

D0002/99B

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

T H E S E

Présentée à

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Ecole Nationale Polytechnique

Département d'Electronique

Pour l'obtention du Grade de

Doctorat d'Etat en Electronique

par

Djamel BENAZZOUZ
Magister en électronique

**Modélisation et évaluation des réseaux
d'interconnexions dynamiques par
les réseaux de Petri stochastiques**

Date de la soutenance : le 27 Juin 1999.

Composition du Jury:

Mr. D. BERKANI
Mr. A. FARAH
Mr. K. HARRICHE
Mr. M. AIDENE
Mr. M. DJEDDI
Mr. E. BERKOUK

Professeur à l'ENP
Professeur à l'ENP
Professeur à l'Université de Boumerdes
M. C. à l'Université de Tizi Ouzou
M. C. à l'Université de Boumerdes
Docteur d'état à l'ENP

Président
Rapporteur
Examinateur
Examinateur
Examinateur
Invité

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

à la mémoire de mon père

à ma femme, et mes enfants: Karim, Lyna, et wacim

à ma mère, mes frères et soeurs.

Remerciements

J'exprime ma profonde gratitude et mes remerciements à Ahcène Farah, Professeur à l'ENP et sous la direction duquel ce travail a été réalisé.

Je tiens à remercier D. Berkani, Professeur à l'ENP pour m'avoir fait l'honneur de présider mon jury.

Kamel Harriche, Mohamed Aidène, Mohamed Djeddi, E. Berkouk ont accepté de participer à mon jury, je les en remercie vivement.

Je remercie ma femme sans qui je n'aurais jamais entrepris cette thèse et qui a supporté mes longues absences du foyer; je lui exprime ma profonde gratitude.

Je remercie ma mère pour ses encouragements constants.

Je remercie toute ma famille de m'avoir soutenu et encouragé.

Je remercie aussi Mme Farida Zerari pour son aide précieuse et son amitié. Qu'elle trouve ici ma profonde gratitude et reconnaissance.

Je remercie enfin Med. Lachemi Boudjemeline qui a l'amitié de revoir la forme de la présente thèse; qu'il trouve ici l'expression de ma profonde gratitude.

Résumé

Cette thèse présente la modélisation des réseaux d'interconnexions dynamiques (RID) par les réseaux de Petri stochastiques (RdPS). Notre approche permet l'évaluation des performances des RID en présence de conflit. Ce conflit a été géré selon deux stratégies définies comme stratégie sans mémoire et stratégie avec mémoire. L'analyse des RdPS se ramène à celle d'un processus de Markov homogène à espace d'état discret à temps continu. Les modèles proposés sont des modèles simplifiés qui limitent considérablement le nombre d'état de la chaîne de Markov. La représentation d'un étage ou tour par une place réduit l'explosion des combinaisons des états de la chaîne de Markov. Les modèles s'adaptent facilement à l'extensibilité du réseau tout en gardant les propriétés de fonctionnement du réseau. Les résultats obtenus montrent qu'il n'y a pas de linéarité, d'une part, entre la bande passante (BW) ou le temps moyen de service (Avt) et la taille du réseau, et d'autre part, entre la BW ou le Avt et la longueur du message. Nous retenons que la stratégie sans mémoire est mieux adaptée aux RID, du fait qu'elle offre une meilleure fluidité de passage de messages à travers le réseau. Notre approche nous a permis de montrer que les RdPS constituent un puissant outil mathématique d'évaluation des performances des RID. Cette évaluation prend en considération, d'une part, la défaillance de la boîte de commutation, et d'autre part, la probabilité de passage ou de blocage des requêtes à travers le réseau.

Mots clés: Réseau de Petri stochastique - Processus de Markov - Réseau d'interconnexion dynamique - Système parallèle - Commutation de circuit.

Abstract

This thesis presents an evaluation method of dynamic interconnection networks (DIN) using stochastic Petri nets (SPN). Our approach allows the performance evaluation of a DIN in presence of conflict. This conflict has been managed into two strategies; defined as unbuffered strategy and buffered strategy. The analysis of SPN is a homogeneous markovian process, and therefore to all SPN one could associate a homogeneous markovian chain (MC). Proposed models are simplified models that limit considerably the number of state of the MC. With the representation of a stage or a pass by a place reduce the state explosion combinations of the MC. Models are adapted easily to the scalability of the network. The obtained results show that there is no linearity, on one hand, between the bandwidth (BW) or the average service time (Avt) and the size of the network, and on the other hand, between the BW or the Avt and the length of the message. We retain that the unbuffered strategy is better adapted to DIN, because it offers a low blocking probabilities of the network. Our approach allows to show that SPN constitute a powerful mathematical performance evaluation tool of DIN. This evaluation takes in consideration, on one hand, the failure of the switching element, and on the other hand, the passing or blocking probability of requests through the network.

Keywords: *Stochastic Petri net - Markov process - Dynamic interconnection network - Parallel processing - Circuit switching.*

Tables des matières

Remerciements	i
Résumé	ii
Abstract	iii
Abréviations	vii
Introduction	1
Chapitre I Les machines parallèles	5
1.1 Architecture d'une machine parallèle	5
1.2 Les machines multiprocesseurs	9
1.2.1 Les machines multiordinateurs	9
Chapitre II Les réseaux d'interconnexions	12
2.1 Les caractéristiques d'un réseau d'interconnexion	12
2.2 Réseau d'interconnexion statique	15
2.3 Réseau d'interconnexion dynamique	16
2.3.1 Description de réseaux simples	17
2.4 Les réseaux d'interconnexions à un seul étage	18
2.5 Les réseaux d'interconnexions à plusieurs étages	22
2.5.1 Réseau Clos	22
2.5.2 Réseau Butterfly	23
2.5.3 Réseau Benès	24
2.5.4 Réseau Oméga et Baseline	24
2.6. Comparaison des performances des réseaux d'interconnexions	25
Chapitre III Les réseaux de Petri	27
3.1 Introduction	27
3.2 Réseaux de Petri : Définitions et Propriétés	28
3.2.1 Définitions de Base	28
3.2.1.1 Marquage	29
3.2.1.2 Règles de fonctionnement	29
3.2.2 Analyse des RdP	30

3.2.2.1 Matrice d'incidence	30
3.2.2.2 Temporisation	32
3.3 Les Réseaux de Petri Stochastiques	32
3.3.1 Pourquoi ?	32
3.3.2 Comment définir les réseaux de Petri stochastiques	33
Chapitre IV Modélisation des réseaux d'interconnexions à plusieurs étages	35
4.1 Introduction	35
4.2 Modélisation d'un réseau d'interconnexions à plusieurs étages	35
4.2.1 Modèle 1 sans mémoire	38
4.2.1.1 Description du modèle 1 sans mémoire	39
4.2.2 Modèle 2 sans mémoire	40
4.2.3 Modèle avec mémoire	40
4.2.3.1 Description du modèle avec mémoire	41
4.3 Processus de Markov	43
4.3.1 Propriété de conservation	43
4.3.2 Conservation du flux en régime stationnaire	43
4.3.3 Formule de Little pour les RdPS	44
4.3.4 Chaîne de Markov	44
Chapitre V Evaluation des performances des réseaux d'interconnexions à plusieurs étages	48
5.1 Evaluation du modèle 1 sans mémoire	48
5.2 Evaluation du modèle 1 avec mémoire	49
5.3 Algorithme de résolution	50
5.4 Applications	51
5.4.1 Modèle 1 sans mémoire	51
5.4.2 Modèle 2 sans mémoire	57
5.4.3 Modèle avec mémoire	61
5.5 Discussion	69

Chapitre VI	Modélisation des réseaux d'interconnexions à un seul étage	72
6.1	Introduction	72
6.2	Modélisation du RIT selon la stratégie sans mémoire	74
6.2.1	Description du modèle sans mémoire	74
6.3	Modélisation du RIT selon la stratégie avec mémoire	78
6.3.1	description du modèle avec mémoire	79
6.4	Algorithme de résolution	82
6.5	Evaluation du modèle sans mémoire	84
6.6	Evaluation du modèle avec mémoire	87
6.7	Discussion	92
Conclusion		94
Bibliographie		96

Abréviations



SISD	Single instruction single data
SIMD	Single instruction multi data
MISD	Multiple instructions single data
MIMD	Multiple instructions multiple data
M	Mémoire
P	Processeur
PE	Processor element
RI	Réseau d'interconnexion
RIP	Réseau d'interconnexion à plusieurs étages
RIT	Réseau d'interconnexion à un seul étage
n	Nombre d'étage pour un RIP et tour pour un RIT
N	Nombre d'entrée ou de sortie d'un RI
CIE	Connexion interétage
BC	Boîte de commutation
RdP	Réseau de Petri
RdPS	Réseau de Petri stochastique
W	Matrice d'incidence
W ⁻	Matrice d'incidence arrière
W ⁺	Matrice d'incidence avant
CM	Chaîne de Markov
Q	Matrice associée aux chaînes de Markov
P _i	Place numéro i d'un RdP, ensemble des Places
T _j	Transition j d'un RdP, ensemble des transitions
M _k	Marquage d'un RdP
M ₀	Marquage initial d'un RdP
M _f	Marquage final d'un RdP
M(P _i)	Marquage de la place P _i
σ	Séquence de franchissement
$\underline{\sigma}$	Vecteur caractéristique de la séquence de franchissement de σ

T_j^0	Ensemble de sortie T_j
P_j^0	Ensemble de sortie P_j
0T_j	Ensemble d'entrée T_j
0P_j	Ensemble d'entrée P_j
μ_i	Probabilité conditionnelle de franchissement, définie comme probabilité de blocage
λ_i	Probabilité conditionnelle de franchissement, définie comme probabilité de passage
φ_i	Probabilité conditionnelle de franchissement, définie comme probabilité de connexion
σ_i	Probabilité conditionnelle de franchissement, définie comme probabilité de déblocage
$E(x)$	Espérance mathématique
F^*_{moy}	Fréquence moyenne de franchissement
D^*_{moy}	Temps moyen de séjour dans les places

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

INTRODUCTION



Introduction

Le développement de l'informatique et de ses applications a entraîné une demande sans cesse croissante en puissance de calcul. Pour répondre à ces besoins, les machines parallèles ont reçu un intérêt tout particulier. De ce fait, de nombreux domaines de recherche nouveaux ont émergé; par exemple, les langages parallèles, la parallélisation automatique, les architectures distribuées, les environnements de programmation, l'analyse de complexité parallèle et l'algorithme parallèle.

Actuellement, la plupart des machines existantes reposent sur des architectures à mémoire distribuée. Une des différences essentielles entre les machines traditionnelles, dites de "Von Neumann", et les réseaux de processeurs que sont les machines à mémoire distribuée, réside dans la gestion des communications. Si on cherche à distribuer le plus possible les traitements élémentaires, on obtient le parallélisme potentiellement le plus efficace, mais les communications induites à travers les réseaux peuvent être alors pénalisantes. Optimiser les communications est donc une des clefs de l'efficacité des algorithmes parallèles.

Si on peut souhaiter une modélisation complète des calculateurs parallèles qui prendrait en compte à la fois le réseau d'interconnexion, les calculs parallèles et les communications associées, la difficulté de sa réalisation apparaît bien vite; seuls quelques résultats particuliers concernant le recouvrement des communications et des calculs existent (dans ce cadre les seuls résultats existants sont en général associés à des instances particulières de machine) [ETU 95]. Notons au passage que le nombre de modèles "globaux" est d'un coup déraisonnablement élevé, puisqu'un modèle complet doit prendre en compte la vitesse relative des instructions de calcul et de communication, leur parallélisme ou non, etc...

Ainsi, à l'heure actuelle, il semble raisonnable d'étudier séparément les calculs et les communications, puis, dans le cas d'une application réelle, tenter de prendre en compte les spécificités de l'architecture utilisée en espérant qu'un bon algorithme, muni de bonnes routines de communications, fournisse un résultat efficace.

