

République algérienne démocratique et populaire  
Ministère de l'enseignement supérieur et la recherche  
scientifique

Université SAAD DAHLAB BLIDA-1

Faculté des sciences

Département d'informatique



**Mémoire de fin d'étude pour l'obtention du diplôme MASTER**  
**Option : systèmes informatiques et réseaux**

Thème :

---

**Génération de topologies personnalisées pour les réseaux sur  
puce 3D**

---

Réalisé par :

**YOUCEF ESSAIDI NASSIM**

**Soutenu devant un jury constitué de :**

- |                    |            |
|--------------------|------------|
| ● Mme Boumahdi     | Présidente |
| ● Mr Ould aissa    | Examineur  |
| ● Mme Toubaline N. | Promotrice |

Année universitaire : 2020-2021

## Résumé

Le développement des technologies des circuits intégrés permet l'intégration de plusieurs composants sur même puce. Ces composants communiquent entre eux à travers des interconnexions de communications (interconnexions point à point ou bus). Cependant, ces derniers trouvent leurs limites en communication. Un nouveau paradigme d'interconnexion est alors apparu (le réseau sur puce –Network on Chip) afin de surmonter les problèmes des méthodes d'interconnexion classiques. Plusieurs problèmes peuvent surgir lors de leur conception du fait du nombre de paramètres qui sont à prendre en compte.

On s'intéresse dans notre travail d'une part à la phase de la personnalisation de topologie. Cette dernière permet à choisir une topologie optimisée. Le but est de proposer une topologie la plus optimale possible, en minimisant le débit du réseau de communications et l'énergie de consommations.

Dans notre travail nous avons étudié le problème d'optimisation de la topologie en considérant a la technologie 3D.

**Mots clés :** système sur puce, réseau sur puce, application spécifique, optimisation, topologie personnalisée et la technologie d'intégration 3D (TSV).

## **Abstract**

The development of integrated circuit technologies allows the integration of several components on the same chip. These components communicate with each other through communications interconnections (point-to-point or bus interconnections). However, the latter find their limits in communication. A new interconnection paradigm then emerged (Network on Chip) in order to overcome the problems of traditional interconnection methods. Several problems can arise during their design due to the number of parameters that must be taken into account.

In our work, we are interested in the phase of topology customization. The latter allows you to choose an optimized topology . The goal is to offer the most optimal topology possible, minimizing the throughput of the communications network and energy consumption.

In our work we have studied the topology optimization problem considering 3D technology.

**Keywords:** system on chip, network on chip, application specific, optimization, custom topology and 3D integration technology (TSV).

## الملخص

يسمح تطوير تقنيات الدوائر الالكترونية بدمج العديد من المكونات على نفس الشريحة. تتواصل هذه المكونات مع بعضها البعض من خلال تواصل الاتصالات (من نقطة إلى نقطة أو توصيلات الباص). ومع ذلك ، فإن هذا الأخير يجد حدوده في التواصل. ثم ظهر نموذج جديد للتواصل ( Network on Chip) من أجل التغلب على مشاكل طرق الترابط التقليدية. يمكن أن تنشأ العديد من المشاكل أثناء تصميمها بسبب عدد المعلمات التي يجب أخذها في الاعتبار.

في عملنا ، نحن مهتمون بمرحلة إعطاء طابع شخصي لطوبولوجيا مناسبة والتي تسمح لنا باختيار هيكل محسن . الهدف هو تقديم أفضل طوبولوجيا ممكنة ، وتقليل إنتاجية شبكة الاتصالات واستهلاك الطاقة.

درسنا في عملنا مشكلة تحسين الهيكل مع الأخذ في الاعتبار التكنولوجيا ثلاثية الأبعاد.  
**الكلمات الرئيسية:** النظام على الرقاقة ، والشبكة على الرقاقة ، والتطبيق المحدد ، والتحسين ، والطوبولوجيا المخصصة ، وتكنولوجيا التكامل ثلاثي الأبعاد (TSV)

## **Remerciements**

**En premier lieu et avant tout nous remercions DIEU << ALLAH>> le tout puissant de nous avoir donné le courage, la patience et la force de réaliser ce projet de fin d'étude.**

**Je tiens à remercier très chaleureusement ma promotrice Docteur. TOUBALINE Nesrine (MCB à l'université SAAD DAHLAB Blida), qui m'a guidé tout au long de ce travail et m'a aidé avec ses conseils, et je la remercie surtout pour la confiance qu'elle a mise en moi pour que je puisse donner le meilleur de moi-même.**

**Je présente mon respect et mon sincère remerciements aux membres de jury qui m'ont fait l'honneur d'avoir accepté de faire partie du jury et de m'avoir consacré de leur temps précieux.**

**Je remercie tous mes amis, pour les conseils, les services et plus particulièrement pour l'amitié qu'ils m'ont témoignée.**

**En terminant, je souhaite démontrer ma grande gratitude à toutes les personnes ayant participé de près ou de loin et plus particulièrement à ma famille a la réalisation de ce projet.**

## **Dédicaces**

**Je dédie le fruit de ce modeste travail comme un geste de gratitude**

**A mes chers parents**

**Pour leur sacrifice, leur amour, leur soutien et encouragements tout au long de mes études**

**A mes très chères sœurs et frères**

**A mes chers amis de l'université avec qui j'ai passé des moments inoubliables avec.**

**A Mes professeurs qui vont voir dans ce travail la fierté d'un savoir bien acquis**

**MERCI**

# Table des matières

RESUME	2
REMERCIEMENT	5
DEDICACES	6
Table des matières	7
Liste des figures	10
Liste des tableaux	12
Introduction générale	13
Chapitre 1 : Généralités sur les réseaux sur puce	15
1 Introduction.....	15
2 Généralités sur les systèmes sur puces.....	16
2.1 Système Mono puce (SoC).....	16
2.2 Système multiprocesseurs sur puce(MPSoC).....	17
2.3 Les types d'interconnexion dans les systèmes sur puce .....	17
3 Architecture du réseau sur puce.....	19
3.1 Architecture en couches.....	20
3.2 Comparaison des Nocs à des réseaux informatiques.....	20
4 Principaux composants du réseau sur puce.....	21
4.1 Les ressources.....	21
4.2 L'Interface Resource-Réseau.....	21
4.3 Le routeur/commutateur.....	22
4.4 Les liens.....	22
5 La topologie.....	22
5.1 Directe et Indirecte.....	23
5.2 Régulière et Irrégulière.....	24
5.3 Topologies Standard 2D.....	24
5.4 Paramètres physiques de la topologie.....	26
5.5 Comparaison entre topologies.....	27
5.6 Topologies en 3D.....	28
6 La communication dans les réseaux sur puces.....	29
6.1 Le routage dans les réseaux sur puces.....	29
6.2 Techniques de commutation.....	32

6.3	Modèles de Trafic (Traffic Patterns).....	34
6.4	Le contrôle de flux .....	35
7	Paramètres de performance du réseau sur puce.....	37
8	Phases de conception d'un réseau sur puce.....	37
9	Outils d'aide à la Conception des Réseaux sur puce.....	38
9.1	NOXIM.....	38
9.2	NS-2.....	38
9.3	SunFloor.....	38
9.4	SunFloor 3D.....	39
9.5	ORION.....	39
9.6	DARSIM.....	39
9.7	Comparaison entre les outils d'aide à la conception.....	39
10	Importance des réseaux sur puce pour les applications futures.....	40
11	Conclusion.....	41
<b>Chapitre 2 Etat de l'art : Optimisation des Topologies</b>		<b>42</b>
1	Introduction.....	42
2	Classification des approches de personnalisation et d'optimisation.....	42
3	Problème d'optimisation.....	43
4	Caractéristiques des topologies régulière et personnalisée.....	44
5	Aperçu de la technologie IC 3D.....	44
5.1	Technologie IC 3D.....	45
5.2	Avantages du réseau à 3 dimensions.....	46
5.3	Technologie de TSVs.....	47
6	Les approches d'optimisation des topologies.....	48
6.1	Topologie régulière.....	49
6.2	Topologie personnalisée.....	49
6.3	Placement des liens verticaux.....	50
6.4	Synthèse sur les approches d'optimisation de topologies.....	51
7	Méthodes et outils d'évaluation de performance du Noc.....	52
7.1	Evaluation des Nocs par des modèles de système.....	53
7.2	Evaluation par simulation.....	54
8	Conclusion.....	55



Chapitre 3 : La conception de la technique proposée	56
1 Introduction.....	56
2 Description du problème d’optimisations .....	56
3 Approche pour la personnalisation des architectures NoCs .....	57
3.1 Hypothèses de base.....	58
3.2 Implémentation de l’algorithme d’insertion de liens.....	60
3.3 Schéma explicatif de l’algorithme d’insertion.....	62
4 Conclusion.....	65
Chapitre 4 : Tests et résultats	66
1 Introduction.....	66
2 Présentation de benchmark VOPD .....	66
3 Le flux de trafic .....	67
4 Les liens candidats.....	68
5 Paramètres de simulation.....	69
6 L’évaluation analytique de l’algorithme d’insertion de lien.....	73
7 Résultats de simulation.....	77
8 Conclusion.....	81
Conclusion Générale	82
Bibliographie	84

## Table des figures

Figure 1.1 : Evolution de l'intégration des transistors[1].....	15
Figure 1.2 : Schéma typique d'un Soc [8].....	16
Figure 1.3 : Architecture d'un système mono-puce [9].....	17
Figure 1.4 : SoC basé sur la communication avec des connexions point à point [10].....	18
Figure 1.5 : Soc basé sur l'infrastructure de communication en bus partagé [11].....	18
Figure 1.6 : Infrastructure de réseaux sur puce [13].....	19
Figure 1.7 : Les éléments du Noc et les couches réseau [14].....	20
Figure 1.8 : La topologie Noc 4x4 Mesh avec les principaux composants identifiés [16].	21
Figure 1.9 : Les parties de L'Interface Resource-Réseau [16].....	22
Figure 1.10 : Exemple de (a) Topologie direct et (b) Topologie indirect [22].....	23
Figure 1.11 : Exemple de topologies en 2D [26].....	25
Figure 1.12 : Topologie 3D [30].....	29
Figure 1.13 : Classification des algorithmes de routage [32].....	30
Figure 1.14 : Situation de deadlock [32].....	32
Figure 1.15 : La commutation de message [34].....	34
Figure 1.16 : Mécanisme de contrôle de flux Ack/Nack [26].....	36
Figure 1.17 : Mécanisme de contrôle de flux Credit-Based [26].....	36
Figure 2.1 : Classification des approches de personnalisation et d'optimisation [26]....	42
Figure 2.2 : Différence entre un optimum global et des optimum local [50].....	43
Figure 2.3 : Différents types de méthodes d'empilage [52].....	46

Figure 2.4 : Méthodes d'empilement TSV [53].....	48
Figure 2.5 : Méthodes et outils d'évaluation d'un réseau sur puce [26].....	53
Figure 3.1 : Exemple de notre supposition [26] .....	57
Figure 3.2: Exemple d'architecture après le mapping du ASIC [91] .....	57
Figure 3.3 : Election de la combinaison gagnante [26].....	61
Figure 3.4 : Algorithme optimisé[26].....	63
Figure 3.5 : Exemple de déroulement de l'algorithme de sélection de liens à insérer....	64
Figure 4.1 : Benchmark VOPD [92].....	67
Figure 4.2 : Les flux de données utilisées, dans le NoC de type 3D 3x3x2 Mesh.....	68
Figure 4.3 : Délai de propagation .....	71
Figure 4.4 : Le degré de clusterisation du NoC de type 2D 3x3 Mesh, lorsque S varie de 0 à 5.....	75
Figure 4.5 : La distance moyenne du NoC de type 2D 3x3 Mesh, lorsque S varie de 0 à 5.....	76
Figure 4.6 : La distance euclidienne du NoC de type 2D 3x3 Mesh, lorsque S varie de 0 à 5.....	77
Figure 4.7 : Comparaison de débit de différentes insertions de lien horizontale.....	79
Figure 4.8 : Comparaison de débit de différentes insertions des TSVs .....	80

## Liste des tableaux

Tableau 1.1 : Comparaison entre les différents supports utilisées dans SoC[13].....	19
Tableau 1.2 : Comparaison des Nocs à des réseaux informatiques [14].....	20
Tableau 1.3 : Paramètres physiques des topologies Noc [27].....	27
Tableau 1.4 : Comparaison entre les outils de conception [38].....	39
Tableau2.1 : Comparaison entre topologie régulière et irrégulière [51].....	44
Tableau 2.2 : résumé des approches d'optimisation (personnalisation).....	51
Tableau 2.3 : Les avantages et les inconvénients des approches.....	52
Tableau 4.1 : Mapping de l'application VOPD.....	67
Tableau 4.2 : Les liens candidats.....	69
Tableau 4.3 : Paramètre de simulation de l'algorithme.....	72
Tableau 4.4 : Evaluation analytique de l'algorithme d'insertion des liens horizontaux.....	73

## Introduction générale

De jour, de nouvelles applications plus complexes et plus personnalisées sont introduites.

Les systèmes complets dits systèmes sur puce (SoC) permettent de répondre aux besoins de ces applications.

Avec le développement technologique [1], il est devenu possible d'intégrer plusieurs composants paradigmes d'interconnexions tels que le point à point et le bus.

Dans ces architectures, le bus est une ressource de communication partagée entre les divers maîtres désirant accéder aux esclaves. Le contrôleur de bus accorde le bus à un seul maître à la fois. Le bus donc pourrait devenir rapidement surchargé [2]. Afin de surmonter le problème de la surcharge d'un bus unique.

C'est dans ce contexte la que le nouveau paradigme réseau sur puce (NoC) est apparu. Cette nouvelle interconnexion inspirée des réseaux informatiques classiques permet une meilleure communication dans le système, en offrant une bonne flexibilité, moins de consommation d'énergie et de surface.

Le Réseau sur puce ou Noc (Network On Chip) [3, 4] pour les systèmes sur puce a été introduit aussi une importante bande passante de communication et une évolutivité très élevée du réseau à travers une structure généralement régulière.

### Domaine abordé :

Contexte général : Optimisation de performances des réseaux sur puce (NoC: Network-on Chip) Contexte précis :

- Développement d'un outil d'évaluation de NoC qui combine l'évaluation analytique et la simulation.
- Mise en œuvre d'une approche adaptative pour la conception des réseaux NoCs :
  - une approche de personnalisation de la topologie du NoC sans tenir compte de l'application.

## Problématique traitée :

La problématique de la mise en œuvre des réseaux sur puce [5, 6] est à la largeur de son espace de conception. Du fait, du nombre de paramètres à prendre en compte. Comme par exemple : le choix de la topologie, l'utilisation des nouvelles technologie(3D).

Il est donc nécessaire de disposer d'un outils d'aide à la conception afin d'assister et de guider le concepteur dans ces choix afin de concevoir un circuit à faible cout tout en optimisant le cout en communication et la surface du circuit.

## Objectif :

Le but de ce travail est de proposer un outil qui maximise les performances des réseaux sur puce dans un cadre multi critères.

## Le plan du document :

Ce mémoire est constitué de quatre chapitres :

- Dans le premier chapitre, nous allons introduire le principe des réseaux sur puce (Network on chip - NoC). Nous verrons quelques types d'interconnexion standards. Puis, nous présenterons l'ensemble des composants et caractéristiques du nouveau paradigme NoC.
- Dans le second chapitre, nous verrons quelques méthodes d'optimisations des topologies et aussi les méthodes d'évaluations de NoC.
- Dans le troisième chapitre, nous présenterons notre technique pour la personnalisation des Topologie.
- Dans chapitre quatre, nous présenterons les résultats obtenus par notre approche d'optimisation et la comparé avec d'autre topologies.

# Chapitre 1

## Généralités sur les réseaux sur puce

### 1.Introduction

L'évolution technologique a permis d'intégrer plus d'un billion de transistors dans le même substrat de silicium, en 2025 la dimension physique de transistor franchira le seuil de 10 nm [1], la figure 1.1 montre l'évolution de l'intégration des transistors dans un même substrat depuis 1970 , cette évolution est en conformité avec la loi de Moore.

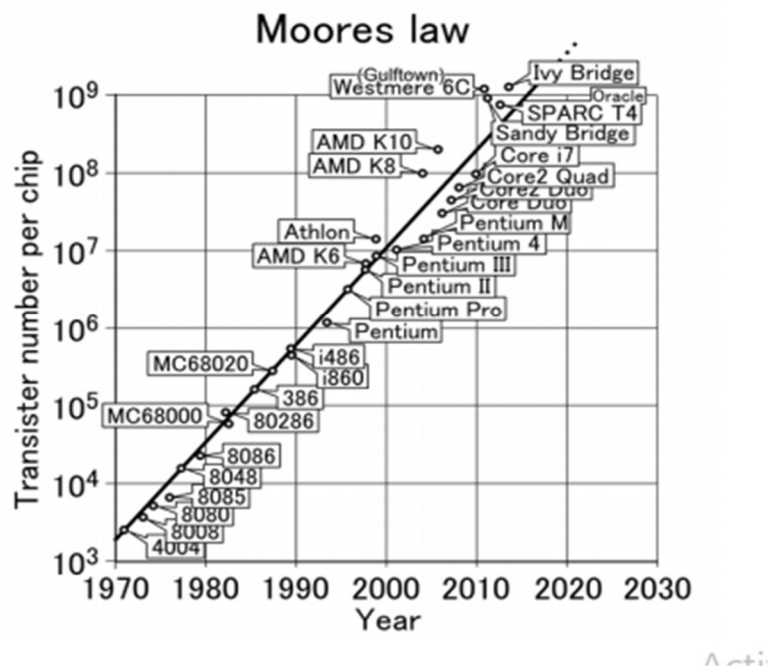


Figure 1.1 : Evolution de l'intégration des transistors [1]

La possibilité d'intégrer un nombre important de transistors dans le même substrat a encouragé les concepteurs des circuits intégrés d'intégrer beaucoup plus de fonctionnalités et d'architectures dans un même circuit afin de réduire le coût de production, en effet un circuit avec une technologie récente peut remplacer des dizaines ou même des centaines des circuits avec des technologies primitives.

Afin de surmonter ces défis, un nouveau paradigme dit réseau sur puce (Network on chip-NoC) est apparu. Inspirer des réseaux informatiques standards, le réseau sur puce a l'avantage d'une bonne flexibilité de communication, d'une consommation d'énergie et de surface réduite.

De nombreux problèmes sont rencontrés lors de la conception d'un réseau sur puce dont : le placement des composants d'une application (IP) sur une architecture, ou le choix d'une topologie tout en considérant certains critères pendant la conception.

## 2. Généralités sur les systèmes sur puce :

Les systèmes à base de puces électroniques font de plus en plus de notre partie de quotidien. Du simple transistor au circuit intégré, ces structures intègrent aujourd'hui des milliers de composants complexes et de fonctionnalités hétérogènes.

L'augmentation constante du nombre de fonctions intégrées dans un seul système a fortement augmenté les contraintes et les exigences d'architecture en terme de composants et les réseaux de communication.

### 2.1. Système Mono puce (Soc) :

L'expression Soc (Système on chip) est utilisée pour désigner un système complet embarqué sur une puce. Il peut comprendre de la mémoire, un ou plusieurs microprocesseurs, des périphériques d'interface, ou tout autre composant nécessaire à la réalisation de la ou les fonctions attendues. On peut intégrer de la logique, des dispositifs (capteurs) mécaniques, optoélectroniques, chimiques ou biologiques ou des circuits radio [7].

De nos jours, nous assistons à un immense progrès de ces systèmes. Ils sont de plus en plus utilisés dans les ordinateurs, téléphones portables ou encore consoles de jeux. Cette évolution a donné naissance à des architectures de plus en plus complexes exécutant plusieurs fonctionnalités à la fois.

Dans la figure 1.2 est présenté un exemple de système mono-puce typique. Il se compose d'un cœur de processeur (CPU), d'un processeur de signal numérique (DSP), de la mémoire embarquée, et de quelques périphériques tels qu'un DMA (direct memory access) et un contrôleur d'E/S.



Figure 1.2 : schéma typique d'un Soc [8]



## 2.2. Système multiprocesseurs sur puce (MP Soc) :

Les systèmes sur puce multiprocesseurs ou MP Soc (Multiprocesseur System-on-chip), sont apparus grâce à l'évolution de la technologie de fabrication des circuits intégrés qui a permis l'intégration de plusieurs processeurs, des nombreux composants numériques spécialisés et hétérogènes (mémoires, périphériques, unités de calcul spécifiques), du logiciel et souvent des circuits mixtes pour fournir un système intégré complet.

Les systèmes mono-puce multiprocesseurs sont intégrés dans divers domaines comme par exemple dans le domaine de l'automobile, des télécommunications et du multimédia et leur diversité ne cesse d'accroître la figure 1.3 montre l'architecture d'un tel système mono-puce multiprocesseur.

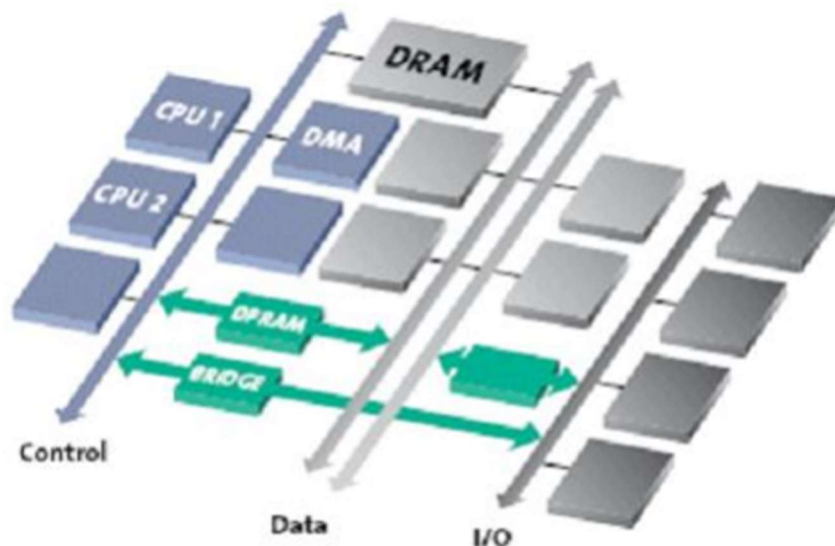


Figure 1.3 : Architecture d'un système mono-puce [9]

## 2.3. Les types d'interconnexion dans les systèmes sur puce :

### 2.3.1. Point à point :

La toute première façon de concevoir l'infrastructure de communication pour les systèmes sur puce était l'utilisation directe de l'interconnexion point à point entre les composants matériels du système.

Cette architecture de communication nécessite beaucoup de broches (PINs) pour chaque IP, une longue durée pendant l'acheminement de trafic et un câblage complexe.

D'autre part, un Soc conçu avec un petit nombre d'IPs basé sur ce type d'interconnexions est susceptible de donner une meilleure performance [10].

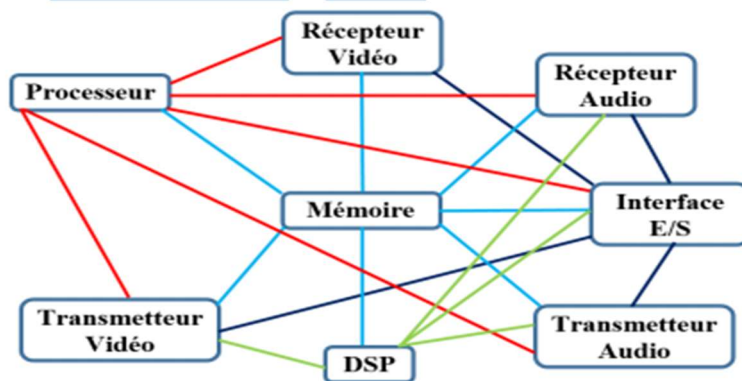


Figure 1.4 : SoC basé sur la communication avec des connexions point à point [10]

### 2.3.2. Bus partagé :

La communication inter-IPs de la majorité des systèmes sur puce est conçue avec des bus en fonction de l'infrastructure de communication.

Les IPs sont connectés au bus par le biais d'une interface. Un arbitre du bus gère la communication et le conflit entre les IPs (figure 1.5).

Le coût et la surface de câblage nécessaire pour la communication sont également réduits, Toutefois, il ne permet aucune forme de parallélisme.

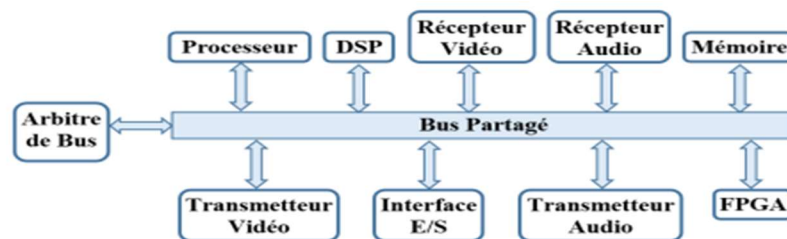


Figure 1.5 : Soc basé sur l'infrastructure de communication en bus partagé [11]

### 2.3.3. Réseaux sur puce (Noc) :

Le Noc est proposé comme une structure de communication évolutive pour les plates-formes à plusieurs cœurs [12].

