about its complexity. We also explore a number of heuristics to solve it in polynomial time.

3 Framework

In this section, we first describe the platform and power consumption model (Section 3.1). Then we formalize the communications that need to be routed (Section 3.2), and we discuss routing rules (Section 3.3). We are then ready to formally define the optimization problem (Section 3.4). Finally, we provide a brief comparison of the routing rules in Section 3.5.

3.1 Platform and power consumption model

The target platform is a CMP (**C**hip **M**ulti**P**rocessor), composed of $p \times q$ homogeneous cores $\mathcal{C}_{u,v}$, with $1 \leq u \leq p$, $1 \leq v \leq q$, arranged along a rectangular grid. There are two unidirectional opposite links between neighbor cores. Hence, vertically, for each $(u,v) \in \{1,\ldots,p-1\} \times \{1,\ldots,q\}$, there is a link $\mathcal{L}_{(u,v)\rightarrow(u+1,v)}$ from $\mathcal{C}_{u,v}$ to $\mathcal{C}_{u+1,v}$ and a link $\mathcal{L}_{(u+1,v)\rightarrow(u,v)}$ from $\mathcal{C}_{u+1,v}$ to $\mathcal{C}_{u,v}$. Similarly, horizontally, for each $(u,v) \in \{1,\ldots,p\} \times \{1,\ldots,q-1\}$, there is a link $\mathcal{L}_{(u,v)\rightarrow(u,v+1)}$ from $\mathcal{C}_{u,v}$ to $\mathcal{C}_{u,v+1}$ and a link $\mathcal{L}_{(u,v+1)\rightarrow(u,v)}$ from $\mathcal{C}_{u,v+1}$ to $\mathcal{C}_{u,v}$.

Let $succ_{u,v}$ be the set of destination cores of the outgoing links of $\mathcal{C}_{u,v}$ (i.e., the neighbor cores). Each link has a maximum bandwidth BW but is scalable: we can choose the fraction $f_{(u,v)\rightarrow(u',v')}$ of the bandwidth of the link from $\mathcal{C}_{u,v}$ to $\mathcal{C}_{u',v'} \in succ_{u,v}$ that is active. This means that $f_{(u,v)\rightarrow(u',v')} \times BW$ bytes can go from $\mathcal{C}_{u,v}$ to $\mathcal{C}_{u',v'}$ during one second, where $0 \leq f_{(u,v)\rightarrow(u',v')} \leq 1$.

We define the set of the active links $\mathcal{A}$ such that

$$\forall (u,v) \in \{1,\ldots,p\} \times \{1,\ldots,q\}, \quad \forall \mathcal{C}_{u',v'} \in succ_{u,v},$$
$$\mathcal{L}_{(u,v)\rightarrow(u',v')} \in \mathcal{A} \;\Leftrightarrow\; f_{(u,v)\rightarrow(u',v')} \neq 0.$$

We model the power consumption of the platform as the sum of a static part (the leakage power), and a dynamic part. The leakage power P_{leak} is the power consumption of a router that is switched on, while the dynamic power depends on the active bandwidth of the link. More precisely, $P_{\text{dyn}}(\mathcal{L}_{(u,v)\rightarrow(u',v')}) = P_0 \times \left(f_{(u,v)\rightarrow(u',v')} BW\right)^{\alpha}$, where P_0 is a constant and $2 < \alpha \leq 3$ [7].

Hence, if $\mathcal{L}_{(u,v)\rightarrow(u',v')} \in \mathcal{A}$, the power dissipated to send communications through $\mathcal{L}_{(u,v)\rightarrow(u',v')}$ is $P_{(u,v)\rightarrow(u',v')} = P_{\text{leak}} + P_0 \times \left(f_{(u,v)\rightarrow(u',v')} BW\right)^{\alpha}$. If $\mathcal{L}_{(u,v)\rightarrow(u',v')}$ is inactive, then $P_{(u,v)\rightarrow(u',v')} = 0$.

3.2 Communications

Since there is no distinction between the applications, we do not have to take care of which application a communication belongs to. And as the mapping of the applications is fixed, the communications can be viewed as follows. We are given a set $\{\gamma_1, \gamma_2, \ldots, \gamma_{n_c}\}$ of n_c different communications; a communication is defined by $\gamma_i = (\mathcal{C}_{usrc(i),vsrc(i)}, \mathcal{C}_{usnk(i),vsnk(i)}, \delta_i)$, where $\mathcal{C}_{usrc(i),vsrc(i)}$ is the source core, $\mathcal{C}_{usnk(i),vsnk(i)}$ is the destination (sink) core, and δ_i is the number of bytes per second required by the message.

The routing of each communication γ_i is described as a path, denoted $path_i$. This path, of length ℓ_i, is a sequence of communication links

$$(\mathcal{L}_{(us_1,vs_1)\rightarrow(ud_1,vd_1)}, \ \ldots, \ \mathcal{L}_{(us_{\ell_i},vs_{\ell_i})\rightarrow(ud_{\ell_i},vd_{\ell_i})}),$$

such that $\mathcal{C}_{us_1,vs_1} = \mathcal{C}_{usrc(i),vsrc(i)}$, $\mathcal{C}_{ud_{\ell_i},vd_{\ell_i}} = \mathcal{C}_{usnk(i),vsnk(i)}$, and for all $\ell \in \{1,\ldots,\ell_i - 1\}$, $\mathcal{C}_{ud_\ell,vd_\ell} = \mathcal{C}_{us_{\ell+1},vs_{\ell+1}}$.

3.3 Routing rules

As stated and motivated earlier, we restrict the study to Manhattan paths, hence to shortest paths. Therefore, the length of any path for communication γ_i between $\mathcal{C}_{usrc(i),vsrc(i)}$ and $\mathcal{C}_{usnk(i),vsnk(i)}$ is $\ell_i = |usrc(i) - usnk(i)| + |vsrc(i) - vsnk(i)|$.

We define diagonals of cores $D_k^{(d)}$ (as illustrated in Figure 1) for all values of $k \in \{1,\ldots,q+p-1\}$, and for $d \in \{1,2,3,4\}$:

- $\mathcal{C}_{u,v} \in D_k^{(1)} \Leftrightarrow u + v - 1 = k$;
- $\mathcal{C}_{u,v} \in D_k^{(2)} \Leftrightarrow u + q - v = k$;
- $\mathcal{C}_{u,v} \in D_k^{(3)} \Leftrightarrow p - u + q - v + 1 = k$;
- $\mathcal{C}_{u,v} \in D_k^{(4)} \Leftrightarrow p - u + v = k$.

Note that each core is in exactly four diagonals (one for each value of d). The index d corresponds to the *direction* of the diagonal.

We also define the direction d_i of communication γ_i, and the index $ksrc(i)$ of the diagonal of direction d_i that $\mathcal{C}_{usrc(i),vsrc(i)}$ belongs to (i.e., $\mathcal{C}_{usrc(i),vsrc(i)} \in D_{ksrc(i)}^{(d_i)}$), as:

- if $usrc(i) \leq usnk(i)$ and $vsrc(i) \leq vsnk(i)$, then $d_i = 1$ and $ksrc(i) = usrc(i) + vsrc(i) - 1$;

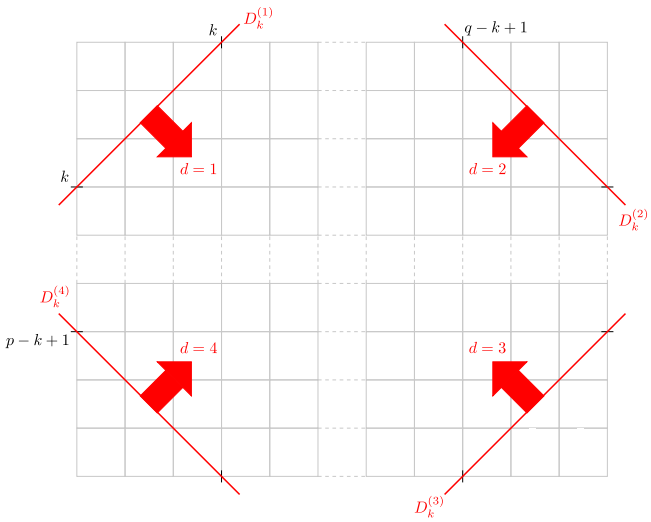


Figure 1: Location of the communications.