Les besoins actuels en matière de calcul qui se manifestent dans différents domaines nécessitent des systèmes informatiques de plus en plus puissants. Ainsi, des projets importants et ambitieux ont pu être mis en oeuvre, tel le programme HPCC (Highly Parallel

Computing and Communication) aux Etats-Unis [TMC 91]. L'objectif de ce type de projets est d'élaborer des calculateurs suffisamment puissants pour faire en sorte que les performances actuellement visées soient atteintes. Ceci est une nécessité pour les applications qui demandent de plus en plus de performances, comme la modélisation du climat, la modélisation des semi-conducteurs, des supraconducteurs,...

Les besoins en calcul sont sans cesse croissants; cela a poussé les architectes de systèmes informatiques à s'intéresser aux systèmes parallèles. En effet, les chercheurs, profitant du parallélisme d'exécution de plusieurs processeurs, espèrent atteindre la machine idéale qui réalise les 1000 TeraFlops (soit 10^{15} opérations par seconde en virgule flottante 64 bits) [REW 97].

La puissance de calcul d'une machine parallèle est potentiellement proportionnelle au nombre de processeurs qu'elle comporte. Cependant, le choix d'une machine pour une application donnée reste toujours posé. L'évaluation des performances des machines parallèles, particulièrement en ce qui concerne les réseaux d'interconnexions, discipline dans laquelle s'inscrit cette thèse, apporte un plus d'information qui permette de répondre à cette question.

L'évaluation des performances d'une machine parallèle consiste en la mesure des grandeurs caractéristiques de son comportement, telles que les vitesses de ses principaux composants (processeur, mémoire, périphérique, etc...), et les liens entre ces principaux composants (disponibilité, capacité de transport d'information, etc...). L'évaluation des performances d'un programme consiste en la mesure des durées caractéristiques de son exécution, comme par exemple la durée totale; ou pour un programme parallèle le temps qu'un processus de ce programme va passer en état bloqué (plus ce temps est long, plus le programme a des chances d'être inefficace).

C'est dans ce contexte que se situe la partie de notre travail dont le but est en effet d'étudier l'évaluation des performances, des réseaux d'interconnexions par les réseaux de Petri stochastiques. Le modèle que nous proposons est constitué de paramètres représentant les différents éléments de la machine tels que les processeurs, les mémoires, les liens de connexions, ainsi que les paramètres architecturaux tels que le nombre de processeurs, la taille mémoire disponible sur chacun des noeuds. Tous ces paramètres caractérisent le type de machine parallèle.

La contribution de notre travail de recherche consiste à la simplicité du modèle proposé. Les systèmes d'une machine parallèle sont tellement complexes que l'évaluation de leurs performances est très difficile. Le choix de ramener les réseaux de Petri stochastiques au processus de Markov à espace d'état discret, simplifie considérablement la modélisation d'un tel système. Notre approche permet aussi d'identifier les paramètres qui interviennent directement ou indirectement dans l'évaluation des performances des réseaux d'interconnexions dynamiques.

Ce travail comprend six chapitres.

Le premier chapitre concerne les différents aspects des machines parallèles. Nous décrirons particulièrement les diverses topologies qui caractérisent l'architecture d'une machine parallèle. Nous donnerons la classification, fondée par Flynn [FLY 72], de ces machines. Puis nous énumérerons quelques exemples de structure de machines parallèles.

Le deuxième chapitre présente les principales caractéristiques d'un réseau d'interconnexion (RI). Nous présenterons quelques topologies d'un RI dynamique particulièrement les RI à plusieurs étages (RIP) et les RI à un seul étage (RIT). C'est dans ce contexte que se situe la partie la plus importante de notre travail de recherche. On trouve beaucoup de travaux sur les RIP mais relativement peu de travaux concernent les RIT.

Le troisième chapitre concerne les réseaux de Petri (RdP) qui sont un outil mathématique très utilisé actuellement dans la modélisation des activités asynchrones ou synchrones, concurrentes et parallèles. Nous citerons les avantages principaux des RdP avec l'ensemble de bases et règles de fonctionnement. Enfin, nous présenterons l'extension des RdP aux réseaux de Petri stochastiques (RdPS). Les RdPS occupent une place privilégiée; d'une part, parcequ'ils permettent d'exprimer facilement les mécanismes premiers des systèmes distribués (le parallélisme et la synchronisation) d'autre part, parcequ'ils fournissent des méthodes d'analyses éprouvées très efficaces [FLO 90]. Par la suite nous utiliserons les RdPS à distributions exponentielles qui ont été suggérés par Molloy [MOL 81], Natkin [NAT 80], et Zuberek [ZUB 85]. L'analyse de ce type de modèle passe par l'étude de la chaîne de Markov continue associée au processus stochastique sous-jacent au modèle. Enfin nous donnerons les motivations quant au choix de cet outil.

Le quatrième chapitre décrira la boîte de commutation et une classe des RIP dont le réseau **baseline** qui est considéré dans le cadre notre approche. Nous présenterons la modélisation des RIP par les RdPS, ainsi que les hypothèses retenues. Dans la première partie de ce chapitre, une généralisation de la modélisation des RIP est développée selon

deux types de stratégies: sans mémoire puis avec mémoire. Dans la deuxième partie nous montrerons que l'analyse d'un RdPS se ramène à celle d'un processus de Markov homogène à espace d'état discret à temps continu.

Dans le cinquième chapitre nous donnerons l'application de notre approche pour des modèles à 3 étages des RIP.

Le sixième chapitre sera consacré à l'application d'une modélisation des RIT selon les deux stratégies retenues pour des modèles à 3 étages.

La dernière partie exposera les conclusions du présent travail.

**LES MACHINES
PARALLELES**

CHAPITRE I

Les machines parallèles

Ce chapitre présente les différents aspects des machines parallèles. Partant du niveau bas, nous décrirons tout d'abord les principales caractéristiques matérielles, c'est à dire les différentes familles de machines parallèles et les différentes topologies (graphes de connexion entre les noeuds de calcul).

L'architecture des ordinateurs a connu d'importantes mutations. Depuis le modèle d'architecture de Von-Neumann proposé dans les années 50 et pendant une vingtaine d'années, les progrès ont essentiellement concerné l'exploitation des apports de l'électronique. Il a fallu attendre les années 80 pour connaître les premières machines RISC (Reduced-Instruction-Set Computing) qui ont permis de lancer des projets précurseurs sur le développement des architectures parallèles. Les évolutions technologiques, notamment les possibilités qu'offrait la micro-électronique avec l'arrivée des premiers circuits VLSI (Very Large Scale Integration), ont été le plus grand stimulant du développement de la nouvelle branche de l'architecture des machines parallèles.

En réalité, le développement des machines parallèles ne fait pas seulement appel au développement architectural mais également à la définition de nouveaux modèles de calcul non séquentiels, c'est à dire ne comportant pas forcément un compteur ordinal qui cadence l'exécution des instructions les unes après les autres. En effet, ces modèles se basent sur une nouvelle organisation des instructions (neuronale, concept 'demand-driven', ...) et s'appuient sur l'organisation intrinsèquement parallèle des applications. Il peut s'agir aussi bien d'un parallélisme de contrôle que de données.

1.1 Architecture d'une machine parallèle

Une machine parallèle est composée de plusieurs processeurs pouvant réaliser des calculs simultanés. Une telle machine comprend donc des processeurs, de la mémoire et des gestionnaires d'entrée/sortie. Il existe actuellement de nombreuses machines parallèles.

Elles peuvent être classées selon divers critères. Une classification générale devrait d'une part, tenir compte des éléments introduits par Flynn dans sa classification qui prend en

compte le flux d' instructions et de données [FLY 72] et d'autre part de procéder à l'évaluation de ces machines.

En effet, certaines machines élaborées ne peuvent pas être classées selon les critères introduits par Flynn: machine à base de processeurs vectoriels pipelinés, machines 'Data-Flow', ...[SAN 91], [DUN 90], [ETU 95]. La classification de Flynn est fondée sur le fait que la machine parallèle possède un ou plusieurs flux de données et d'instructions. La combinaison entre ces deux paramètres aboutit à quatre types d'architectures résumées dans le Tableau 1

		Flux de données	
		Simple	Multiple
Flux d'instructions	Simple	SISD	SIMD
	Multiple	MISD	MIMD

Tableau 1. Taxinomie des architectures parallèles proposée par Flynn.

Nous pouvons considérer les machines parallèles comme étant divisées en deux grandes classes: les machines synchrones et les machines asynchrones.

a) Machines synchrones

Ces machines comportent des unités de calcul (processeurs câblés) qui exécutent la même instruction par unité de contrôle centralisée commune (voir Figure 1.1). Un réseau d'interconnexion permet d'assurer la communication des processeurs entre eux ou des processeurs aux mémoires.

La classe SIMD (Single Instruction Multiple Data) est la machine pour laquelle tous les processeurs exécutent la même instruction sur des données différentes. Le comportement de ce type de machines est synchrone. Son fonctionnement est très intéressant pour le traitement des images bidimensionnelles, où la même opération est effectuée sur tous les pixels. Un exemple bien connu de machines SIMD correspond à la CM2 (Connection Machine 2 de Thinking Machine et la Maspar) [TMC 91].

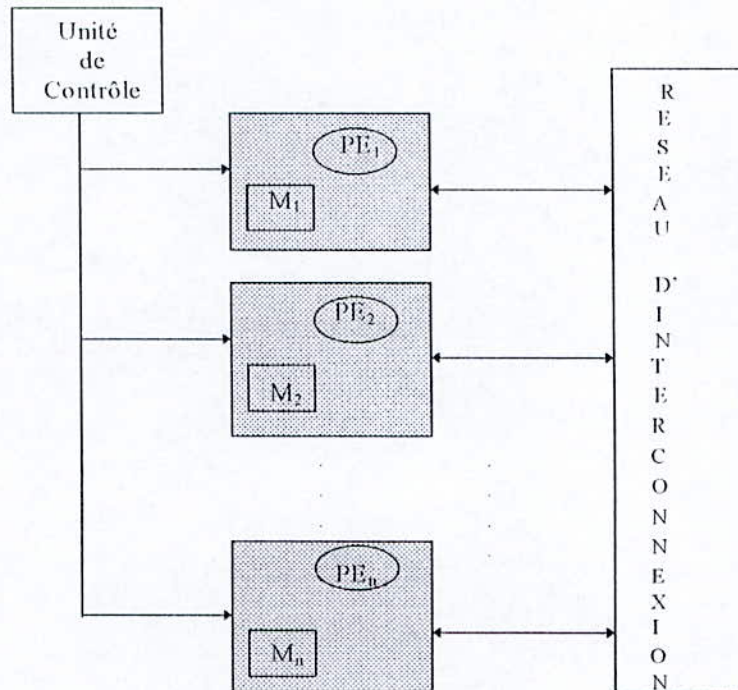


Fig. 1.1 Organisation générale d'une machine SIMD. (PE: processor element, M: memory).

Les machines synchrones présentent l'avantage d'un modèle de programmation simple qui se rapproche du modèle de programmation séquentiel classique. Cela facilite leur exploitation. Néanmoins, en vue d'une utilisation massivement parallèle, les machines SIMD restent limitées surtout en ce qui concerne les contraintes technologiques relatives à la diffusion d'une instruction à des milliers de processeurs [ETU 95].

En effet, il est pratiquement très difficile d'assurer un contrôle centralisé des instructions quand le nombre de processeurs dépasse une certaine limite [UPL 94].

b) Machines asynchrones

La classe MIMD (Multiple Instruction Multiple Data) pour laquelle les processeurs sont plus indépendants, et peuvent exécuter des instructions différentes sur des données différentes. Le comportement de ces machines est asynchrone.

Remarque:

La plupart des machines fonctionnent aujourd'hui en mode MIMD; cependant elles sont souvent programmées en mode SPMD (Single Program Multiple Data). Ce mode

intermédiaire est dérivé des deux modes précédents; il est très répandu actuellement, et signifie que tous les processeurs exécutent le même code sur des données différentes. Il faut faire attention au fait que ce mode caractérise la programmation et non le fonctionnement matériel de la machine.

Un autre critère est utilisé pour classer les machines parallèles; il concerne l'architecture de leur mémoire et l'on distingue alors deux types de machines:

- Soit que la mémoire correspondante est partagée et/ou les processeurs partagent donc les données d'un programme; l'organisation d'une telle machine est schématisée par la figure 1.2 et se présente de telle façon que tous les processeurs accèdent à la même mémoire.