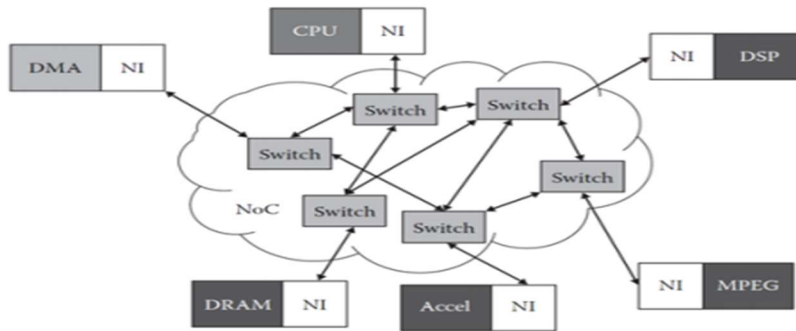


Figure 1.6 : Infrastructure de réseaux sur puce [13]

Une comparaison entre les différents supports qui peuvent être utilisées dans un système sur puce est donnée par le Tableau 1.1.

Tableau 1.1 : Comparaison entre les différents supports utilisées dans SoC [13]

	Point à point	Bus	NoC
parallélisme	++	--	++
Consommation	+	-	++
surface	+	++	-
Scalabilité	--	-	++
Réutilisation	-	++	++
Qualité de service	++	+	++

++ : très bon

+: bon

-- : très mauvais

- : mauvais

D'après le tableau ci-dessus, nous remarquons clairement que le support NoC est plus efficace par rapport aux supports du point à point et Bus dans tous les paramètres sauf il est défaillant en surface.

Cette comparaison nous permet de conclure l'importance de l'évolution des supports.

### 3. Architecture du réseau sur puce :

Les protocoles utilisés dans le Noc sont généralement des versions simplifiées des protocoles de communication utilisés dans les réseaux informatiques. De même, la plupart des définitions qui sont utilisées dans les réseaux de communication informatiques sont également applicables dans le Noc.

### 3.1. Architecture en couches :

La représentation des couches du modèle OSI classique n'est pas adaptée à la description d'un réseau Noc, car contrairement aux réseaux informatiques où les protocoles sont complexes vu qu'ils sont utilisés pour une topologie inconnue.

Principalement quatre couches sont considérées telles que : la couche physique, la couche liaison de données, la couche réseau et la couche transport.

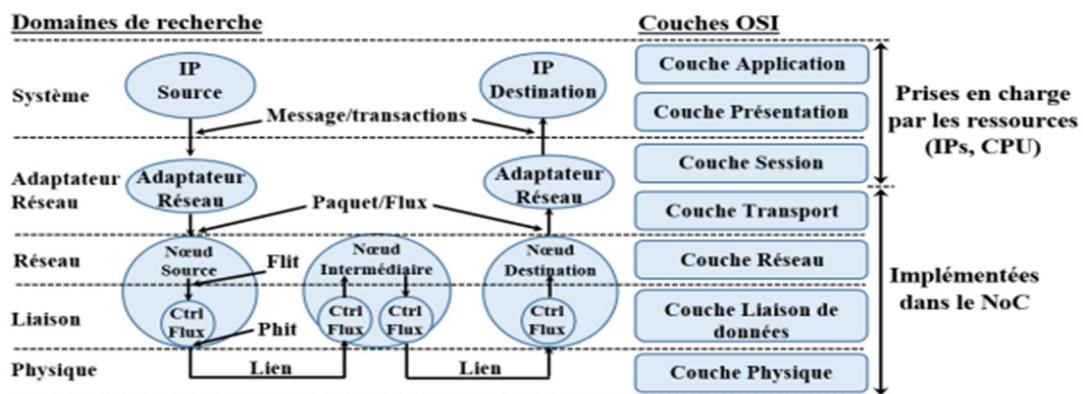


Figure 1.7 : Les éléments du Noc et les couches réseau [14]

### 3.2 Comparaison des Nocs à des réseaux informatiques :

Le tableau ci-dessous compare les architectures Nocs à des réseaux informatiques.

Les réseaux sur puce sont d'une technologie spécifique et nécessite des ressources plus évoluer et plus puissante.

Tableau 1.2 : Comparaison des Nocs à des réseaux informatiques [14]

Réseaux sur puce	Réseaux informatiques
Un degré élevé d'hétérogénéité	Hétérogène(multiplateformes)
Adapter à des applications	Usage générale d'applications
Importantes contraintes D'énergie (des techniques sophistiquées de routage et de commutation sont nécessaire	L'énergie n'est pas le souci majeur
Une topologie prédéfinie (fixée par le concepteur)	La topologie n'est pas prédéfinie
Routeurs sophistiquées	Routeurs et switches

## 4. Principaux composants du réseau sur puce :

Il existe trois principaux types de composants à savoir les routeurs/commutateurs, les ressources également appelés les cœurs (appelés aussi en anglais Intellectuel Propriet (IP)) et les interfaces ressource-réseau abrégées IRR (ou en anglais RNI : Resource Network Interfaces).

Dans la suite du mémoire on utilisera l'appellation "nœud", où un nœud est composé d'une ressource connectée à un routeur via une interface ressource-réseau IRR, comme le montre le zoom illustré sur la figure 1.8[15, 16, 17].

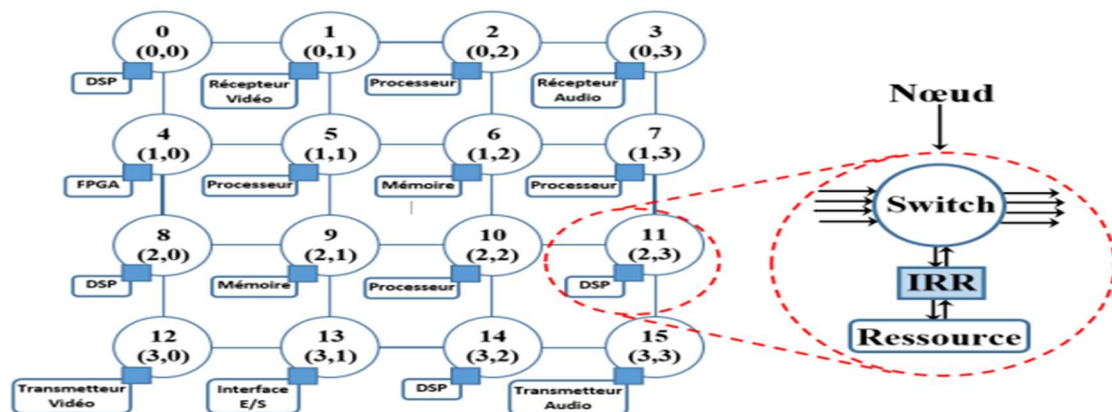


Figure 1.8 : La topologie NoC 4x4 Mesh avec les principaux composants identifiés [16]

### 4.1. Les ressources :

Une ressource peut être un processeur d'usage général, un DSP (de l'anglais "Digital Signal Processor"), une mémoire, un composant matériel d'application spécifique, contrôleur d'E/S, contrôleur graphique, le module de signal mixte, la fréquence radio unitaire (RF: radio fréquence), etc...

Le concepteur peut construire des ressources propres ou réutiliser les ressources commerciales disponibles chez différents fournisseurs de composants [18].

### 4.2. L'Interface Resource-Réseau (IRR) (ou en anglais RNI : Resource Network Interface) :

L'IRR relie une ressource à un routeur réseau. Par conséquent, elle permet à la ressource d'envoyer des données au routeur. Le rôle de l'IRR dans les NoCs est le même rôle que celui de la carte réseau pour l'Internet [19].

L'IRR se compose de deux parties, la partie dépendante des ressources et la partie indépendante des ressources comme le montre la figure 1.9.

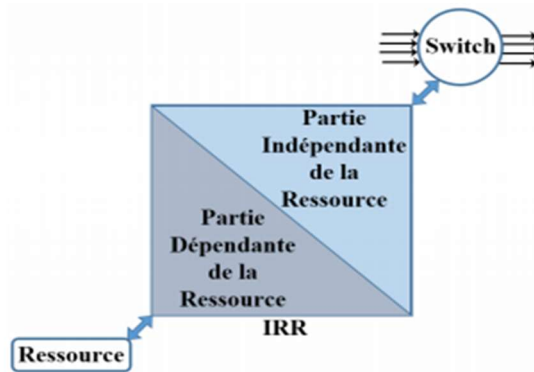


Figure 1.9 : Les parties de L'Interface Ressource-Réseau [16]

### 4.3. Le routeur/commutateur (Router/Switch) :

Les routeurs sont constitués de ports d'entrée/sortie contenant (ou pas) des files d'attente pour stocker les données qui ne pourront pas être transmises tout de suite, d'une matrice de commutation pour relier physiquement les ports d'entrée aux ports de sorties, des blocs de contrôle et d'arbitrage des paquets et enfin, d'une table de routage ou bien d'un bloc de calcul de la fonction de routage si ce-dernier est distribué [20].

La logique de contrôle peut servir à des tâches comme la détection/correction d'erreurs ou bien pour le contrôle de flux.

### 4.4. Les liens :

Les liens sont des connexions logiques entre deux (ou plusieurs) éléments communicants [21]. Ils permettent les connexions de type point à point entre les ressources IPs et les interfaces ressource-réseau IRRs, entre l'IRR et les routeurs et entre les routeurs.

## 5. La topologie :

Une topologie d'un NoC est généralement modélisée sous forme de graphique  $G=(v, E)$ , où  $V$  l'ensemble des sommets, représentent les nœuds, qui contiennent les éléments de traitement et de routage, et  $E$  l'ensemble des arêtes, représentent les liens de communication interconnectant les nœuds.

La topologie décrit comment les nœuds sont connectés les uns aux autres via des liens de communication. Une topologie peut être « logique » et/ou « physique ». La topologie logique illustre la façon dont les données circulent entre les nœuds au sein du réseau tandis que la topologie physique indique le placement des divers nœuds sur la zone de la puce.

Les topologies existantes pour les NoC peuvent être classées en directes ou indirectes, et peuvent également être classées comme régulières et irrégulières. Le choix d'une topologie donnée pour un SoC dépend souvent des exigences de communication de l'application [22].

### 5.1. Directe et Indirecte :

Dans la littérature, les topologies sont usuellement réparties en trois classes : directes, indirectes et hybrides. Cette classification est issue des réseaux d'ordinateurs. Dans les réseaux directs, chaque routeur est connecté à au moins un élément communicant alors que dans les réseaux indirects, les routeurs peuvent être connectés seulement à d'autres routeurs [23].

Enfin, les topologies hybrides sont des combinaisons des deux types de réseaux précédents. Alors que les réseaux directs offrent de meilleures performances en surface utilisée, les réseaux indirects sont une alternative pour la conception d'interconnexions extensibles. En effet, dans ces réseaux, le nombre de routeurs est plus élevé et ils ne sont, pour la plupart, pas reliés aux éléments de traitement. C'est pourquoi, ils améliorent la connectivité et gèrent mieux l'augmentation du nombre d'éléments à interconnecter.

Des exemples de topologies indirectes bien connues incluent les papillons, deBruin et le réseau Clos. Figure 1.10(a) et 1.10(b) montre un exemple de topologie direct and indirect respectivement :

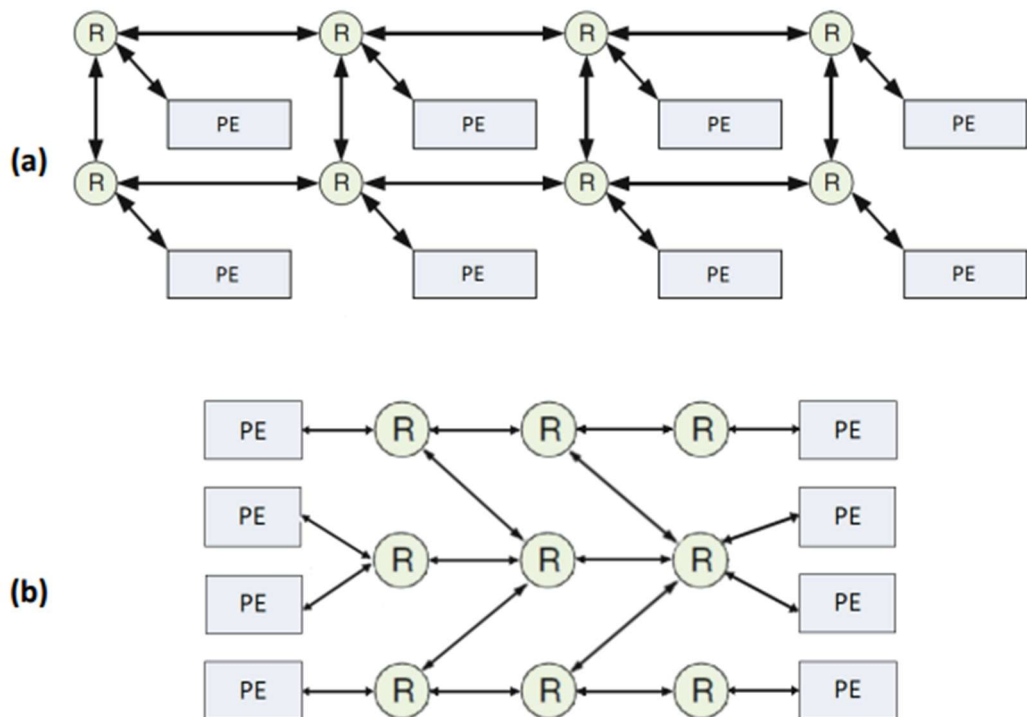


Figure 1.10 : Exemple de (a) Topologie direct et (b) Topologie indirect [22]

## 5.2. Régulière et irrégulière :

L'autre critère permettant de classer les topologies est la régularité. En effet, la topologie est dite régulière lorsque tous les nœuds du graphe du réseau ont exactement le même nombre d'arcs les reliant aux nœuds adjacents. Dans le cas contraire, la topologie est irrégulière [24]. Parce qu'il est difficile de concevoir un réseau avec des performances précises à partir d'une architecture non dédiée, les topologies irrégulières sont plus appropriées pour des réseaux spécifiques à une application. Un autre moyen d'obtenir des NoCs irréguliers optimisés est de supprimer les routeurs et les liens non nécessaires d'un réseau régulier.

## 5.3. Topologies Standard 2D :

Chaque topologie possède des avantages et des inconvénients. Afin de comparer les topologies entre elles, différents paramètres physiques sont utilisés [25]. Parmi les paramètres les plus couramment considérés dans les publications on note le degré du routeur (nombre de ports), le nombre total de liens, le diamètre du réseau, la distance moyenne du réseau, le degré de clustérisations, la régularité, la symétrie, l'extensibilité, la fatalité, la diversité de chemins de routage, et la largeur de bisection.

Comme pour les réseaux informatiques, de nombreuses topologies sont envisageables pour construire un NoC. La figure 1.11 illustre les topologies les plus couramment utilisées dans la conception des NoC.



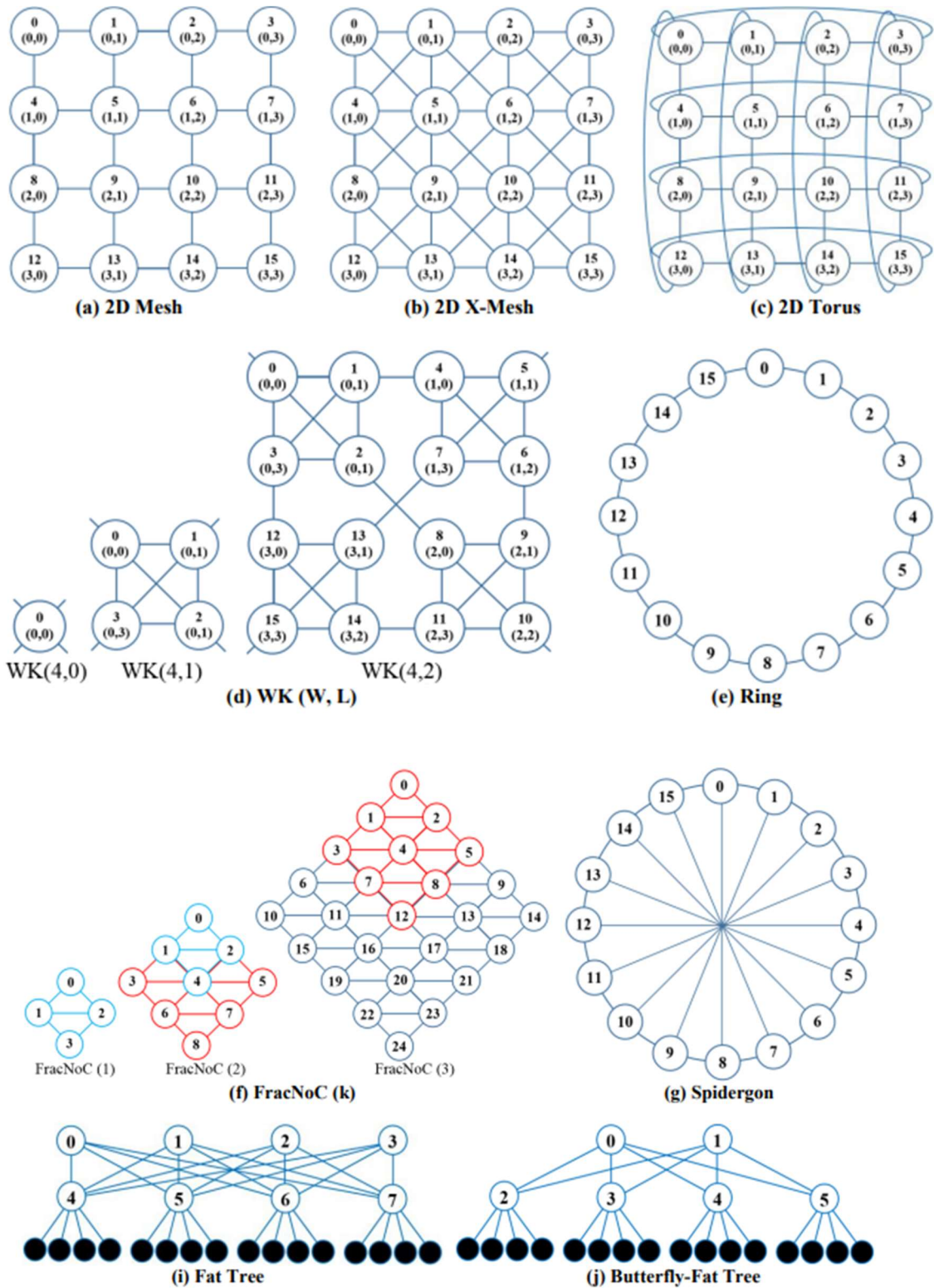


Figure 1.11 : Exemple de topologies en 2D [26]

## 5.4. Paramètres physiques de la topologie :

### A. Le diamètre :

C'est le plus grand nombre de sauts dans tous les chemins les plus courts. Ce critère peut être utilisé comme indicateur pour l'évaluation de la latence maximale. Un petit diamètre permet une communication rapide entre les nœuds les plus éloignés. En d'autres termes, la latence maximale est proportionnelle au nombre maximal de sauts, c'est pour cette raison que le diamètre est considéré comme une mesure importante.

### B. La distance moyenne :

C'est le nombre moyen de sauts entre tous les nœuds dans le réseau. Ce critère peut être une indication pour évaluer la latence moyenne.

### C. Le degré :

C'est le nombre de voisins directs d'un nœud dans le NoC. Il indique si l'architecture NoC est régulière. Un degré élevé permet à de nombreux voisins proches d'effectuer des communications rapides. Toutefois, afin de réduire les coûts matériels et d'augmenter la capacité d'extension, un degré plus petit et fixe, est exigé.

### D. La bisection :

Représente le nombre minimal de liaisons à couper pour séparer le réseau en deux portions égales. Ce critère permet d'évaluer le coût de transfert des données de la moitié du réseau vers l'autre moitié. La bande passante collective de ces liens est nommée bande passante de bisection, et elle permet de mesurer la performance d'une topologie. Il s'agit d'un indicateur d'évaluation du débit. Par exemple, une grande bisection est nécessaire afin de fournir plus de chemins entre deux sous-réseaux, et améliore donc la performance globale du réseau.

### E. Le nombre de liens :

C'est le nombre de liens bidirectionnels du NoC. Ce critère peut être une indication de la charge du réseau. En outre, l'ajout de liens dans un NoC pourrait réduire le diamètre, d'améliorer la communication, et réduire la latence de communication. Cependant, le NoC avec un nombre de liens plus élevé (ex., un NoC entièrement connecté) est coûteux en termes de surface de silicium.

### F. Le degré de clustérisations :

Permet de spécifier la façon dont les nœuds sont interconnectés les uns aux autres, En d'autres termes, il indique la disponibilité des chemins de routage alternatifs afin de re-router des paquets congestionnés à un nœud donné, ce qui permet de réduire la congestion dans les nœuds intermédiaires. Un degré de cautérisation plus élevé favorise aussi la communication locale.

### G. La régularité :

Un réseau est dit régulier lorsque tous les sommets ont le même nombre de voisins, c'est-à-dire le même degré ou valence. Un réseau régulier dont les sommets sont de degré  $K$  est appelé un réseau  $k$ -régulier ou réseau régulier de degré  $k$ .

### H. La symétrie :

Un réseau est dit symétrique si quel que soit la paire de nœuds  $(i,j)$ , il existe autant de liens  $l$  de la forme  $lij$  que  $lji$ .

### I. La diversité de chemins de routage :

Un réseau a une diversité de chemins de routage si la plupart des couples de routeurs ont de multiples chemins de routage entre eux, La diversité de chemins de routage ajoute de la robustesse au réseau.

## 5.5. Comparaison entre topologies :

Le tableau 1.3 montre une comparaison entre les topologies les plus couramment utilisées par rapport à certains paramètres physiques.

Tableau 1.3 : Paramètres physiques des topologies Noc [27]

OCI/ Propriété	Links	Diamètre	Degré	Bissection	Degré de Clustering	Distance Moyenne
2D Mesh	$2N - 2\sqrt{N}$	$2(\sqrt{N} - 1)$	2, 3, 4	$\sqrt{N}$	0	$2\sqrt{N}/3$
Torus	$2N$	$2(\sqrt{N} - 1)$	4	$2\sqrt{N}$	0	$\sqrt{N}/2$
WK(w, k)	$w * (w^l - 1)/2$	$2^l - 1$	3, 4	$(W/2)^2$	Min = 0,5 Max= 1	Non Défini
Spidergon	$(3/2) N$	$N/4$	3	$(N/2) + 2$	0	$\frac{N^2 + 4N - 8}{8N - 8}$
x-Mesh	$4N - 6\sqrt{N} + 2$	$(\sqrt{N} - 1)$	3, 5, 8	$3\sqrt{N} - 1$	Min = 0,42 Max= 1	Non Défini
FracNoC	$3N - 4\sqrt{N} + 1$	$2(\sqrt{N} - 1)$	2, 3, 4, 5, 6	$2\sqrt{N} - 1$	Min = 0,4 Max= 1	Non Défini

- Plus le diamètre d'un réseau est petit, moins le temps qu'il faut pour envoyer un message d'un nœud au nœud le plus éloigné dans le réseau est grand.
- Toutes ces topologies maillées (2D-Mesh, 2D Torus et 2D Folded Torus) ont un problème avec la latence réseau associée et, par conséquent, avec la consommation d'énergie.

## 5.6. Topologies en 3D :

L'empilage de puces tridimensionnelles (3D) est présenté comme la technologie miracle qui peut maintenir l'élan de Moore et alimenter la prochaine vague de produits électroniques grand public. Outre la flexibilité imposée par ce nouveau paradigme de conception, l'un des principaux défis auxquels les concepteurs sont confrontés en matière d'intégration 3D est de réaliser l'interconnexion entre les composants au sein d'une couche et entre les couches de manière évolutive et efficace.

Ce nouveau concept permet aussi d'appliquer presque toutes les topologies 2D mentionnées précédemment au domaine 3D, il est alors possible de faire la transition des architectures 2D vers les architectures 3D NoC.

Réaliser un NoC en 3D apporte plusieurs avantages. Pour commencer, il est possible de disposer plus de routeurs dans une surface réduite grâce aux étages supplémentaires. Cela augmente la densité du NoC et réduit la distance entre les nœuds, réduisant la latence et également la consommation d'énergie comparé à son homologue en 2D [28].

La condition obligatoire afin de concevoir un NoC en 3D est d'avoir des interconnexions verticales entre les couches. Ces interconnexions verticales sont nommées TSV pour Through-Silicon-Via [29], ces liaisons supplémentaires augmentent la connectivité du réseau et la bande passante est également augmentée.

Ce type de topologie apporte cependant de nouvelles problématiques. L'augmentation de la densité du NoC provoque une forte hausse de la température à cause de composants superposés dissipant beaucoup de puissance. Une température élevée a des conséquences critiques dans un circuit. Cela peut nuire à la fiabilité de la puce en générant des erreurs et réduire aussi sa durée de vie. Il existe différentes méthodes pour réduire la température, tel qu'un placement intelligent considérant la température ou encore de nouvelles structures de refroidissement.