- if $usrc(i) \leq usnk(i)$ and $vsrc(i) > vsnk(i)$, then $d_i = 2$ and $ksrc(i) = usrc(i) + q - vsrc(i)$;
- if $usrc(i) > usnk(i)$ and $vsrc(i) > vsnk(i)$, then $d_i = 3$ and $ksrc(i) = p - usrc(i) + q - vsrc(i) + 1$;
- if $usrc(i) > usnk(i)$ and $vsrc(i) \leq vsnk(i)$, then $d_i = 4$ and $ksrc(i) = p - usrc(i) + vsrc(i)$.

With those definitions, since the paths are shortest paths, communications always move along the same direction. Formally, the ℓ^{th} communication link of $path_i$ goes from a core in $D^{(d_i)}_{ksrc(i)+\ell-1}$ to a core in $D^{(d_i)}_{ksrc(i)+\ell}$. Therefore, the index $ksnk(i)$ of the diagonal of direction d_i that $\mathcal{C}_{usnk(i),vsnk(i)}$ belongs to is $ksnk(i) = ksrc(i) + \ell_i$, i.e., $\mathcal{C}_{usnk(i),vsnk(i)} \in D^{(d_i)}_{ksrc(i)+\ell_i}$.

We are now ready to describe the different routing rules:

• *XY routing* (XY). Each communication goes horizontally first, then vertically.

• *Single-path Manhattan routing* (1-MP). The communication can take any path as described above.

• *s-paths Manhattan routing* (s-MP). A communication γ_i can be split into $s' \leq s$ distinct communications $\gamma_{i,1}, \gamma_{i,2}, \ldots, \gamma_{i,s'}$, of sizes $\delta_{i,1}, \delta_{i,2}, \ldots, \delta_{i,s'}$, where:

1. for each $s'' \in \{1, \ldots, s'\}$,
$$\gamma_{i,s''} = (\mathcal{C}_{usrc(i),vsrc(i)}, \mathcal{C}_{usnk(i),vsnk(i)}, \delta_{i,s''});$$
2. $\sum_{s''=1}^{s'} \delta_{i,s''} = \delta_i$.

Note that for each $i \in \{1, \ldots, n_c\}$, since all $\gamma_{i,j}$ (for $j \in \{1, \ldots, s\}$) have the same source core and sink core, they all have the same length ℓ_i and direction d_i. However, since communications have been split, we can now choose different paths for each part of the former communications.

• *max-paths Manhattan routing* (max-MP). This is a special case of s-MP where the number of paths is not bounded, i.e., a communication can be split into any number of distinct communications. We bound this number in Section 4.

3.4 Problem definition

We are given a CMP, a set of communications $\{\gamma_1, \ldots, \gamma_{n_c}\}$, and a routing rule (XY or s-MP), with a maximum number s of paths for a single communication. A routing is defined by:

- for each $i \in \{1, \ldots, n_c\}$, a splitting into $\{\gamma_{i,1}, \ldots, \gamma_{i,s}\}$ if $s > 1$, otherwise $\gamma_{i,1} = \gamma_i$ for XY or 1-MP;
- for each $j \in \{1, \ldots, s\}$, the path $path_{i,j}$ of $\gamma_{i,j}$;
- for all $(u, v) \in \{1, \ldots, p\} \times \{1, \ldots, q\}$ and $\mathcal{C}_{u',v'} \in succ_{u,v}$, the fraction of bandwidth $f_{(u,v)\to(u',v')}$ used for the communication from $\mathcal{C}_{u,v}$ to $\mathcal{C}_{u',v'}$.

Our goal is to find a routing that minimizes the total power consumption, while ensuring that link bandwidths are not exceeded. This last constraint adds the volume of communication going through each link and checks that the fraction of bandwidth available is not exceeded: for all $(u,v) \in \{1,\dots,p\} \times \{1,\dots,q\}$ and $\mathcal{C}_{u',v'} \in succ_{u,v}$,

$$\sum_{\substack{i \in \{1,\dots,n_c\}, j \in \{1,\dots,s\} \\ \mathcal{L}_{(u,v)\to(u',v')} \in path_{i,j}}} \delta_{i,j} \quad \leq \quad f_{(u,v)\to(u',v')} \times BW.$$

3.5 Comparison of routing rules

Note first that XY routing is a restriction of 1-MP routing, which is itself a restriction of s-MP routing.

We give here an example such that there exists a 1-MP routing that is better than the XY routing, and there exists a s-MP routing that is better than any 1-MP routing. We set $P_{\text{leak}} = 0$, $P_0 = 1$, $\alpha = 3$, $BW = 4$, and we consider two communications $\gamma_1 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 1)$ and $\gamma_2 = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 3)$. The XY routing is shown in Figure 2(a), and it leads to a power $P_{\text{XY}} = 2 \times 4^3 = 128$. The best 1-MP routing is depicted in Figure 2(b), and leads to a power $P_{1-\text{MP}} = 2 \times (1^3 + 3^3) = 56$. In the best 2-MP routing, γ_2 is split into $\gamma_{2,1} = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 1)$ and $\gamma_{2,2} = (\mathcal{C}_{1,1}, \mathcal{C}_{2,2}, 2)$ (see Figure 2(c)). The consumed power is then $P_{2-\text{MP}} = 2 \times (2^3 + 2^3) = 32$.

4 Theoretical results

In this section, we first show (Section 4.1) how much power we can save if Manhattan routing can be used instead of XY routing. Then, we prove the NP-completeness of the problem of finding a Manhattan routing in Section 4.2.

4.1 Manhattan vs XY

Throughout this section we let $P_{\text{leak}} = 0$ and $P_0 = 1$, so that routing policies aim at load-balancing communications as well as possible on all communication links. This scenario corresponds to communication-intensive applications: as the total communication volume increases, the dynamic part of the power consumption becomes more and more predominant. Note that if P_{leak} is very large and P_0 very small, then the problem becomes completely different, since the objective would be to group many communications on the same links, in order to minimize the total number of links that would be used in the end.

We start by counting the number of Manhattan paths going from $\mathcal{C}_{1,1}$ to $\mathcal{C}_{p,q}$, hence enabling us to characterize the maximum number of paths that can be used by a max-MP routing.

Lemma 1 *There are* $\binom{p+q-2}{p-1}$ *Manhattan paths going from* $\mathcal{C}_{1,1}$ *to* $\mathcal{C}_{p,q}$.

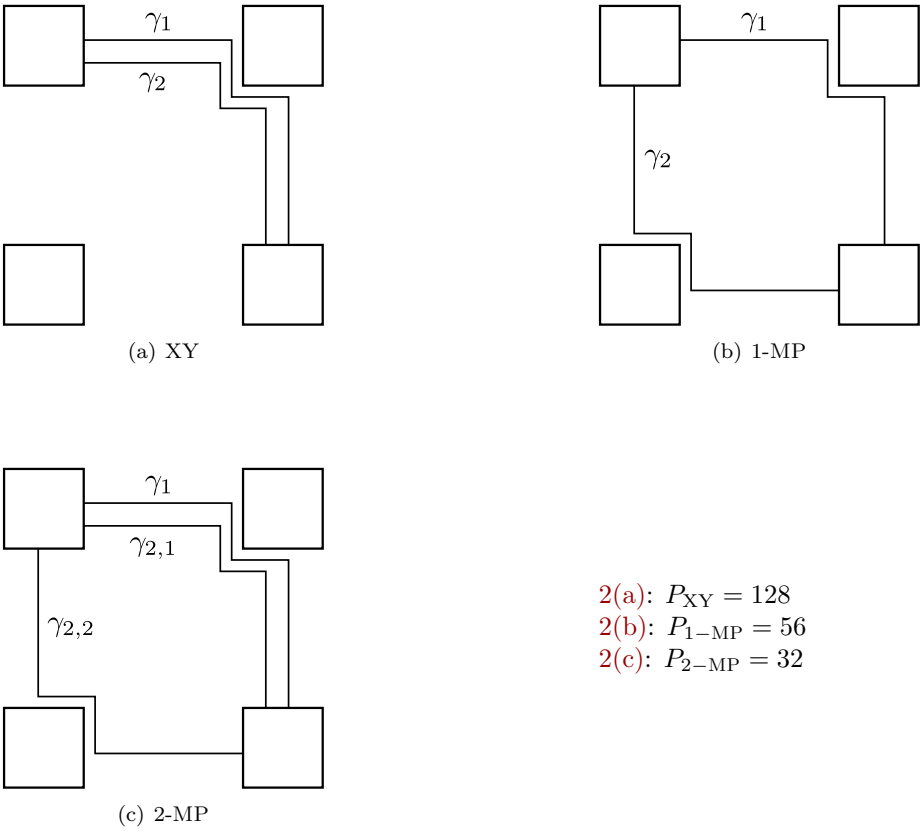


Figure 2: Comparison of routing rules.