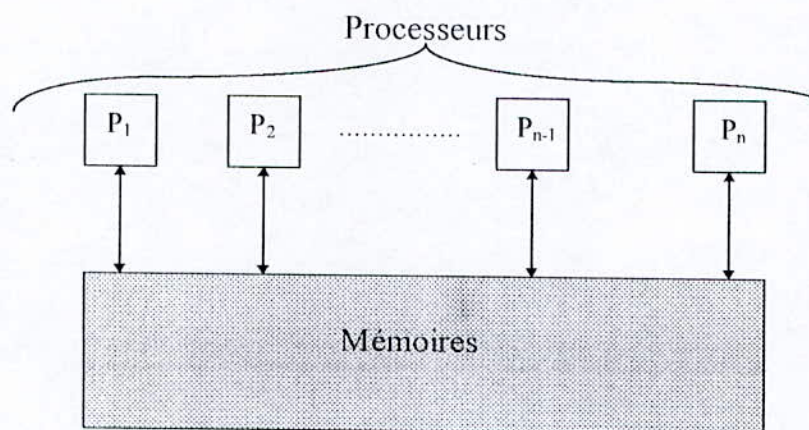


Fig. 1.2 Architecture schématique d'une machine parallèle à mémoire partagée.

- Soit que la mémoire correspondante est distribuée et/ou chacun des processeurs possède sa mémoire. La coopération (échange de données) et la synchronisation entre processeurs ne peut alors se faire que par l'intermédiaire de liens de communications; c'est à dire par l'échange de messages (ou par communication). L'ensemble de liens de communication d'une machine est appelé le **réseau de communication** ou **réseau d'interconnexion**. L'organisation d'une telle machine est schématisée par la figure 1.3.

L'architecture des machines parallèles à mémoire partagée, ou distribuée sont dites respectivement machines multiprocesseurs ou multiordinateurs.

Les multiordinateurs sont plus adaptées à une perspective de machines massivement parallèles, car dès que le nombre de processeurs atteint une certaine limite, le partage d'une

mémoire devient pratiquement impossible. La structure physique de la mémoire peut ne pas directement correspondre au modèle de programmation et d'exécution.

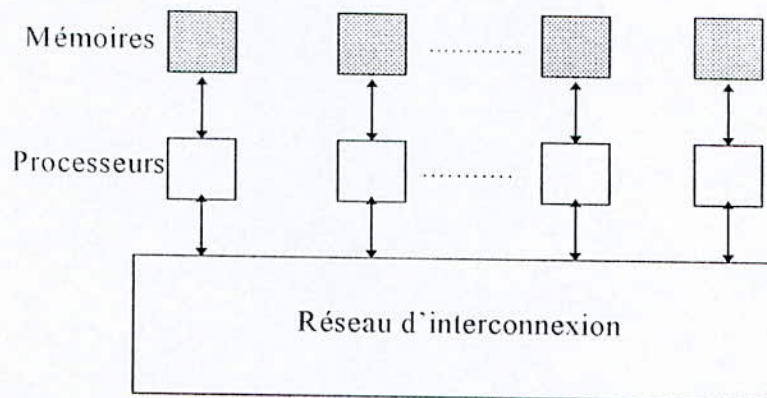


Fig. 1.3 Architecture schématique d'une machine parallèle à mémoire distribuée.

1.2 Les machines multiprocesseurs

Bien que ces machines ne soient pas adaptées à une perspective de machines massivement parallèles, leur évolution présente un double intérêt. D'une part, leur modèle de programmation reste très souple et très utilisé dans les machines à multiordinateurs (en se basant sur un espace d'adressage unique). D'autre part, des performances intéressantes sont atteintes par ces machines, notamment celles formées à base de groupes ou "clusters" telle que la machine EXEMPLAR. Comme exemple de machines multiprocesseurs, on peut citer la KSR (Kendell Square Research), les machines vectorielles CRAY (XMP et YMP),...[RAM 93]

1.2.1 Les machines multiordinateurs: Machines MIMD à passage de messages

Ces machines dont l'organisation générale est présentée sur la figure 1.4, sont dites massivement parallèles. Parmi les machines MIMD développées et commercialisées, nous citons la CM-5 de TMC (Thinking Machine Corporation), et la Paragon d'INTEL. L'exemple de la CM-5 [TMC 92] représente bien ce type de machines. En effet, elle tire profit des modèles d'exécution d'autres architectures (SPMD susceptible de se substituer au SIMD sur des processeurs standards) et exploite également les performances de processeurs SPARC ainsi que des mémoires dynamiques (surmontant ainsi le coût des mémoires statiques).

Le but des constructeurs est de mettre de plus en plus de processeurs dans les machines puisqu'a priori cela multiplie d'autant la puissance de calcul. Les très grosses mémoires étant difficiles à construire et souvent plus lentes, les machines parallèles à mémoire partagée comportent peu de processeurs (moins de 64) et la majorité des machines présentées actuellement sur le marché sont à mémoire distribuée.

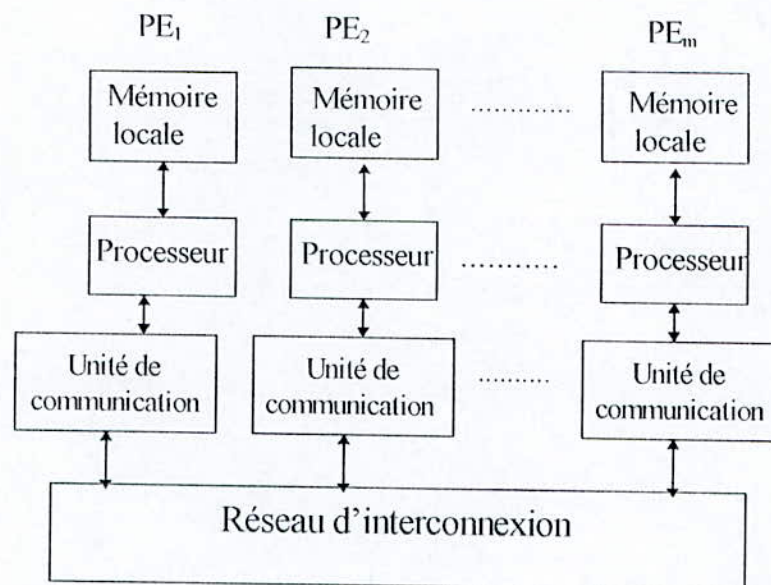


Fig. 1.4 Organisation générale des machines MIMD à passage de messages

L'augmentation de la puissance de calcul est, en fait, fonction de plusieurs paramètres; on citera les paramètres essentiels tel que le type de mémoire, les processeurs, l'architecture adoptée et le réseau d'interconnexion. On voit beaucoup de recherches orientées vers les performances des mémoires et des processeurs mais relativement peu de recherche portant particulièrement sur les réseaux d'interconnexions.

Dans cette thèse nous nous intéressons particulièrement aux RI. Ces réseaux sont généralement choisis pour respecter deux contraintes: facilement extensibles et assurer un lien de connexion entre chaque entrée et sortie. L'augmentation de la puissance de calcul des machines parallèles passe théoriquement par l'ajout de processeurs, il faut donc que cet ajout de noeuds soit facile et ne change pas fondamentalement les propriétés du réseau, c'est ce que nous appelons la contrainte *d'extensibilité*.

On sait donc construire des réseaux complètement connectés que l'on appelle *commutateurs* mais pour un petit nombre de processeurs (<64) [TMC 92]. Pour avoir un nombre de processeurs intéressant, les constructeurs se sont d'abord tournés vers d'autres

réseaux facilement extensibles, telles que les grilles ou tores 2D ou 3D (par exemple Computing Surface de Meiko, Transputers de Telmat), et les hypercubes (par exemple n Cube, CM2, iPSC) [RAM 93].

On distingue aussi une nouvelle génération de réseaux tels que le réseau de la dernière née de Thinking-Machine (fat-tree) [LEI 85], et les réseaux de la CS2 et de la SP1 qui sont des réseaux Omega. On reviendra avec plus de détail sur ces types de réseaux dans le chapitre II qui présentera les RI.



**LES RESEAUX
D'INTERCONNEXIONS**

CHAPITRE II

Les réseaux d'interconnexions

2.1 Les caractéristiques d'un réseau d'interconnexion

Les réseaux d'interconnexions d'une machine parallèle se présente comme un réseau d'interconnexion point-à-point. Chaque point ou noeud est généralement matérialisé par une entité complexe composée de différents modules: processeur (s), mémoire (s) externes et/ou composants de gestion et routage ou routeur des communications. La figure 2.1 schématise un modèle de noeud d'un réseau.

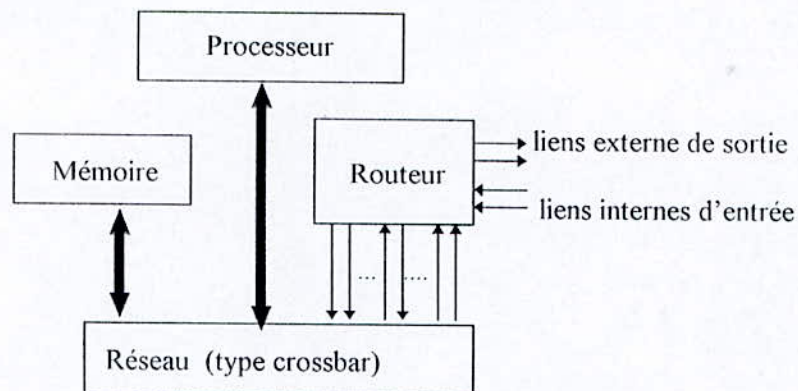


Fig. 2.1 Schématisation d'un noeud.

Dans une telle schématisation, nous considérons que le processeur (généralement de calculs) échange des messages avec le reste du réseau par l'intermédiaire d'un commutateur ou routeur. Dans ce cas, certains liens alimentent le routeur depuis le processeur, ou utilisent des messages en provenance du réseau. Ces liens sont des canaux internes du noeud. D'autres au contraire permettent le transit de messages par le routeur, ce sont les canaux externes. Le rôle d'un routeur est d'acheminer les messages entre noeuds non voisins. Les routeurs réalisent de façon distribuée un algorithme de routage qui spécifie le chemin à suivre dans le réseau pour se rendre de tout noeud x vers tout noeud y . Cet algorithme de routage est décrit par une fonction de routage [FRA 95].

Généralement on distingue 4 caractéristiques dans un RI [FEN 81], [BHU 87]:

a) Temps (Timing)

Le temps peut être synchrone ou asynchrone. Les systèmes synchrones sont utilisés par exemple dans une configuration de noeuds en ligne (array). Une horloge centrale est utilisée pour que les processeurs dans les noeuds communiquent de façon synchrone. Les systèmes asynchrones ne disposent pas d'une horloge centrale qui contrôle les communications. Les requêtes de connexions sont issues des processeurs de façon asynchrone.

b) Mode de commutation (Switching)

Dans un RI point-à-point, la transmission d'un message entre processeurs non directement voisins, doit être routé via des noeuds intermédiaires. Comme schématisé sur la figure 2.1, nous considérons ici que le routeur de message se fait par l'intermédiaire de routeur implanté dans chaque noeud. La commutation d'un routeur est le phénomène consistant en la réception d'une adresse de destination, puis au décodage de cette adresse pour déterminer le canal de sortie convenable et à l'envoi de cette adresse sur ce canal. Mais il existe différents mécanismes physiques ou technologiques de commutation utilisés par les routeurs. Ces différences de mécanismes sont en partie dues au progrès et à la maîtrise technologiques acquises par les différents constructeurs au cours du temps.

Néanmoins, tous les types de commutations ont, dans certains cas, des avantages et dans d'autres, des inconvénients. On trouve une présentation des diverses techniques de commutation classiques dans le rapport Uplal [UPL 94] et du livre de J. de Rumeur [RUM 94].

Nous citons les différents modes classiques qui nous permettent de mieux comprendre l'échange d'information entre processeurs et mémoires à travers un RI. Ces modes de commutations sont présentés en partie dans le livre de J. de Rumeur [RUM 94].