Nous présentons brièvement deux topologies 3D prometteuses de la littérature vis-à-vis de la consommation d'énergie :

— 3D mesh [30], ou grille 3D comme représenté à la figure 1.12. C'est l'adaptation de la grille 2D apportant une topologie 3D simple à réaliser. Cependant, la taille du routeur augmente car deux nouvelles directions peuvent être empruntées par les paquets ce qui implique un crossbar 7×7 au lieu d'un crossbar de taille 5×5 en 2D.

— Honeycomb [30] est un NoC exploitant une structure d'interconnexions alvéolée. Cela permet de réduire le coût matériel du réseau car le nombre d'interconnexions est moins important que dans une grille 2D classique comme le montre la figure 1.12.a. De cette manière le nombre de directions par routeur est réduit, diminuant la taille du crossbar. Ainsi, ce procédé est reproduit avec une dimension supplémentaire, la figure 1.12.b montrant une structure alvéolée en 3D. Cette topologie réduit le coût de 40 % comparé à

une topologie en grille 3D. Ces réductions se traduisent par une réduction significative de la consommation d'énergie.

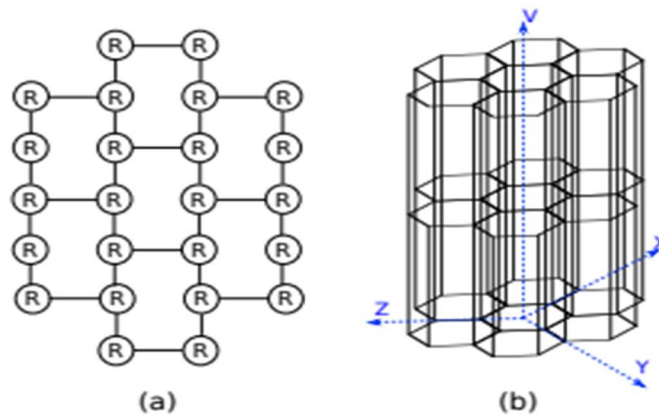


Figure 1.12 : Topologie 3D [30]

–(a) Topologie en grille 3D. (b) Structure alvéolée basée sur une grille 2D.

## 6. La communication dans les réseaux sur puces

### 6.1. Le routage dans les Réseaux sur puces :

L'algorithme de routage détermine le chemin à suivre pour le paquet entre la source et la destination, le choix de l'algorithme de routage est une tâche critique dans la conception des réseaux. En effet on doit prendre en considération tous les paramètres caractéristiques de réseau vus précédemment à savoir la topologie, le mode de commutation ....

Il y a quelques propriétés des algorithmes de routage qui sont essentiellement requis pour les réseaux d'interconnexion à savoir la connectivité, l'adaptabilité, la liberté face aux interblocages deadlock et livelock, ainsi que la tolérance aux pannes [31]. La connectivité est la capacité d'acheminer des paquets provenant de n'importe quel nœud source à n'importe quel nœud destination. L'adaptabilité est la possibilité d'acheminer des paquets à travers des chemins alternatifs dans la présence des contentions. La liberté de Deadlock et de livelock est la capacité de garantir que les paquets ne seront pas bloqués ou de circuler à travers le réseau pour toujours sans la possibilité d'atteindre la destination. La tolérance aux pannes est la capacité d'acheminer les paquets en présence de commutateurs défectueux.

### 6.1.1. Classification des algorithmes de routage :

On trouve une diversité des algorithmes de routage appliqués dans le développement des réseaux sur puce qu'on peut classer selon différents critères, La figure 1.13 présentée ci-dessous montre ces différents critères :

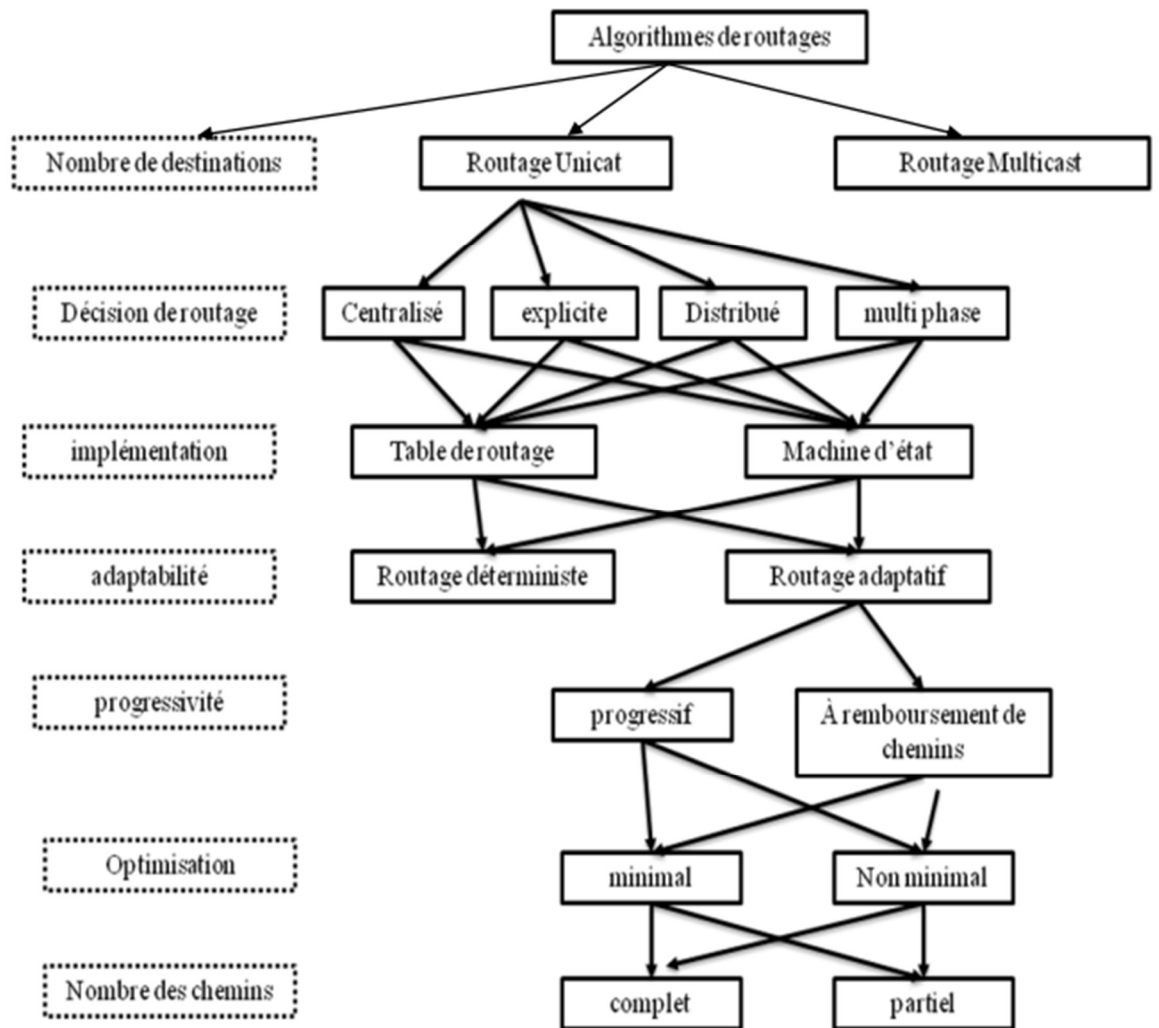


Figure 1.13 : Classification des algorithmes de routage [32]

#### A. Routage source vs distribué (déterministe vs adaptatif) :

Le routage dans le NoC peut être source ou distribué (déterministe ou adaptatif). Dans le routage source, l'itinéraire de l'ensemble du trajet depuis la source vers la destination est pré-calculé et fourni dans l'entête du paquet contrairement au routage distribué, où l'entête du paquet ne contient que l'adresse de destination et que la trajectoire est calculée dynamiquement par la participation des routeurs

intermédiaires du chemin [32] . Dans le routage distribué, de multiples chemins de la source vers la destination sont possibles. Quand un paquet entre dans un routeur, l'adresse de destination est lue à partir de l'entête et en conséquence, la fonction de routage calcule tous les ports de sortie possibles où ce paquet peut être transmis. Ensuite, une fonction de calcul d'itinéraire sélectionne un des ports de sortie admissible pour transmettre le paquet. La sélection du port de sortie dépend des conditions dynamiques du réseau tel que la congestion et les défauts dans les liens.

#### B. Algorithmes de routage à base du modèle de tours (turn-model) :

Un grand nombre d'algorithmes de routage distribués pour le NoC ont été proposées dans la littérature [32]. Dans cette section, nous considérons uniquement les algorithmes de routage "turn-model". Dans ce modèle, certains tours sont restreints pour la communication en fonction des règles utilisées. La caractéristique la plus importante à prendre en considération dans un algorithme de routage est l'absence d'interblocage. Tous les modèles d'algorithmes de routage à base de tours sont sans interblocage. L'absence d'interblocage ainsi que divers algorithmes de routage à base du modèle de tours sont présentés dans ce qui suit.

### 6.1.2. Problèmes de routage :

#### A) Situation de deadlock :

Lorsque les paquets des données circulent à travers le réseau, ils occupent certaines ressources entre les différents chemins.

Lorsque plusieurs paquets sont mutuellement à l'attente des uns des autres afin de libérer certaines ressources, on est dans une situation de deadlock et alors tous les paquets concernés seront bloqués définitivement.



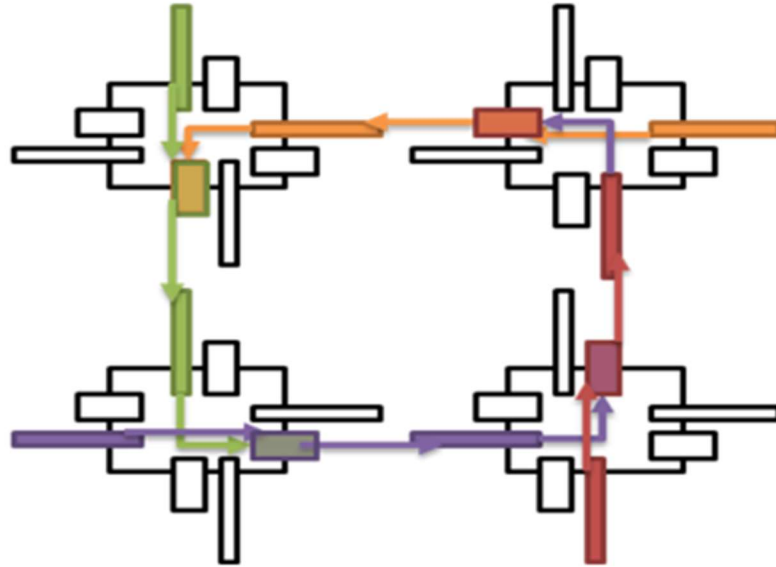


Figure 1.14 : Situation de deadlock [32]

#### B) Situation de livelock :

Ce genre de problème survient lorsque les paquets circulent autour d'une destination sans l'atteindre. Donc, les données ne peuvent pas être acheminées vers la destination. Ce genre de problème se produit dans le cas des algorithmes de routage non minimaux où on cherche à délivrer le paquet sans prendre en considération le plus court chemin. Afin d'améliorer le débit, ce genre de problème doit être évité.

#### C) Situation de starvation :

L'attribution des priorités différentes pour les paquets de données dans le NOC peut causer ce genre de problème. Donc les paquets de priorité élevée peuvent facilement atteindre leur destination, alors que les paquets à faible priorité ne seront jamais délivrés à leur destination. Ce genre de situation se produit parce que les paquets de haute priorité réservent les ressources et les paquets de faible priorité restent en attente de la disponibilité de ces derniers. Ce type de problème peut être évité en utilisant un algorithme de routage approprié qui réserve une partie de la bande passante de réseau au service des paquets à faible priorité.

## 6.2. Techniques de commutation :

Les techniques de commutation sont utilisées pour acheminer des données depuis la source vers la destination dans un réseau de communication [33], Il existe plusieurs techniques de commutation, telles que :



### 6.2.1. La commutation de circuits (circuit switching) :

La commutation circuits est la technique de commutation la plus ancienne et dans ce cas de commutation, les données ne sont pas divisées et le chemin de routage est établi de la source à la destination avant de procéder à la transmission des données ; ce chemin est consacré seulement à une transmission à la fois et ne sera libéré qu'à la fin de la transmission de la totalité du message, l'avantage de (CS) est la garantie de la latence de transmission une fois la connexion entre la source et la destination est assurée, l'inconvénient est la faible utilisation des différentes broches du réseau ; cet inconvénient explique pourquoi ce type de commutation est faiblement utilisé dans les NOC.

Avantages :

- Gain de temps s'il y a plusieurs envois
- Pas de partage de la bande passante

Inconvénients :

- Occupation durable (blocage) des ressources
- Toute erreur de transmission nécessite de refaire intégralement la transmission

### 6.2.2. La commutation de messages (messages switching) :

Dans la commutation de messages, aucun chemin physique n'est établi entre les deux ressources IPs. Chaque message est envoyé individuellement par le réseau d'une source vers une destination en passant par un ou plusieurs routeurs de commutation, comme illustré sur la figure 1.15. Chacun de ces routeurs attend la réception complète du message avant de le remettre sur le réseau. Cela demande de grands tampons sur chaque équipement, d'au moins la taille du message, ainsi qu'un contrôle de flux pour éviter les engorgements.

Avantages (par rapport à la commutation de circuits) :

- Une meilleure utilisation des lignes
- Le transfert est assuré même si le correspondant distant est occupé ou non connecté
- La diffusion d'un même message à plusieurs correspondants
- Le changement de format des messages
- L'adaptation des débits

Inconvénients :

- Risque de débordement des tampons de stockage, tout dépend de la taille des messages

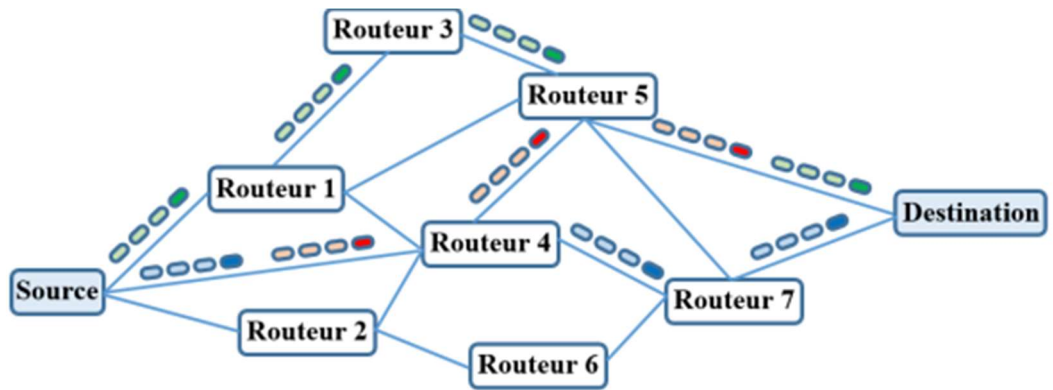


Figure 1.15 : La commutation de message [34]

### 6.2.3. La commutation de paquets (packets switching) :

Il existe plusieurs mécanismes de commutation basés sur le principe de commutation des paquets dont les plus connus sont store and forward, le wormhole et Virtual-cut-through (VCT) , ces trois modes de commutation partagent le principe de diviser le message à transmettre en un ensemble de paquets où chaque paquet lui-même est formé par un ensemble de FLIT, un FLIT lui-même est composé par un certain nombre de phits (habituellement unitaires), la taille de ce dernier, dans la plupart de temps, est adaptée à la taille des liens d'interconnexions du réseau (ceci est défini par le nombre de bits de données pouvant être transmis en parallèle entre deux routeurs voisins), un autre point commun entre ces trois modes de commutation est le fait que les différents paquets formant un message peuvent avoir des différents chemins de routage.

### 6.3. Modèles de Trafic (Traffic Patterns) :

Le NoC se comporte différemment selon les algorithmes de routage utilisant différent types de modèles de trafic [35]. Les chercheurs ont utilisé différents types de générateurs de trafic pour l'évaluation des NoC Les modèles de trafic les plus largement utilisées pour l'analyse de la consommation d'énergie et la latence dans les réseaux d'interconnexion sont :

#### 1.Uniforme

Chaque nœud envoie des messages à d'autres nœuds avec une probabilité égale. Les nœuds de destination sont choisis de façon aléatoire à l'aide d'une fonction uniforme de distribution de probabilité.

#### 2.Hotspot

Chaque nœud envoie des messages à d'autres nœuds avec une probabilité égale, sauf pour un nœud spécifique (appelé Hotspot) qui reçoit des messages avec une plus grande probabilité. Le pourcentage de messages supplémentaires qu'un nœud Hotspot reçoit par rapport aux autres nœuds est indiqué après le nom Hotspot (par exemple, Hotspot 10%).

### 3. Transpose

Chaque nœud envoie des messages uniquement vers une destination avec la moitié haute/basse de sa propre adresse transposée. Soit la source avec l'adresse  $(n_1, n_2, \dots, n_{m-1}, n_m)$  communique avec la destination dont l'adresse est donnée par  $(n_{m/2}, n_{(m/2)+1}, \dots, n_m, n_1, n_2, \dots, n_{(m/2)-1})$ .

### 4. Bit-Complement

Chaque nœud envoie des messages uniquement à celui qui est le complément de sa propre adresse, c'est à dire, la source avec l'adresse  $(n_1, n_2, \dots, n_{m-1}, n_m)$  communique avec la destination dont l'adresse est donnée par  $(n_1, n_2, \dots, n_{m-1}, n_m)$ .

### 5. Bit-Reversal

Bit-Reversal Chaque nœud envoie seulement au nœud dont l'adresse de l'expéditeur est inversé bit par bit, c'est à dire, l'adresse de destination est donnée par  $(n_m, n_{m-1}, \dots, n_2, n_1)$  pour la source  $(n_1, n_2, \dots, n_{m-1}, n_m)$ .

## 6.4. Le contrôle de flux :

Le contrôle de flux est utilisé pour synchroniser la transmission de données à l'intérieur d'un routeur, entre les routeurs voisins et entre les IPs communicantes. Il assure qu'une IP source n'engloutit pas l'IP de destination (ou le port d'entrée n'engloutit pas le port de sortie dans un routeur) avec des données. Ainsi, le but du contrôle de flux est de garantir un transport de données sûr, sans perte de paquets du fait d'un dépassement de capacité de l'un des tampons parcourus, depuis l'IP source jusqu'à l'IP de destination. Il existe plusieurs protocoles de contrôle de flux [36], nous détaillerons dans ce qui suit les trois principales techniques.

#### 1. Contrôle de flux Ack/Nack:

Le contrôle de flux Ack/Nack ne nécessite aucun compteur dans le nœud émetteur en-amont (upstream) pour indiquer la disponibilité du tampon dans le nœud récepteur en-aval (downstream). C'est un protocole très simple où la logique de contrôle se charge d'acquitter chaque flit reçu du côté du nœud récepteur en-aval. Le nœud émetteur en-amont quant à lui doit donc conserver chaque flit émis jusqu'à ce qu'il ait été acquitté. Les flits sont envoyés avec optimisme dès qu'elles sont disponibles, si le nœud récepteur en-aval dispose d'un emplacement libre dans la file d'attente, il accepte le flit en renvoyant un signal Ack au nœud émetteur en-amont, sinon il ne l'accepte pas et il renvoie un signal Nack au nœud émetteur en-amont. Le mécanisme de contrôle de flux Ack/Nack est traditionnellement considéré comme inefficace en termes de stockage (chaque flit transmis doit attendre un accusé de réception) et la bande passante (en raison des retransmissions potentiels) [36].

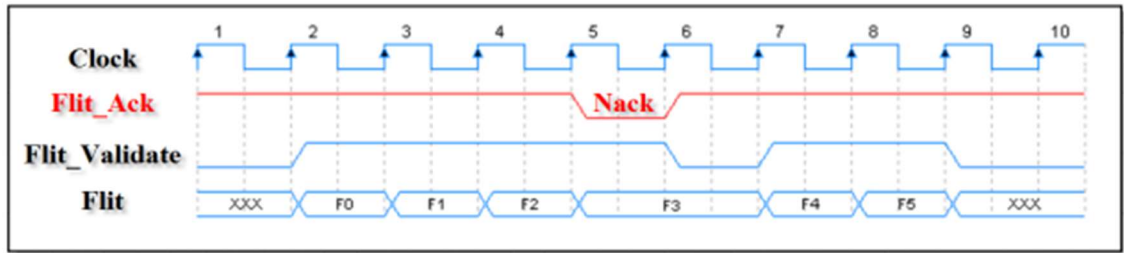


Figure 1.16 : Mécanisme de contrôle de flux Ack/Nack [26]

2. Contrôle de flux par credit-base :

Credit-Based est un mécanisme de contrôle de flux, où le nœud émetteur a un compteur pour suivre le nombre d'emplacements libres disponibles dans la file d'attente du nœud récepteur (le crédit), ce nombre d'emplacements est équivalent au nombre de flits pouvant être émis en séquence. Le compteur est décrémenté à chaque fois qu'un flit est transmis et incrémenté à chaque fois qu'un signal de crédit arrive du nœud récepteur, qui à son tour envoie le crédit à chaque fois qu'il a réussi à transmettre un flit de sa file d'attente pour le prochain saut. Par rapport au contrôle de flux On/Off, le contrôle de flux Credit-Based nécessite plus de signalisation de contrôle, mais de plus petites files d'attente. En particulier, pour la justesse du protocole, une  $Q_{min} = 1$  est suffisante. Toutefois, cette petite taille conduit à l'insertion de flits vides (des bulles) à chaque saut, vu qu'un seul crédit est disponible sur chaque canal à un moment donné. Par conséquent, un seul flit peut être transmis à chaque aller-retour de ce crédit, et plus la valeur de la latence du canal  $K > 1$  est élevée, plus la performance est mauvaise. Pour éviter l'insertion de bulles, la file d'attente doit être dimensionnée en fonction de la latence aller-retour.

La figure 1.17 montre un exemple du mode de fonctionnement de ce mécanisme de contrôle de flux.

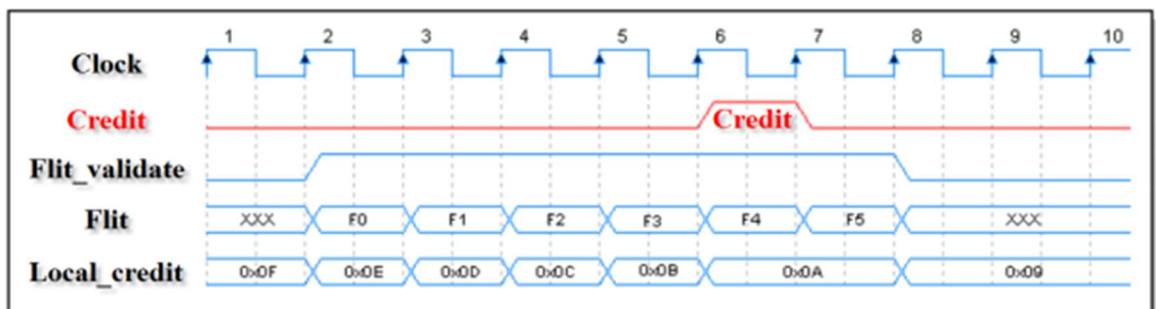


Figure 1.17 : Mécanisme de contrôle de flux Credit-Based [26]

## 7. Paramètres de performance du réseau sur puce :

Comme d'autres réseaux, la performance d'un Noc est évaluée par de nombreux critères tels que :

1/ La latence : La latence est définie par le délai moyen (en nombre de cycles) nécessaire pour transférer une unité de données (message, paquet ou fit) de la source vers destination.

La latence est le nombre de cycles écoulés depuis la première tentative d'injection du fit d'entête (header) jusqu'à la réception du dernier fit (tai) par la destination [37].

La valeur de la latence va dépendre de deux choses :

- La position relative de l'émetteur par rapport au récepteur
- La congestion dans le réseau

2/La superficie : La surface est une des premières métriques qui est évaluée car elle contribue directement au coût du circuit à fabriquer.

3/Le débit : C'est le nombre total de paquets qui atteignent leur destination par unité de temps.

4/La charge du lien : La charge du lien est définie comme la quantité de données circulant sur chaque lien à chaque direction, Le retard du réseau dépend fortement de la charge des liens.

5/La consommation d'énergie : La consommation d'énergie a toujours été un paramètre important de la performance dans tous types de réseaux. Les algorithmes de routage du Noc sont également évalués sur la base de l'énergie consommée par les routeurs correspondants.

## 8. Phases de conception de réseau sur puce :

Il est critique de concevoir un Noc efficace pour exploiter pleinement le nombre et la puissance de traitement des IPs dans un même circuit.

La conception d'un Noc passe généralement par les phases suivantes :

- Modélisation du trafic généré par l'application ;
- Choix de la topologie adaptée ;
- Assignation des tâches aux IPs ;
- Mapping des IPs sur la Topologie ;
- Allocation des chemins de routage et réservation des ressources ;

- Test de performance ;
- Implémentation ;

## 9. Outils d'aide à la conception des réseaux sur puce :

La conception d'un réseau sur puce est une tâche complexe, il est donc nécessaire d'avoir un outil d'aide à la conception pour les Noc. Cet outil peut être automatique ou semi-automatique afin de permettre au concepteur d'avoir la possibilité de forcer des contraintes dans le flot de conception,

On va illustrer quelque outil d'aide à la conception :

### 9.1. NOXIM :

Cet outil a été proposé par la computer architecture team [38], il permet à l'utilisateur de définir une architecture Noc 2D-mesh avec différents paramètres :

- Taille du réseau
- Taille des tampons
- Taille du paquet
- Algorithme de routage
- Taux d'injection des paquets

### 9.2. NS-2 :

NS-2 d'abord été développé pour le prototypage et la simulation de réseaux informatiques ordinaires.

