

# FlexOE : Un Algorithme de Routage Adaptatif pour le contrôle de Congestion dans les Réseaux sur Puce

OTÁVIO ALCÂNTARA DE LIMA JÚNIOR<sup>1</sup>, VIRGINIE FRESSE<sup>1</sup>, FRÉDÉRIC ROUSSEAU<sup>2</sup>

<sup>1</sup> Laboratoire Hubert Curien UMR CNRS 5516

18 rue du Professeur Benoît Lauras, 42000 Saint-Étienne, France

<sup>2</sup> TIMA Laboratory, UJF/CNRS/Grenoble INP, SLS Group

46 avenue Félix Viallet, 38031 Grenoble, France

[otavio.alcantara.de.lima.junior@univ-st-etienne.fr](mailto:otavio.alcantara.de.lima.junior@univ-st-etienne.fr), [virginie.fresse@univ-st-etienne.fr](mailto:virginie.fresse@univ-st-etienne.fr),

[frederic.rousseau@imag.fr](mailto:frederic.rousseau@imag.fr)

**Résumé** - Les réseaux sur puce sont actuellement les architectures de communications les plus adaptées pour les systèmes embarqués multicœurs. Ces réseaux doivent être capables de supporter différents flux de données en temps réel. Leurs performances dépendent directement de la stratégie de routage adoptée. Dans ce travail, nous proposons un nouvel algorithme de routage adaptatif appelé FLEXOE basé sur l'information de congestion de voisinage, sur le routage multi-chemins et sur le modèle Odd-Even. L'algorithme FLEXOE est développé puis intégré dans un réseau sur puce (NoC Hermes) avant d'être implanté sur FPGA. Les résultats des évaluations mettent en évidence des performances supérieures aux algorithmes de routage de référence pour certains scénarios de trafic.

**Abstract** - Networks-on-Chip (NoCs) are currently the most appropriate communication structure for many-core embedded systems. These networks support many real-time data flows. Their performances directly depend on the routing strategy. In this paper, we present a new congestion-aware routing algorithm (FlexOE) based on simple and flexible scheme of prioritized sets of rules. These sets of rules are based on the Odd-Even turn model, minimal paths checking, congestion information from adjacent routers and the availability of output paths. The algorithm FlexOE is integrated on a Hermes NoC, and then implemented on an FPGA. The evaluation results point out that FlexOE has greater performances than reference algorithms for some test scenarios.

## 1 Introduction

Les réseaux sur puce (NoC Network On Chip) sont actuellement les structures de communications les plus adaptées pour les systèmes embarqués multicœurs. Ces réseaux doivent être capables de supporter différents flux de données en temps réel. Les performances du réseau dépendent directement de la stratégie de routage.

La congestion d'un réseau est un facteur essentiel dans ses performances globales. Généralement, un NoC congestionné peut être considéré comme un réseau sur-utilisé dans lequel circule un trop grand nombre de paquets dans une région. Cette surcharge amène à une dégradation de performances de la structure (goulot d'étranglement et interblocages). Une solution pour réduire cette dégradation consiste à utiliser un algorithme de routage basé sur les informations de congestion permettant de gérer au mieux les flux de données dans le réseau. Ainsi, le routage utilisant ces informations de congestion va permettre de router les paquets vers un chemin moins congestionné pour équilibrer les flux dans le réseau et éliminer les goulots d'étranglement. Ces types d'algorithmes sont basés sur un ensemble précis de règles définies (pas d'adaptation sur le nombre de règles), telles que le schéma de routage adaptatif, la détection de la congestion et bien d'autres. La règle de congestion est considérée en priorité, les autres règles étant secondaires.

Dans ce papier, nous proposons un nouvel algorithme de routage pour les réseaux 2D maillé. Cet algorithme appelé FlexOE est un algorithme adaptatif et basé sur un

ensemble de règles priorisées telles que l'information de congestion de voisinage, sur le routage multi-chemins et sur le modèle Odd-Even. Dans notre cas, toutes les règles sont considérées au même niveau de priorité. On applique un « filtrage adaptatif et dynamique » sur les ensembles de règles pour trouver le chemin de routage le plus approprié. L'algorithme FlexOE développé est ensuite intégré dans un NoC existant (NoC Hermes [1]), puis implanté sur FPGA.