*** Commutation de messages (Store-and-forward)**

Il s'agit certainement du modèle le plus simple et le plus ancien (modélisant la quasi-totalité des machines parallèles jusqu'au début des années 90).

Dans ce cas, les communications s'effectuent entre voisins [SAA 89]. Les messages avancent dans le réseau vers leur destination en transitant dans les noeuds intermédiaires. Un en-tête contenant un identificateur du destinataire est ajouté au message. Ainsi, le

message est stocké avant d'être réémis (d'où le terme anglais consacré dans le cadre des machines distribuées 'store-and-forward') par un noeud intermédiaire. A chaque étape, le canal emprunté est aussitôt libéré. Un défaut de cette technique est de nécessiter une taille de registre importante pour stocker le message sur les processeurs intermédiaires. En fait, ces messages sont généralement stockés en mémoire. Mais les temps d'accès à la mémoire, proportionnels à la taille des messages, ralentissent alors fortement les communications. Exemples, la Computing Surface de Meiko; le T-node et le Mega-node de Telmat, les Voltex d'Archipel, ainsi que les autres machines à base de transputers utilisent un routage par commutation de messages [UPL 94].

*** Commutation par paquets (Packet Switching)**

Ce mode dérive du précédent. En effet, cette fois les messages sont découpés en paquets de taille fixe; on bénéficie alors d'un effet pipeline. Cela signifie que les paquets sont envoyés les uns à la suite des autres: le second n'attend pas que le premier soit arrivé à destination. Les paquets sont routés indépendamment les uns et des autres, car chacun possède un en-tête l'adresse de destination, ils peuvent même emprunter des chemins différents. Notons que ceci implique, par rapport à la commutation de messages, un surplus de communications engendré par le supplément d'information ajouté à chaque paquet. De plus, la reconstitution du message à partir des différents paquets demande également le transport d'un supplément d'information sur chaque paquet. Par contre, l'avantage de ce mode de commutation est de pouvoir utiliser des registres de petite taille, locaux au routeur. La capacité de stockage nécessaire sur un noeud pour un canal se limite à la taille d'un paquet.

*** Commutation de circuit (Circuit Switching)**

Dans ce mode, l'en-tête est envoyé en premier et s'il peut atteindre la destination, il réserve le chemin par lequel il est passé. Le message est alors envoyé directement de la source à la destination, sans être stocké sur les processeurs intermédiaires qui le laissent passer. Les messages qui voudraient emprunter tout ou partie de ce chemin restent bloqués jusqu'à ce qu'ils soient libérés. C'est le principe du téléphone. On établit d'abord la liaison (cela consiste en la matière à réserver une suite de canaux); la conversation commence ensuite.

Exemple: le direct-connect, qui est proposé par Intel sur les hypercubes de la série iPSC/2 et iPSC/860 est un routage par commutation de circuits.

c) Stratégie de contrôle

La stratégie de contrôle d'un RI peut être centralisée ou distribuée. Dans un contrôle centralisé, un contrôleur général reçoit toutes les requêtes de communications et assure la commutation nécessaire pour établir la connexion entre la source et la destination. Dans un contrôle distribué, un contrôleur local gère les requêtes de communications de façon distribuée.

d) Topologie

La topologie d'un réseau détermine l'interconnectibilité entre deux noeuds (entrée et sortie). Il existe deux classes de topologies; statique et dynamique. La topologie statique possède des liens dédiés entre les noeuds; ces liens sont figés et ne peuvent être reconfigurables. Par contre, la topologie dynamique possède des liens entre les noeuds qui peuvent être reconfigurés.

Nous nous intéressons dans cette thèse à la topologie dynamique. Beaucoup de recherches ont investi la topologie statique; on trouve peu de recherches sur la topologie dynamique; en fait, il s'agit de la nouvelle génération de machines parallèles.

Plusieurs structures de RI existent [FEN 81]. Ces réseaux ont une topologie statique, adoptent un temps asynchrone, utilisent un mode de commutation par paquet, basé sur un contrôle distribué.

2.2 Réseau d'interconnexion statique

La figure 2.2 montre quelques RI à topologie statique. La figure 2.2 (a) montre une topologie unidimensionnelle définie comme topologie linéaire (linear array). C'est la plus simple des topologies et peut être utile dans les systèmes à pipeline. La figure 2.2 (b), (c) et (d) montre une topologie bi-dimensionnelle. Ceci est obtenu en fermant la topologie linéaire de la figure 2.2 (a) pour former un anneau (ring). L'anneau est un réseau très utilisé à cause de sa structure simple et de sa facilité d'extension (par exemple, le Token-ring d'IBM). De plus, de nombreux algorithmes ont été écrits pour l'anneau [RUM 94].

La topologie en arbre binaire (Tree) de la figure 2.2 (c) est une topologie régulière mais asymétrique. La figure 2.2 (e) et (f) montre une topologie tridimensionnelle. L'arbre binaire complet (Fully connected) est le plus complexe; chaque noeud est connecté à un autre noeud. Il s'agit là aussi d'un réseau très courant, en particulier pour interconnecter un petit nombre de processeurs. De nombreux algorithmes ont été écrits pour cette structure.

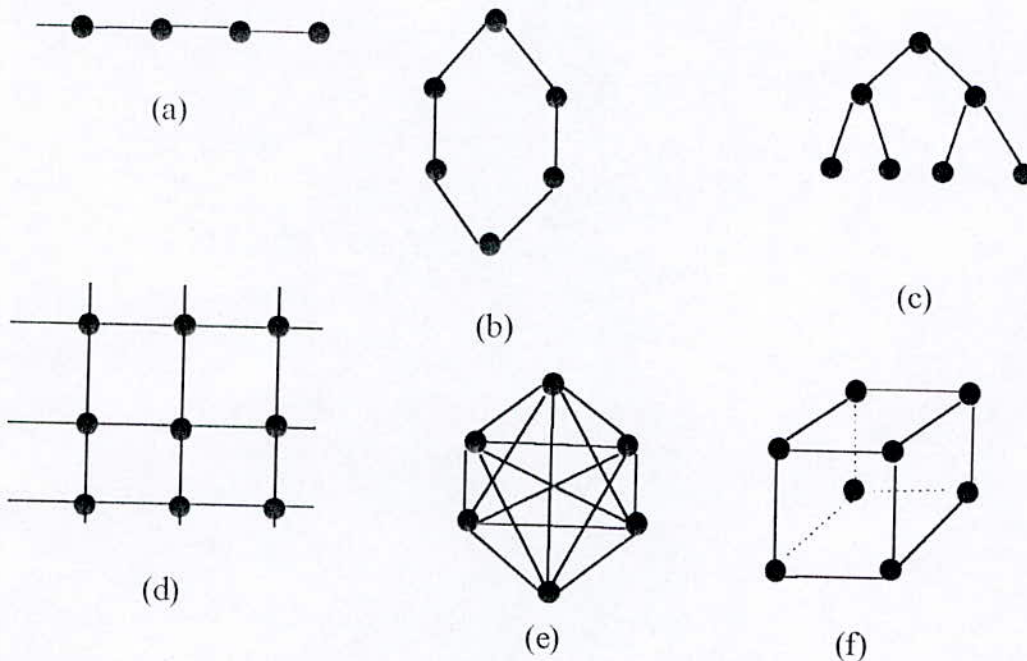


Fig. 2.2 Exemples de topologie statique. (a) arbre linéaire ou bus, (b) anneau ou cycle, (c) arbre binaire complet, (d) mesh, (e) anneau complet, (f) 3-cube.

Il existe aussi d'autre type de topologie statique tel que l'hypercube qui est un réseau utilisé dans plusieurs machines parallèles comme les ncube de nCUBE, les iPSC-1, 2 et 860 d'Intel ou les connection Mahines (CM-1 ou CM-2).

2.3 Réseau d'interconnexion dynamique

Nous venons de présenter la construction des réseaux de processeurs les plus usuels. Ces réseaux ont une configuration statique, dans le sens où la topologie est fixée une fois pour toutes. Il existe une seconde catégorie de réseaux dont la topologie est dynamique, c'est à dire pour lesquels le support physique est fixé, mais où des commutateurs (switches), généralement appelé boîte de commutation, permettent de modifier le schéma de connexion. On dit que ces réseaux sont reconfigurables par opposition aux réseaux statiques.

En effet, ces réseaux comprennent des boîtes de commutations qui peuvent être positionnées de manière différente dans le temps et qui, suivant leur position, permettent d'obtenir la connexion désirée. Dans ce type de réseau on distingue deux classes: réseau d'interconnexion à un seul étage (RIT) et réseau d'interconnexion à plusieurs étages ou multiétages (RIP). Le terme 'multiétage' signifie que les commutateurs, qui sont les briques de base du réseau, sont organisés en étages. Un étage est composé de commutateurs. Chaque commutateur possède un certain nombre de liens d'entrée et de sortie. Plusieurs étages sont juxtaposés. Si un réseau est constitué de n étages, numérotés en général de 0 à $n-1$, les sorties des commutateurs de l'étage i (que l'on appelle plus brièvement sorties de l'étage i) ne peuvent être directement liées qu'aux entrées de l'étage $i+1$ (entrées de commutateurs de l'étage $i+1$), ceci pour tout i , $0 \leq i \leq n-2$. L'ensemble des liens d'entrée de l'étage 0 constitue des liens d'entrée du réseau.

Dans un réseau multiétages, il n'est possible d'établir une nouvelle liaison qu'entre une entrée libre du réseau (processeur source) et une sortie libre du réseau (un banc mémoire destination), c'est à dire une entrée et une sortie qui ne sont pas en cours de communication.

2.3.1 Description de réseaux simples

La complexité des réseaux multiétages se mesure en nombre de commutateurs élémentaires ou point de croisement mis en jeu dans le réseau. Le plus simple de ces réseaux est le *bus* utilisé en temps partagé, représenté dans la figure 2.3. Tout processeur (source) est relié à toute mémoire (destination) par l'intermédiaire du même bus, mais une seule requête peut être envisagée à la fois. En effet, dès que le bus est utilisé pour relier un processeur à une mémoire, il faut attendre qu'il se libère pour relier un autre processeur à une autre mémoire. La mise en oeuvre est en général facile.

Le réseau le plus complexe est le **crossbar**. Dans ce type de réseau, il existe toujours une liaison possible entre tout couple de noeuds libres (processeur vers banc mémoire), indépendamment des liaisons déjà existantes. On peut représenter un réseau crossbar par une grille (P, M) , les P processeurs étant en tête des lignes, les M bancs mémoires en tête de colonnes, telque représenté dans la figure 2.4, avec un interrupteur à chaque point de croisement entre une ligne et une colonne. Chaque processeur P_i est alors relié à la mémoire M_j en fermant l'interrupteur se trouvant au point de croisement de la ligne i et la colonne j .

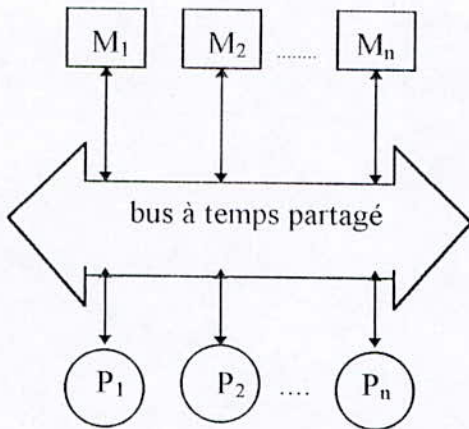


Fig. 2.3 Liaison par bus à temps partagé

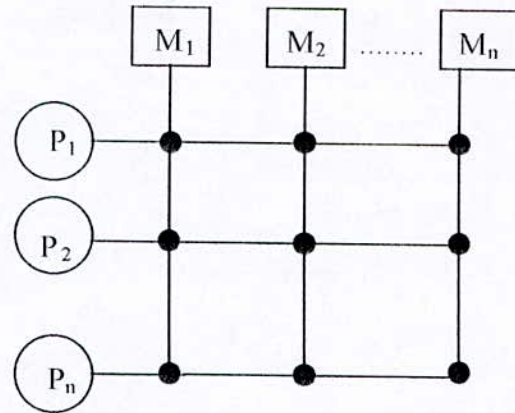


Fig. 2.4 Topologie crossbar

Le réseau crossbar est malheureusement le réseau le plus coûteux qui permette de réaliser toute permutation en établissant des liaisons dans un ordre quelconque. Le nombre de points de croisement est égal au produit du nombre de processeurs par le nombre de bancs mémoires, c'est à dire de l' $O(N^2)$ en supposant N-entrées / N-sorties.