NS-2 a été largement utilisé par de nombreux chercheurs pour simuler le Noc [39], utilisé dans de nombreuses études sur les Noc comme outil de simulation, ce qui est une référence fiable pour comparer les performances de deux architectures différentes.

Enfin NS-2 est un simulateur open source piloté par événements discrets et développé en C++ et OTCL,

Cette modularité et cette disponibilité ont facilité sa diffusion entre les chercheurs.

### 9.3. SunFloor :

Est un outil d'aide la conception des Nocs. Il peut être utilisé dès les premières phases de conception pour synthétiser les topologies avec ces contraintes en entrée (modèle, énergie et espace, objectifs de conception). A partir de ces données, SunFloor

génère une spécification système prête à être traduite en architecture complète, généralement en langage System [40].

#### 9.4. SunFloor 3D :

Est une extension de la version précédente. La principale fonctionnalité ajoutée est la génération de spécifications pour les Nocs [41].

#### 9.5. ORION :

Il a été proposée par une équipe de l'université dans [42] , il s'agit d'un simulateur dédié principalement à l'estimation de la puissance et de l'espace pour les architectures Noc.

#### 9.6. DARSIM :

Est un simulateur Noc développé au (MIT), cet outil permet la simulation d'architecture mesh 2 et 3 dimensions. Il propose une multitude de configurations de simulation pour Noc avec différents paramètres [43].

#### 9.7. Comparaison entre les outils d'aide à la conception :

Tableau 1.4 : comparaison entre les outils de conception [38]

Outil	Année	Paramètre					Disponibilité
		TR	TT	TP	AR	TIP	
<b>Noxim</b>	<b>2010</b>	+	+	+	+	+	+
<b>NS-2</b>	<b>1995</b>	+	+	+	+	+	+
<b>SunFloor-3D</b>	<b>2006 2009</b>	+	-	-	-	-	-
<b>ORION 1 et 2</b>	<b>2003 2009</b>	-	+	-	-	-	-
<b>DARSIM</b>	<b>2009</b>	+	+	+	+	+	-

TR : taille du réseau. TT : Taille des tampons. TP : Taille des paquets

AR : Algorithme de routage. TIP : Taux d'injection des paquets

+ : OUI - : NON

Tous les outils de cette liste ont été présentés par la communauté scientifique à travers des équipes de recherche. Néanmoins, ils ne sont pas tous accessibles, on trouve que Noxim qui est toujours maintenu par ses développeurs et NS-2 avec sa nouvelle version NS-3. Toutefois, il y a d'autres propositions de l'industrie, comme FLexNoC présenté par ARTERIS, the CHAIN works tool suite concut par silistix ou encore iNoC. Ces outils sont disponibles mais ils sont payants sauf NS-2.

## 10. Importance des réseaux sur puce pour les applications futures :

Plusieurs systèmes sur puce utilisent aujourd'hui le réseau comme support de communication entre les différents composants qu'ils gèrent, ceci n'impose pas que tous les SOC utilisent les NOC, mais la tendance actuelle est son utilisation systématique pour les applications futures. En effet une étude faite par Sonics en 2013 [32] montre que 47% des organismes concernés par la conception de SOC ont déjà utilisé des NOC ou ont l'intention d'évaluer des NOC commercialisés.

On trouve plusieurs réseaux développés mais seuls quelques-uns ont été commercialisés. Ceci est dû à la complexité du processus de conception et le coût qu'il présente. Par la suite, on montrera quelques exemples des réseaux commercialisés ou faisant partie des systèmes commercialisés.

- Epiphany 9 : c'est un coprocesseur (comprenant 64 cœurs) développé par Adapteva implémenté autour d'un réseau Mesh à deux dimensions, on s'attend à ce que la future version contienne plus de 4096 cœurs. Des solutions comme la maquette Parallella et la maquette ZED Board intègrent le composant Epiphany 9 [44].

- Net speed orion : c'est un réseau dont l'architecture est définie par l'outil NOCSTUDIO, cet outil permet de déterminer la topologie, la structure du routeur et l'algorithme de routage permettant d'atteindre les performances les plus optimisés selon les exigences des applications à intégrer [45].

- NoC d'Arteris: est le premier réseau commercial. La plupart des paramètres de conception de NoC d'Arteris sont définis par l'utilisateur à savoir la topologie du réseau, l'algorithme de routage et le nombre de ports d'entrée et de sortie sur les commutateurs. Le contrôle de flux du réseau peut être optimisé pour les besoins des applications en combinant des différentes méthodes de contrôle [46].

- STNoC est un réseau commercial fait par STMicroelectronics. C'est une application simple de mise en œuvre qui utilise le principe du wormhole pour l'acheminement des données avec un routage déterministe à la source et qui dispose d'une topologie Spidergon [47].

- Processeur IBM Cell : L'architecture de processeur IBM cell adopte un algorithme de routage déterministe en raison de la simplicité de sa topologie ayant un petit nombre de nœuds. Il s'agit d'un ensemble de huit processeurs qui sont reliés dans une topologie en anneau avec une largeur de 4 connections, une fonction d'arbitrage est adoptée afin d'empêcher le parcours de plus que la moitié du diamètre du réseau (un parcours maximal est limité par un chemin à 4 sauts (hops)) [48].

- SonicsGNest un réseau de Sonics appelé aussi (SGN), est un réseau configurable avec une topologie flexible, ce réseau est compatible avec les interfaces OCP et AMBA , la configuration du réseau est faite à partir de l'outil de développement SonicsStudio® Director qui permet au concepteur, à l'aide d'une méthodologie bien précise, de créer, optimiser, et vérifier le fonctionnement du réseau [49].



## 11. Conclusion :

Dans ce premier chapitre, nous avons présenté les principales caractéristiques des réseaux sur puce. Puis nous avons identifié les problématiques et les limitations des architectures de communication intégrées sur le bus partagé, largement utilisés jusqu'à maintenant, avec l'accroissement des besoins en bande passante, en performances ainsi que les contraintes liées aux nouvelles technologies d'intégration.

En effet, les Noc offrent de grandes perspectives de par leur structure mais également par la possibilité de proposer une qualité de service pour les communications de l'application. Ces réseaux peuvent avoir des topologies différentes mais également des gestions de flux de données différentes influant directement sur la complexité matérielle du Noc mais également sur latence du transport des données. Ces Noc sont donc extrêmement paramétrables.

Ainsi, de nombreux paramètres sont à prendre en compte lors de leur conception, car chacun d'entre eux peut influencer directement ou indirectement les performances globales de l'architecture.

Notre objectif dans ce mémoire est d'étudier la performance des topologies des NoCs en maximisant les paramètres physiques.

## Chapitre 2

### Etat de l'art : Optimisation des Topologies

#### 1.Introduction

Nous présentons dans ce chapitre un état de l'art sur les approches d'optimisation et de personnalisation des Topologies des NoCs basé sur la topologie, Ensuite nous présentons leurs avantages et inconvénients, Enfin, nous présentons une synthèse qui résume nos choix d'optimisations.

#### 2. Classification des approches de personnalisation et d'optimisation :

Nous en s'intéresse juste au niveau physique de la personnalisation de la topologie.

Les approches d'optimisation de performance du Nocs, dans la phase de conception et dans la phase d'exécution, peuvent être classées en fonction du niveau auquel l'optimisation s'effectue, c'est à dire au niveau physique, au niveau communication et au niveau application, comme l'illustre la figure 2.1 [26].

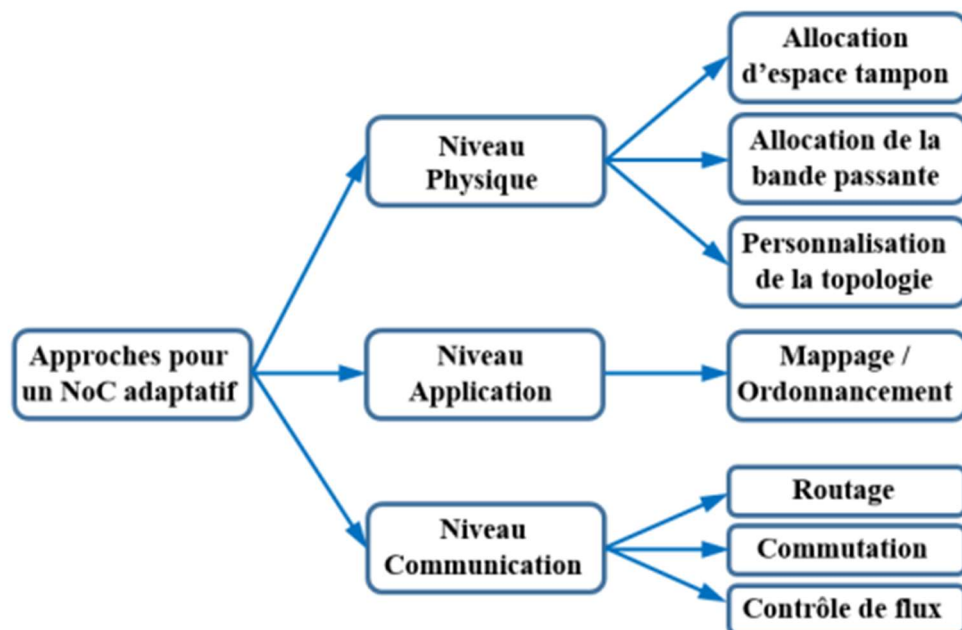


Figure 2.1 : Classification des approches de personnalisation et d'optimisation des NoCs [26]

### 3. Probleme d'optimisation :

Les problèmes d'optimisation ont été définis par un espace d'état, une ou plusieurs fonction(s) objective(s) et un ensemble de contrainte [50].

1. L'espace d'état est défini par l'ensemble de domaines de définition des variables du problème.
2. Les variables du problème peuvent être de nature diverse (réelle, entière, booléenne, etc...).
3. Une fonction objective représente le but à atteindre pour le décideur (minimisation de cout, de durée, d'erreur, ... Elle définit un espace de solutions potentielles au problème.
4. L'ensemble de contraintes définit des conditions sur l'espace d'état que les variables doivent satisfaire. Ces contraintes sont souvent des contraintes d'inégalité ou d'égalité et permettent en général de limiter l'espace de recherche.
5. Une méthode d'optimisation cherche le point ou un ensemble de points de l'espace d'état possible qui satisfait au mieux un ou plusieurs critères. Le résultat est appelé valeur optimale ou optimum.
6. Optimum / valeur optimale est le résultat obtenu à la fin du processus (minimum ou bien le maximum).

Cependant, il peut exister des solutions intermédiaires, qui sont également des optimums, mais uniquement pour un sous-espace restreint de l'espace de recherche : nous parlons alors d'optimums locaux. Cette notion est illustrée dans la figure 2.2.

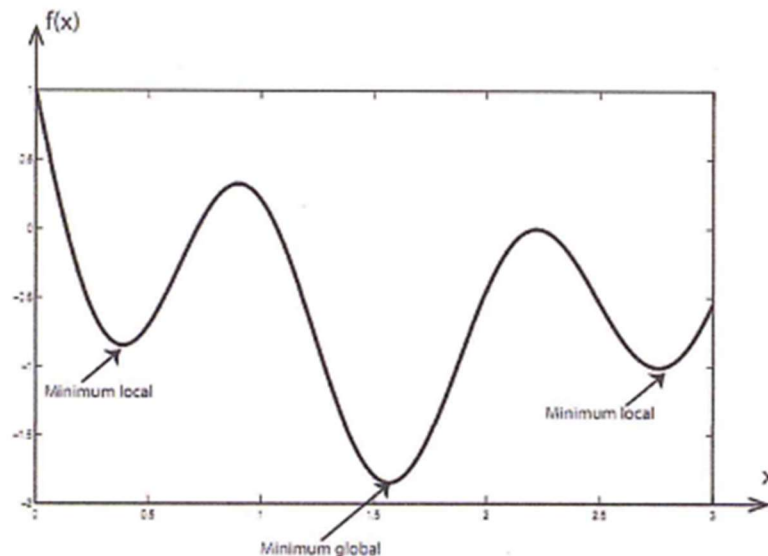


Figure 2.2 : Différence entre un optimum global et des optimum local [50]

#### 4. Caractéristique des topologies régulière et personnalisée :

Un récapitulatif de comparaison des différents points entre une topologie régulière et autre personnalisée est présenté dans le tableau 2.1 de Xinyu Li [51].

Tableau 2.1 : comparaison entre topologie régulière et irrégulière [51]

Topologie régulière	Topologie personnalisée
Maillée, Torus, . . .	Topologie irrégulière
Soc utilité générale	Soc application spécifique
Routeurs homogènes	Routeurs hétérogènes
Topologie réutilisées	Conception des routeurs réutilisées
Peu de temps de conception	Long temps de conception
Performances basses	Performances élevées
Energie élevées	Energie basse

Les NoCs de la littérature utilisent généralement des topologies régulières. La topologie standard la plus utilisée est la topologie en grille 2D (2D mesh).

Une structure régulière présente l'intérêt d'être une topologie de structure mathématique simple, ce qui permet d'utiliser des règles de routage simples. Cependant, en pratique, le placement et la taille des composants IP du SoC permettent rarement d'intégrer une topologie régulière.

Une topologie irrégulière ou personnalisée permet plus de liberté et ainsi de tailler précisément le réseau requis. Elle peut être issue d'une topologie régulière qui a été retaillée au plus juste de façon à enlever les éléments non utilisés. Une topologie irrégulière nécessite en revanche une plus grande attention pour le routage car les règles à appliquer ne sont plus triviales. Ainsi, dans la mesure du possible nous privilégions l'utilisation d'une topologie régulière, mais nous devons disposer d'une solution qui permette l'utilisation plus générale d'une topologie quelconque.

#### 5. APERÇU DE LA TECHNOLOGIE IC 3D :

La technologie 3D IC est la nouvelle approche pour augmenter les performances de l'appareil en empilant plusieurs matrices ou plaquettes les unes sur les autres. Il est maintenant considéré comme une solution viable, car le nœud technologique est approché la technologie submicronique très profonde où de nombreux problèmes qui ne

sont pas un problème dans l'ancienne technologie deviennent les paramètres critiques qui doivent être pris en considération en raison des effets sur les performances.

La conception d'un réseau sur puce à 2 dimensions a été l'objet de recherche d'un grand nombre des travaux ; mais l'intégration de plus de cœurs d'IP afin de répondre aux exigences requises par les applications modernes, des réseaux de grand taille doivent être mis en place pour communiquer ces différents cœurs, cependant le fait d'agrandir la taille de réseau a montré des mauvaises conséquences. L'apparition de la technologie des circuits intégrés à 3 dimensions (3D IC), qui est définie par des multiples couches en silicium empilés ensemble et communiquant à travers des liens d'interconnexions verticaux qu'on nomme TSV (Through Silicon Via), a encouragé les chercheurs à implémenter leurs réseaux en utilisant cette nouvelle technologie pour en exploiter les avantages par rapport aux circuits à 2 dimensions.

### 5.1. Technologie IC 3D :

L'intégration 3D offre une méthode moins difficile pour obtenir une intégration de transistor plus élevée pour les besoins de l'application actuelle par rapport à la mise à l'échelle du transistor vers des nœuds technologiques plus petits. Plutôt que définissant de nouveaux processus et recherchant des solutions aux nouveaux défis créés par les nouvelles technologies de processus, l'intégration 3D nous permet d'utiliser une technologie plus ancienne et mature, par exemple 65 nm, pour doubler la densité de transistors et ainsi fournir plus de fonctionnalités. Bien que l'intégration 3D ait été étudiée depuis 1986 dans [52], elle n'était qu'au stade de la recherche sans commercialisation en raison de la limitation technologique à l'époque. Avec les technologies des semi-conducteurs d'aujourd'hui, la technologie 3D peut être conçue et mise en œuvre à un coût relativement faible. Cependant, pour que cette technologie soit largement adoptée dans l'industrie, plusieurs obstacles critiques doivent être résolus.

Plusieurs moyens peuvent être utilisés pour fournir des connexions électriques entre les matrices dans l'empilement 3D, tels que l'utilisation du TSV, du câblage filaire ou sans contact, comme illustré à la Figure 2.3. Les méthodes d'empilement déterminent le compromis en termes de densité et de coût d'interconnexion. La technique de connexion par fil fournit une connexion puce à puce depuis l'extérieur de la puce à l'aide de fils de connexion qui se trouvent normalement dans l'emballage 3D

Les puces sont empilées à l'aide de la technologie d'emballage en se collant les unes aux autres. Cette méthode est limitée à la résolution des wire bonders, par exemple 35  $\mu\text{m}$  pour un fil de 15  $\mu\text{m}$ . Cela le rend plus difficile lorsque le nombre d'E/S pour chaque puce de la pile augmente. Ce type de structure a été démontré en utilisant plus de deux filières. Pour une communication inter-matrice sans contact en 3D l'empilement, qu'il s'agisse d'un circuit capacitif ou inductif, peut être utilisé pour fournir une communication sans fil entre les puces de l'empilement. La communication par couplage capacitif est limitée à quelques micromètres tandis que le couplage inductif peut fournir une distance plus longue.

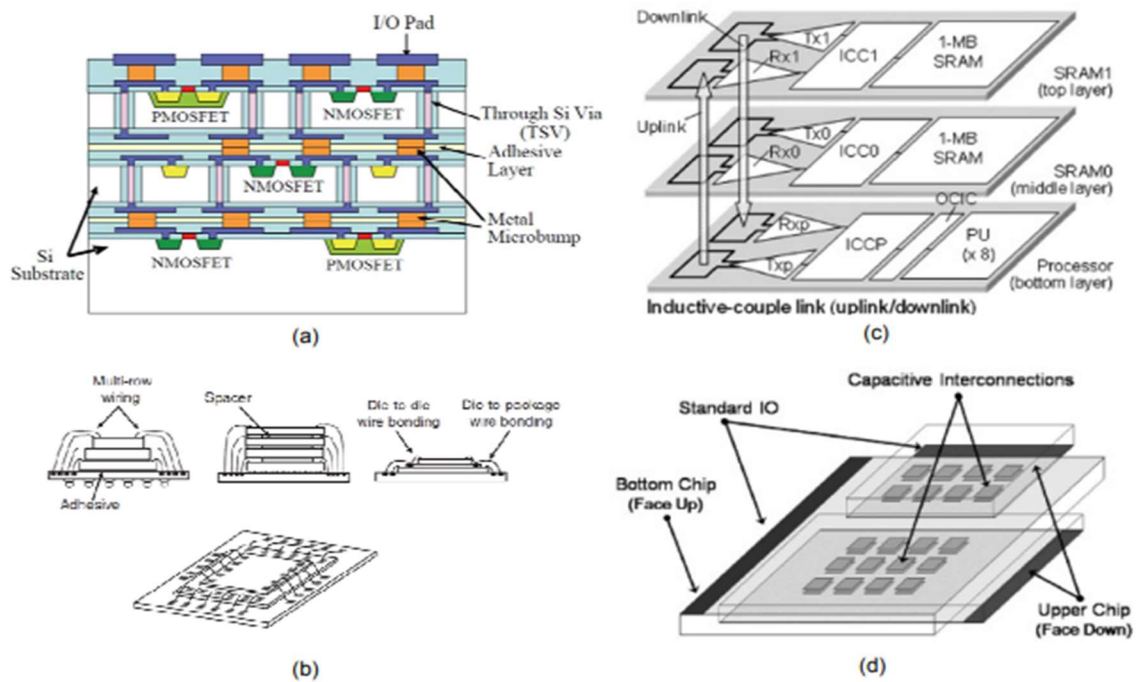


Figure 2.3 : Différents types de méthodes d'empilage [52]

## 5.2. Avantages du réseau à 3 dimensions :

Les améliorations des performances de réseaux 3D par rapport au réseau à 2 dimensions sont établies grâce à la réduction de nombre moyen de hops.

La réduction du nombre moyen de Hops est due à

- L'augmentation du nombre de liens d'interconnexions et à l'augmentation du degré de routeur.
- En effet, le routeur dans un réseau à 3 dimensions a au moins un lien de connexion supplémentaire par rapport à un routeur qui appartient à un réseau de deux dimensions, les équations qui correspondent au calcul du nombre de liens d'interconnexions et le nombre moyen de hop des réseaux a 3 dimensions sont données par l'équation Eq1 et l'équation Eq2 ci-dessous :

$$links = N_1 N_2 (N_3 - 1) + N_1 N_3 (N_2 - 1) + N_2 N_3 (N_1 - 1) \quad (1)$$

Avec  $N_i$  est le nombre des commutateurs dans la dimension  $i$  :

$$hops_{noc} = \frac{n_1 n_2 n_3 (n_1 + n_2 + n_3) - n_3 (n_1 + n_2) - n_1 n_2}{3(n_1 n_2 n_3 - 1)} \quad (2)$$

Pour comprendre l'importance de la réduction du nombre moyen de hops, prenons l'exemple d'un réseau mesh-2 D de taille 8\*8 et un réseau mesh à 3D de taille (4\*4\*4). Le nombre moyen de hops pour ces deux réseaux sont respectivement 5.33 et 3.81 alors on a 40% plus de hops pour un réseau à 2 dimensions par rapport à un réseau à 3 dimensions.

### 5.3. Technologie des TSVs :

TSV est une méthode qui utilise des via à travers différentes couches de silicium actif. Les matériaux utilisés pour le TSV sont le tungstène (W), le cuivre (Cu) et le poly-silicium (Poly-Si) [53]. Le matériau Poly-Si est stable et a moins d'effet sur les caractéristiques de l'appareil que les autres matériaux. Cependant, le cuivre ou le tungstène conviennent mieux au TSV en raison d'une résistance plus faible. Le cuivre est le plus couramment utilisé car il a une bonne conductivité thermique par rapport au tungstène et au poly-si. le cuivre TSV crée un effet de contrainte en raison d'une grande différence de coefficient de dilatation thermique (CTE) entre le substrat de silicium et le cuivre, ce qui n'est pas le cas pour le tungstène TSV. Une comparaison détaillée du matériau de remplissage de via peut être trouvée dans.

Le tungstène a un délai plus long par rapport au cuivre TSV pour n'importe quelle taille de diamètre et n'est donc utilisé que dans la recherche.

Il existe différentes techniques pour implémenter TSV pour empiler plusieurs niveaux tels que via-first, via middle ou via-last. Chaque méthode présente des caractéristiques différentes. Dans la technique via-first, le TSV est formé avant la structure BEOL. Nous avons donc une taille relativement petite par rapport aux deux autres méthodes. L'approche par voie intermédiaire crée le TSV après la formation de BEOL et avant la formation de FEOL.

La taille de TSV se situe entre la taille des méthodes via-first et via-last. Alors que dans l'approche via-last, le TSV est formé après la formation de BEOL, ce qui entraîne une grande taille de TSV. Cette méthode aussi pose un grand défi pendant la formation du processus de formation du TSV afin d'éviter d'endommager les dispositifs qui ont déjà été formés. Toutes les méthodes de formation du TSV sont illustrées dans la figure 2.4.

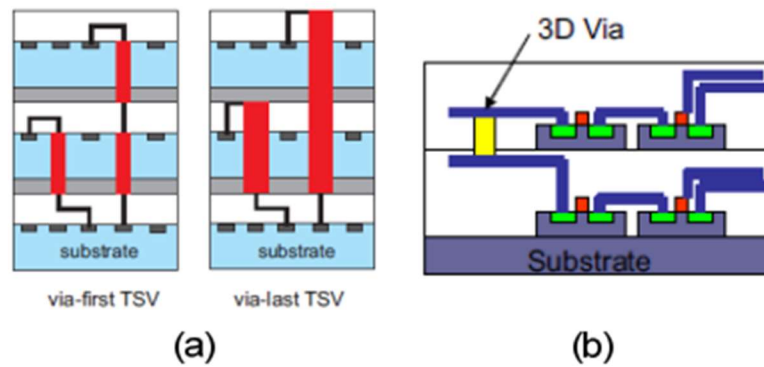


Figure 2.4 : Méthodes d'empilement TSV [53]

(a) via-first et via-last dans le CMOS en masse (b) via-first TSV dans SOI

## 6. Les approches d'optimisation des topologies :

Les approches peuvent être classées en deux catégories principales : des approches de synthétisation et des approches de personnalisation :

La première classe concerne les techniques proposées pour personnaliser entièrement l'architecture, elles sont plus appropriées pour une application spécifique, et elles ne sont pas nécessairement adaptées aux topologies régulières (ex., 2D-Mesh) [54] [55]. Par exemple, dans [56], un algorithme génétique est présenté pour la synthèse d'une application spécifique pour le Soc avec l'objectif d'optimiser la consommation d'énergie et de l'espace de conception tout en maximisant les performances (ex., la latence, le débit, etc.). Cet algorithme est itératif et permet la production et la sélection d'un nombre constant de configurations intermédiaires qui satisfont aux critères de fitness proposés. Cependant, trouver la meilleure configuration peut poser de sérieux problèmes, car la précision dépend de la fonction de fitness. En outre, malgré les capacités de ces techniques de maximiser la performance, en offrant des techniques pour incorporer les architectures d'interconnexion sur puce synthétisées qui sont plus adaptées pour une application spécifique, ils peuvent souffrir du fait de fournir des architectures avec une grande régularité dans un délai raisonnable.