## 2 Etat de l'art et motivation

### 2.1 Etat de l'art

Ce papier porte principalement sur les algorithmes de routage basé sur les informations de congestion. Un algorithme de routage basé sur les informations de congestion route les paquets vers un chemin moins congestionné et ceci en fonction de l'état des routeurs du NoC. Il existe de nombreux algorithmes de routage de ce type, partageant certaines caractéristiques telles que le routage adaptatif et des mécanismes de détection de congestion.

Ces algorithmes de routage utilisent l'information provenant des routeurs adjacents. L'information de congestion est basée sur l'utilisation de buffers d'entrée ou/et sortie du routeur. Des algorithmes adaptatifs basés sur le routage XY sont facilement mis en oeuvre [2][3][4]. Le modèle Odd-Even est également utilisé en [5] et [6] du fait de sa grande adaptabilité par rapport aux autres algorithmes partiellement adaptatifs [7]. Le

schéma de routage DyAD [8] combine les avantages des algorithmes adaptatifs et ceux déterministes. L'information de congestion est prise en compte afin de choisir la meilleure stratégie de routage. Lorsque le réseau n'est pas congestionné, un routage déterministe est exploité. Dans le cas d'un réseau congestionné, un algorithme de routage adaptatif est mise en oeuvre.

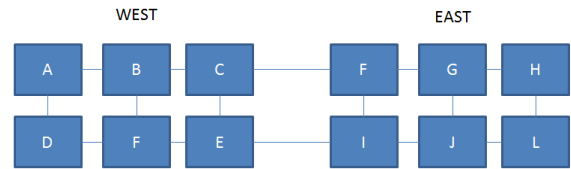
DyXY [9] est un algorithme de routage basé sur l'information de congestion. Cet algorithme envoie les paquets à travers le chemin plus court entre le nœud source et le nœud destination. Dans le cas de plusieurs chemins plus courts, le chemin le moins congestionné est sélectionné. Cet algorithme nécessite deux canaux virtuels pour éviter des inter-blocages. Une extension du modèle DyXY, appelé EDXY, est proposée en [10]. Cet algorithme peut éviter les congestions lorsque le router courant est adjacent au rang (colonne) du nœud de destination. Cette modification nécessite l'insertion de deux fils et d'un circuit dédié à la transmission d'information de congestion pour chaque rang et pour chaque colonne. Le travail de [5] présente un algorithme de routage basé sur l'information de congestion pour des NoC maillé. Cet algorithme est basé sur le modèle Odd-Even et permet le routage à travers des chemins minimaux et non minimaux. Chaque router possède des informations sur l'utilisation des buffers de ses routeurs voisins. Il calcule un score basé sur ces informations (taux d'utilisation des buffers). Le choix du router de sortie dépend du score de chaque voisin, la décision se faisant sur le même modèle que l'algorithme odd-even.

## 2.2 Motivation

Un algorithme de routage basé sur l'information de congestion est mis en oeuvre afin de réduire la latence des paquets à travers la sélection du chemin le moins congestionné. Certains cas critiques peuvent se présenter. On considère le réseau de la Fig. 1 avec un réseau qui est découpé en deux parties, une partie Ouest et une partie Est. On suppose que tout le trafic entre les parties Ouest et Est s'effectue sur le lien entre le router C et le router F (correspondant au lien CF sur la figure), ce qui génère un lien congestionné. La mise à jour de l'information de congestion sur le lien CF entraîne le routage du trafic vers le lien entre le router E et le router I (correspondant au lien EI), rendant ce lien congestionné et décongestionnant le lien CF. Une mise à jour entraîne une modification sur les informations de congestion du réseau, le lien CF devenant le chemin moins congestionné. Ainsi, les deux liens deviennent successivement congestionnés et non congestionnés, les trafic étant routé vers le lien CF puis le lien EI, Le phénomène de congestion ne se dissipe pas dans le réseau, il migre d'un lien vers un autre. Un des deux liens reste toujours sous utilisé, l'autre sur-utilisé. Il devient donc intéressant de proposer un algorithme qui supprime les goulots d'étranglement et répartit les trafics dans le réseau.