Dans les réseaux dynamiques on cherche à construire des commutateurs dans lesquels chaque processeur soit potentiellement connecté à tous les bancs mémoire via le réseau multiétages [RUM 94] [HWA 93].

Le réseau crossbar est considéré comme un réseau à un seul étage du fait qu'il est construit avec un seul commutateur au point de croisement.

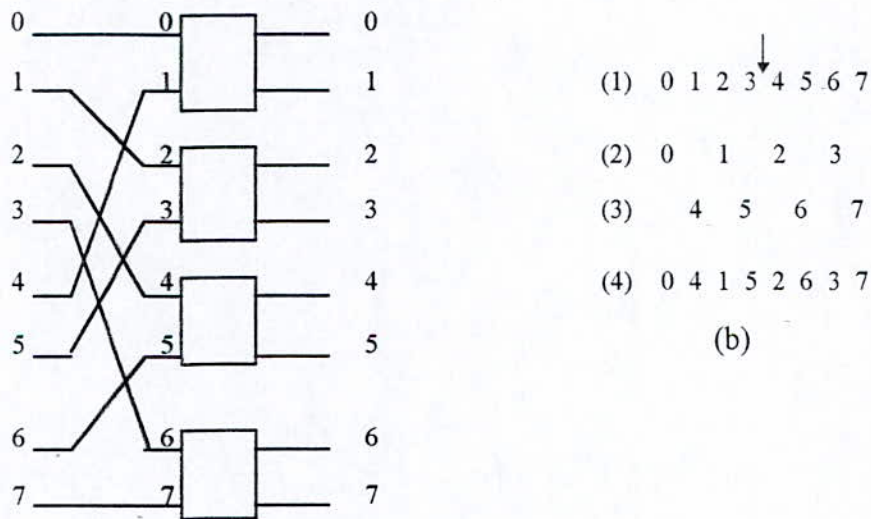
les réseaux crossbar (P, M) permettent de réaliser toutes les permutations des entrées. En d'autres termes, quelle que soit la permutation π de $\{0, 1, 2, \dots, p-1\}$, il existe un état du réseau qui permet simultanément d'associer à toute entrée E_i , la sortie $S_{\pi(i)}$. Un état du réseau correspond à une position donnée de chaque interrupteur, et donc, dans ce cas, la permutation π est réalisée en fermant tout i , l'interrupteur situé à l'intersection de la ligne i et de la colonne $\pi(i)$.

Il est difficile aujourd'hui de construire des réseaux crossbar avec plus de 32 entrées à cause de la complexité des coûts et des délais de réalisation. Aussi les constructeurs proposent-ils d'autres alternatives que sont les RIT et les RIP.

2.4 Les réseaux d'interconnexions à un seul étage

La figure 2.5 montre un RIT. Ce réseau à $N=8$ entrées et $N=8$ sorties. Le réseau consiste à un seul étage avec quatre commutateurs. La liaison entre ces commutateurs

détermine la topologie du réseau. Chaque commutateur a 2 entrées et 2 sorties (on dit que c'est un commutateur 2x2), qui peut être contrôlé pour établir une connexion interne en **direct** ou en **croisé**. L'entrée de ces commutateurs est désignée par S_0 à S_{N-1} . La figure 2.5 montre un modèle du **shuffle-exchange parfait** pour $n=3$ (avec $N=2^n$, où n représente le nombre de tours ou de passes). Cette configuration est obtenue de la façon suivante: l'entrée E_x doit être connectée à l'entrée $2*x$ modulo (2^n-1) du commutateur; c'est à dire l'entrée E_x est connectée à la sortie $S[2*x \text{ modulo } (2^n-1)]$. Une méthode pratique est souvent utilisée pour déterminer l'interconnectabilité du shuffle-exchange parfait (perfect shuffle-exchange), première étape, on numérote la séquence normale selon n (fig. 2.5 (b), ligne 1).



Fig; 2.5 a) Réseau à un seul étage type Shuffle-exchange parfait.

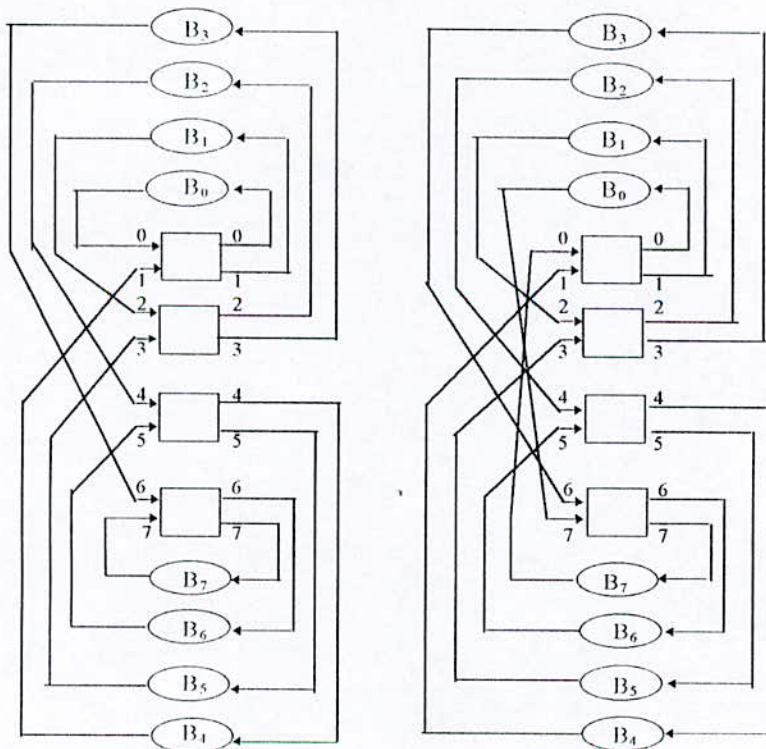
Cette séquence est divisée en deux moitiés égales, après quoi, la première moitié reportée sur la ligne 2, entre chaque numéro, on laisse un espace. La deuxième moitié est reportée sur la ligne 3 et enfin la ligne 4 montre la configuration shuffle exchange parfait en fusionnant la ligne 2 et 3 [HWA 93].

Dans un réseau à un seul étage, les sorties doivent obligatoirement être connectées aux mêmes numéros d'entrée via des mémoires tampons (B_i). Ceci est dû essentiellement au fait qu'une transmission de donnée à l'entrée du réseau peut ne pas être connectée à la sortie en un seul cycle, c'est à dire en un seul tour. Ces données peuvent recirculer à travers le réseau jusqu'à n tours (un tour dans un réseau à un seul étage est équivalent à un étage dans un

réseau à plusieurs étages). Par exemple, si l'entrée E_0 veut transmettre des données à la sortie S_2 (fig. 2.5.a), deux cycles sont nécessaires: dans le premier cycle, les données sont transmises de l'entrée E_0 à la sortie S_1 en positionnant le commutateur en **croisé**; dans le deuxième cycle, les données passent de l'entrée E_1 vers la sortie S_2 en mettant le commutateur dans la position **directe**. On trouve d'autres types de configuration donnés par Burke [BUR 91].

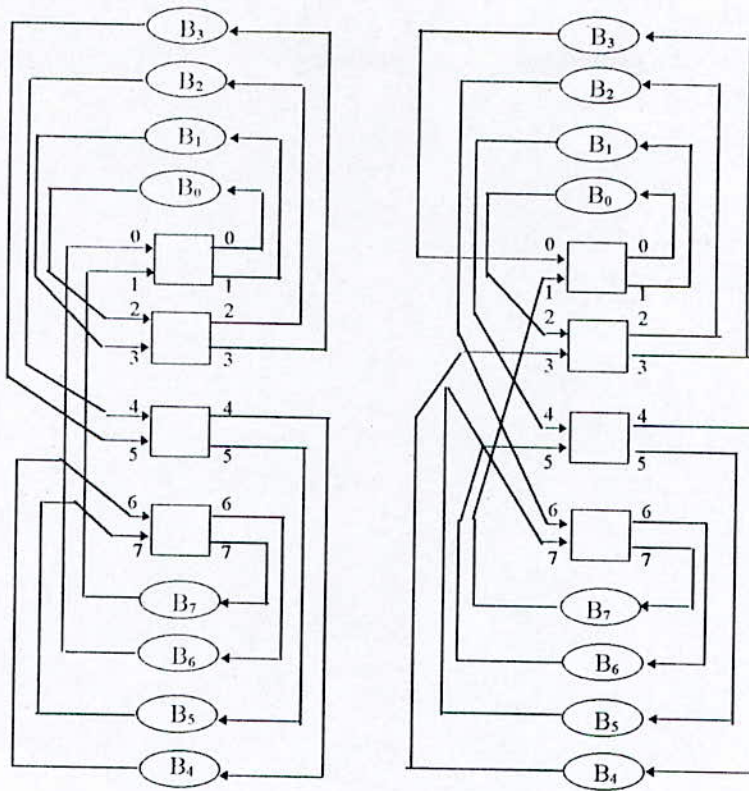
On distingue d'autres types de RIT; on énuméra quelques uns qui sont représentés dans la figure 2.6. Le principe de la structure du réseau reste la même; ce qui change c'est la liaison entre les entrées et les sorties qui définit généralement la topologie du réseau.

La complexité du RIT est de l'ordre de $O(N)$; par contre celui d'un RIP est $O(N \log_2 N)$ [HUA 87] ou N est le nombre d'entrée. On a observé que les RIT ont un chemin (path) moyen plus court. Ce chemin n'est autre que le nombre de tours utilisé dans le réseau pour établir une connexion entre la source et la destination. Dans un RIP, chaque requête doit faire $\log_2 N$ étages pour atteindre sa destination. Par contre dans un RIT, le nombre maximum de tour pour une quelconque connexion entre entrée et sortie est inférieur ou égale à $\log_2 N$. Ceci est un avantage considérable pour ce type de réseau.



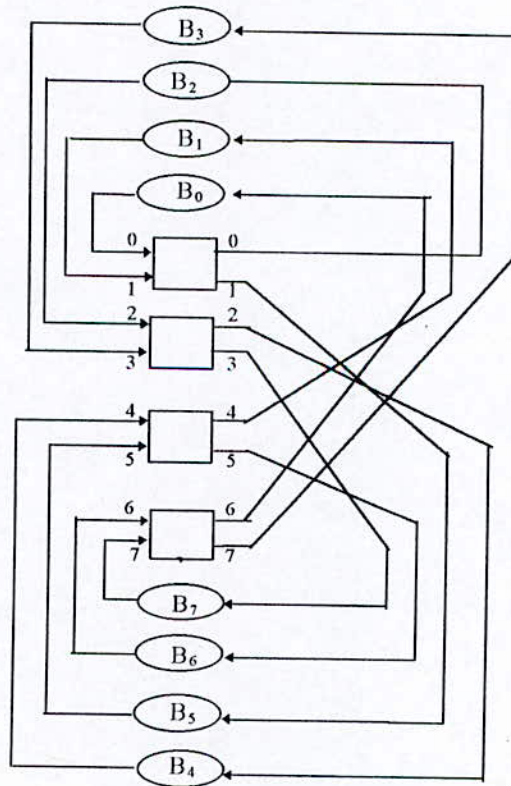
a) Réseau d'interconnexion à un seul étage topologie Shuffle-exchange

b) Topologie Shuffle-exchange modifié



c) Topologie Rot

d) Topologie Zeta



e) Topologie Beta

Fig. 2.6 Quelques exemples des réseaux d'interconnexions à un seul étage

2.5 Les réseaux d'interconnexions à plusieurs étages

Les RIP ont été bien définis par J. de Rumeur [RUM 94]; c'est une nouvelle génération de machines qui ont pris de l'essor ces dernières années. Ces réseaux présentent une connexion complète entre l'entrée et la sortie; ils sont facilement extensibles. Un réseau multiétage généralisé est présenté dans la figure 2.7. Un nombre de commutateur axb est utilisé dans chaque étage. Une connexion fixe entre étage adjacent est établie. Cette connexion interétage définit la topologie du réseau. Les commutateurs disposent de commandes qui permettent d'établir la connexion entre la source et la destination désirée.