L'autre classe concerne les approches de personnalisation partielle de la topologie. En revanche, elles commencent principalement par une architecture d'interconnexion sur puce régulière comme le 2D-Mesh. Puis, elles personnalisent partiellement l'architecture de base en insérant des longs liens. Par exemple, une approche de personnalisation d'un 2D-Mesh standard en insérant quelques longs liens spécifiques à l'application a été proposée dans [57].

L'algorithme commence par une architecture régulière de 2D mesh et ajoute progressivement des liens à longue portée entre les commutateurs selon la fréquence de



communication la plus élevée jusqu'à ce que toutes les ressources (liens) disponibles sont utilisées. Les résultats analytiques et de simulation ont été présentés pour montrer que cette approche permet de réduire considérablement la latence moyenne des paquets. Toutefois, le choix stratégique des liens à insérer sur le réseau d'interconnexion sur puce qui convient le mieux aux besoins de plusieurs applications est difficile sans connaître à l'avance leurs correspondances avec les modèles de trafic.

### 6.1. Topologie régulière :

La majorité des travaux déjà cités utilisent la topologie 2D-mesh, on va citer quelque approche :

- L'approche proposée par Chatha et al. [58] consiste à partir du placement des IPs allouer à chaque IP un routeur parmi les routeurs de ces coins. Ensuite, éliminer les routeurs non utilisés, cette dernière a subi un changement en nombre des liens associées à ces routeurs, donc c'est une approche de suppressions des liens basée sur une topologie régulière 2D-mesh.
- Une méthode dans [59] en ajoutant des liens supplémentaires a une topologie régulière 2D-mesh, pour alléger les routeurs qui présentent un goulot d'étranglement dans le système.
- Dans [60], la cible optimise la topologie du réseau, en utilisant une méthode de décomposition pour une entière personnalisation et l'insertion de longs liens aléatoires sur la topologie standard 2D-Mesh.

### 6.2. Topologie personnalisée :

- Dans [61], les auteurs présentent une méthodologie pour la personnalisation du crossbar STBUS (STBUS: Serial Telecom BUS) , nous présentons une méthode dérivé de la combinaison de deux mappages initiaux avec deux types de personnalisation de la topologie basée sur l'insertion de liens croisés ,Le premier type de personnalisation repose sur l'insertion de liens croisés génériques( connexions entre n'importe quel couple de nœuds) , le deuxième type est basé sur l'insertion de liens croisés spidergon (qui sont des connexions entre un nœud et son homologue diagonal dans le réseau).
- Des approches pour des réseaux sur puce reconfigurables qui permettent l'adaptation en cours d'exécution de la topologie aux exigences des applications sont proposées dans [62] [63] [64]. Ces approches sont compilées sur des composants FPGA (Field-Programmable Gate Array) et sont, cependant, limitée à

des composants reconfigurables. Une architecture Noc, appelée ReNoC (ReNoc: Reconfigurable Noc), pour permettre à la topologie du réseau d'être configurée par l'application exécutée sur l'architecture Soc. ReNoC combine la commutation de paquets et la commutation de circuits dans la même topologie. Cela permet de créer des topologies d'applications spécifiques dans une plateforme générale de SoC à base de NoC.

- CustomTopo, une approche général pour la génération de topologie WRONoC (wavelength-routed optical network-on-chip ) proposées dans [65] qui optimise l'utilisation des ressources pour les conceptions spécifiques à l'application. Nous analysons le problème de génération de topologie sous trois aspects : longueur d'onde l'utilisation, l'utilisation de l'ADF et la perte d'insertion, Custom Topo est applicable aux réseaux symétriques et asymétriques. Spécifiquement, pour les réseaux asymétriques,
- Un algorithme génétique a été proposé dans [66], dont l'objectif de minimiser la consommation d'énergie et la surface de silicium, en générant plusieurs topologies personnalisées. Il commence à former des configurations (architectures) aléatoirement, affecter les IPs d'une application aléatoirement, puis invoquant un algorithme appelé MSP ( Modified Shortest Path) afin de générer le chemin le plus court entre deux extrémités de communication source et destination.

### 6.3. Placement des liens verticaux :

Xu et al dans [67] on fait des expérimentations sur le placement des liens verticaux avec un nombre de TSVs a 100%, 50%, 25%, dans les architecture réseaux sur puce 3D (4\*4). Les résultats ont montré une réduction de 5.24% et 2.18% de débit (pour 100% et 50% respectivement) par rapport au 25% de liens.

Paricha a proposé dans [68] une technique de sérialisation pour réduire le nombre de TSVs. L'idée consiste à mettre un seuil de nombre de TSV. S'il dépasse ce seuil, une sérialisation est adoptée afin de réduire la bande passante de certains liens au lieu de les supprimer totalement. Cependant, cette technique entraine une grande latence de transfert de paquets ainsi qu'une grande consommation en énergie.

Une autre étude a permis de présenter l'impact des TSV sur les NoC dans xiangyu et al [69]. Ils ont montré que la surface totale consommé par les TSVs augmente chaque fois que la taille du réseau augmente. Dans le cas général, la taille du réseau désigne le nombre de couches verticales (plus de 3 couches peut causer une hausse de température.

## 6.4. Synthèse sur les approches d'optimisation des topologies :

En résumé ci-dessous les optimisations des topologies les plus utilisées dans les NoCs :

Tableau 2.2 : résumé des approches d'optimisation (personnalisation)

	auteur	Descriptions Méthodes	spécification	Type ,Topologie	objectif	Années
1	[Palermo, G 61]	ST Noc (basé sur insertion)	MP Soc	Statique ; Ring ,spider gon en 2D	réduire la distance du réseau	2007
2	[Ogres, U. Y., & Macules, R ,70]	Long-Range Link Insertion	ASIC	Statique ; Mesh-2D	réduire latence	2006
3	[Chen, G., Li, F., & Kantemir, M, 71]	hybride compilateur-exécution	ASIC	Dynamique Mesh-2D	Réduire la consommation d'énergie	2007
4	[ Mengchu Li, Tsun-Ming , 65]	WRONoCs (réseaux optiques sur puces à acheminement par longueur d'onde)	ASIC	Personnalisée en 2D (Optical)	Densité de communication réseau, réduire longueur d'onde	2018
5	[ Srinivasan, K., Krishnan , 66]	algorithme génétique (basé sur le regroupement)	MP Soc	Statique ; personnalisée	Réduire la consommation d'énergie	2005
6	[Guz, Z., Walter, I. H., Bolotin, E., Cidon, I., Ginosar ,72]	L'ALLOCATION DE CAPACITÉ DE LIAISON	MP Soc	Statique ; Mesh-2D	Réduction en termes de zone de liaison et de puissance	2007
7	[Chatha et al. , 58]	éliminer les routeurs non utilisés	ASIC	Statique ; Mesh-2d	Réduction d'énergie	2008
8	[Jiao, J., & Fu, Y , 73]	BRANCH BOUND ALGORITHM(based on insertion link)	MP Soc	Mesh-2D	la flexibilité, convergence , faible latence	2011
9	[Venkataraman, N. L., & Kumar, R, 74]	ASNoC	ASIC	personnalisée	Moins de surface, moins d'énergie	2019
10	[ Gulzari, U. A., Anjum, S., Aghaa, 75]	cross-by-pass link(basé sur insertion)	MP Soc	Statique ; Mesh-2D	réduire la latence moyenne et augmenter le débit	2017
11	[Yan, S., & Lin, B, 76]	Ripup-Reroute-andRouter-Merging (RRRM)	ASIC	Dynamique Personnalisée en 3D	Réduire l'énergie	2011

D'après le tableau en peut déduire notre but a focalisé la personnalisation de la Topologie :

Tableau 2.3 : Les avantages et les inconvénients des approches

Approche	Avantages	inconvénients
Insertion des liens en 2d	++ Augmentation en débit	+ Plus énergie
Suppression des liens en 2d	- Moins d'énergie (par rapport au topologie initial) Moins de cout	-Réduction en débit
Insertion des liens TSVs	+ Augmentation en débit	+ plus de surface + Plus énergie
Suppression des liens des TSVs	-- Moins d'énergie -- Surface	- moins débit

- diminution      - - plus de diminution

+ augmentation    ++ plus d'augmentation

D'après ce tableau en distingue que chacune d'elle apportent des avantages et des inconvénients, Nous nous intéresseront sur les deux mesures suivantes :

- Augmentation en débit par rapport à l'insertion des liens en 2D.
- Réduction de la Surface par rapport à la suppression des liens des TSVs en 3D.

## 7. Méthodes et outils d'évaluation de performance du Noc :

Les architectures NoC peuvent être évaluées soit par expérimentation avec les systèmes existants, soit par expérimentation avec des modèles de systèmes (modèles de simulation ou d'évaluation analytique), comme le montre la figure 2.5.

Cependant, l'expérimentation des performances des NoCs avec des vrais systèmes sur FPGA est très coûteuse. Par conséquent, la modélisation est la solution la plus efficace pour l'évaluation et l'optimisation de l'architecture du NoC avant sa mise en œuvre sur la puce.

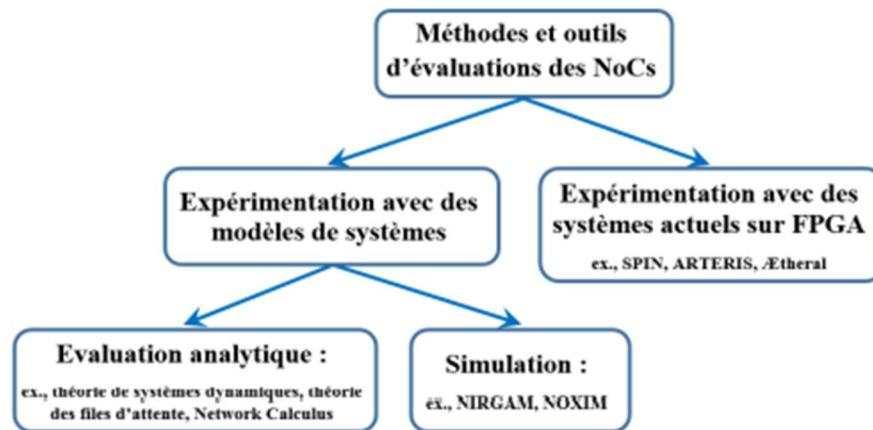


Figure 2.5 : Méthodes et outils d'évaluation d'un réseau sur puce [26]

## 7.1. Evaluation des Nocs par des modèles de système :

### 7.1.1. Evaluation analytique :

Il existe plusieurs modèles analytiques d'évaluation de performance appropriés, ils peuvent estimer très tôt dans la phase de conception des indicateurs de performance souhaités en une fraction de temps que prendrait la simulation. Bien que l'utilisation des modèles de haut niveau recèle beaucoup d'aspects technologiques complexes, elle facilite l'exploration rapide de l'espace de conception NoC.

Parmi les modèles les plus couramment utilisés dans la littérature [77], on peut citer :

- la théorie des files d'attente (QT: Queuing Theory),
- le Network Calculus (NC: Network Calculus),
- l'analyse d'ordonnancement (SA: Schedulability Analysis),
- l'analyse du flot de données (DF: DataFlow Analysis).
- La théorie des systèmes dynamiques (DS: Dynamique Systems)
- La modélisation par réseau à compartiments (CFF: Compartmental Fluid-Flow)

### 7.1.2. Résumé sur les modèles d'évaluation analytique :

Le compromis général entre l'abstraction et la précision est peut-être observé dans la comparaison entre ces quatre formalismes. Comme chacun des formalismes examinés a des avantages et des inconvénients comme le montre le tableau 2.4 :

## 7.2. Evaluation par simulation :

### 7.2.1. Classification des simulateurs Noc :

Cette section a pour but d'analyser les différentes solutions de simulation de NoC existantes afin de choisir l'une d'entre elles qui offre un bon rapport de simplicité, rapidité, possibilité de personnalisation de l'architecture NoC (topologie, espace tampon, etc.), possibilité d'évaluation des performances (latence et consommation d'énergie).

#### a. Simulateurs de systèmes :

Ils proposent des plateformes de simulation MPSoC mais ils se focalisent sur un point particulier de l'architecture (processeur, tolérance aux fautes...) et ne permettent pas d'explorer tous les paramètres de la plateforme, comme : SimOS, Simics, RSIM

#### b. Simulateurs de réseaux d'interconnexion :

Cette catégorie de simulateurs autorise l'exploration du réseau d'interconnexions. En effet, ces simulateurs sont, par nature, conçus pour explorer les réseaux selon un maximum de paramètres. On peut citer Atlas [78], µSpider [79], NoCGen [80], Noxim [81],

#### c. Simulateurs mixtes :

Cette catégorie de simulateur permet une exploration complète d'une architecture multiprocesseur, du réseau d'interconnexions aux processeurs. Dans cette catégorie, on trouve deux approches afin de permettre une simulation totale.

La première approche consiste à coupler un simulateur de la première et de la deuxième catégorie. Ainsi on obtient un simulateur complet qui permet une simulation de toutes les parties du multiprocesseur. On peut citer l'association Sicosys+RSIM [82] ou encore GEMS+GARNET [83]. Dans les deux cas, le point délicat consiste à fournir une interface adaptant les protocoles de communication des deux simulateurs.

La deuxième approche rectifie cette faiblesse en proposant des simulateurs complets destinés à l'exploration architecturale. Mentor Graphics [84] et Coware [85] proposent, chacun, une plate-forme complète mais payante permettant l'exploration architecturale. ASIM, MCSIM, MPARM et Soclib sont des simulateurs MPSoC complets.

### 7.2.2. Simulateurs Noc existants :

L'étude de l'existant de la section précédant (Classification des simulateurs Noc) nous a permis de voir les différentes approches pour simuler une plate-forme MP Soc/NoC. Il apparaît qu'au moment du choix aucune solution ne correspondait exactement à nos besoins de simulations de réseau d'interconnexion. Cette étude met en évidence que dans la majorité des cas, la modélisation du réseau d'interconnexions est réalisée en décrivant chaque élément de cette interconnexion. Ceci peut poser un problème si on envisage de simuler un réseau de grande taille. Par exemple en SystemC, il en résulte un grand nombre de threads SystemC ce qui risque de ralentir la simulation.

Pour cette raison nous avons voulu voir plus en détail les simulateurs de réseau d'interconnexion les plus proches de nos besoins de simulation, leur description est présentée dans cette section, afin de pouvoir choisir un simulateur qui convient le plus à la simulation de nos travaux sur le Noc :

1. ARTERIS [86] est le premier outil de conception de réseaux sur puce à avoir été commercialisé. Il est une suite d'outils développés par la compagnie ARTERIS. Cette suite comprend deux outils d'aide à la conception et à l'exploration architecturale.
2. BookSim [87] est un simulateur à cycle précis de réseaux d'interconnexion. Initialement développé et introduit avec les principes et pratiques d'un livre de réseaux d'interconnexion, depuis sa première apparition sa fonctionnalité a été étendue en permanence. La version actuelle, BookSim 2.0.
3. HNoCS [88] est une implémentation open-source d'une plateforme de simulation NoC en utilisant OMNeT++ . Le moteur de simulation à base d'événements OMNeT++ offre des APIs C++ pour un ensemble de services qui peuvent être utilisés pour modéliser, configurer, décrire la topologie, recueillir des données de simulation et effectuer des analyses.
4. Noxim [89] est un outil de simulation MSoC/NoC développé à l'Université de Catane (Italie). L'outil est disponible gratuitement sur le site Noxim via SourceForge sous conditions de la licence GPL. Il est compilé pour un environnement Linux. Noxim est écrit en SystemC.
5. NIRGAM [90] est simulateur SystemC à base d'événements discrets et de cycle précis pour la conception de NoC en termes d'algorithmes de routage et d'applications sur diverses topologies (2D Mesh et 2D Torus actuellement mises en œuvre).

## 8.Conclusion :

Dans le deuxième chapitre nous avons cités les classifications des approches de personnalisation et d'optimisation des architectures topologies, ensuite nous avons vu les caractéristiques des topologies régulières et personnalisées, après on a cités les différentes approches d'optimisation. Et à la fin on a vu les Méthodes et outils d'évaluation de performance du Noc.

Ces recherches ont souligné aussi l'importance de maintenir un équilibre entre les performances et l'utilisations de la technologie 3D, et aussi l'importance des TSVs.

Nous proposons dans le chapitre suivant, une approche d'optimisation pour les mesures citées précédemment (débit, énergie de consommation) en considérant la technologie 3D en se basant sur l'insertions des liens horizontaux dans une topologie 2D (inter-couche).

# Chapitre 3

## La conception de la technique proposée

### 1.Introduction

Le nouveau paradigme de communication (réseau sur puce) permet un gain considérable en performance.

Cependant, plusieurs critères et paramètres influencent sur ces performances, nous citons le coût en surface de silicium et énergie de consommation et latence, ces mesures et cela que nous prenons en considération pour optimiser la conception.

Dans ce chapitre, notre idée consiste à la phase de personnalisation de topologie en utilisant la technologie 3D en insérant des liens horizontaux entre les routeurs de même couche en 2D.

La formulation mathématique de notre problématique ainsi que les détails de notre solution sont expliqués dans les paragraphes suivants.

### 2.Description du problème d'optimisations :

Dans notre travail nous avons considéré un réseau 3D mesh complètement connecté entre deux niveaux seulement, Chaque niveau doit être symétrique par rapport au nombre des routeurs, nombre des liens.

La personnalisation de la topologie est un problème d'optimisation multi-objectif à pour but d'augmenter les critères de performance (débit, énergie de consommation).

Dans la conception proposée on va ajouter des liens entre routeurs mais cette optimisation par insertion a des inconvénients (plus énergie et plus de surface) à cet effet on ajoute la contrainte suivante :

Pour chaque deux nœuds adjacents dont chacun a un lien vertical TSV on élimine un des liens verticaux.

Cette contrainte a pour but réduit la surface de puce.



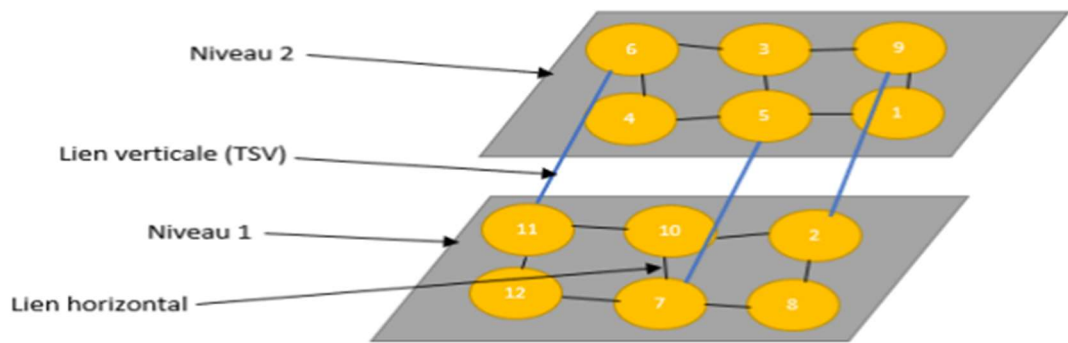


FIG. 3.1 : exemple de notre supposition [26]

Dans notre conception, on ne va pas maitre le placement des IPs (mapping), on met on considération Ce calcul de placement des IPs est déjà fait par applications données comme illustre la figure suivante :

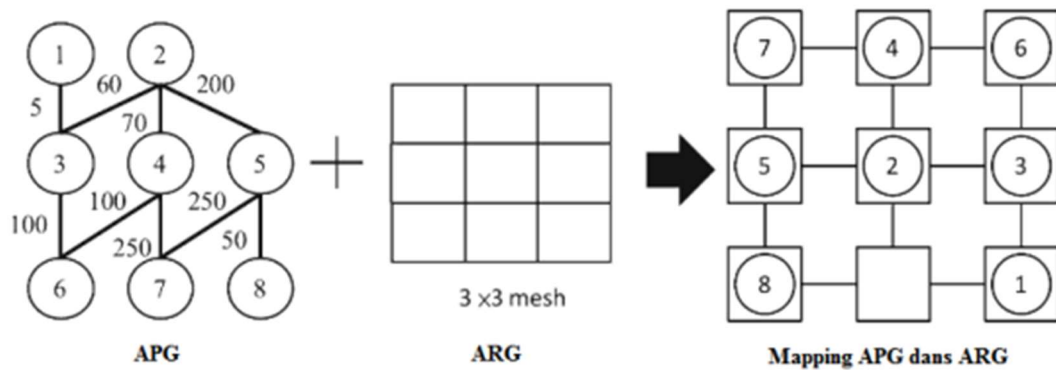


Figure 3.2 : exemple d'architecture après le mapping du ASIC [91]

### 3. Approche pour la personnalisation des architectures NoCs :

Dans cette approche nous avons construire architecture optimiser basé sur un topologie standard 3D mesh.

L'insertion des liens horizontaux : Consiste à insérer itérativement des liens stratégique (par un nombre d'insertion donné) pour chaque couche, donc chaque itération en insère un nombre de liens égal au nombre de niveaux.

Pour chaque insertion le but est d'optimiser le mieux possible les critères suivants :

- Augmenter le degré de clusteration (Cn)
- Réduire la distance Moyenne (Dm)

Lorsque on choisit le lien optimal à insérer on l'ajoute ce lien parallèlement deux fois dans chaque couche.

### 3.1. Hypothèses de base :

Pour permettre aux concepteurs d'adapter un réseau NoC donné afin qu'il corresponde à une grande charge de travail de différentes applications, nous avons proposé un algorithme d'insertion de liens. Cet algorithme se résume en deux étapes ; la première étape consiste à sélectionner les liens stratégiques à ajouter sur le NoC ; la deuxième étape consiste à la mise en œuvre de ces liens sur l'architecture NoC de base souhaitée.

Afin de personnaliser la topologie du NoC sans avoir recours au modèle de trafic, nous avons voulu améliorer les caractéristiques physiques de la topologie du NoC, ce qui aura forcément un impact direct sur les performances de communication de celui-ci. Nous avons donc mis en place plusieurs hypothèses, comme par exemple :

- Un petit diamètre permet une communication rapide entre les nœuds les plus éloignés. En d'autres termes, la latence maximale est proportionnelle au nombre maximal de sauts, c'est pour cette raison que le diamètre est considéré comme une mesure importante.
- De même, une faible distance moyenne du NoC peut avoir un impact fort sur la latence moyenne de la communication sur puce.
- Un degré moyen élevé des nœuds permet d'avoir des communications rapides entre les voisins proches. Toutefois, plus le degré moyen est faible plus les coûts matériels sont réduits et la capacité d'extension du NoC est augmentée.
- Pour fournir plus de chemins entre deux sous-réseaux, et améliorer donc la performance globale du réseau, il est nécessaire d'avoir une grande bissection.
- L'ajout de liens dans un NoC pourrait réduire le diamètre, d'améliorer la communication, et de réduire la latence de communication. En revanche, plus le NoC a de liens, plus le coût en surface de silicium est élevé
- Un degré de clustérisations plus élevé indique la disponibilité des chemins de routage alternatifs afin de ré-router des paquets congestionnés à un nœud donné, ce qui permet de réduire la congestion de la mémoire tampon des nœuds intermédiaires. Un degré de clustérisations plus élevé favorise aussi la communication locale.

Notre algorithme de personnalisation du réseau d'interconnexion sur puce ne prend en compte donc que les propriétés physiques de la topologie, et il les améliore afin que le NoC supporte des applications NoC avec différents modèles de trafic de communication. Parmi les caractéristiques physiques de la topologie du NoC qui nous semblent les plus significatives, nous avons choisi de travailler sur la distance moyenne (notée :  $D_m$ ) et le degré de

clustérisations (noté :  $Cn$ ). Plus précisément, notre stratégie vise à maximiser le degré de clustérisations ( $Cn$ ) et à minimiser la distance moyenne ( $Dm$ ). Donc, l'objectif est de trouver un compromis d'amélioration de ces deux mesures pour un Noc initial donné par l'étape précédente.

### 3.1.1. Supposition :

L'unité de calcul de la distance des liens est calculée par le nombre de saut.

La sélection des liens à insérer se fait par sélectionné tous les liens non disponibles avec la vérification des contraintes suivante :

- La distance des liens sélectionné doit être inférieure ou égal à trois sauts
- Les liens ajouter a au-moins un routeur associé avec qui relie avec un lien vertical (les routeurs qui ont relié avec des TSV, ils ont plus de trafic).
- S'il s'agit du nombre de voisins (dégrée) maximal sur chaque nœud, qui est limité à 7 voisins

### 3.1.2. Formulation du problème :

Le problème de trouver un compromis d'amélioration de la distance moyenne et du degré de clustérisations ( $Max(Cn)$  &  $Min(Dm)$ ) peut être considéré comme un problème d'optimisation, Il peut être défini par la fonction de fitness suivante :

$$Max(\text{dégrée de clusterisation}), Min(\text{distance moyenne}) \text{ avec } \sum_{l \in Ls} s(l) \leq (1/2)S \quad (3.1)$$

Où  $Ls$  est la liste des liens qui peuvent être ajoutés au Noc de base,  $l$  est un lien sélectionné appartenant à cette liste de liens,  $s(l)$  est la distance du lien sélectionné  $l$  et  $S$  est le budget initial (la distance globale des liens d'une architecture donnée par l'application).