Pour résoudre ce problème, nous proposons un algorithme de routage basé sur un schéma simple et flexible de règles priorisées, qui permet d'équilibrer des

charges de trafic dans le réseau. Ces ensembles de règles sont basées sur le modèle Odd-Even, la vérification des chemins minimaux, les informations de congestion des routeurs adjacents et la disponibilité du chemin de sortie. L'équilibre de charge est obtenu grâce à des choix du chemin le moins congestionné qui varie périodiquement. Le chemin de sortie est choisi grâce à la vérification d'un ensemble de règles.



**Figure 1- Architecture de réseau découpé en deux parties (ouest et est). Un trafic fort entre les deux entraîne une congestion sur les deux liens.**

## 3 Principe de FlexOE

FlexOE est un algorithme de routage adaptatif qui permet de sélectionner le chemin de sortie le plus approprié pour un paquet. Le choix du lien de sortie se fait en fonction d'un ensemble de règles priorisées. Ces règles sont classées avec différents niveaux de priorité de type descendant. Le premier critère associé doit être un algorithme de routage multi-chemins. Les autres critères sont des informations de mesure du réseau ou du routeur. Le nombre et type de règles sont définis par l'utilisateur. Ces règles sont analysées de manière séquentielle (mais vérifiées de manière parallèle). Le principe de l'algorithme consiste à vérifier tout d'abord que toutes les règles sont vraies pour un ou plusieurs chemins de sortie. Si aucun chemin de sortie ne répond à cet ensemble de règles, la règle la moins importante est supprimée et ce nouvel ensemble de règles est de nouveau testé pour chaque sortie. Les chemins de sortie sont vérifiés dans une séquence prédéterminée qui change de manière cyclique.

Dans ce travail, nous avons choisi quatre critères de sélection pour implanter cet algorithme de routage basé sur les informations de congestion. Le premier critère vérifie si un chemin de sortie respecte les restrictions du modèle Odd-Even [7]. Le modèle Odd-Even est un algorithme de routage multi-chemins sans interblocage, modèle utilisé dans les réseaux 2D maillé. Le deuxième critère vérifie que le chemin de sortie est minimal. Ensuite, le troisième critère vérifie la faible congestion du chemin de sortie. Le dernier critère teste si le chemin de sortie est libre pour une éventuelle utilisation.

La Fig. 2 montre l'organisation des règles et la gestion de ces règles. Tout d'abord, on définit un ensemble de règles ainsi que le niveau de priorité pour chacune. Le premier ensemble de règles englobe toutes les règles prédéfinies. A l'arrivée d'un paquet dans le routeur, l'ensemble des règles sont appliquées pour l'ensemble des sorties du routeur par rapport à l'adresse de destination associée au paquet entrant. Les règles sont vérifiées dans l'ordre de priorité précédemment défini. Si l'ensemble des règles est vrai pour une sortie,

le routeur envoie le paquet vers cette sortie. Si l'ensemble des règles n'est vrai pour aucune des sorties, l'algorithme élimine la règle la moins importante et ré-vérifie le nouvel ensemble de règles pour chacune des sorties. A chaque échec de vérification des règles sur les sorties, une règle est supprimée et le nouvel ensemble de règles est de nouveau testé sur l'ensemble des sorties.