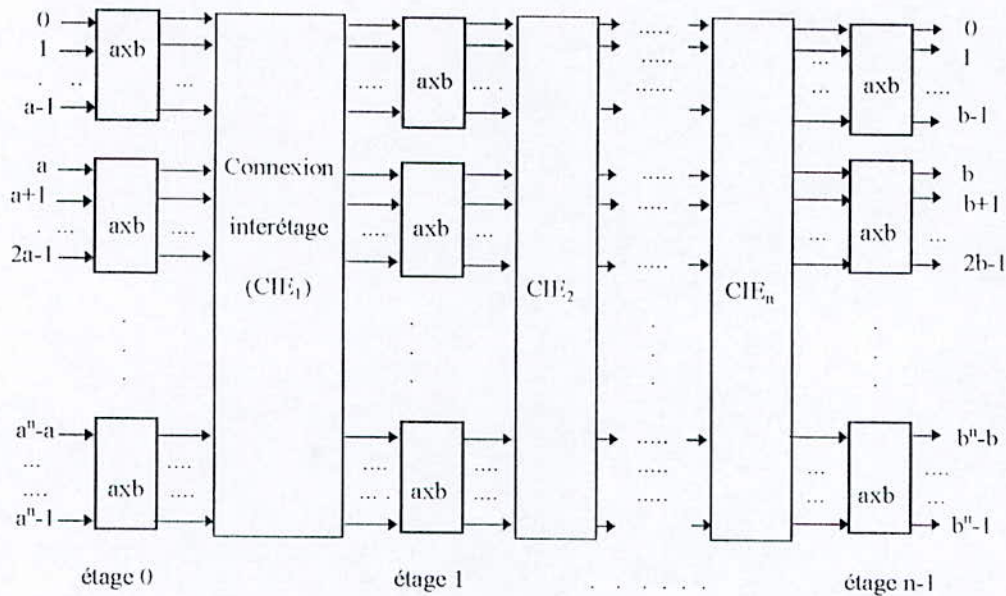


Fig. 2.7 Structure générale d'un réseau d'interconnexion multiétage avec un module de commutation axb et une connexion interétage $CIE_1, CIE_2, \dots, CIE_n$.

Il existe plusieurs types de RIP, ils diffèrent dans le choix du module de commutation et le choix de la connexion interétage (CIE). Le plus simple du module de commutation est le commutateur 2×2 , comme le montre la figure 2.5 (a). Nous allons présenter dans ce qui suit les CIEs avec quelques exemples les plus répandues.

2.5.1 Réseau Clos

Un réseau Clos, de paramètres a, b et r , noté $CL(a, b, r)$ est constitué de trois étages: le premier étage (étage 0) est constitué de r commutateurs identiques, dits (a, b) ,

chacun reliant **a** entrées vers **b** sorties (nous avons donc $r \cdot a$ entrées et $r \cdot b$ sorties au premier étage); le deuxième étage (étage 1) comporte b commutateurs (r, r) identiques (nous avons donc ici $b \cdot r$ entrées et autant de sorties); et le troisième étage (étage 2) est le symétrique du premier et consiste en r commutateurs (b, a) identiques. La figure 2.8 illustre ce réseau. Nous allons présenter maintenant quelques exemples de RIP les plus communs et qui sont tous à base de commutateurs (2×2) .

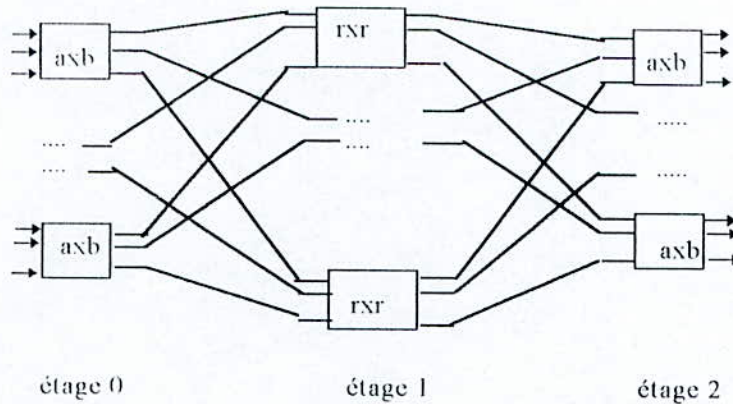


Fig. 2.8 Réseau Clos CL(a, b, r)

2.5.2 Réseau Butterfly

Le réseau multiétages Butterfly de dimension n est un réseau formé de $n+1$ étages ayant chacun 2^n commutateurs (2×2) . Il existe donc 2^{n+1} liens d'entrées et 2^{n+1} liens de sorties. Les étages sont généralement étiquetés par les entrées de 0 à n . La figure 2.9 représente un réseau Butterfly de dimension 2.

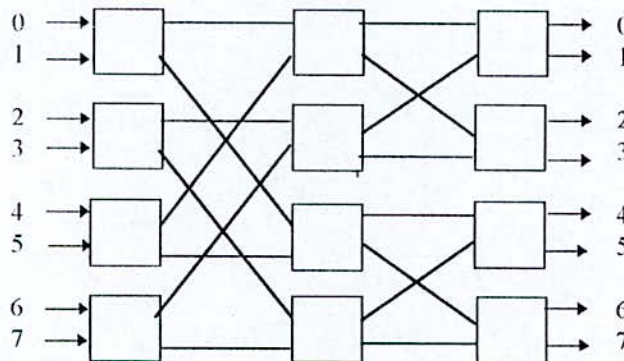


Fig. 2. 9 Réseau Butterfly de dimension 2

2.5.3 Réseau Benès

Un réseau Benès de dimension n , noté B_n , est constitué d'un réseau Butterfly de dimension n , suivi d'un réseau Butterfly de dimension n inversé (c'est à dire obtenu par une opération miroir), de sorte que l'étage n du premier soit confondu avec l'étage n du second (figure 2.10). Il a donc $2n+1$ étages de 2^n commutateurs (2×2) chacun.

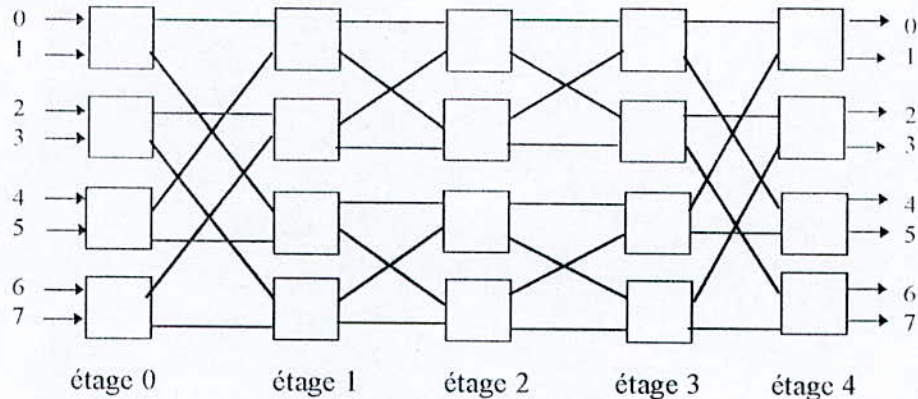


Fig. 2.10 Réseau Benès de dimension 2

2.5.4 Réseau Oméga et Baseline

Le réseau oméga de dimension n est formé de $n+1$ étages de 2^n commutateurs (2×2), les sorties de 2^{n-1} premiers commutateurs d'un étage quelconque sont reliées aux entrées hautes des commutateurs de l'étage suivant, dans le même ordre. La figure 2.11 représente un réseau oméga de dimension 2.

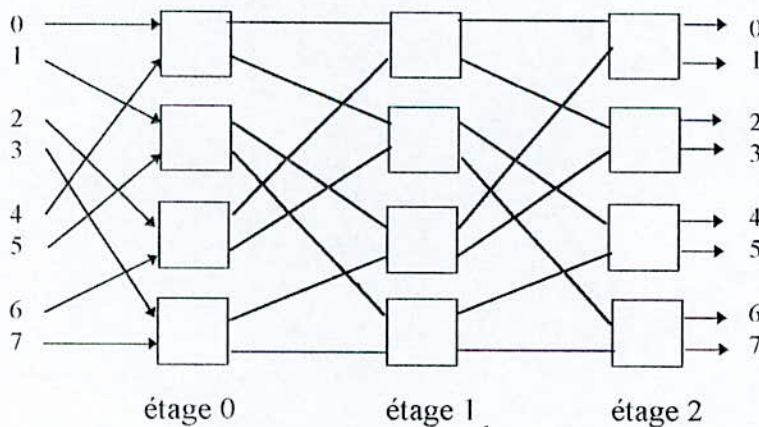


Fig. 2.11 Réseau oméga de dimension 2

Le réseau oméga est équivalent au réseau butterfly [RUM 94]. Il n'est pas le seul réseau équivalent topologiquement au réseau butterfly; on peut trouver le réseau baseline. Pour

l'équivalence entre réseaux il y a lieu de se reporter à la Thèse de J. M. Fourneau [FOR 85]. Le réseau baseline de dimension n est également formé de $n+1$ étages de 2^n commutateurs (2×2). Le premier étage est constitué d'un block de $2^n \times 2^n$, et le deuxième étage est constitué de deux sous-blocks de $2^{n-1} \times 2^{n-1}$ identifié par C_0 et C_1 . La construction de ce processus peut être récursive jusqu'au dernier block de 2×2 . La figure 2.13 montre un réseau baseline de dimension 2.

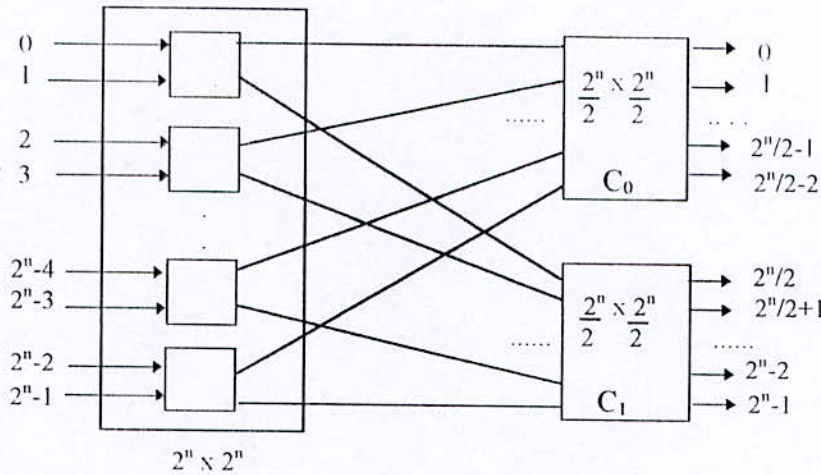


Fig. 2.12 Construction récursive

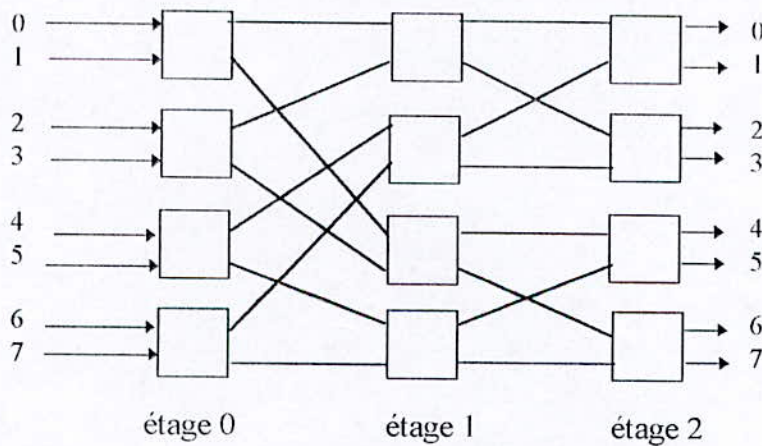


Fig. 2.13 Réseau baseline de dimension 2

2.6. Comparaison des performances des réseaux d'interconnexions

Par exemple, si on considère le réseau crossbar, on sait qu'il nécessite N^2 boîte de commutation (BC) pour pouvoir établir une connexion entre N -entrées/ N -sorties. Par contre, dans les RIP type baseline par exemple, cela nécessite $N/2(\text{Log}_2 N)$ BC [REW 97]. L'avantage majeur du crossbar est la vitesse de connexion. Dans un cycle de temps, une

connexion peut s'établir entre le processeur source et la mémoire destination. Par contre les RIP nécessitent Log_2N cycles pour établir une connexion. La figure 2.14 illustre les performances du réseau crossbar, RIP et la structure classique, liaison par bus. L'axe vertical représente la bande passante (BW) qui est un des paramètres qui permet de mesurer les performances d'un réseau. La BW représente le nombre d'accès de requête que le réseau satisfera par une unité de temps (qu'est le cycle dans notre cas). L'axe horizontal représente le taux d'arrivée des requêtes. La figure 2.14 montre que les performances des RIP sont très proches des performances du réseau crossbar pour une complexité matérielle moindre (nombre de boîte de commutation). C'est l'une des raisons fondamentales qui a laissé bon nombre de chercheur de s'intéresser aux RIP.