L'équation (3.1) vise à optimiser les performances physiques de la topologie dans un premier temps, indépendamment des applications qui devraient être exécutées sur le Soc basé sur ce Noc optimisé. Cette équation d'optimisation vise par ailleurs à optimiser les performances de communication de différentes applications qui peuvent être exécutées sur les Socs basées sur ce Noc personnalisé. En effet, la formule de la fonction de fitness (3.1) peut être convertie sous la forme :

$$Max(\text{débit}) \text{ avec } \sum_{l \in Ls} s(l) \leq (1/2)S \quad (3.2)$$

Sachant que le degré de clustérisations d'un nœud  $i$  est donnée par l'équation suivante :

$$Cn_i = \frac{2l_i}{n_i(n_i-1)} \quad (3.3)$$

Où  $l_i$  est le nombre de liens entre les voisins du nœud  $i$ , et  $n_i$  est le nombre de voisins du nœud  $i$ .

Ainsi, le degré de clustérisations moyen est obtenu par la formule suivante :

$$Cn = (\sum_{i=1}^N Cn_i) / N \quad (3.4)$$

Où  $N$  est le nombre total des nœuds dans le Noc.

Par ailleurs, la distance moyenne du Noc est peut-être calculée par la formule :

$$Dm = \left( \sum_{i=1}^N \sum_{j=1}^N d_{ij} \right) / (N(N - 1)) \quad (3.5) \text{Acti Accé}$$

Où  $d_{ij}$  est la distance entre le nœud  $i$  et le nœud  $j$ , et  $N$  est le nombre de nœuds dans le Noc. Plusieurs distances peuvent être considérées pour le calcul de la distance entre deux nœuds  $i$  et  $j$ , notamment le nombre de sauts entre eux ou bien la distance euclidienne.

Dans l'ensemble de notre travail, chaque topologie est modélisée par une matrice d'adjacence. La distance considérée est celle du nombre de sauts. Donc, la distance entre deux nœuds connectés est équivalente à un saut. Si deux nœuds ne sont pas directement connectés, leur distance est équivalente au nombre de sauts du plus court chemin entre eux.

### 3.2. Implémentation de l'algorithme d'insertion de liens :

A partir des deux fonctions d'optimisation (3.1) et (3.2), on a défini deux plans d'actions. En premier lieu, nous avons développé un algorithme itératif pour la recherche des liens stratégiques à insérer sur le Noc. Les liens sélectionnés doivent optimiser au maximum les propriétés physiques de la topologie du Noc, tel que diminuer la distance moyenne et augmenter le degré de clusterisation (équation 1). L'optimisation de la topologie du NoC peut avoir un impact direct sur les performances de QoS de la communication sur puce (débit, latence, énergie, etc.), tout en respectant la contrainte relative au budget en termes de ressources (liens disponibles).

En second lieu, nous avons procédé à évaluer les performances de communication (débit, à la latence, la charge des liens et à la consommation de l'énergie) du NoC optimisé. Plusieurs modèles de trafic (applications) peuvent être considérés pour cet effet, notamment les modèles Bit-Reversal, Transpose, Shuffle, et Uniform, Hot-Spot, etc. Le routage du flux de trafic d'un modèle donné est effectué par l'algorithme de routage "Source" associé à l'algorithme Dijkstra pour la sélection du plus court chemin entre la source et la destination.

L'algorithme de personnalisation de NoC présenté ici (Figure 3.4) permet de concevoir un réseau d'interconnexion sur puce qui peut servir comme une toile de communication pour le maximum d'applications du NoC tout en optimisant les performances de communication des applications SoC.

Pour choisir judicieusement les liens à insérer sur le Noc, l'algorithme commence à partir d'une configuration standard initiale  $C0$  du Noc utilisé dans l'étape précédente (l'architecture après le mapping) et les ressources disponibles  $S$  (ex.,  $S = 4$ ). Ensuite, l'algorithme passe par plusieurs étapes :

Etape 1 : lors de la première étape, la matrice d'adjacence ( $A$ ) de la topologie du NoC est calculée,  $A_{ij} = 1$  si et seulement s'il y a un lien entre les nœuds  $i$  et  $j$ ,  $A_{ij} = 0$  sinon.

Etape 2 : lors de la deuxième étape, une liste  $Ls$  de tous les liens qui pourraient être ajoutés à la configuration initiale  $C0$  est générée (le lien  $L_{ij}$  est ajouté à la liste  $Ls$ , si  $A_{ij} = 0$  et  $i \neq j$ ). La liste  $Ls$  est filtrée par la suite de façon à ne conserver que les liens  $L_{ij}$  qui ont une distance inférieure ou

égal à la moitié de la distance initiale et améliorent simultanément le degré de clusterisation ( $Cn$ ) et la distance moyenne ( $Dm$ ).

Etape 3 : lors de la troisième étape, les liens conservés sont testés un-par-un sur la configuration actuelle  $Ck$ , c'est ainsi que leur couple  $(Cn, Dm)$  est calculé. En suite la liste  $Ls$  sera trié par ordre décroissant, de tel sorte que les liens qui améliore le plus le couple  $(Cn, Dm)$  se trouvent en premier. Pour cela, nous avons défini une solution optimale de référence. Pour des performances physiques idéales du NoC, il faut que celui-ci soit complètement connecté. Un NoC complètement connecté offre une meilleure distance moyenne équivalente à "1" saut (un seul saut de n'importe quel nœud vers n'importe quel autre nœud), ainsi qu'un meilleur degré de clusterisation équivalent lui aussi à "1". Par conséquence, la solution optimale est définie par  $S_{opt} (Dm_{opt}, Cn_{opt}) = (1,1)$ . Les deux meilleures liens est celui qui offre une solution  $S_i (Dm_i, Cn_i)$  qu'est la plus proche à la solution optimale  $S_{opt} (Dm_{opt}, Cn_{opt})$ , (cf. figure 3.3). Ainsi, nous avons calculé la distance euclidienne de chaque solution  $(Dm_i, Cn_i)$  (qui correspond au lien  $i$ ,  $i$  allons de 1 à  $n$ , où  $n$  est le nombre total de liens dans  $Ls$ ) par rapport à la solution optimale  $(Dm_{opt}, Cn_{opt})$ . La distance euclidienne est peut-être obtenue comme suite :

$$DIS\ EUC(S_i, S_{opt}) = \sqrt{(Cn_i - Cn_{opt})^2 + (Dm_i - Dm_{opt})^2} \quad (3.7)$$

Le lien qui offre la plus faible distance euclidienne est celui qui optimise le mieux simultanément les deux paramètres  $Cn$  et  $Dm$ , à savoir,  $Min(Dm)$  et  $Max(Cn)$ . En cas où plusieurs liens ont la même distance euclidienne, l'un d'entre eux sera sélectionné aléatoirement.

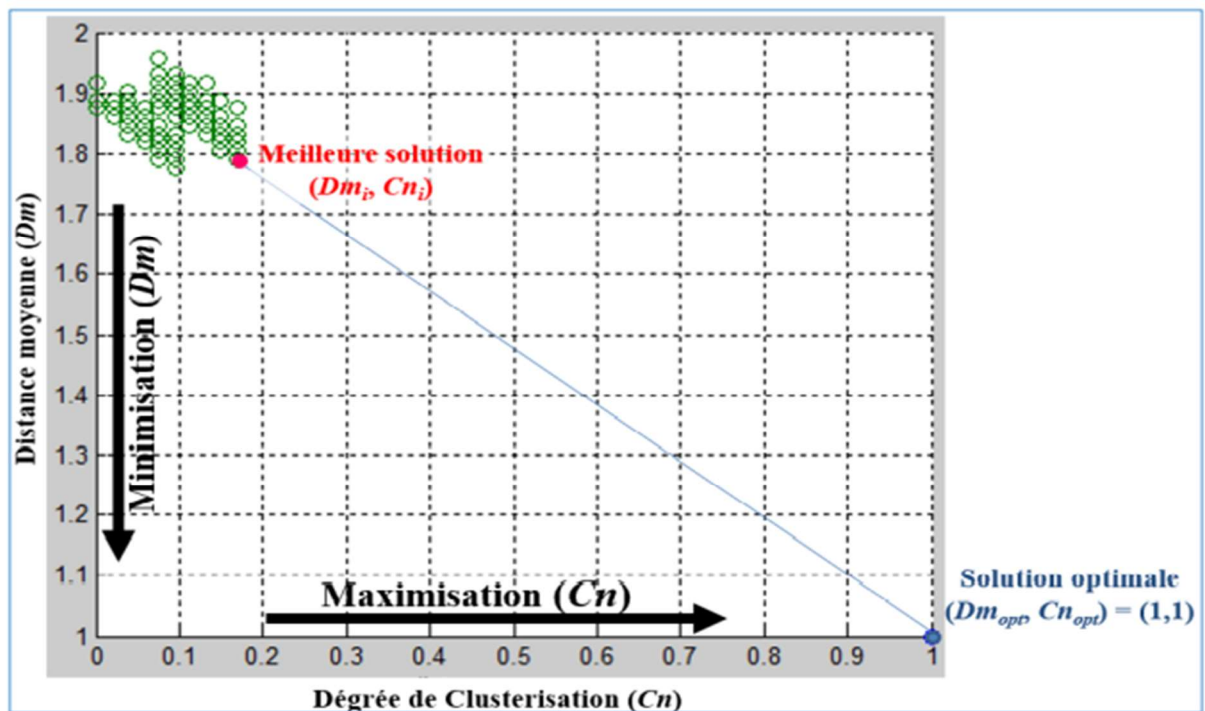


Figure 3.3 : Election de la combinaison gagnante [26]

Etape 4 : lors de cette étape, les deux premiers liens sont insérés définitivement sur la configuration actuelle  $C_k$  pour obtenir une nouvelle architecture.

Etape 5 : Tant que le budget n'est consommé, retour à l'Etape 3.

### 3.3. Schéma explicatif de l'algorithme d'insertion :

L'algorithme présenté ici est composé de deux parties principales (cf. figure 3.4). La première partie concerne la sélection et l'insertion des liens stratégiques sur le Noc de base.

Cet algorithme prend en entrée un Noc donné après les étapes précédente et tout type de modèle de trafic. Pour nos évaluations et nos simulations.

Le calcul des route-codes pour le routage est automatique, et les paramètres de simulation sont écrits directement sur les fichiers de configuration du simulateur NS-2.

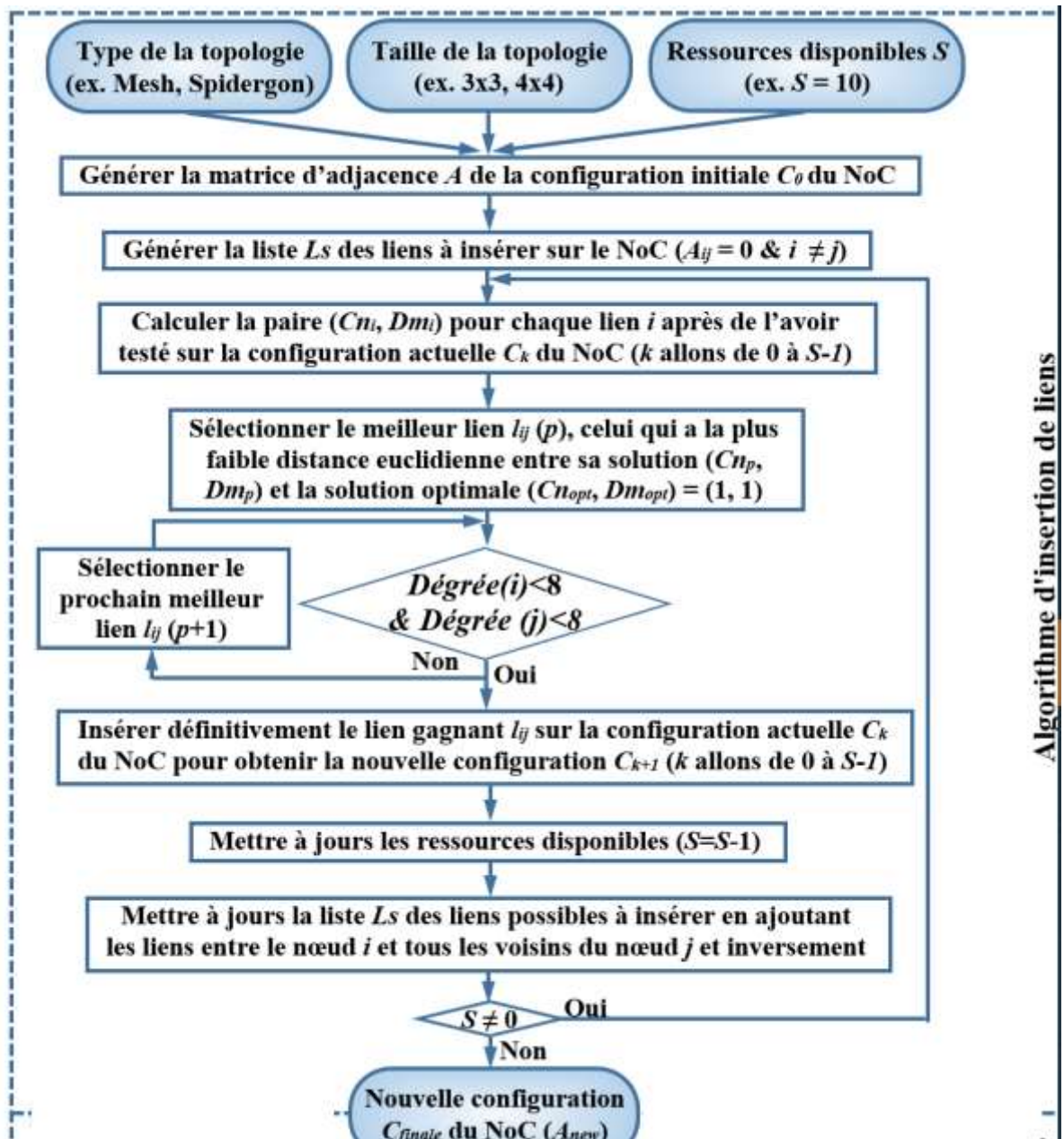


Figure 3.4 : Algorithme optimisé [26]

La figure 3.5 illustre l'exécution, étape par étape, de la première partie de l'algorithme de personnalisation sur un NoC de type 2D 3x3 Mesh, avec un budget  $S = 4$ .



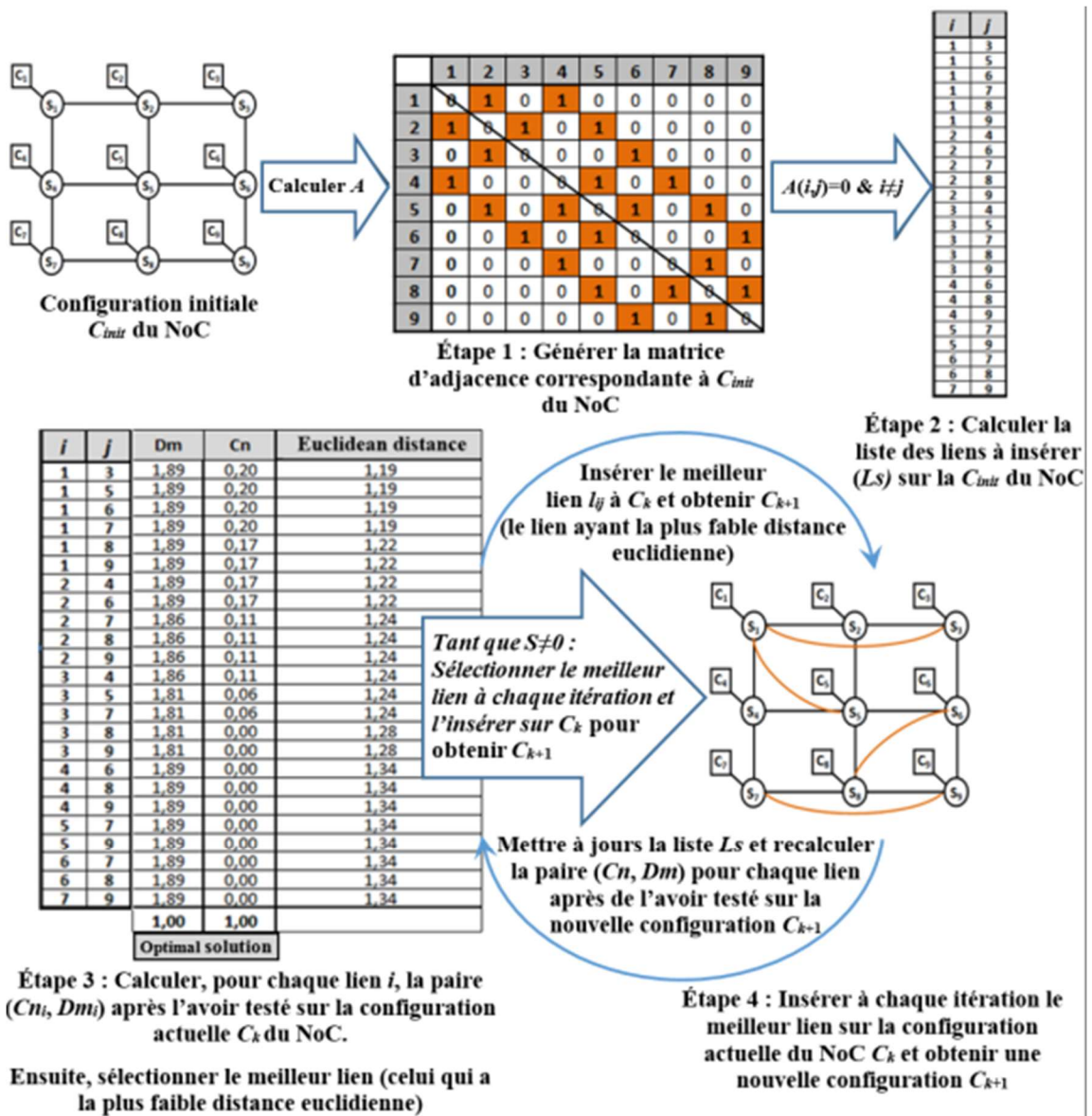


Figure 3.5 : Exemple de déroulement de l'algorithme de sélection de liens à insérer [26]



## 4.Conclusion :

Dans ce chapitre, nous avons présenté une approche qui permet l'optimisation de topologie en insérant des liens a pour objectif de réduire le débit de communications moyen.

Nous avons commencé par introduire notre problème, puis on a présenté notre supposition, on a décrit les algorithmes utilisés.

Nous utiliserons un simulateur NS-2 afin de tester la validation de notre proposition et nous réaliserons une étude comparative de notre étude.

# Chapitre 4

## Tests et résultats

### 1. Introduction :

Après avoir implémenté les différents algorithmes détaillés dans le chapitre précédent, nous présentons les différents résultats obtenus lors de l'expérimentation. Nous commençons par une présentation de flux de trafic correspond, puis nous allons citer les différents paramètres de simulateurs NS2, ensuite nous allons étudier les résultats basés sur la comparaison de différentes insertions (Stratégique, aléatoire, mauvaise insertion, sans insertion, insertion 100% des TSV, insertion 50% et 25% d'insertion).

Les tests sont réalisés sur un simulateur NS-2 à l'aide de deux scripts NSG2.1 et jTrcezar respectivement pour la création d'un fichier. TCI et la transformation de fichier.tr en résultats donné en débit, se simulateur est réalisé sur une machine équipée d'un processeur Intel Core i5 7Gen avec 2.5GHz et d'une RAM de 8 GO sous le système d'exploitation Ubuntu20 de 64 bits.

Avant de tester les résultats il faut d'abord calculer les deux mesures (degré de clusteration  $C_n$  et la distance moyenne  $D_m$ ) et le résultat de ces mesures en fonction de la distance euclidienne, pour cela on a créé un programme java dans Eclipse 20 qui calcule ces résultats.

### 2. benchmark VOPD (video object plane decoder) :

Dans notre expérimentation on a utilisé benchmark standards VOPD pour construire le flux de notre travail :

#### 2.1. Présentation de VOPD :

L'application VOPD est composée de 16 composants (IPs) qui communiquent entre eux à travers 21 liens.

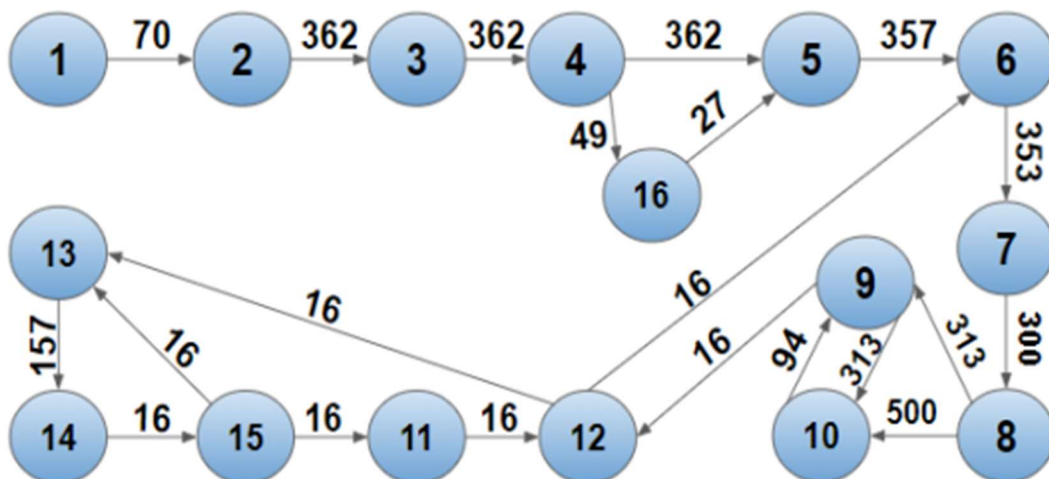


Figure 4.1 : Benchmark VOPD [92]

## 2.2 résultats de mapping de VOPD :

D'après les recherches réalisées par abbad salim et taieb selma en 2019/2020 sur le mapping on a pris l'un des résultats de mapping concernant le VOPD décrite en tableau ci-dessous :

Tableau 4.1 : mapping de l'application VOPD

	Plan1								Plan2							
Tuile	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IP	IP6	IP5	IP7	IP1	IP8	IP4	IP3	IP2	IP11	IP10	IP13	IP14	IP12	IP9	IP16	IP15
TSV	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

## 3. Le flux de trafic :

Nous considérons un Noc de type 3D Mesh 3x3x2 pour notre cas d'étude.

On choisit un flux qui correspond l'application benchmark VOPD :

La figure 4.5 illustre les flux de données transmis entre les cœurs participants à la communication :

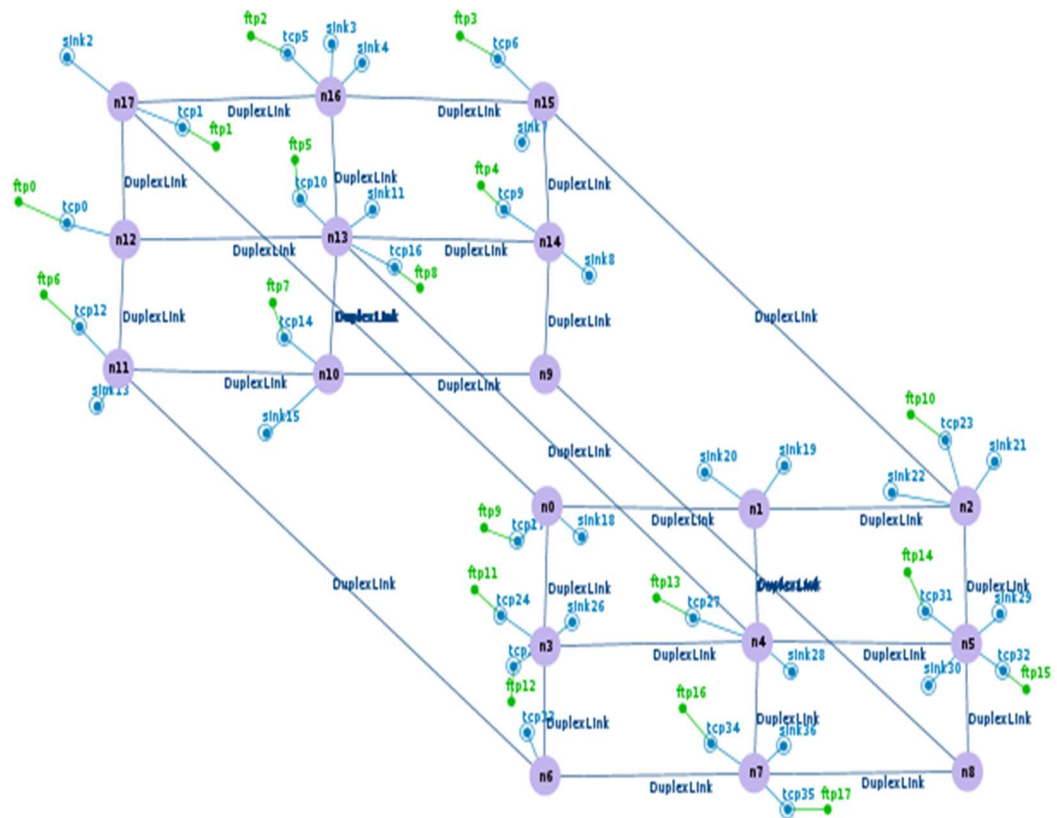


Figure 4.2 : Les flux de données utilisées, dans le NoC de type 3D 3x3x2 Mesh

L'application est représentée comme un processus parallèle de communication déjà mis en place entre les cœurs IP. Les cœurs IP sélectionnés à partir de l'application VOPD.