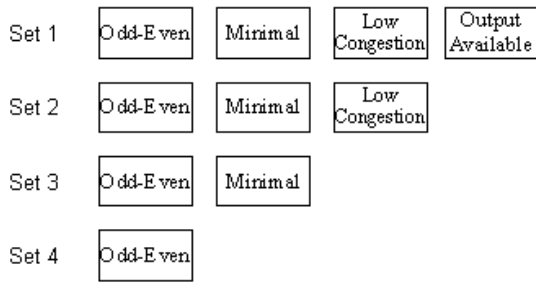


Figure 2- Organisation de l'ensemble des règles

Le processus est répété en supprimant successivement la règle la moins significative. L'implémentation de cet algorithme est réalisée au moyen d'un IP intégré dans chaque routeur pour vérifier chaque règle individuellement et l'ensemble des règles. Ce bloc récupère les informations de congestion des routeurs adjacents via des liens dédiés, les informations étant récupérées sur chaque cycle d'horloge. Les autres règles sont vérifiées en parallèle dans cet IP. Les informations de disponibilité des sorties sont enregistrées dans une table de correspondance interne qui est mise à jour pour chaque routage de paquet. De plus, ce bloc dédié met à jour un tableau adressable pour un index, ce tableau étant mis à jour à chaque cycle d'horloge. Ce tableau indique les règles qui sont vraies pour chaque chemin de sortie. Après la mise à jour du tableau, le circuit de vérification fait l'analyse du tableau en un cycle d'horloge (l'ensemble des règles est vérifié en parallèle), permettant ainsi de définir le chemin de sortie choisi sur le front d'horloge suivant.

Afin d'équilibrer les paquets dans le réseau, l'ordre des liens sélectionnés change dans le temps. Par exemple, sur le premier cycle d'horloge la séquence est : SUD, NORD, OUEST, EST. La séquence est mise à jour sur chaque cycle d'horloge dans le réseau. Ainsi, en cas de trafic fortement congestionné, ce qui correspond à l'arrivée d'un paquet sur chaque cycle, la modification de l'ordre des liens permet de répartir les données de manière équilibrée dans le réseau.

## 4 Expérimentations

Les expérimentations sont effectuées à travers l'intégration de l'algorithme de routage FlexOE dans le NoC Hermes en intégrant deux canaux virtuels. Les algorithmes de routage classiques: XY, DyXY et DyAD-OE [8] sont également intégrés dans le Hermes afin de pouvoir comparer les performances de l'algorithme de routage proposé par rapports aux algorithmes de routage standard existant.. Les architectures d'évaluation sont implantées sur la plateforme d'évaluation ML605, qui contient un FPGA Virtex 6 XC6LXFPGA. Les expérimentations sont

effectuées en utilisant les environnements logiciels Atlas, ModelSim et Xilinx ISE 12.1. Les scénarii des tests sont basés sur les besoins en communication pour des applications d'imagerie. La latence moyenne pour transmettre en ensemble de paquet est extraite pour chaque scenario de trafic.

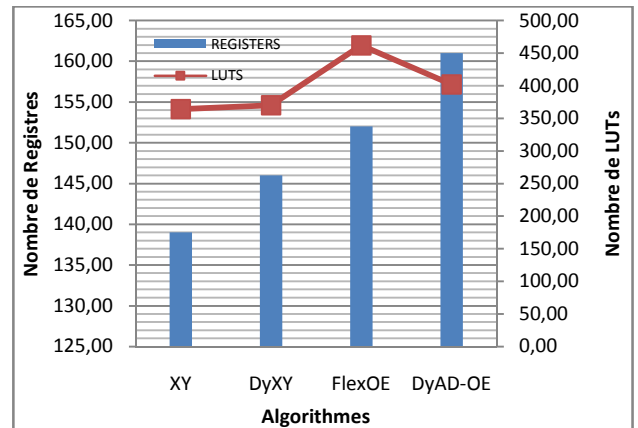


Figure 4 – Nombre de registres et LUTs

### 4.1 Performances en termes de ressources

Nous présentons l'analyse de ressources des algorithmes implémentés sur FPGA. La Figure 4 présente le nombre de registres et LUTs nécessaires pour un routeur du NoC Hermes, le routeur intégrant chaque algorithme présenté. On constate que le nombre de ressources nécessaires est légèrement supérieur que celui du DyXY et inférieur que celui de DyAD.

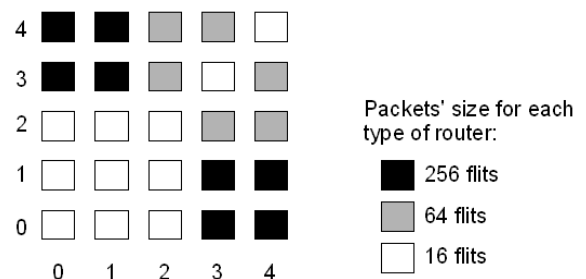


Figure 3 – Scenario de transfert de données dans un NoC 5x5 basé sur une application d'imagerie.