Proof 1 *Let $N(u,v)$ be the number of paths going from $\mathcal{C}_{1,1}$ to $\mathcal{C}_{u,v}$. For all $(u,v) \in \{2,\ldots,p\} \times \{2,\ldots,q\}$, we have $N(u,v) = N(u-1,v) + N(u,v-1)$ (one path finishing vertically and one finishing horizontally). In addition, for each $v \in \{1,\ldots,q\}$, $N(1,v) = 1$ and for each $u \in \{1,\ldots,p\}$, $N(u,1) = 1$ (one single horizontal or one single vertical path). By immediate recursion, we have, for all $(u,v) \in \{1,\ldots,p\} \times \{1,\ldots,q\}$, $N(u,v) = \binom{u+v-2}{u-1} = \binom{u+v-2}{v-1}$.*

Single source and single destination. We start the comparison with communications that share the same source core and the same destination core. We study the worst case of an XY routing versus a multi-path Manhattan routing, in which the maximum number of communications is the number of different paths in the processor. This corresponds to the max-MP routing rule.

Theorem 1 *Given a $p \times q$ CMP with $q \geq p$, $q = O(p)$, and a set of communications to be routed from $\mathcal{C}_{1,1}$ to $\mathcal{C}_{p,q}$, the minimum upper bound for the ratio of the power consumed by an XY routing (P_{XY}) over the power consumed by a max-*MP *routing ($P_{\max}$) is in $O(q)$.*

Note that the result holds true for a $p \times p$ square CMP, or for a CMP with $p \geq q$ and $p = O(q)$ (with a minimum upper bound in $O(p)$).

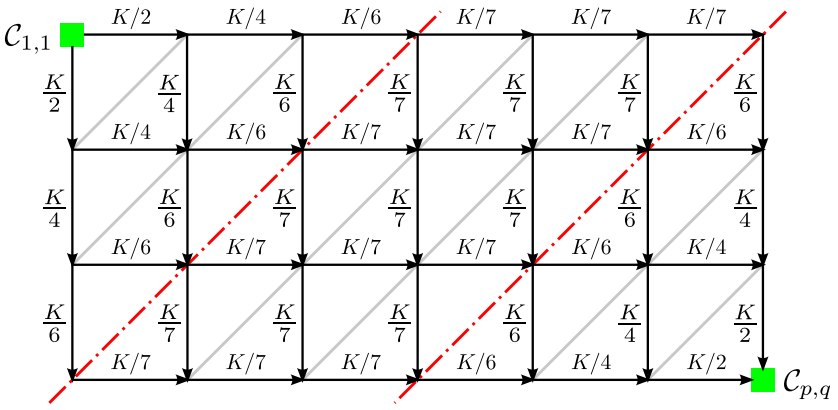


Figure 3: Ideal sharing of one communication.

Proof 2 *We first prove that an upper bound of* $P_{\text{XY}}/P_{\max}$ *is in* $O(q)$*. Then, we show that this bound can be achieved on a square CMP.*

Let K *be the total size of the communications to route (that is to say* $K = \sum_{i\in\{1,\dots,n_c\}} \delta_i$*). The XY routing is forwarding all these communications along the same route, leading to a power consumption* $P_{\text{XY}} = (p+q) \times K^\alpha$*, and therefore* P_{XY} *is in* $O(p+q) = O(q)$*.*

*All communications, even if split in multiple paths (as allowed with a max-*MP *routing), follow the same diagonals in direction* 1*. For each* $k \in \{1, \dots, q+p-2\}$*, we define by* $K_k^{(1)}$ *the sum of the* γ_i *for all* $i \in \{1, \dots, n_c\}$ *such that* $ksrc(i) \leq k$ *and* $ksnk(i) > k$*. Since all communications have the same source and destination,* $K_k^{(1)} = K$ *for each* k*. For a given* $K_k^{(1)}$*, the ideal way to map those communications is to distribute them among all the communication links from* $D_k^{(1)}$ *to* $D_{k+1}^{(1)}$ *(see Figure 3). Such a splitting cannot be achieved but provides a bound on how to load-balance the communication across the links. We have:*

$$P_{\max} \geq \sum_{k=1}^{p-1} 2k \left(\frac{K_k^{(1)}}{2k}\right)^\alpha + \sum_{k=p}^{q-1} (2p-1) \left(\frac{K_k^{(1)}}{2p-1}\right)^\alpha + \sum_{k=q}^{q+p-2} 2(q+p-k-1) \left(\frac{K_k^{(1)}}{2(q+p-k-1)}\right)^\alpha,$$

and, since $K_k^{(1)} = K$ *and* $\sum_{k=1}^{p-1} k^{1-\alpha} \geq \int_1^p dx/x^{1-\alpha}$*,*

$$P_{\max} \geq K^\alpha \left(2 \times \frac{1}{2^{\alpha-1}} \frac{1}{2-\alpha} \left(1 - p^{2-\alpha}\right) + \frac{q-p}{(2p-1)^{\alpha-1}}\right),$$

and hence $P_{\max} = O(1)$*, since* $\alpha > 2$ *and* $q = O(p)$*.*

Finally, we conclude that the worst ratio $P_{\text{XY}}/P_{\max}$ *is at most in* $O(q)$*, providing us an upper bound on this ratio.*

*We now exhibit an instance of the problem on a square CMP and a max-*MP *routing such that the ratio (in* $O(p)$*) is realized, when all communications go from the same source core to the same destination core. Let* $p = 2 \times p'$*, and*

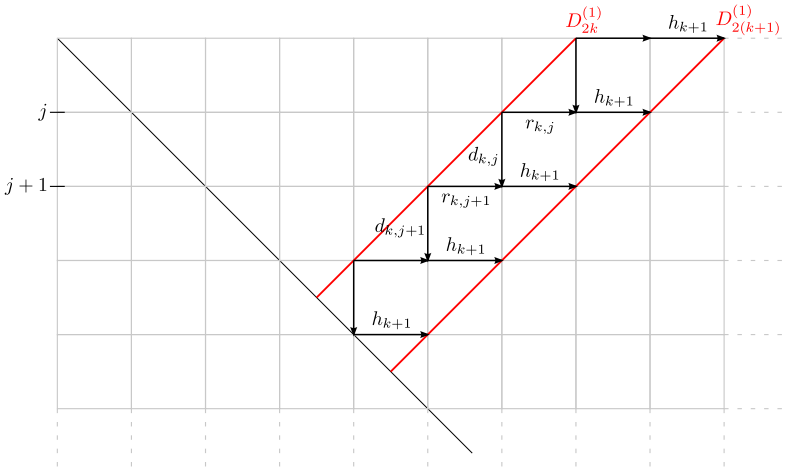


Figure 4: Routing pattern.

K be the total size of the communications to route. The power consumed with an XY routing is $P_{\text{XY}} = 2p \times K^\alpha$, and therefore P_{XY} is in $O(p)$.

Now we consider the routing pattern depicted in Figure 4. We deal with the cores in diagonal. On semi-diagonal $D^{(1)}_{2k}$, for $j \in \{1, \ldots, k\}$, the core $\mathcal{C}_{j,2k+1-j}$ on line j is sending $r_{k,j}$ communications to its right core, and $d_{k,j}$ to its down core. Between $D^{(1)}_{2k}$ and $D^{(1)}_{2(k+1)}$, for $j \in \{1, \ldots, k+1\}$, the core $\mathcal{C}_{j,2k+2-j}$ on line j is sending h_{k+1} communications to its right core.