Dans le graph Les nœud tcp sink représente les cœurs de destination et les nœud tcp représente les cœurs source des IPs.

Comme le montre la figure 4.5, les flux de données sont représentés à l'aide du protocole ftp qui est représenté dans la figure pour chaque cœurs IP.

#### 4. Les liens candidats :

L'algorithme commence par une configuration NoC standard  $Ck=0$  (3D 3x3x2 Mesh), la liste  $L_s$  des liens qui pourraient être ajoutés au NoC est générée (48 liens sont possibles) par l'ajout de tous les liens inexistant dans la topologie initiale voire dans le tableau ci-dessous :

Tableau 4.2 Les liens candidats

La liste des liens	
0	2, 4, 5, 6, 7, 8
1	3, 5, 6, 7, 8
2	3, 4, 6, 7, 8
3	5, 7, 8
4	6, 8
5	6, 7
6	8
9	11, 13, 14, 15, 16, 17
10	12, 14, 15, 16, 17
11	12, 13, 15, 16, 17
12	14, 16, 17
13	15, 17
14	15, 16
15	17

## 5.Paramètres de simulation d'un réseau de communication :

Nous avons effectué des simulations en utilisant le simulateur NS-2. L'objectif principal est d'évaluer l'impact de l'amélioration des performances physiques du réseau Noc sur ces performances en QoS de communication (débit, latence, charge et consommation d'énergie).

Nous avons conduit ces simulations en utilisant différents paramètres, Les principaux paramètres configurables de simulation sont [93] :

### 1.La taille de la topologie :

Représente le nombre des nœuds dans une topologie aux fonctions de (i, j, k) tel que i = la dimension des colonne, j la dimension des ligne, k = le nombre des couches dans une topologie 3D.

## 2. Planificateur de paquets (Queue type) :

Également appelé planificateur de réseau ou discipline de file d'attente (qdisc) ou algorithme de file d'attente est un arbitre sur nœud dans un réseau de communication, il gère la séquence des paquets réseau dans les files d'attente de transmission et de réception de la pile de protocoles.

Il existe plusieurs types de planificateur de paquets :

### Tail Drop :

Est un algorithme de gestion de file d'attente simple utilisé par le planificateur de réseau dans l'équipement réseau pour décider quand supprimer les paquets. Avec la suppression de la queue, lorsque la file d'attente est remplie à sa capacité maximale, les paquets nouvellement arrivés sont abandonnés jusqu'à ce que la file d'attente ait suffisamment de place pour accepter le trafic entrant.

Le nom résulte de l'effet de la politique sur les paquets entrants. Une fois qu'une file d'attente a été remplie, le routeur commence à éliminer tous les datagrammes supplémentaires, abandonnant ainsi la queue de la séquence de paquets.

### RED :

Random Early Drop est un mécanisme de mise en file d'attente qui essaie d'éviter la congestion du réseau en contrôlant la taille moyenne de la file d'attente. La taille moyenne de la file d'attente est comparée à deux seuils : un seuil minimum ( $\min_{th}$ ) et maximum ( $\max_{th}$ ). Si la taille moyenne de la file d'attente ( $\text{moyenne}_q$ ) est inférieure au seuil minimum, aucun paquet n'est abandonné. Lorsque la taille moyenne de la file d'attente est supérieure au seuil maximal, tous les paquets entrants sont supprimés. Mais si la taille moyenne de la file d'attente se situe entre les seuils minimum et maximum, les paquets sont abandonnés au hasard avec une probabilité  $P_d$ .

### SFQ :

La mise en file d'attente d'équité stochastique (SFQ) est assurée par des algorithmes de hachage et de répétition alternée. Un flux de trafic peut être identifié de manière unique par 4 options (adresse src, adresse dst, port src et port dst), ces paramètres sont donc utilisés par l'algorithme de hachage SFQ pour classer les paquets dans l'un des 1024 sous-flux possibles. Ensuite, l'algorithme à tour de rôle commencera à distribuer la bande passante disponible à tous les sous-flux, à chaque tour donnant **sfq-allot** octets de trafic. L'ensemble de la file d'attente SFQ peut contenir 128 paquets et 1024 sous-flux sont disponibles.

## 3. Capacité des liens :

La capacité d'un lien, c'est la quantité d'information qu'il peut transporter. C'est fortement relié au débit d'une connexion réseau.

Une question importante est de savoir combien d'informations on peut faire circuler d'un nœud, **S**, vers un autre, **T**. C'est le problème du **S-T flot maximum**.

Chaque arc **(u,v)** a une capacité qui limite le flot qui circule sur cet arc entre **u** et **v**.

Dans chaque nœud en dehors de S et T, aucune information n'est perdue, et aucune n'est créée. Ainsi, tout ce qui rentre, ressort.

Une liaison 100Mbps est capable de supporter un débit théorique de 12.5Mo/s. J'imagine que les auteurs de ces lignes ont naïvement calculé que l'octet étant traditionnellement valorisé à 8 bits, il suffisait de faire une petite division pour obtenir le débit en Mo/s.

#### 4. délai de propagation :

Le délai de propagation, ou délai, est une mesure du temps nécessaire à un signal pour se propager d'une extrémité du circuit à l'autre. Le délai de propagation du réseau est mesuré en nanosecondes (nS). Le délai de propagation typique pour l'UTP de catégorie 5e est un peu inférieur à 5 nS par mètre (le pire des cas autorisé est de 5,7 nS/m). Un câble de 100 mètres peut avoir un retard comme indiqué ci-dessous.

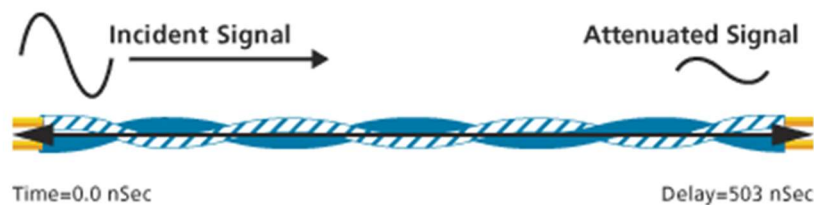


Figure 4.3 : Délai de propagation

Le délai de propagation est la principale raison d'une limitation de longueur dans le câblage LAN. Dans de nombreuses applications de mise en réseau, telles que celles utilisant CSMA/CD, il existe un délai de propagation maximal qui peut être pris en charge sans perdre le contrôle des communications.

#### 5.Queue size :

La longueur maximale d'une file d'attente peut être limitée à un nombre défini de messages, ou à un nombre défini d'octets (le total de toutes les longueurs de corps de message, en ignorant les propriétés du message et les frais généraux), ou les deux.

#### 6.canal de communication :

Un canal de communication ou canal de transmission est un support (physique ou non) permettant la transmission d'une certaine quantité d'information, depuis une source (ou émetteur) vers un destinataire (ou récepteur). Souvent, le canal altère l'information transmise, par exemple en ajoutant un bruit aléatoire. Il existe de type de canal de communication full-duplex et half-duplex.

Un canal de communication full-duplex est un canal de communication qui transporte l'information dans les deux sens (bidirectionnel). Selon que l'information peut être transportée simultanément dans les deux sens ou non.

Un canal de communication half-duplex également appelé à l'alternat). Un canal qui transporte l'information dans un seul sens est appelé simple (monodirectionnel).

7. Protocole de communication :

Il existe 2 type de protocole de communication les plus utilisé dans le réseau :

- TCP (Transport Control Protocol) assure un service de transmission de données fiable avec une détection et une correction d'erreurs de bout en bout.
- UDP (User Datagram Protocol) offre un service de transmission de datagrammes sans connection.

8. Type d'application pour la communication :

Il existe de 2 protocole utilisé pour le transfert des fichier d'une communication réseaux :

- File Transfer Protocol (protocole de transfert des fichiers), ou FTP, est un protocole de communication destiné au partage des fichiers sur un réseau TCP/IP.
- CBR (Constant Bit Rate) Débit binaire constant : Le respect d'un contrat de trafic CBR signifie que le client émet ses cellules à un rythme inférieur ou égal à PCR, tout en respectant la tolérance de gigue. Le trafic non conforme est écrêté. Le service CBR donne une bande passante constante.

Les paramètres de simulation choisis sont dans le tableau ci-dessous :

Tableau 4.3 : Paramètre de simulation de l'algorithme

Topologie	Mesh 3D
La taille de la topologie	3x3x2
Queue type	Drop tail
Capacité des liens	100 MBps
Propagation delay	10 ms
Queue size	50
Canal de communication	Bidirectionnel (duplex)
Protocole Communication	TCP
Le flux de trafic	TCP source , TCP sink



Packet size	1500 bytes
Type d'applicaton	FTP
Paramètre de simulations	Temps de simulations :50ms Trace file : out.tr Name file :out.nam

## 6.L'évaluation analytique de l'algorithme d'insertion de lien :

Pour montrer l'intérêt de notre algorithme, nous l'avons évalué analytiquement. Pour ce faire, nous avons pris, d'une part, des liens sélectionnés par notre algorithme, et d'autre part des liens aléatoires à insérés, avec bien sur le même nombre total de liens ajoutés pour chaque application donnée.

Les résultats obtenus sont illustrés jusqu'à un budget de  $S = 5$  liens supplémentaires dans une architecture 2D  $3 \times 3$  :

Tableau 4.4 : Evaluation analytique de l'algorithme d'insertion des liens horizontaux

Architecture personnalisé par l'ajout des liens sélectionnés par notre algorithme					Architecture personnalisé par l'ajout des liens sélectionnés aléatoirement					Liens insérés		
Degrée de cluster (Cn)	Distance moyenne (Dm)	Distance euclidienne	lien		Pourcentage de Eclidie	Degrée de cluster (Cn)	Distance moyenne (Dm)	Distance euclidie	lien			
			i	j					i	j		
0.00	1.56	1.14			0.00	0.00	1.56	1.14			0.00	S=0
0.20	1.44	0.91	1	3	20.2	0.17	1.44	0.94	4	6	17.5	S=1
0.41	1.36	0.69	5	7	20.4	0.33	1.39	0.78	0	4	17	S=2
0.55	1.29	0.54	1	5	21.7	0.48	1.33	0.62	4	8	20.5	S=3
0.70	1.24	0.39	3	7	27.8	0.52	1.32	0.58	6	8	6.5	S=4
0.75	1.21	0.32	1	7	18	0.54	1.26	0.53	5	7	8.6	S=5

Architecture personnalisée par l'ajout des liens les plus mauvais sélectionnés						
Degré de cluster (Cn)	Distance moyenne (Dm)	Distance euclidienne	lien		Pourcentage	Liens insérés
			i	j		
0.00	1.56	1.14			0.00	S=0
0.00	1.5	1.12	0	5	1.1	S=1
0.00	1.47	1.11	0	7	3.3	S=2
0.00	1.44	1.09	1	8	5.5	S=3
0.00	1.42	1.08	3	8	9	S=4
0.00	1.31	1.05	1	6	12.6	S=5

D'après le tableau en remarque que le taux d'amélioration de la valeur optimale (distance euclidienne) de l'approche proposée est 41% meilleur par rapport à des insertions aléatoires et de 51% par rapport à des insertions les plus mauvaises après l'ajout de 10 liens.

Les figures 4.4 et 4.5 illustrent l'évolution du degré de clusterisation et de la distance moyenne, respectivement, en fonction des liens ajoutés (un 2D 3x3 Mesh ( $N = 9$  nœuds) a initialement 24 liens, Nous pouvons constater aussi que le NoC idéal est obtenu lorsque le couple  $(Dm, Cn)$  atteint la valeur  $(1, 1)$ , c'est à dire, tous les nœuds sont entièrement connectés.

### NoC 2D 3x3 Mesh personnalisé par l'ajout des liens

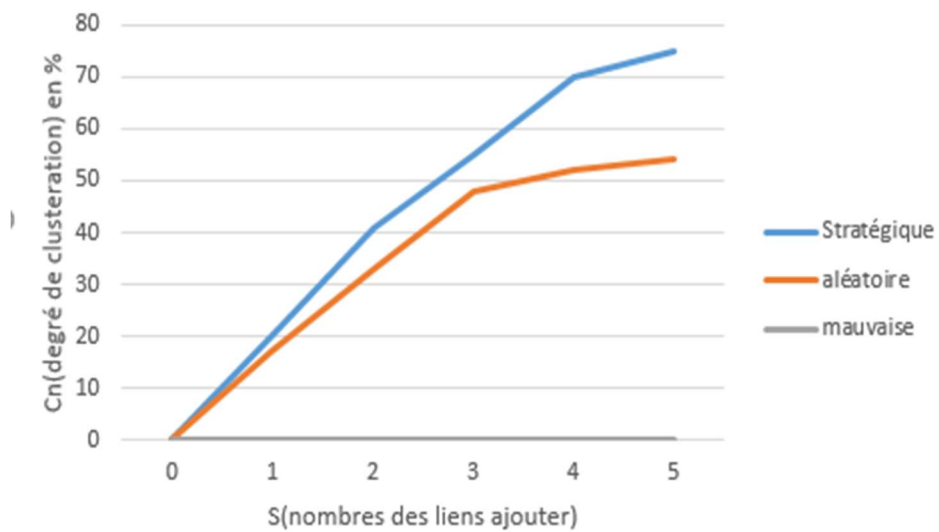


Figure 4.4 : Le degré de clusterisation du NoC de type 2D 3x3 Mesh, lorsque S varie de 0 à 5

D'après La figure 4.4 on remarque que liens stratégique est En continuelle augmentation et évolue mieux en fonction du degré de clusteration Cn par rapport aux deux autres insertions (aléatoire, mauvaise) , C'est ce qui donne un point de vue globale sur l'impact de l'approche proposé.

Sur cette figure, le NoC 2D 3x3 Mesh a initialement 24 liens de base, après l'ajout de dix liens, soit 34 liens ce qui représente 60% du budget globale (56 liens).

## NoC 2D 3x3 Mesh personnalisé par l'ajout de liens

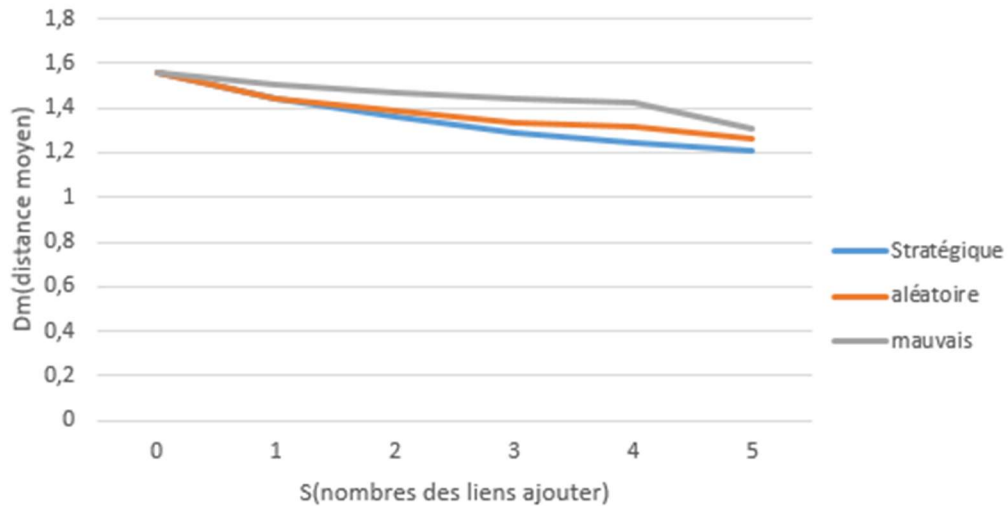


Figure 4.5 : La distance moyenne du NoC de type 2D 3x3 Mesh, lorsque S varie de 0 à 5

La figure 4.5 : la distance moyenne des insertions stratégique et aléatoire ne montre aucune différence selon le graphe et sont meilleurs que l'insertion les plus mauvaise.

La figure 4.6 montre la différence entre les solutions optimales des différentes topologies (par insertion stratégique et par insertion aléatoire et insertions les plus mauvaises) qui représente la distance euclidienne de deux mesures (la distance moyenne et le degré de clusteration) en fonction de chaque insertion.

La meilleure solution optimale est celle qui a une distance euclidienne minimale, et d'après la figure 4.6 elle confirme l'efficacité de l'approche proposée.

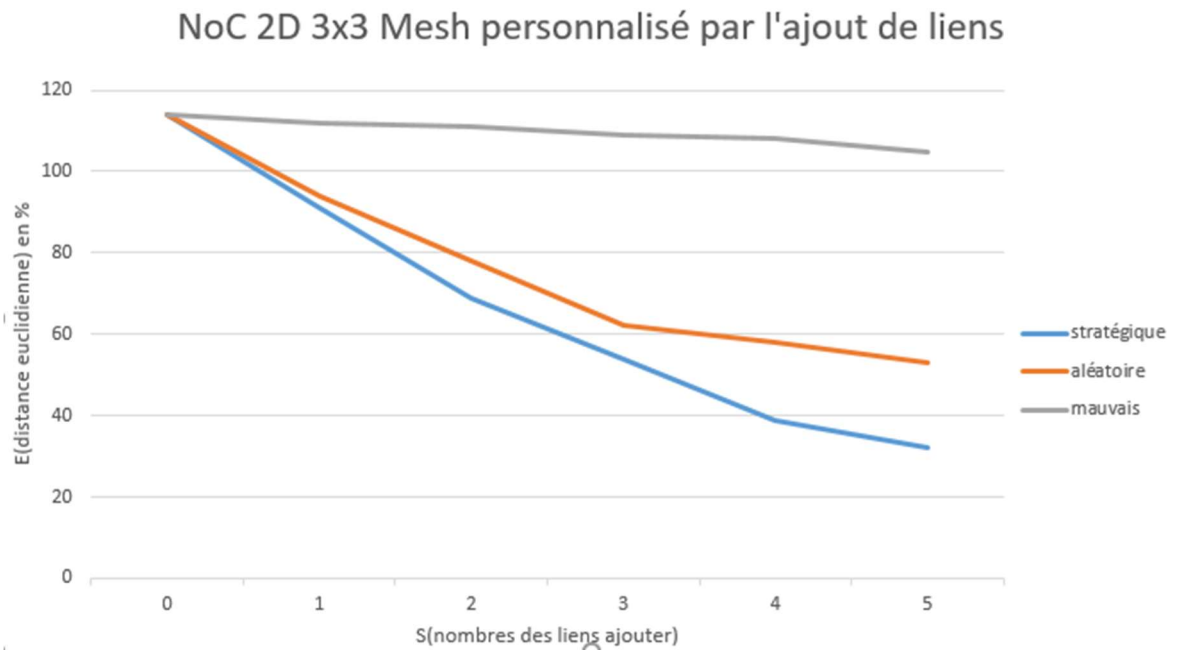


Figure 4.6 : La distance euclidienne du NoC de type 2D 3x3 Mesh, lorsque S varie de 0 à 5

Ces résultats montrent que l'amélioration des performances physiques du NoC de type 2D 3x3Mesh est très grande dans le cas de notre algorithme d'insertion de lien que dans le cas où les liens sont insérés aléatoirement, ce qui montre l'efficacité de notre algorithme qui cible les liens les plus bénéfiques à la topologie du NoC.

## 7. Résultats de simulation :

Les résultats de simulation correspondant aux paramètres de performance le débit, sont illustrés respectivement par les figures 4.7, 4.8 Sur chaque figure correspondante à un paramètre de performance.

Nous avons effectué une comparaison entre l'insertion de cinq liens par couche stratégiquement et l'insertion de cinq liens par couche aléatoirement avec les mêmes mesures choisies.

Les résultats de simulations sont exprimés en deux graph qui corresponde à la comparaison.

Le débit :

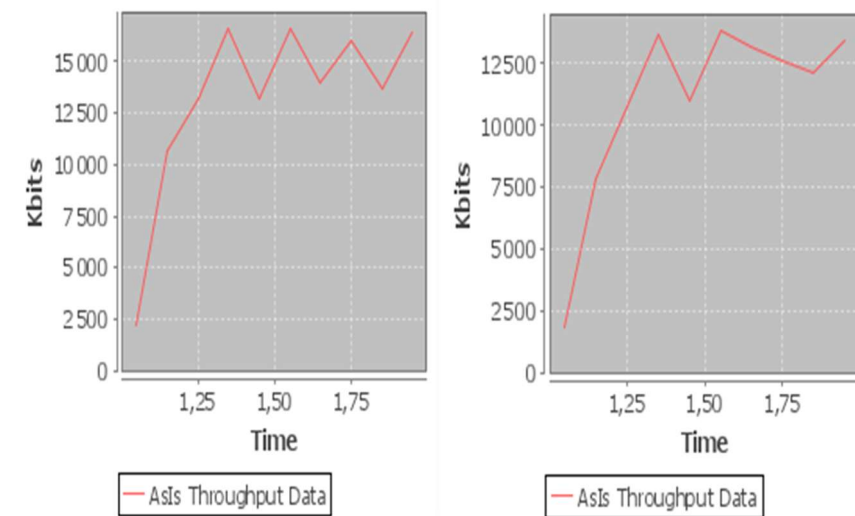
Le débit pour chaque nœud représente le nombre de bits arrivé à ce nœud (Kbps). Le débit total  $T(t)$  est calculé en faisant la moyenne du nombre de flits reçus avec succès par leur destination par seconde pendant le temps de simulation.

6.1 : comparaison entre le choix des insertions de lien à insérer dans la topologie :

Premièrement on a fait une comparaison entre 4 différentes topologies avec le même nombre des TSVs(on a pris 5 liens de TSV pour chaque topologie) :

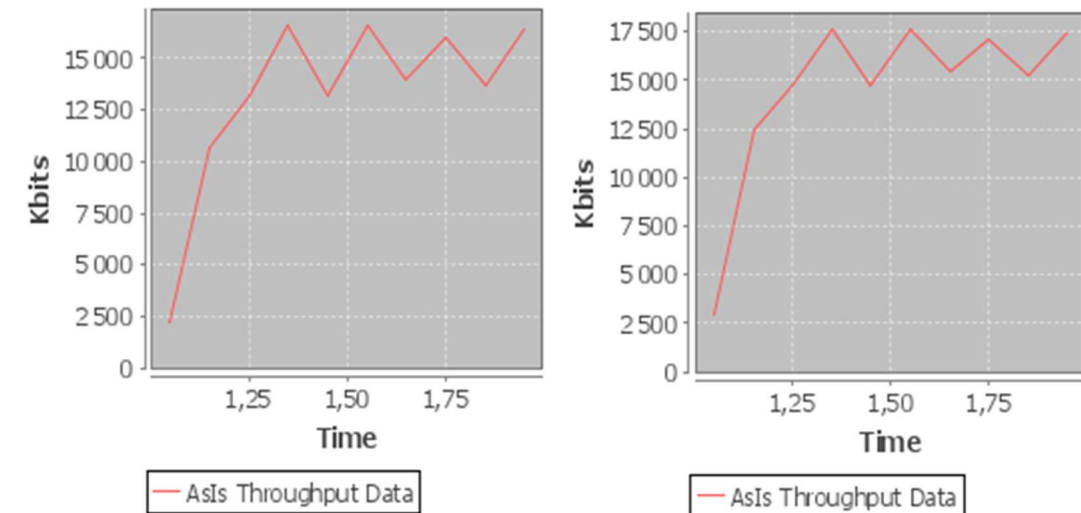
- Topologie avec des insertions stratégiques
- Topologie avec des insertions aléatoires
- Topologie par des insertions des plus mauvais liens
- Topologie sans insertions

La figure 4.7 illustre la variation du débit sous différents débits d'injection (flit-intervalle égal à 0.1 cycles d'horloge) de différentes insertions à partir de topologie 3D.



a. Par mauvaise insertions

b. sans insertion



c. insertions aléatoire

d. insertions Stratégique

Figure 4.7 : Comparaison de débit de différentes insertions de lien horizontale

Le débit augmente proportionnellement en fonction du Temps dans la figure 4.7.

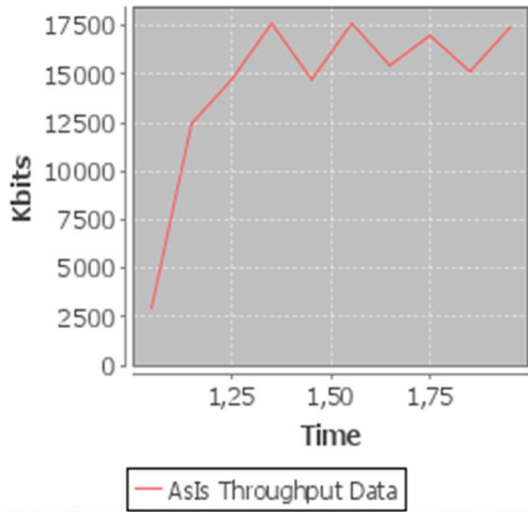
D'après la figure 4.7 on remarque que dans l'insertion stratégique le débit est le plus élevé dans l'intervalle [1.25, 1.9] par rapport aux autres insertions qui atteignent jusqu'à 17500 kbit/s.