Comparé à l'algorithme XY (nécessitant le moins de ressources), FlexOE nécessite 13 registres et 170 LUTs supplémentaires. Ceci correspond à 13% de registres en plus et 100% de LUTs en plus (les LUTs étant utilisées pour les vérifications des règles et du choix de chemin de sortie). Comparé à l'algorithme le plus gourmand en termes de ressources (DyAD), FlexOE utilise 94% des registres et 75% des LUTs de DyAD, soit 152 registres et 340 LUTs. Cependant, les ressources mises en œuvre sont faibles par rapport aux ressources disponibles dans les FPGAs actuels, soit 2432 registres et 5440 LUTs pour un NoC de taille 4x4. Par comparaison, l'algorithme XY nécessite 2224 registres et 2720 LUTs.

### 4.2 Performances temporelles

Pour l'évaluation temporelle nos expérimentations portent sur le NoC Hermes 4x4, Plusieurs paquets de

tailles différentes (16, 64 et 256 flits) sont envoyés de plusieurs nœuds sources vers plusieurs nœuds destination, comme représenté en Figure 4.. Chaque router transmet un ensemble de 30 paquets. La latence totale pour transmettre un ensemble de paquets est mesurée pour une charge injectée variant de 10% jusqu'à 100%. Le modèle de trafic transpose est utilisé. On calcule la latence moyenne de tous les paquets.

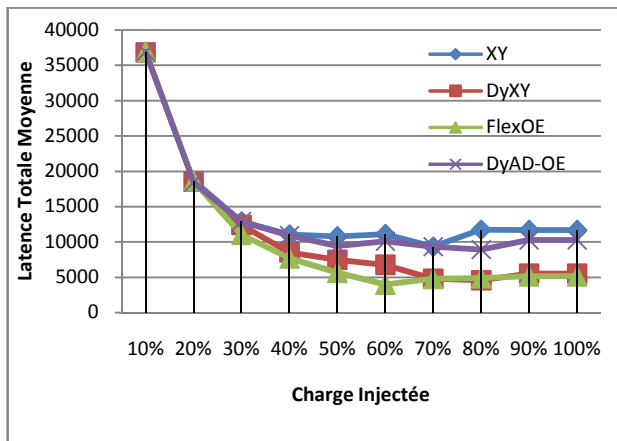


Figure 5 – Latence totale moyenne (cycles d'horloge) pour un modèle de trafic transpose uniforme (30 paquets)

La Fig. 4 montre la relation entre la charge injectée et la latence totale. En dessous d'une charge de 30%, les latences moyennes sont les mêmes (ce qui s'explique par le fait que cela ne permet pas de congestionner le réseau). FlexOE donne les meilleures performances entre 30 et 70%. Même si DyXY semble proche, la latence moyenne est d'environ 4 000 cycles pour FlexOE et 7 000 cycles pour DyXY (soit une augmentation de 280% pour 3 000 cycles supplémentaires). Au-delà de 70% les deux algorithmes donnent des performances temporelles identiques.

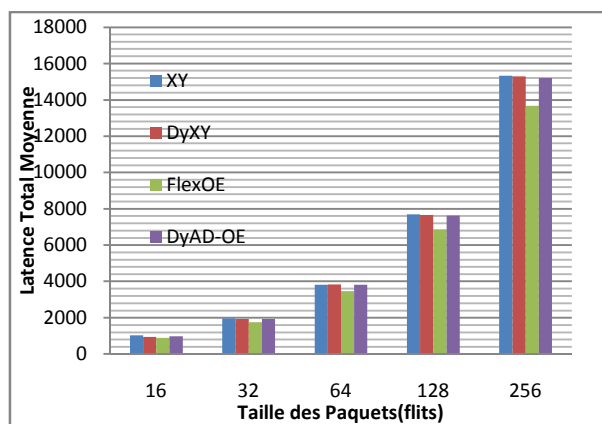


Figure 6 - Latence totale moyenne (cycles d'horloge) pour trafic transpose uniforme (30 paquets, 50% charge)

Pour le deuxième scénario de test (de type transpose également), un NoC 3x3 est utilisé. Chaque router transmet un ensemble de 30 paquets dont les tailles varient de 16 à 256 flits. La charge injectée est de 50%. La Fig. 6 montre la relation entre la latence totale moyenne et la taille des paquets (taille donnée en flits). FlexOE présente la plus faible latence totale moyenne pour ce type de scénario et ceci quelque soit la taille des

paquets. Les latences moyennes sont d'environ 10% plus faibles que celles obtenues par les autres algorithmes de routage et ceci pour des tailles de paquets variant entre 32 et 256 flits.