We set:

- *for $k \in \{1, \ldots, p'\}$, $h_k = \dfrac{K}{k}$;*
- *for $k \in \{1, \ldots, p'-1\}$ and $j \in \{1, \ldots, k\}$,*

$$r_{k,j} = \frac{k+1-j}{k(k+1)}K \quad \text{and} \quad d_{k,j} = \frac{j}{k(k+1)}K \ .$$

We show that the splits and merges of communications are valid:

- *for $k \in \{1, \ldots, p'-1\}$ and $j \in \{2, \ldots, k\}$,*

$$\frac{1}{K}\left(r_{k,j} + d_{k,j-1}\right) = \frac{k}{k(k+1)} = h_{k+1} \ ;$$

- *for $k \in \{1, \ldots, p'-1\}$, $r_{k,1} = h_{k+1}$ and $d_{k,k} = h_{k+1}$;*
- *for $k \in \{1, \ldots, p'-1\}$ and $j \in \{1, \ldots, k\}$,*

$$\frac{1}{K}\left(r_{k,j} + d_{k,j}\right) = \frac{k+1}{k(k+1)} = h_k \ .$$

*What is the dissipated power with this max-*MP *routing? The total power consumption is twice the power consumed until diagonal $D^{(1)}_{2p'}$ (we define sym-*

metrical routes for the other half of the routing). Therefore, we have:

$$\frac{1}{2}P_{\max} = \sum_{k=1}^{p'} k\,(h_k)^\alpha + \sum_{k=1}^{p'-1}\sum_{j=1}^{\alpha}((d_{k,j})^\alpha + (r_{k,j})^\alpha)$$
$$\leq \sum_{k=1}^{p'} k\,(h_k)^\alpha + \sum_{k=1}^{p'-1}\sum_{j=1}^{k}(d_{k,j}+r_{k,j})^\alpha \ .$$

Also, we know that for $k \in \{1,\ldots,p'-1\}$ *and* $j \in \{1,\ldots,k\}$, $d_{k,j}+r_{k,j} = h_k$. *Therefore,*

$$\frac{1}{2}P_{\max} \leq \sum_{k=1}^{p'} k\,(h_k)^\alpha + \sum_{k=1}^{p'-1} k\,(h_k)^\alpha \leq 2K^\alpha \sum_{k=1}^{p'} \frac{1}{k^{\alpha-1}}$$
$$\leq 2K^\alpha\,(1+(1-1/p'))\ .$$

Finally, since P_{XY} *is in* $O(1)$, *the ratio* $P_{\mathrm{XY}}/P_{\max}$ *is in* $O(p)$, *which concludes the proof.*

This shows that even with an exponential number of paths, using multi-paths routing on a square CMP, in which all communications have the same source core and the same destination core, leads to a power improvement factor of up to $O(p)$, compared to an XY routing. Moreover, this factor can be reached with a max-MP routing. We did not succeed to derive this factor with a single-path routing (1-MP), and this is left as an open problem.

In the next paragraph, we investigate whether this factor can be improved when communications must be routed from/to different core pairs.

Multiple sources and multiple destinations. We now consider that several communications with different sources and destinations must be routed on the CMP. The upper bound on the improvement factor when using (multiple) Manhattan paths then becomes $O(p^{\alpha-1})$, and this ratio is reached even for a 1-MP single-path routing.

Theorem 2 *Given a* $p \times q$ *CMP with* $q \geq p$, $q = O(p)$, *and a set of communications, the minimum upper bound for the ratio of the power consumed by an XY routing (*P_{XY}*) over the power consumed by a max-*MP *routing (*$P_{\max}$*) is in* $O(p^{\alpha-1})$.

Proof 3 *Similarly to the proof of Theorem 1, we first show that an upper bound of* $P_{\mathrm{XY}}/P_{\max}$ *is in* $O(p^{\alpha-1})$. *The tightness result is given in Lemma 2, for a* 1-MP *routing.*

We start by providing a lower bound of $P_{\max}$, *following the same line of reasoning as in the proof of Theorem 1. This time, we have to consider diagonals going into each of the four possible directions: for each* $k \in \{1,\ldots,q+p-2\}$ *and for each* $d \in \{1,\ldots,4\}$, $K_k^{(d)}$ *is the sum of the* δ_i *such that* $d_i = d$, $ksrc(i) \leq k$ *and* $ksnk(i) > k$.

For a given $K_k^{(d)}$, *the* ideal *way to map those communications (with as many paths as desired) is to distribute them equally among all the communication links*

from $D_k^{(d)}$ to $D_{k+1}^{(d)}$, hence providing us with a lower bound on $P_{\max}$. Thus, if all communications go in direction d, we have:

$$\begin{aligned} P_{\max}^{(d)} \geq & \sum_{k=1}^{p-1} 2k \left(\frac{K_k^{(d)}}{2k}\right)^{\alpha} + \sum_{k=p}^{q-1} (2p-1) \left(\frac{K_k^{(d)}}{2p-1}\right)^{\alpha} \\ & + \sum_{k=q}^{q+p-2} 2(q+p-k-1) \left(\frac{K_k^{(d)}}{2(q+p-k-1)}\right)^{\alpha} \\ \geq & \frac{1}{(2p)^{\alpha-1}} \sum_{i=1}^{q+p-2} \left(K_i^{(d)}\right)^{\alpha}. \end{aligned}$$

Note that for a given communication link that is between two successive diagonals in a direction, there exists another direction such that this link is between two successive diagonals in this direction. For instance $\mathcal{L}_{(1,1)\rightarrow(1,2)}$ goes from $D_1^{(1)}$ to $D_2^{(1)}$ but also from $D_p^{(4)}$ to $D_{p+1}^{(4)}$.

However, because of the convexity of the power function, the power dissipated by a routing is less than the power dissipated if the communications in each direction would not interfere:

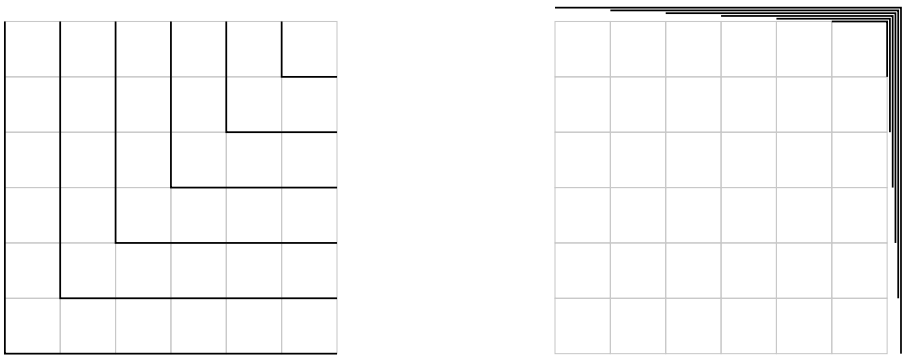
$$P_{\max} \geq \sum_{d=1}^{4} P_{\text{s-MP}}^{(d)} = \frac{1}{(2p)^{\alpha-1}} \sum_{d=1}^{4} \sum_{i=1}^{q+p-2} \left(K_i^{(d)}\right)^{\alpha}.$$

There remains to find an upper bound on P_{XY}, which is more difficult to achieve than in the single source/destination case. First, for a given sum of communications $K_k^{(d)}$ and a given occupation of the links from $D_k^{(d)}$ to $D_{k+1}^{(d)}$, note that the worst case would be to map the whole $K_k^{(d)}$ onto the maximum occupied link, because of the convexity of the power function. Let us consider now the direction 1. We relax the problem by saying that the set of communication links from $D_k^{(1)}$ to $D_{k+1}^{(1)}$ has a non empty intersection with any set of links from $D_{k'}^{(2)}$ to $D_{k'+1}^{(2)}$, $k' \in \{1, \ldots, q+p-2\}$, and with any set of links from $D_{k''}^{(4)}$ to $D_{k''+1}^{(4)}$, $k'' \in \{1, \ldots, q+p-2\}$. We keep on relaxing by placing the $K_k^{(1)}$ both on a link of the first set and on a link of the second set.