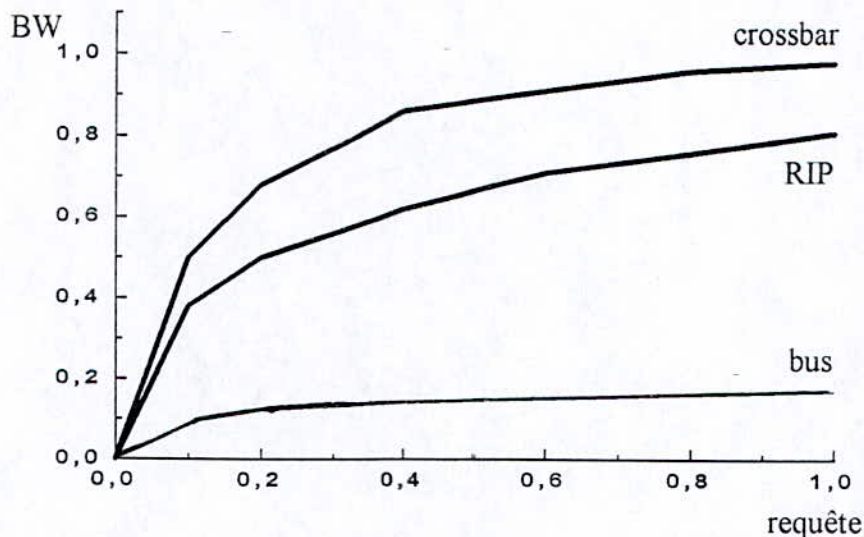



Fig. 2.14 Comparaison des performances des RI entre le crossbar, RIP et le bus.
[Hesham El-Rewini & Ted G. Lewis, 'Distributed and parallel computing', Ed. Manning, 1997, pp. 102].



**LES RESEAUX
DE PETRI**

CHAPITRE III

Les réseaux de Petri

Depuis longtemps déjà, l'homme a ressenti le besoin de comprendre le fonctionnement du monde qui l'entoure. C'est ainsi que très rapidement, il lui a fallu modéliser les phénomènes physiques afin d'en isoler les principaux ressorts. La modélisation lui a permis d'analyser des rapports de cause à effet, d'établir des lois de comportement aboutissant à une prédiction possible de ces phénomènes. Cependant, un dilemme subsiste entre l'élaboration d'un modèle trop simpliste autorisant une analyse facile de son comportement, mais un peu éloigné du comportement réel du système, et un modèle qui est très proche du système réel, mais dont l'étude est trop complexe, voire impossible.

C'est dans cet esprit que Carl Adam Petri a défini dans les années 60, un modèle consacré à l'étude des systèmes à événements discrets: les réseaux de Petri (RdP) [PET 62]. Ce modèle, initialement utilisé pour décrire le fonctionnement de systèmes informatiques, possède des propriétés structurelles que l'on peut facilement mettre en évidence et qui sont immédiatement transportables sur le système réel [BRA 83], [FLO 91], [MUR 89]. Très rapidement ce modèle a été doté d'extensions et a vu son domaine d'application s'étendre. En particulier, l'évaluation des performances des systèmes de production fait aujourd'hui largement appel aux RdP. Ces dernières années ont vu se réaliser des applications dans le domaine de l'informatique; c'est dans ce cadre que le présent travail a été effectué.

3.1 Introduction

Les réseaux de Petri sont des outils de modélisation des activités asynchrones ou synchrones, concurrentes et parallèles.

Les avantages principaux des RdP sont les suivants:

- (i) ils autorisent une représentation graphique relativement proche des systèmes réels
- (ii) leur dynamique s'exprime sous forme d'équations d'état simples.
- (iii) certains types de RdP possèdent des propriétés fortes (conservation de marquages, temps de séjour, ...) qui sont des propriétés interprétables en termes de blocage d'informations dans les réseaux de communications informatiques.

De plus, les moyens dont on dispose pour l'étude des systèmes discrets sont très peu

nombreux et sont parfois soumis à des hypothèses fortes (c'est le cas des files d'attente, par exemple) [FLO 90].

Dans cette partie, nous donnerons les définitions et propriétés de bases des réseaux de Petri en insistant plus particulièrement sur les réseaux de Petri stochastiques (RdPS) qui sont une extension des RdP. C'est avec les RdPS que nous avons étudié l'évaluation d'architecture des systèmes parallèles. Nous présenterons par la suite les motivations du choix de cet outil mathématique.

3.2 Réseaux de Petri : Définitions et Propriétés

3.2.1 Définitions de Base

Un réseau de Petri est un graphe biparti constitué de places, de transitions (correspondant aux sommets du graphe) et d'arcs qui relient les transitions aux places et les places aux transitions. Il est représenté par un quadruple $[P, T, I, O]$ ou:

- P désigne l'ensemble (fini) de places $P = \{P_1, P_2, \dots, P_n\}$
- T est l'ensemble (fini) de transitions $T = \{T_1, T_2, \dots, T_m\}$
- I est une application de $P \times T \rightarrow \mathbb{IN}$ (ensemble des entiers naturels) correspondant aux arcs directs des places vers les transitions.

$I(p, t) > 0$ signifie qu'il existe un arc orienté de p vers t et $I(p, t)$ est la valuation de cet arc.

$I(p, t) = 0$ indique l'absence d'arc orienté reliant la place p à la transition t .

- O est une application de $T \times P \rightarrow \mathbb{IN}$ correspondant aux arcs directs reliant les transitions aux places.

$O(t, p) > 0$ signifie qu'il existe un arc orienté de t vers p et $O(t, p)$ est la valuation de cet arc.

$O(t, p) = 0$ indique l'absence d'arc orienté reliant la transition t à la place p .

On notera t° (resp. ${}^\circ t$) l'ensemble des places de sorties (resp. d'entrée) de la transition t .

$${}^\circ t = \{ p \in P; I(p, t) > 0 \}$$

$$t^\circ = \{ p \in P; O(t, p) > 0 \}$$

De même p° (resp. ${}^\circ p$) désignera l'ensemble de transitions de sortie (resp. d'entrée) de la place p .

$${}^\circ p = \{ t \in T; O(t, p) > 0 \}$$

$$p^\circ = \{ t \in T; I(p, t) > 0 \}$$

3.2.1.1 Marquage

Les places peuvent contenir zéro ou plusieurs marques (ou jetons). Le marquage d'un RdP est une application M de P dans \mathbb{N} (ensemble des entiers naturels).

$$M : P \rightarrow \mathbb{N}$$

$M(p)$ désigne le nombre de marques (jetons) contenues dans la place p .

M peut être représenté par un vecteur colonne indicé par les numéros des places.

Un réseau de Petri marqué est un RdP noté R muni d'un marquage initial M_0 ; la figure 3.1 est un exemple de réseau de Petri marqué.

Nous avons pour l'exemple de la figure 3.1:

- $n = 4$ (nombre de places) $m = 5$ (nombre de transitions)
- $O(t_1, p_2) = 2$ $O(t_4, p_3) = 3$ $O(t_2, p_1) = O(t_3, p_3) = O(t_5, p_1) = O(t_5, p_4) = 1$
- $I(p_1, t_1) = I(p_2, t_3) = 1$ $I(p_2, t_4) = I(p_3, t_5) = I(p_4, t_2) = 2$
- et $I(p, t) = O(t, p) = 0$ pour tout autre $p \in P$ et $t \in T$
- $p_2^\circ = \{t_3, t_4\}$ ${}^\circ p_2 = \{t_1\}$ $t_5^\circ = \{p_1, p_4\}$ ${}^\circ t_5 = \{p_3\}$
- ${}^t M_0 = (1, 2, 0, 1)$

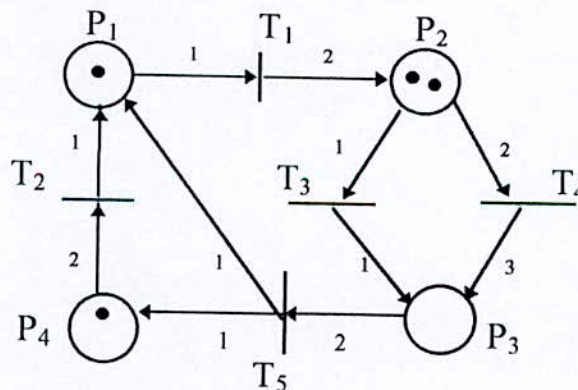


Fig. 3.1 Un exemple de RdP marqué

3.2.1.2 Règles de fonctionnement

Le passage d'un marquage à un autre s'effectue de façon instantanée par franchissement d'une ou plusieurs transitions. Une transition est dite tirable (ou franchissable) si chacune de ses places d'entrée contient un nombre de jetons au moins égal à la valuation de l'arc joignant cette place à la transition.

La transition t est franchissable si :

$$\forall p \in {}^{\circ}t \quad M(p) \geq I(p, t)$$

Revenons encore à la figure 3.1. La transition T_1 est franchissable car sa place d'entrée P_1 contient un nombre de jetons égal à la valuation de l'arc (p_1, t_1) ($I(p_1, t_1) = 1$).

Par contre, la transition t_2 n'est pas tirable puisque $I(p_4, t_2) = 2 > M(p_4) = 1$

Le tirage d'une transition t a pour conséquence:

(i) de retirer $I(p, t)$ jetons de chaque place d'entrée p de la transition t .

(ii) d'ajouter $O(t, p)$ jetons dans chaque place de sortie p de la transition t .

Le franchissement d'une transition t provoque alors le passage à un marquage M' tel que:

$$\forall p \in P \quad M'(p) = M(p) + O(t, p) - I(t, p)$$

Le marquage M' du RdP de la figure 3.1, obtenu après franchissement de la transition t_1 est:

$${}^tM' = (0, 4, 0, 1)$$

Une séquence de transitions $\sigma = (t_{i1}, t_{i2}, \dots, t_{is})$ est franchissable pour le marquage M_0 si $t_{ik}(k=1, 2, \dots, s)$ est franchissable par le marquage $M^{(k-1)}$ obtenu par le franchissement de $\sigma_{k-1} = (t_{i1}, t_{i2}, \dots, t_{i(k-1)})$. L'ensemble des marquages qu'il est possible d'atteindre partant du marquage M_0 en franchissant une séquence de transitions, sera noté $\mathcal{M}(R, M_0)$. Bien entendu, si $M \in \mathcal{M}(R, M_0)$, il existe au moins une séquence $\sigma = (t_{i1}, t_{i2}, \dots, t_{is})$ franchissable qui permet de passer du marquage M_0 au marquage M .