## 5 Conclusion

Cet article présente un nouvel algorithme de routage basé sur les informations de congestion pour les réseaux sur puce. Cet algorithme est basé sur un schéma mettant en œuvre un ensemble de règles priorisées et un ordonnancement des sorties de type « round robin ». Quatre règles sont définies avec des priorités de type descendant : modèle Odd-Even, chemin de sortie minimal, faible congestion et disponibilité du chemin de sortie. L'algorithme de routage développé et intégré dans le NoC Hemes montre que cet algorithme permet d'obtenir des latences de paquets plus rapides et une meilleure latence moyenne que les algorithmes de routage traditionnels sans nécessiter plus de ressources.

## 6 Remerciements

Ce travail est supporté par la Région Rhône-Alpes et par le Capes Brésil (procès 245340/2012-2).

## 7 Références

- [1] F. Moraes, N. Calazans, A. Mello, L. Möller, and L. Ost, "HERMES: an infrastructure for low area overhead packet-switching networks on chip," *Integr Vlsi J*, vol. 38, no. 1, pp. 69–93, Oct. 2004.
- [2] M. Valinataj, S. Mohammadi, J. Plosila, and P. Liljeberg, "A fault-tolerant and congestion-aware routing algorithm for Networks-on-Chip," in *2010 IEEE 13th International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, 2010, pp. 139–144.
- [3] C. Wang, W.-H. Hu, and N. Bagherzadeh, "Congestion-aware Network-on-Chip router architecture," in *15th CSI International Symposium on Computer Architecture and Digital Systems*, 2010, pp. 137–144.
- [4] C. Wang and N. Bagherzadeh, "Design and Evaluation of a High Throughput QoS-Aware and Congestion-Aware Router Architecture for Network-on-Chip," in *Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euromicro International Conference on*, 2012, pp. 457–464.
- [5] P.-T. Huang and W. Hwang, "An adaptive congestion-aware routing algorithm for mesh network-on-chip platform," in *SOC Conference, 2009. SOCC 2009. IEEE International*, 2009, pp. 375–378.
- [6] A. Mehranzadeh, A. Khademzadeh, and A. Mehran, "FADyAD-Fault and congestion aware routing algorithm based on DyAD algorithm," in *2010 5th International Symposium on Telecommunications (IST)*, 2010, pp. 274–279.
- [7] G.-M. Chiu, "The odd-even turn model for adaptive routing," *Ieee Trans. Parallel Distrib. Syst.*, vol. 11, no. 7, pp. 729–738, Jul. 2000.
- [8] J. Hu and R. Marculescu, "DyAD - Smart Routing for Networks-on-Chip," in *In ACM/IEEE Design Automation Conference*, 2004, pp. 260–263.
- [9] M. Li, Q.-A. Zeng, and W.-B. Jone, "DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip," in *2006 43rd ACM/IEEE Design Automation Conference*, 2006, pp. 849–852.
- [10] P. Lotfi-Kamran, A. M. Rahmani, M. Daneshalab, A. Afzali-Kusha, and Z. Navabi, "EDXY - A low cost congestion-aware routing algorithm for network-on-chips," *J Syst Arch.*, vol. 56, no. 7, pp. 256–264, Jul. 2010.
- [11] H. S. Kia and C. Ababei, "A new fault-tolerant and congestion-aware adaptive routing algorithm for regular Networks-on-Chip," in *2011 IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 2465–2472.
- [12] J. Tan, V. Fresse, and F. Rousseau, "Emulation platform for an adaptive NoC-based MPSoC architecture dedicated to spectral imaging for art authentication," in *International Conference on Image Processing Theory, Tools and Applications*, 2012.