Then, for $d = 2$ and $d = 4$, $\sigma_{1,d}$ is the permutation of $\{1, \ldots, q+p-2\}$ such that $\sum_{k=1}^{p+q-2} \left(K_k^{(1)} + K_{\sigma_{1,j}(k)}^{(d)}\right)^{\alpha}$ is maximum. We map $K_k^{(1)}$ and $K_{\sigma_{1,j}(k)}^{(d)}$ onto the same link, thus $K_{\sigma_{1,j}(k)}^{(d)}$ cannot interfere anymore with another $K_{k'}^{(1)}$, hence the permutation.

We define $\sigma_{3,2}$ and $\sigma_{3,4}$ in the same way and obtain that:

$$\begin{aligned} P_{\text{XY}} \leq & \sum_{k=1}^{p+q-2} \left(K_k^{(1)} + K_{\sigma_{1,2}(k)}^{(2)}\right)^{\alpha} + \left(K_k^{(1)} + K_{\sigma_{1,4}(k)}^{(4)}\right)^{\alpha} \\ & + \left(K_k^{(3)} + K_{\sigma_{3,2}(k)}^{(2)}\right)^{\alpha} + \left(K_k^{(3)} + K_{\sigma_{3,4}(k)}^{(4)}\right)^{\alpha}. \end{aligned}$$



(a) YX (b) XY

Figure 5: Proof of Lemma 2.

Indeed, we account for all communications, in any direction. Since for all (a, b), $(a+b)^\alpha \leq (2a)^\alpha + (2b)^\alpha$, *we deduce that*

$$P_{\mathrm{XY}} \leq 2 \times 2^\alpha \sum_{k=1}^{p+q-2} \sum_{j=1}^{4} \left(K_k^{(j)}\right)^\alpha,$$

and hence P_{XY} *is in* $O(1)$.

Finally we conclude that the ratio $P_{\mathrm{XY}}/P_{\max}$ *is at most in* $O(p^{\alpha-1})$. *We prove that this ratio can indeed be achieved in Lemma 2.*

Lemma 2 *The ratio in* $O(p^{\alpha-1})$ *of Theorem 2 can be achieved with a* 1-MP *routing on a square CMP.*

Proof 4 *We consider a* $p \times p$ *CMP, where* $p = p' + 1$, *and a set of* p' *communications* $\gamma_1, \ldots, \gamma_{p'}$, *where for all* $i \in \{1, \ldots, p'\}$, $\gamma_i = (\mathcal{C}_{1,i}, \mathcal{C}_{i,p'+1}, 1)$.

The XY routing depicted in Figure 5(b) has a power consumption of $P_{\mathrm{XY}} = 2\sum_{i=1}^{p'} i^\alpha$. *We have:*

$$(p')^{\alpha+1} \leq \frac{P_{\mathrm{XY}}}{2(\alpha+1)} \leq (p'+1)^{\alpha+1} - 1,$$

hence P_{XY} *is in* $O((p')^{\alpha+1})$.

The 1-MP *routing depicted in Figure 5(a) is a YX routing, and its power consumption is:*

$$P_{1-\mathrm{MP}} = \sum_{i=1}^{p'} 2i \times 1^\alpha = p'(p'+1).$$

We conclude that in this example the ratio $P_{\mathrm{XY}}/P_{1-\mathrm{MP}}$ *is in* $O(p^{\alpha-1})$, *hence matching the upper bound.*

4.2 NP-completeness

Theorem 3 *Finding a s-MP routing that minimizes the total power consumption while ensuring that link bandwidths are not exceeded is a NP-complete problem.*

Proof 5 *Consider the associated decision problem: given a power threshold P, is there a s-MP routing that does not exceed any link bandwidth, and such that the total power consumption is not greater than P? The problem is obviously in NP: given a routing, it is easy to check in polynomial time that it is a s-MP routing (each communication is split in at most s communications), that the bandwidth on each link is not exceeded, and that the total power consumption is not greater than P.*

In fact, even without any power consideration, we prove that the problem of matching the bandwidth constraints is NP-complete. The associated decision problem is as follows: is there a s-MP routing that does not exceed any link bandwidth?

To establish the completeness, we use a reduction from 2-PARTITION. *We consider an instance $\mathcal{I}_1$ of* 2-PARTITION*: we are given n strictly positive integers $a_1, a_2, \ldots, a_n$, does there exist a subset I of $\{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$? Let $S = \sum_{i=1}^{n} a_i$.*

We build an instance $\mathcal{I}_2$ of our problem. The CMP is of size $p \times q$, with $p = 2$ and $q = (s-1)n + 2$, and the maximum bandwidth of communication links is $BW = S/2 + (s-1)n$. We have $n_c = n + q$ communications $(\gamma_1, \gamma_2, \ldots, \gamma_{n_c})$ to route. The first n communications are traversing the CMP: γ_1 goes from $\mathcal{C}_{1,1}$ to $\mathcal{C}_{p,q}$; γ_2 starts from $\mathcal{C}_{1,s}$, and so on: for each $i \in \{1, \ldots, n\}$, $\gamma_i = (\mathcal{C}_{1,(i-1)(s-1)+1}, \mathcal{C}_{p,q}, a_i + s - 1)$. The last q communications are one-hop vertical communications: for each $i' \in \{1, \ldots, q-2\}$, $\gamma_{n+i'} = (\mathcal{C}_{1,i'}, \mathcal{C}_{2,i'}, BW - 1)$; $\gamma_{n_c-1} = (\mathcal{C}_{1,q-1}, \mathcal{C}_{2,q-1}, BW - \frac{S}{2})$, and $\gamma_{n_c} = (\mathcal{C}_{1,q}, \mathcal{C}_{2,q}, BW - \frac{S}{2})$.

Note that since the routing is using only shortest paths, we do not have any choice for the routing of communications $\gamma_{n+1}, \ldots, \gamma_{n_c}$: each communication must follow the vertical link, as shown in Figure 6.

Clearly, the size of $\mathcal{I}_2$ is polynomial in the size of $\mathcal{I}_1$. We now show that $\mathcal{I}_2$ has a solution if and only if $\mathcal{I}_1$ does. Suppose first that $\mathcal{I}_1$ has a solution and let I be a subset of $\{1, \ldots, n\}$ such that $\sum_{i \in I} a_i = S/2$. For each $i \in \{1, \ldots, n\}$, we split the communication γ_i into $\gamma_{i,1}, \ldots, \gamma_{i,s}$ such that $\delta_{i,s} = a_i$ and for all $k \in \{1, \ldots, s-1\}$, $\delta_{i,k} = 1$. To define completely a path, we just have to decide for the vertical link that is used. For each $i \in \{1, \ldots, n\}$ and each $k \in \{1, \ldots, s-1\}$, $\gamma_{i,k}$ uses $\mathcal{L}_{(1,(i-1)(s-1)+k) \to (2,(i-1)(s-1)+k)}$. For each $i \in I$, $\gamma_{i,s}$ uses $\mathcal{L}_{(1,q-1) \to (2,q-1)}$ and for each $i \in \{1, \ldots, n\} \setminus I$, $\gamma_{i,s}$ uses $\mathcal{L}_{(1,q) \to (2,q)}$. No link bandwidth is exceeded and we obtain a solution to $\mathcal{I}_2$.