L'insertion aléatoire et les mauvaises insertions ont presque le même débit et le débit prend des valeurs les plus basses dans la topologie sans insertion.

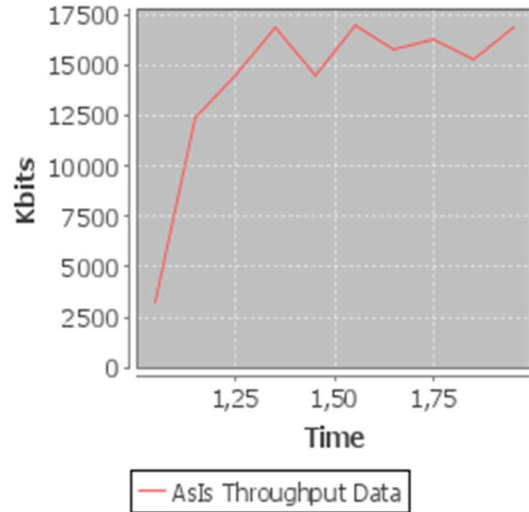
Nous remarquons clairement que le débit de l'insertion des liens stratégiques est plus grand que l'insertion aléatoire et les plus mauvaises insertions et sans insertion de 10 liens horizontaux sous différents débits d'injection de 1.2 à 1.9.

6.2 : comparaison entre les différents nombres d'insertions des TSVs dans la topologie d'insertion stratégique :

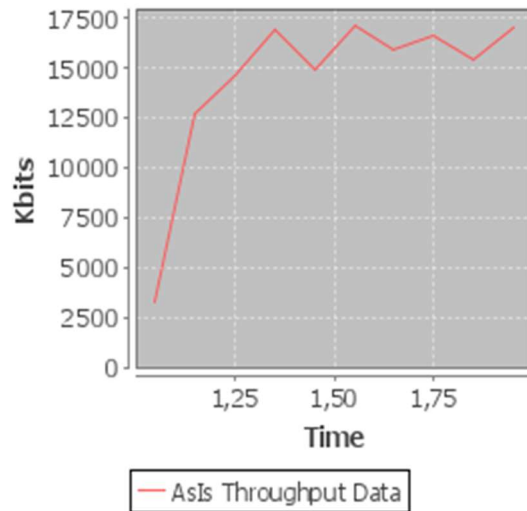
On a comparé dans la 2<sup>ème</sup> étape entre l'insertion 100%, 50% et 25% des TSVs qui représentent respectivement (9, 4, et 2) dans une topologie 3D mesh 3x3x2 avec les liens stratégiques quand on a pris précédemment, les résultats sont dans la figure 4.8 suivante :



a. 100% d'insertions



b. 50% d'insertions



c. 25% d'insertions

Figure 4.8 : Comparaison de débit de différente insertions des TSVs

Dans la figure 4.8 on remarque que les trois graphes ont presque le même débit.

Ce qui fait que l'ajout des liens vertical n'a aucun effet sur le débit lors de l'utilisation d'un algorithme stratégique.



## Discussion :

D'après les graphes précédents on peut remarquer que l'algorithme proposé donne des résultats optimaux en terme de débit, qui pourrait compenser la perte causée par la suppression des liens verticaux (TSVs) pour l'optimisations de la surface des puces.

Malheureusement on ne peut pas extraire les résultats en terme d'énergie de consommations et en surface à cause de 2 chose

1. Le simulateur choisit (NS2) ne permet pas l'extraction des résultats de ces mesures.
2. Par manque de temps on ne peut pas modifier le script écrit en TCL de NS2.

Et aussi le simulateur (NS2) ne fait pas la différence entre les liens verticaux et les horizontaux, ce qui ne permet pas de donner des résultats efficaces sur la technologie 3D.

## 8.Conclusion

Dans ce chapitre, nous avons commencé par présenter le flux de trafic de l'application VOPD puis on a évalué analytiquement notre algorithme par les caractéristiques de la topologie, en suite nous avons exploité les résultats de simulation de la comparaison de différentes topologies basé sur l'insertion stratégique proposé par notre algorithme et l'insertion aléatoire et le pire cas des insertions de lien, en suite on a fait une comparaison des topologies pour l'insertions des TSVs(100%, 50%, 25%).

Notre solution prouve son intérêt avec ces valeurs en donnant des bonnes solutions en terme de débit, et qui peut remplacer la perte de débit causée par la suppression des liens verticaux (TSVs).

## Conclusion générale :

Aujourd'hui, les systèmes embarqués sur puce SoCs sont devenus de plus en plus complexes grâce à l'évolution de la technologie des circuits intégrés. Les performances de ces SoCs dépendent fortement de l'architecture d'interconnexion sur puce et du protocole de communication entre les cœurs IPs. Avec la technologie d'intégration croissante, la conception d'un SoC efficace est critique pour exploiter pleinement le nombre et la puissance de traitement des cœurs IPs dans un même circuit. La quantité de données échangées dans les systèmes est ainsi augmentée. Dans ce contexte, le paradigme NoC est devenu la solution privilégiée pour la communication dans les SoCs complexes. Cependant, faute de méthodologies et d'outils d'expérimentation et d'aide à la conception adaptés, l'évaluation de performance de ces réseaux sur puce constitue un défi majeur pour l'industrialisation de ces systèmes. Des études récentes ont montré que pour améliorer les performances du NoC, l'architecture de celui-ci pouvait être personnalisée, soit au moment de la conception, soit au moment de l'exécution.

L'objectif principal de cette thèse était donc d'implémenter de nouvelles approches pour améliorer les performances des NoCs, notamment la latence, le débit, la consommation d'énergie, la surface silicium et la simplicité de mise en œuvre en minimisant le nombre de lien vertical(TSV) utilisé.

Dans les expérimentations réalisées, nous avons pu observer l'efficacité de notre méthode en comparant ses résultats avec différentes insertions, Néanmoins, la qualité des solutions trouvées par notre solution dépend du paramètre de simulation et du nombre de liens à insérer.

Un des obstacles remarqué pendant notre recherche est que le simulateur NS2 n'effectue pas des résultats en énergie de consommation et en surface.

C'est dans nos perspectives, que nous voulons approfondir plus dans ce domaine et améliorer notre contribution, en terme de :

- Valider notre contribution par la communauté scientifique.
- Réaliser une interface simple utilisateur, en automatisant la phase d'étude de paramètres.
- Procéder au reste des phases de conception des réseaux sur puce 3D, tel que le routage et le mapping et floorplanning.
- Améliorer notre solution algorithmique en cherchant d'autre approche de personnalisations (statiques, dynamique).

## Bibliographie

- [1] SWAPNA S , EFFICIENT ROUTER DESIGN FOR NETWORK ON CHIP, National Institute Of Technology Rourkela , , 2013 .
- [2] KanishkaLahiri, SujitDey and AnandRaghunathan.Evaluation of the traffic–performance characteristics of system–on–chip communication architectures.In Proceedings of 14th International Conference on VLSI Design, pp. 29–35, 2001.
- [3] M. Nakajima et al. A 400MHz 32b embedded microprocessor core AM34– 1 with 4.0GB/s cross-bar switch for SoC.In Proceedings of International Solid State Circuits Conference, 2002.
- [4] SudeepPasricha, Mohamed Ben–Romdhane, and Nikil D. Dutt, BMSYN: Bus matrix communication architecture synthesis for MPSoC. IEEE Transactions on Computer–Aided Design of Integrated Circuits and Systems, 26 (8), pp. 1454–1464, 2007.
- [5] William J. Dally and Brian Towles. Route packets, not wires: On–chip interconnection networks. In Proceedings of the Design Automation Conference (DAC.01), pp. 681–689, 2001.
- [6] A. Hemani, A. Jantch, S. Kumar, A. Postula, J. Öberg, M. Millberg, and D. Lindqvist. Network on a Chip: An architecture for billion transistor era. In Proceedings of the IEEE NorChip Conference, 2000.
- [7] Keating, Michael, et al. "Standard Low Power Methods." *Low Power Methodology Manual: For System-on-Chip Design* (2007): 13-19.
- [8] Messaoudi, Kamel. *Traitement des signaux et images en temps réel:" implantation de H. 264 sur MPSoC"*. Diss. Dijon, 2012.
- [9] Wolf, Wayne, Ahmed Amine Jerraya, and Grant Martin. "Multiprocessor system-on-chip (MPSoC) technology." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27.10 (2008): 1701-1713.
- [10] D. GaoMing, Z. DuoLi, Y. YongSheng, M. Liang, G. LuoFeng, S. YuKung et G. MingLun, «FPGA prototype design of Network on Chips,» chez Anti-counterfeiting, Security and Identification, 2nd International Conference on, 2008.
- [11] Jovanovic, Slavisa. *Architecture reconfigurable de système embarqué auto-organisé*. Diss. Université Henri Poincaré-Nancy 1, 2009.
- [12] S. Suboh, M. Bakhouya, J. Gaber et E.-G. El-Ghazawi, «An interconnection architecture for network-on-chip systems,» Telecommunication Systems, vol. 37, n° 11-3, p. 137– 144, 2008.

- [13] Abdelmalek, B. O. U. G. U. E. T. T. A. Y. A. *Génération d'un réseau sur puce au format VHDL RTL à partir d'un modélisation de haut niveau UML par raffinement*. Diss. Université Badji Mokhtar de Annaba, 2017.
- [14] X. TRAN, «Méthode de Test et Conception en Vue du Test pour les Réseaux sur Puce Asynchrones : Application au Réseau ANOC,» Institut polytechnique de Grenoble, laboratoire CEA-LETI, France, 2008.
- [15] TAKKI CHEBIHI, M. *Protocoles de rééducation en traumatologie orthopédie* (Doctoral dissertation), 2009.
- [16] R. Holsmark et M. Högberg, «Modeling and Prototyping of Network on Chip,» Sweden, 2002.
- [17] Wang, K., Gu, H. and Wang, C., "Study on Hybrid Switching Mechanism in Network on Chip". In proceedings of the 7th International Conference on ASIC, pp. 914-917 2007.
- [18] K. Lahiri, A. Raghunathan et S. Dey, «Evaluation of the traffic-performance characteristics of system-on-chip communication architectures,» chez VLSI Design, 2001. Fourteenth International Conference on, Bangalore, 2001.
- [19] A. Varghese, B. Edwards, G. Mitra, A. P. Rendell, "Programming the Adapteva Epiphany 64-core Network-on-chip Coprocessor", in Proceedings of the 28th International Parallel & Distributed Processing Symposium Workshops, pp.984-992, Phoenix, USA, May 19-23, 2014.
- [20] Delorme, Julien. *Méthodologie de modélisation et d'exploration d'architecture de réseaux sur puce appliquée aux télécommunications*. Diss. INSA de Rennes, 2007.
- [21] H. Moussa, «Architectures des réseaux sur puce pour décodeurs canal multiprocesseurs,» l'école nationale supérieure des télécommunications de Bretagne, Bretagne, 2009.
- [22] Tatas, Konstantinos, et al. *Designing 2D and 3D network-on-chip architectures*. No. IKEEBOOK-2017-046. New York: Springer, 2014.
- [23] Duato, Jose, Sudhakar Yalamanchili, and Lionel Ni. *Interconnection networks*. Morgan Kaufmann, 2003.
- [24] Umamaheswari, S., et al. "Comparing the performance parameters of network on chip with regular and irregular topologies." *Trends in Network and Communications* (2011): 177-186.
- [25] P. Pande, C. Grecu, M. Jones, A. Ivanov et R. Saleh, «Performance evaluation and design tradeoffs for network-on-chip interconnect architectures,» Computers, IEEE Transactions on, vol. 54, n° 18, p. 1025–1040, 2005.
- [26] Chariete, Abderrahim. *Approches d'optimisation et de personnalisation des réseaux sur puce (NoC: Networks on Chip)*. Diss. Université de Technologie de Belfort-Montbéliard (UTBM), 2014.
- [27] S. Murali, D. Atienza, P. Meloni, S. Carta, L. Benini, G. De Micheli et L. Raffo, «Synthesis of Predictable Networks-on-Chip-Based Interconnect Architectures for Chip Multiprocessors,» chez Very Large Scale Integration (VLSI) Systems, 2007.
- [28] Feero, Brett Stanley, and Partha Pratim Pande. "Networks-on-chip in a three-dimensional environment: A performance evaluation." *IEEE Transactions on computers* 58.1 (2008): 32-45.
- [29] Duato, Jose, Sudhakar Yalamanchili, and Lionel Ni. *Interconnection networks*. Morgan Kaufmann, 2003.

- [30] Lau, John H. "TSV manufacturing yield and hidden costs for 3D IC integration." *2010 Proceedings 60th electronic components and technology conference (ECTC)*. IEEE, 2010.
- [31] Lu, Zhonghai. *Using wormhole switching for networks on chip: Feasibility analysis and microarchitecture adaptation*. Diss. KTH, 2005.
- [32] Chatmen, Mohamed Fehmi. *Conception d'un réseau sur puce optimisé en latence*. Diss. Université de Bretagne Sud, 2016.
- [33] Ali, Muhammad, Michael Welzl, and Sybille Hellebrand. "A dynamic routing mechanism for network on chip." *2005 NORCHIP*. IEEE, 2005.
- [34] A. Mello, L. Tedesco, N. Calazans et F. Moraes, «Virtual Channels in Networks on Chip: Implementation and Evaluation on Hermes NoC,» chez in Proc. the Symposium on Integrated Circuits and Systems Design, 2005.
- [35] A. Patooghy et H. Sarbazi-Azad, «Performance comparison of partially adaptive routing algorithms,» chez Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on, 2006.
- [36] A. Mostéfaoui, «Architectures Flexibles pour la Validation et l'Exploration de Réseaux Sur Puce,» Institut National Polytechnique De Grenoble, 2009.
- [37] J. Flich, P. Lopez, M. Malumbers et J. Duato, «Improving the Performance of Regular Networks with Source Routing,» chez Proceeding of the IEEE International Conference
- [38] Achballah, Ahmed Ben, and Slim Ben Saoud. "A survey of network-on-chip tools." *arXiv preprint arXiv:1312.2976* (2013).
- [39] Lemaire, Romain, et al. "Performance evaluation of a NoC-based design for MC-CDMA telecommunications using NS-2." *16th IEEE International Workshop on Rapid System Prototyping (RSP'05)*. IEEE, 2005.
- [40] Pullini, Antonio, et al. "NoC design and implementation in 65nm technology." *First International Symposium on Networks-on-Chip (NOCS'07)*. IEEE, 2007.
- [41] Seiculescu, Ciprian, et al. "SunFloor 3D: A tool for networks on chip topology synthesis for 3-D systems on chips." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 29.12 (2010): 1987-2000.
- [42] Wang, Hang-Sheng, et al. "Orion: A power-performance simulator for interconnection networks." *35th Annual IEEE/ACM International Symposium on Microarchitecture, 2002. (MICRO-35). Proceedings..* IEEE, 2002.
- [43] Lis, Mieszko, et al. "DARSIM: a parallel cycle-level NoC simulator." *MoBS 2010-Sixth Annual Workshop on Modeling, Benchmarking and Simulation*. 2010.
- [44] A. Varghese, B. Edwards, G. Mitra, A. P. Rendell, "Programming the Adapteva Epiphany 64-core Network-on-chip Coprocessor", in Proceedings of the 28th International Parallel & Distributed Processing Symposium Workshops, pp.984-992, Phoenix, USA, May 19-23, 2014.
- [45] 'NetSpeed ORION: A New Approach to Design On – chip Interconnects', 26-Aug-2013.
- [46] Arteris, <http://www.arteris.com/>
- [47] STMicroelectronics. <http://www.st.com>

- [48] M. Gschwind, H. Hofstee, B. Flachs, M. Hopkin, Y. Watanabe, and T. Yamazaki, "Synergistic processing in cell's multicore architecture," *Micro*, IEEE, vol. 26, no. 2, pp. 10–24, 2006.
- [49] sonics ,<http://sonicsinc.com/resources/product-briefs/>
- [50] Boufar, Nassima, Assia Taghribet, and Karima Sid. "L'utilisation des systèmes auto organisés pour analyser les images médicales." (2016).
- [51] Li, Xinyu. *Méthodologie de conception automatique pour multiprocesseur sur puce hétérogène*. Diss. Université Paris Sud-Paris XI, 2009.
- [52] K. Park et W. Willinger, *Self-similar network traffic and performance evaluation*, WileyBlackwell, Éd., John Wiley and Sons (ISBN: 0471319740), 2000, p. 576.
- [53] W. Leland, M. Taqqu, W. Willinger et D. Wilson, «On the self-similar nature of Ethernet traffic (extended edition),» *IEEE/ACM Transactions on Networking*, vol. 2, n° 11, pp. 1-15, 1994.
- [54] K. Srinivasan, K. Chatha et K. Isis, «ISIS: A genetic algorithm based technique for custom on-chip interconnection network synthesis,» chez *Proceedings of the 18th International Conference on VLSI Design, 4th International Conference on Embedded Systems Design (VLSID'05)*, 2005.
- [55] K. Srinivasan, K. Chatha et G. Konjevod, «Linear programming based techniques for synthesis of network on chip architectures,» *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, n° 14, p. 407–420, 2006.
- [56] Srinivasan, Krishnan, and Karam S. Chatha. "ISIS: a genetic algorithm based technique for custom on-chip interconnection network synthesis." *18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*. IEEE, 2005.
- [57] Z. Guz, I. HarWalter, E. Bolotin, I. Cidon, R. Ginosar et A. Kolodny, «Network delays and link capacities in application-specific wormhole NoCs,» *VLSI Design*, vol. 2007, p. 15, 2007.
- [58] Chath, k. s. et al.(2008). automated techniques for synthesis of application-specific network-on-chip architecture. *IEEE Transactions on computer-aided design of Integrated circuits and systems*, Vol. 27, N.8.
- [59] Chariete, A., et al. "Une approche de personnalisation des architectures pour réseaux d'interconnexion sur puce." *Les Journées Doctorales d'Informatique et Réseaux (JDIR)* (2011).
- [60] G. Chen, F. Li et M. Kandemir, «Reducing Energy Consumption of On-Chip Networks Through a Hybrid Compiler-Runtime Approach,» chez *Parallel Architecture and Compilation Techniques. 16th International Conference on*, 2007.
- [61] Palermo, Gianluca, et al. "Mapping and topology customization approaches for application-specific stnoc designs." *2007 IEEE international conf. on application-specific systems, architectures and processors (ASAP)*. IEEE, 2007.
- [62] T.Pionteck, R. Koch et C. Albrecht, «Applying partial reconfiguration to networks-onchips,» chez *Field Programmable Logic and Applications, 2006. FPL '06. International Conference on*, 2006.
- [63] Hubner, M. Ullmann, L. Braun, A. Klausmann et J. Becker, «Scalable applicationdependent network on chip adaptivity for dynamicalrecon\_gurable real-time systems,» chez *International Conference on Field Programmable Logic and Applications (FPL'04)*, 2004.

- [64] A. Pinto, L. Carloni et A. Sangiovanni-Vincentelli, «Efficient synthesis of networks on chip,» chez Proceedings of the 21st International Conference on Computer Design (ICCD), 2003.
- [65] Li, Mengchu, et al. "CustomTopo: A topology generation method for application-specific wavelength-routed optical NoCs." *Proceedings of the International Conference on Computer-Aided Design*. 2018.
- [66] Srinivasan, Krishnan, and Karam S. Chatha. "ISIS: a genetic algorithm based technique for custom on-chip interconnection network synthesis." *18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*. IEEE, 2005.
- [67] Xu, Thomas Canhao, Pasi Liljeberg, and Hannu Tenhunen. "A study of through silicon via impact to 3D network-on-chip design." *2010 International Conference on Electronics and Information Engineering*. Vol. 1. IEEE, 2010.
- [68] Pasricha, Sudeep. "Exploring serial vertical interconnects for 3D ICs." *2009 46th ACM/IEEE Design Automation Conference*. IEEE, 2009.
- [69] Dong, Xiangyu, and Yuan Xie. "System-level 3D ic cost analysis and design exploration." *Three Dimensional Integrated Circuit Design*. Springer, Boston, MA, 2010. 261-280.
- [70] Ogras, U. Y., & Marculescu, R. (2006). "It's a small world after all": NoC performance optimization via long-range link insertion. *IEEE Transactions on very large scale integration (VLSI) systems*, 14(7), 693-706.
- [71] Chen, G., Li, F., & Kandemir, M. (2007, September). Reducing Energy Consumption of On-Chip Networks Through a Hybrid Compiler-Runtime Approach. In *16th International Conference on Parallel Architecture and Compilation Techniques (PACT 2007)* (pp. 163-174). IEEE.
- [72] Guz, Z., Walter, I. H., Bolotin, E., Cidon, I., Ginosar, R., & Kolodny, A. (2007). Network delays and link capacities in application-specific wormhole NoCs. *VLSI design, 2007*.
- [73] Jiao, J., & Fu, Y. (2011, December). B2RAC: a physical express link addition methodology for network on chip. In *Proceedings of the 4th International Workshop on Network on Chip Architectures* (pp. 17-22).
- [74] Venkataraman, N. L., & Kumar, R. (2019). Design and analysis of application specific network on chip for reliable custom topology. *Computer Networks*, 158, 69-76.
- [75] Gulzari, U. A., Anjum, S., Aghaa, S., Khan, S., & Sill Torres, F. (2017). Efficient and scalable cross-by-pass-mesh topology for networks-on-chip. *IET Computers & Digital Techniques*, 11(4), 140-148.
- [76] Yan, S., & Lin, B. (2011). Design of application-specific 3D networks-on-chip architectures. In *3D Integration for NoC-based SoC Architectures* (pp. 167-191). Springer, New York, NY
- [77] S. Suboh, M. Bakhouya, S. Lopez-Buedo et T. El-Ghazawi, «Simulation-based approach for evaluating on-chip interconnect architectures,» chez Programmable Logic, 2008 4th Southern Conference on, 2008.
- [78] GAPH-group, «Atlas webpage,» [En ligne]. Available: [http://www.inf.pucrs.br/~gaph/AtlasHtml/AtlasIndex\\_us.html](http://www.inf.pucrs.br/~gaph/AtlasHtml/AtlasIndex_us.html)
- [79] S. Evain, J. Diguët et D. Houzet, «ySpider: a CAD Tool for Efficient NoC Design,» Université de Bretagne Sud, LORIENT, 2004.



- [80] J. Chan et S. Parameswaran, «NoCGEN: A Template Based Reuse Methodology for Networks on Chip Architecture,» chez VLSI Design, 2004. Proceedings, 17th International Conference on, 2004.
- [81] J. Flich, P. Lopez, M. Malumbres et J. Duato, «Improving the Performance of Regular Networks with Source Routing,» chez Proceeding of the IEEE International Conference on Parallel Processing, 2000.
- [82] M. Palesi, D. Patti et F. Fazzino, «Noxim»(2002).
- [83] V. Puente, J. Gregorio et R. Beivide, «Sicosys: an integrated framework for studying interconnection network performance in multiprocessor systems,» chez Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on, 2002.
- [84] A. r. r. © Mentor Graphics, «Mentor Graphics,» [En ligne]. Available: <http://www.mentor.com/>.
- [85] J. Emer, P. Ahuja, E. Borch, A. Klauser, C.-K. Luk, S. Manne, S. Mukherjee, S. Mukherjee, H. Patil, S. Wallace, N. Binkert, R. Espasa et T. Juan, «Asim: A Performance Model Framework,» IEEE Computer, vol. 35, n° 12, pp. 68-76, 2002.
- [86] . Thid, M. Millberg, E. Nilsson et A. Jantschi, «NNSE: Nostrum Network-onChip Simulation Environment,» Royal Institute of Technology, Sweden.
- [87] N. Jiang, G. Michelogiannakis, D. Becker, B. Towles et W. Dally, «BookSim 2.0,» 2013.
- [88] Y. Ben-Itzhak, E. Zahavi, I. Cidon et A. Kolodny, «HNOCS: Modular Open-Source Simulator for Heterogeneous NoCs,» chez Embedded Computer Systems (SAMOS), International Conference on, 2012.
- [90] G. Beltrame, C. Bolchini, L. Fossati, A. Miele et D. Sciuto, «ReSP: A non-intrusive Transaction-Level Reflective MPSoC Simulation Platform for design space exploration,» chez Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific, Seoul, 2008.
- [91] R. Lemaire, «Conception et modélisation d'un système de contrôle d'applications de télécommunication avec une architecture de réseau sur puce (NoC),» Institut National Polytechnique de Grenoble, 2006.
- [92] Abbad salim, tileb selma. optimisation multiobjectif des performances des réseaux sur puce , memoire de master université blida (2019).
- [93] Royer, Alexandre. *Evaluation de performances de réseaux de communication à l'aide de chaînes de Markov hybrides*. Diss. Institut National Polytechnique de Grenoble-INPG, 2006.