Suppose now that $\mathcal{I}_2$ has a solution. All source cores are on line 1, all destination cores are on line 2, and the sum of all communications is equal to the total available bandwidth of the vertical links. Therefore, each vertical link must be fully utilized, up to the maximum bandwidth BW. Since communication γ_1 is the only one that can use links $\mathcal{L}_{(1,1) \to (2,1)}$ to $\mathcal{L}_{(1,s-1) \to (2,s-1)}$, it must send a communication with $\delta_{1,k} = 1$ on each of these links, for $1 \leq k \leq s-1$. After that, this communication cannot be split anymore because the routing must use at most s paths. Because the available bandwidth of the vertical links until the

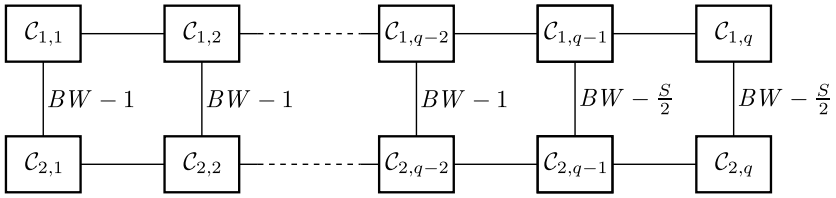


Figure 6: NP-completeness proof.

last two ones is $BW-1$*, the* a_1 *remaining bytes must wait until* $\mathcal{C}_{1,q-1}$ *or* $\mathcal{C}_{1,q}$ *to go down. We can reiterate this reasoning on the next communications* $\gamma_2, \ldots, \gamma_n$*. Finally the 2-partition comes from the fact that at most* $S/2$ *bytes can go down through* $\mathcal{L}_{(1,q-1)\to(2,q-1)}$ *and the vertical links have to be saturated. This concludes the proof.*

5 Heuristics

We present in this section several heuristics to solve the 1-MP problem. Note that we restrict ourselves to single-path routing heuristics because of the overhead incurred by routing a given communication across several paths; with the packets following different paths, reconstructing the message becomes a time-consuming task and may well involve complicated buffering policies. Instead, we envision a table-driven scheduling algorithm, which the system can safely call each time there is a new set of applications to be routed along the CMP. Furthermore, thanks to the theoretical results of Section 4, we hope significant gains over XY routing when using single-path routing, as is shown in Lemma 2.

In all the heuristics, when we deal with the communications greedily, these are sorted by decreasing number of bytes per second δ_i, which we call *weight* in the following. We have considered variants of the heuristics, where communications are sorted according to another criterion (as for instance their length, or the ratio of their weight over their length). It turns out that decreasing weights gives the best results, hence we report only this variant. The source code for all heuristics and simulations is available at [15].

5.1 Simple greedy (SG)

We route communications one by one, and for each communication, we build the path from the source core to the destination core hop by hop, the next used link being the least loaded link among the one or two possible next links. If there is a tie, we choose the link that gets closer to the diagonal, from the source core to the sink core.

5.2 Improved greedy (IG)

We pre-route the communications as if all possible links between two diagonals could be used and if we could share each communication among all those links, similarly to Figure 3. As mentioned in Section 4.1, such a pre-routing cannot be achieved, and we merely use it as a virtual initial distribution. We sort

the communications by decreasing weights, and deal with the communications greedily.

When processing a communication γ_i, we first remove all its contributions to the loads of the links (remove its pre-routing) and then find a unique route for this communication (with the pre-routing loads of the yet un-processed communications still on the links). Starting from the source core, we choose at each step the next link that will be used in the following way (there are at most two possible links). Recall that d_i is the direction of γ_i, and let k_0 be such that the current core $\mathcal{C}_{u,v}$ belongs to $D_{k_0}^{(d_i)}$. If $u = usnk(i)$ (resp. $v = vsnk(i)$), we have no choice, the next link is horizontal (resp. vertical). Otherwise, we choose the one of the two links between diagonals $D_{k_0}^{(d_i)}$ and $D_{k_0+1}^{(d_i)}$ that could lead to the lowest power consumption. For each of the two possible links, we compute a lower bound on the power consumption to reach the sink core after the chosen link: for each $k \in \{k_0 + 1, \ldots, usnk(i) + vsnk(i) - 1\}$, we keep the least loaded possible link between $D_k^{(d_i)}$ and $D_{k+1}^{(d_i)}$, and we compute the power consumption if we add communication γ_i. The lower bound is obtained by summing all these power consumptions, together with the power consumption of the link chosen between $D_{k_0}^{(d_i)}$ and $D_{k_0+1}^{(d_i)}$. Finally, we choose the link with the smallest lower bound, and we iterate until the destination core is reached.

5.3 Two-bend (TB)

We authorize at most two bends for the routing of a given communication. Once again, we sort the communications by decreasing weights. For each communication γ_i, we try all possible routings (there are at most $|usrc(i) - usnk(i)| + |vsrc(i) - vsnk(i)|$ different two-bend routings), and we keep the best one (in terms of power consumption).

5.4 XY improver (XYI)

The idea is to start with an XY-routing and to try to decrease the load of the most loaded links. We first route the communications using XY-routing, and we build a list of links, containing all the links, from the most loaded one to the least loaded one. We take the first link in the list. For each communication going through this link, we try to move it, so that it avoids this highly loaded link. More precisely, if the link is vertical, we use instead the horizontal link going to the same core, from the core that is the closest to the source core of the communication. If the link is horizontal, we instead use the vertical link going from the same core, and going to the core that is closest to the sink core of the communication. If the communication cannot be moved without violating the Manhattan path constraint, it is removed from the list of the communications going through this link.

For each communication, we compute the power consumption with the modified routes. If none of the modifications lead to a lower power consumption (or simply if no modification is available), we remove the link from the list, and iterate with the next link in the list. If at least one modification leads to a power improvement, we keep the new routing that consumes the lowest power, update the load of the links, and we sort again the list of links by decreasing

load. We then iterate. Note that there are at most $p \times q$ modifications per communication.

5.5 Path remover (PR)

Similarly to heuristic **IG**, we first assume that each communication is (virtually) pre-routed with all paths from its source node to its destination node, as in Figure 3. Then, we iteratively remove links for the communications, until there remains only one path for each of them. While there remains a communication with two or more paths, we consider the most loaded link, and the largest communication that uses this link. We remove this link from the list of links used by this communication, unless this removal would break its last remaining path for this communication. Otherwise, we consider removing the second communication, and so on.

After removing a link for a communication γ_i, we need some path cleaning operation. We update the array of possible links for γ_i (initially, it contains all Manhattan paths), in such a way that it is easy to check, when considering a subsequent deletion, if there remains a path for γ_i. For example, assume that $d_i = 1$. If we delete $\mathcal{L}_{(u,v)\rightarrow(u,v+1)}$, and if the link $\mathcal{L}_{(u,v)\rightarrow(u+1,v)}$ has already been removed, we delete as well the links $\mathcal{L}_{(u-1,v)\rightarrow(u,v)}$ and $\mathcal{L}_{(u,v-1)\rightarrow(u,v)}$. Also, if we delete $\mathcal{L}_{(usrc(i),v)\rightarrow(usrc(i),v+1)}$, then all the links $\mathcal{L}_{(usrc(i),v')\rightarrow(usrc(i),v'+1)}$ for all $v' \in \{v, \ldots, vsnk(i) - 1\}$, and $\mathcal{L}_{(usrc(i),v'')\rightarrow(usrc(i)+1,v'')}$ for all $v'' \in \{v, \ldots, vsnk(i)\}$, can be deleted. Finally, we can remove a link between diagonals $D_k^{(d)}$ and $D_{k+1}^{(d)}$ only if there are at least two valid links between those two diagonals. Please refer to [15] for further details on the implementation.

6 Simulations

As mentioned earlier, the source code for the simulations is available at [15]. The CMP is of size 8×8. Given that implementing continuous frequencies is not practical, we use the characteristics of the links described in [7]. The given discrete values for the frequencies fit our model with $P_{\text{leak}} = 16.9\,\text{mW}$, $P_0 = 5.41$ and $\alpha = 2.95$. We have then three possible frequencies: 1 Gb/s, 2.5 Gb/s and 3.5 Gb/s. Note that the heuristics presented in the previous section work with both continuous frequencies and discrete frequencies; in this latter case (which is the case of these simulations), each time that we compute the power consumption, we pick the first frequency in the set of possible frequencies higher than the required continuous frequency. We use random source and sink nodes for the communications.

In addition to the heuristics described in Section 5 (**SG**, **IG**, **TB**, **XYI**, **PR**), we run the **XY** heuristic, and we define the **BEST** heuristic as the best heuristic among all six ones on the given problem instance. Each point of the graph is obtained by averaging on 50000 sets of communications. For each simulation, we plot the inverse of the power of each heuristic (which we set to 0 if the heuristic fails), that we normalized by the inverse of the power of **BEST**, and the ratio of failures (instances where the heuristic does not find a solution).

6.1 Sensitivity to the number of communications

We first assess the impact of the number of communications, for both small, mixed and big communications. Results are reported in Figure 7.

6.1.1 Small communications

We draw the weight of each communication uniformly between 100 Mb/s and 1500 Mb/s. Concerning the capacity of the heuristics to find a solution, the failure ratio defines a clear hierarchy among the heuristics. From the worst one to the best one, we have **XY**, **SG**, **TB**, **IG**, **XYI** and finally **PR**. **XY** begins to fail with less than 10 communications. With 80 communications, **XY** and **SG** fail almost all the time, while **PR** succeeds four times out of five, **XYI** half the time, **IG** every fifth time and **TB** every tenth time. **PR** succeeds almost every time when at least one heuristic succeeds.

The power inverse keeps this hierarchy, except that **PR** is not the best heuristic when the constraints are low, because it does not care about static power. **PR** stays at 80% of **BEST** for any number of communications, but **XYI** is the best heuristic when there are less than 70 communications, and then its performance drops.

6.1.2 Mixed communications

We draw the weight of each communication uniformly between 100 Mb/s and 2500 Mb/s. With these parameters, we reach more or less the same conclusions, except that **TB** and **IG** now have almost the same results.

6.1.3 Big communications

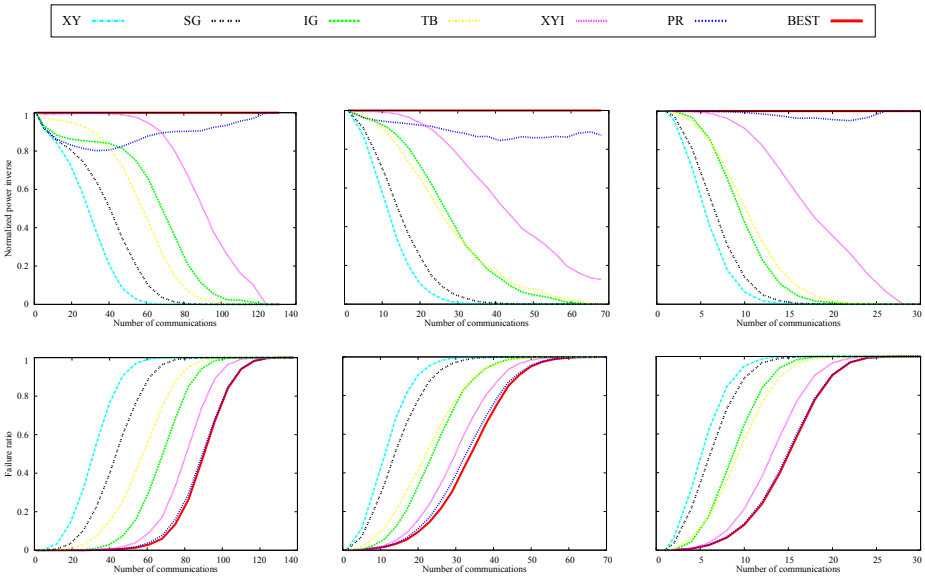
We draw the weight of each communication uniformly between 2500 Mb/s and 3500 Mb/s. With such large communications, **PR** is still the best heuristic, and it is closer to **BEST** than previously: it is always within 95% of **BEST**.

6.2 Sensitivity to the size of communications

Here we study the behavior of the heuristics, when the size of communications gets larger, for three different sizes of the communication set. Results are reported in Figure 8.

6.2.1 Few communications

In this experiment, we draw 10 communications. **XYI** is clearly the best heuristic if the average weight is less than 1600 Mb/s, otherwise **PR** is the best: in their best range, their inverse power always is up to 98% of **BEST**. One can remark that the performance of all heuristics is suddenly decreasing around 1750 Mb/s. This comes from the fact that as soon as the weight of every communication reaches 1751 Mb/s, then two communications cannot share the same link any more.



(a) Small communications (b) Mixed communications (c) Big communications

Figure 7: Sensitivity to the number of communications.

6.2.2 Some communications

We now draw 20 communications. Even though **XYI** is always at 99% of **BEST** when the average weight is less than 1750 Mb/s, it falls at only 35% of **BEST** for weights larger than 2000 Mb/s. Conversely **PR** is not affected.

6.2.3 Numerous communications

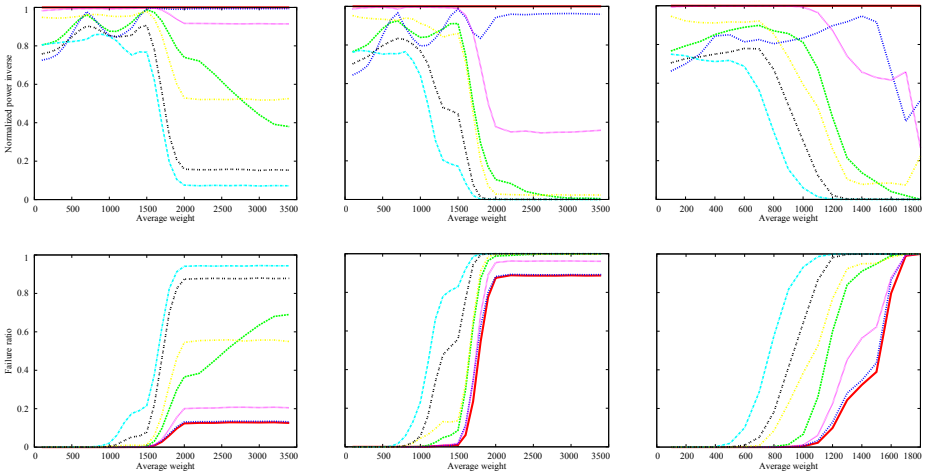
Finally we draw 40 communications. Here **XYI** is at 90% of **BEST** until 1100 Mb/s, and then falls down. **PR** is always at 60% of **BEST**.

6.3 Sensitivity to the average length of communications

Finally, we study the influence of the length of the communications, i.e., the Manhattan distance between the source core and the destination core, on the performance of the various heuristics. In both previous simulation sets, we have drawn the source core and the sink core randomly, regardless of the length of the communication. Now we draw only communications whose length is around the target average length. Results are reported in Figure 9.

6.3.1 Numerous small communications

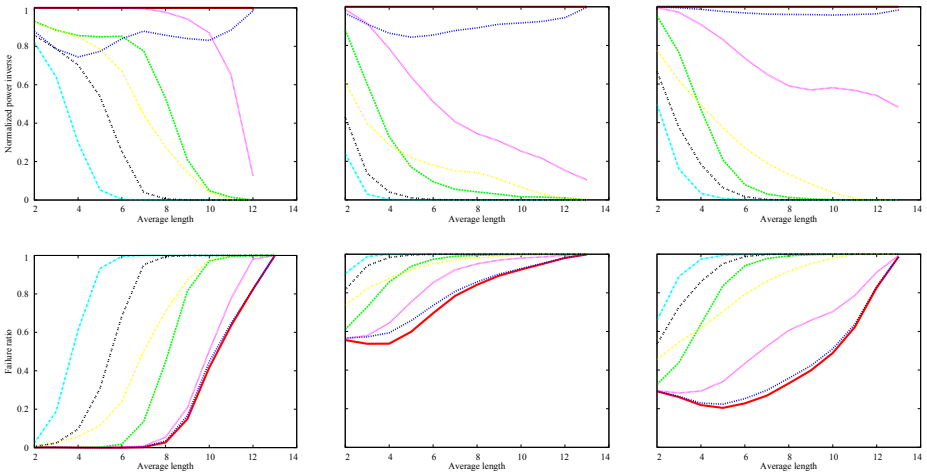
We draw 100 communications, whose weight is between 200 Mb/s and 800 Mb/s. We see that **XYI** is the best heuristic until the average length is 10, and stays at least within 90% of **BEST**. Moreover, **PR** is around 80% of **BEST** before a length of 10 and then becomes the best heuristic.



(a) Few communications (b) Some communications (c) Numerous communications

Figure 8: Sensitivity to the size of communications.





(a) Numerous small communications (b) Some mixed communications (c) Few big communications

Figure 9: Sensitivity to the length of communications.

6.3.2 Some mid-weighted communications

We draw 25 communications, whose weight is between 100 Mb/s and 3500 Mb/s. Except for a length of 2, **PR** is the best heuristic, and stays at least within 85% of **BEST**. We observe that **XYI** is the second best heuristic, decreasing regularly from 95% to 10%.

6.3.3 Few big communications

We draw 12 communications, whose weight is between 2700 Mb/s and 3300 Mb/s. For any length, **PR** is the best heuristic, within about 90% of **BEST**. Compared to **BEST**, **XYI** decreases from 95% to 40%. **IG** is slightly better than **TB** for communications of length less than 5, and after that, **TB** is better than **IG**.

The number of failures of **BEST** decreases from communications of length 2 to communications of length 5: this is because short communications are more likely to occur on X-axis or Y-axis; in this case, if two communications are on the same axis, we do not have any choice to separate these communications.

6.4 Summary of simulations

Altogether, **XYI** and **PR** are the best two heuristics: **XYI** is better than **PR** when the problem is not severely constrained, but **PR** is more and more competitive, compared to the other heuristics, when the problem becomes constrained. This last observation holds true for any constraint type, be it a high number of communications, or heavily-weighted communications. **TB** is slightly better than **IG** in almost all situations, and these heuristics return a solution in fewer cases; in addition, whenever they succeed, their solution is worse than those of **XYI** and **PR**. Finally, **SG** improves the solution given by **XY**, but this solution is far from **BEST**.

On average, over all problem instances, **XY** succeeds only 15% of the times, while **XYI** and **PR** succeeds respectively 46% and 50% of the times. This last value confirms that **PR** is the best heuristic to find a valid solution, because **BEST** succeeds 51% of the times. A first conclusion is that Manhattan routing finds three times more solutions than XY routing, which is a very significant result.

Concerning the absolute inverse of power consumption, its average value is 2.44 (resp. 2.57) times higher in **XYI** (resp. **PR**) than in **XY**, and even 2.95 times higher in **BEST**. Moreover, this dramatic gain of energy is achieved within quite a reasonable time: in average, the solution is obtained in 24 ms for **XYI**, and in 38 ms for **PR**.

We conclude this section with an interesting statistical value: averaging over all the experiments, static power accounts for 1/7-th of the total power (and dynamic power accounts for the remaining 6/7-th fraction). These fractions obviously depend upon (i) the absolute values of the parameters, and (ii) the total communication volume. For instance a lower value of the ratio P_{leak}/P_0 would favor **PR** over other heuristics.

7 Conclusion

In this paper, we have investigated the problem of routing communications in chip multiprocessors. While the most natural and widely used algorithm to handle communications is XY routing, we have shown that the consumed power can be dramatically reduced when using Manhattan routing instead of XY routing, and this with both a theoretical and a practical perspective.

On the theoretical side, we establish the NP-completeness of the problem of finding a Manhattan routing that minimizes the dissipated power, and we exhibit the minimum upper bound of the ratio of the power consumed by an XY routing over the power consumed by a Manhattan routing. We consider either that multiple paths may be used to route a single communication, or that a unique Manhattan route must be chosen (single-path). When several concurrent communications should be routed, it turns out that the worst case ratio of power consumption can be achieved even when restricting to single-path Manhattan routing.

On the practical side, we design several single-path polynomial time heuristics, and we compare them through extensive simulations. The use of a Manhattan path allows us to find valid routing solutions more than three times more often than the XY routing. Moreover, the power consumed by a Manhattan routing is always much lower than that consumed by an XY routing. Thanks to our two best heuristics, **XYI** and **PR**, power efficient solutions can be achieved in a reasonable time.

As future work, we still need to estimate how much can be gained by a single-path Manhattan routing when all communications share the same source and destination nodes. Also, we would like to establish a bound on the optimal solution for single-path Manhattan routings (or even compute the optimal solution for small problem instances), so that we could give an insight on the absolute performance of our heuristics. Finally, it may be interesting to design multi-path heuristics, since these may allow for an even better load-balance of communications throughout the CMP. Of course, one would then need to account for their potential overhead at the system level.

Acknowledgments. A. Benoit and Y. Robert are with the Institut Universitaire de France. This work was supported in part by the ANR *RESCUE* project.

References

[1] M. Alonso, S. Coll, J.-M. Martínez, V. Santonja, P. Lòpez, and J. Duato. Dynamic power saving in fat-tree interconnection networks using on/off links. In *Proceedings of HPPAC 2006*. IEEE Computer Society, 2006.

[2] G. Blake, R. Dreslinski, and T. Mudge. A survey of multicore processors. *Signal Processing Magazine, IEEE*, 26(6):26 –37, Nov. 2009.

[3] J. Duato. A new theory of deadlock-free adaptive routing in wormhole networks. *IEEE Trans. Parallel Distrib. Syst.*, pages 1320–1331, 1993.

[4] L. Gravano, G. D. Pifarré, P. E. Berman, and J. L. C. Sanz. Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks. *IEEE Transactions on Parallel and Distributed Systems*, pages 1233–1251, 1994.

[5] K. Gunther. Prevention of deadlocks in packet-switched data transport systems. *IEEE Transactions on Communications*, 29(4):512 – 524, Apr. 1981.

[6] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. A 5-ghz mesh interconnect for a teraflops processor. *Micro, IEEE*, 27(5):51 –61, Sept. 2007.

[7] J. Kim and M. A. Horowitz. Adaptive supply serial links with sub-1V operation and per-pin clock recovery. In *Proceedings of Int. Solid-State Circuits Conf.*, pages 1403–1413, 2002.

[8] M. A. Kinsy, M. H. Cho, T. Wen, E. Suh, M. van Dijk, and S. Devadas. Application-aware deadlock-free oblivious routing. In *Proceedings of the 36th annual Int. Symp. on Computer architecture*, pages 208–219. ACM, 2009.

[9] C.-Y. Lee and N. Jha. FinFET-based dynamic power management of on-chip interconnection networks through adaptive back-gate biasing. In *Proceedings of ICCD 2009, the IEEE Int. Conf. on Computer Design*, pages 350 –357, Oct. 2009.

[10] S. E. Lee and N. Bagherzadeh. A variable frequency link for a power-aware network-on-chip (NoC). *Integration*, pages 479–485, 2009.

[11] F. Li, G. Chen, and M. T. Kandemir. Compiler-directed voltage scaling on communication links for reducing power consumption. In *Proc. ICCAD'05*, pages 456–460, 2005.

[12] J. Li, W. Huang, C. Lefurgy, L. Zhang, W. E. Denzel, R. R. Treumann, and K. Wang. Power shifting in thrifty interconnection network. In *Proceedings of HPCA 2011*, pages 156–167, 2011.

[13] N. Michael, M. Nikolov, A. Tang, G. E. Suh, and C. Batten. Analysis of application-aware on-chip routing under traffic uncertainty. In *Proceedings of NOCS*, pages 9–16. IEEE Computer Society, 2011.

[14] J. D. Owens, W. J. Dally, R. Ho, D. N. J. Jayasimha, S. W. Keckler, and L.-S. Peh. Research challenges for on-chip interconnection networks. *IEEE Micro*, 27:96–108, 2007.

[15] P. Renaud-Goud. Source code for the simulations.

[16] D. Seo, A. Ali, W.-T. Lim, and N. Rafique. Near-optimal worst-case throughput routing for two-dimensional mesh networks. In *Proceedings of ISCA'05, the 32nd Int. Symp. on Computer Architecture*, pages 432 – 443, June 2005.

[17] L. Shang, L.-S. Peh, and N. Jha. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proceedings of 9th Int. Symp. on High-Performance Computer Architecture*, pages 91 – 102, Feb. 2003.

[18] D. Shin. Power-aware communication optimization for networks-on-chips with voltage scalable links. In *Proceedings of CODES+ISSS'04*, pages 170–175, 2004.



Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399