3.2.2 Analyse des RdP

3.2.2.1 Matrice d'incidence

La structure d'un RdP peut être définie par une matrice à éléments entiers, appelée matrice d'incidence. Pour un RdP avec n places et m transitions, la matrice d'incidence $W = (c_{ij})$ est une matrice de $(n \times m)$, définie par:

$$W(i, j) = w_{ij} = O(t_j, p_i) - I(p_i, t_j)$$

La matrice d'incidence du RdP de la figure 3.1 est donnée par:

$$W = \begin{bmatrix} -1 & 1 & 0 & 0 & 1 \\ 2 & 0 & -1 & -2 & 0 \\ 0 & 0 & 1 & 3 & -2 \\ 0 & -2 & 0 & 0 & 1 \end{bmatrix}$$

Nous définirons de même la matrice d'incidence avant W^+ (resp. arrière W^-) par:

$$W^+(i, j) = O(t_j, p_i)$$

$$W^-(i, j) = I(p_i, t_j)$$

D'où l'égalité matricielle suivante $W(i, j) = W^+(i, j) - W^-(i, j)$

pour l'exemple de la figure 3.1 nous avons:

$$W^+ = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad W^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Soit une séquence de transitions franchissables à partir d'un marquage initial M_0 . Le vecteur caractéristique de la séquence σ , noté $\underline{\sigma}$ est le vecteur colonne à coefficients entiers de dimension m , où $\underline{\sigma}_i$ ($i=1, 2, \dots, m$) est le nombre de fois que la transition t_i apparaît dans σ . Alors nous avons l'équation fondamentale suivante :

$$M_f = M_0 + W \cdot \underline{\sigma} \tag{3.1}$$

où M_0 est le marquage initial. M_f est le marquage obtenu à partir de M_0 après que la séquence σ ait été franchie, W étant la matrice d'incidence du RdP.

Considérons le RdP de la figure 3.1 avec le marquage initial $M_0 = (1, 2, 0, 1)$. La séquence $\sigma = (t_1, t_4, t_5, t_2, t_1, t_3)$ est une séquence franchissable à partir de M_0 et son vecteur caractéristique est donnée par $\underline{\sigma} = (2, 1, 1, 1, 1)$.

Appliquons l'équation fondamentale Eq.3.1. Il vient:

$$M_f = M_0 + W \cdot \underline{\sigma} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 0 & 1 \\ 2 & 0 & -1 & -2 & 0 \\ 0 & 0 & 1 & 3 & -2 \\ 0 & -2 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 0 \end{bmatrix}$$

Remarque:

L'équation fondamentale Eq.3.1 donne le marquage final connaissant le marquage initial et la séquence des transitions franchies. Mais elle ne permet pas de vérifier si le franchissement d'une séquence donnée est possible.

3.2.2.2 Temporisation

Aucune durée n'est liée au franchissement des transitions ni au séjour dans les places pour ce qui concerne les RdP que nous avons présentés précédemment. Il existe plusieurs extensions possibles des RdP. Parmi ces extensions nous pouvons citer les RdP temporisés et les RdP stochastiques. Cette classe appartient à la classe des RdP non autonomes (la classe des RdP permettant de lier l'évolution d'un réseau à celui d'un environnement).

Dans les RdP temporisés et les RdP stochastiques, on associe aux places ou aux transitions des durées de séjour ou de franchissement: déterministes dans le premier cas, aléatoires dans le second.

Dans notre approche, nous avons utilisé les RdP stochastiques comme outil mathématique. Nous allons, tout d'abord, préciser les motivations qui nous ont amené à retenir cet outil comme moyen d'évaluation des performances dans les systèmes (performances fonctionnelles et/ou de sûreté de fonctionnement). Nous rappellerons ensuite quelques définitions et propriétés relatives à une classe particulière de processus stochastiques à savoir les chaînes de Markov, qui sont le substrat fondamental des analyses faites avec les RdPS.

3.3 Les Réseaux de Petri Stochastiques

3.3.1 Pourquoi les RdPS ?

Les premières études d'évaluation de performances de systèmes informatiques ont été essentiellement basées sur des techniques qui font appel à des méthodes de simulation ou à des méthodes d'analyse basée sur la théorie des files d'attente.

Les méthodes de simulation permettent de présenter des systèmes à configurations réalistes. Cependant, d'une part, la convergence de ce type de méthodes est souvent assez coûteuse et assez aléatoire, et d'autre part, la validation du modèle est assez difficile.

Les réseaux de files d'attente permettent de modéliser uniquement des comportements de pur parallélisme. Un client arrive dans la queue, attend un service, reçoit le service et entre dans une autre queue ou quitte le réseau de files d'attente. Donc à un moment donné, plusieurs clients sont servis en parallèle dans plusieurs queues. Ce modèle est très bien adapté pour la modélisation du comportement asynchrone des clients; cependant le seul type d'interaction pouvant être décrit entre ces comportements

correspond au contentieux à l'accès au service [FLO 91].

Dans le domaine de la sûreté de fonctionnement, les processus stochastiques et particulièrement les chaînes de Markov sont les principaux outils de modélisation. Et comme tout graphe d'états peut être utilisé pour représenter le comportement du modèle, les chaînes de Markov permettent en théorie de décrire de très nombreuses architectures de systèmes, ceci étant dû à l'absence de restrictions sur les schémas de synchronisation mis en oeuvre. Malheureusement, le processus de modélisation implique l'énumération de tous les états possibles et de toutes les possibilités de transitions entre états; ceci est d'autant plus pénalisant, que même si cette fastidieuse étape combinatoire est rendue possible, il est très difficile d'éviter, en particulier les erreurs d'omission.

Il apparaît donc, qu'un outil d'un niveau d'abstraction acceptable est nécessaire. Un tel outil doit combiner, d'une part, un niveau d'expression qualitatif assez élevé du parallélisme, de la compétition et de la synchronisation et, d'autre part, doit être capable de représenter fidèlement le processus stochastique. Ces deux aspects permettent à la fois de faire une validation qualitative du modèle et une évaluation quantitative des performances et des critères de sûreté de fonctionnement (fiabilité, sécurité, disponibilité, ...).

3.3.2 Comment définir les réseaux de Petri stochastiques

Une synthèse de la définition idéale des RdPS a été présentée par Florin [FLO 90]. Celle-ci doit tenir compte des principes suivants:

- Le point essentiel concerne la capacité du modèle à représenter n'importe quel ensemble d'activités parallèles à ressource partagées et, plus généralement, soumises à tout schéma de synchronisation possible.
- La durée d'une opération étant une variable aléatoire, sa distribution ne doit pas être restreinte à un type particulier, et doit pouvoir dépendre du contexte courant d'exécution.

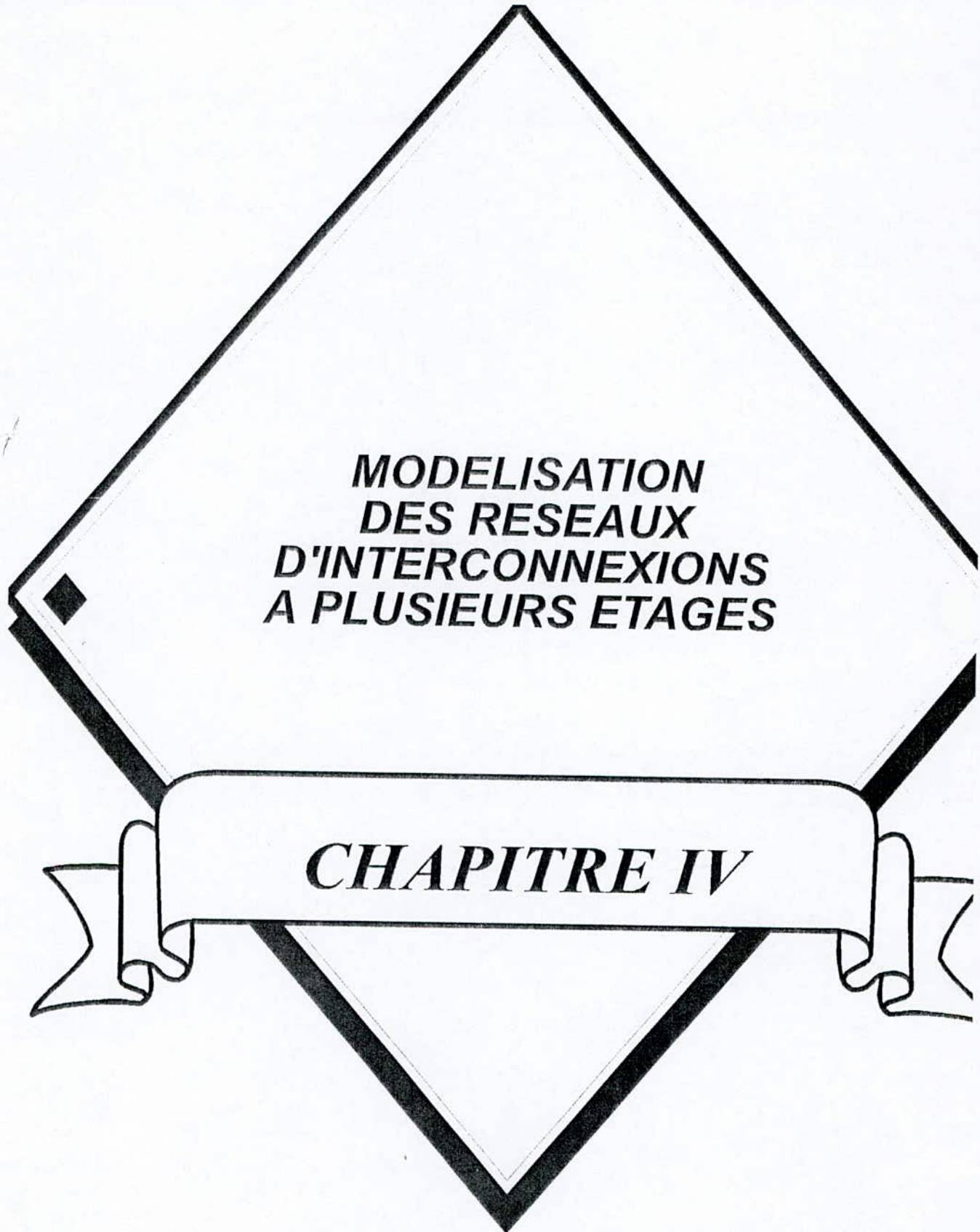
Dans notre approche, nous avons choisi que la variable aléatoire suive une loi de distribution exponentielle. Ce choix s'explique par le fait de travailler dans l'intervalle de temps ou l'état 'future' dépend de l'état 'présent' du système (i.e. un système sans mémoire). L'analyse de ce type de modèle passe par l'étude d'une chaîne de Markov continue. En effet, il a été montré que le processus stochastique sous-jacent au RdPS est une chaîne de Markov continue [FLO 91].

- Chaîne de Markov [FLO 91]

Un processus de Markov est un processus stochastique $X(t)$ tel que, étant donné trois instants t, u, s , où $t > u > s$, la valeur de $X(t)$ lorsque $X(u)$ est connue ne dépend que de $X(u)$ et pas de $X(s)$. Autrement dit, les états futurs d'un processus de Markov ne dépendent que de l'état présent et non de la trajectoire ayant permis d'atteindre cet état:

$$P[X(t) = E_i / X(u), X(s)] = P[X(t) = E_i / X(u)]$$

Nous reviendrons avec plus de détail sur l'ensemble des probabilités du processus stochastique quand nous développerons les modèles dans le chapitre suivant, particulièrement lorsque nous poserons les hypothèses qui seront évidemment fortes pour pouvoir appliquer le processus de Markov.



**MODELISATION
DES RESEAUX
D'INTERCONNEXIONS
A PLUSIEURS ETAGES**

CHAPITRE IV

Modélisation des réseaux d'interconnexions à plusieurs étages

4.1 Introduction

Dans le cadre des travaux de cette thèse, nous avons développé deux modèles de réseaux d'interconnexions dynamiques [BEN 94], [BEN 96]. Généralement dans les réseaux d'interconnexions dynamiques, on utilise des commutateurs ou boîte de commutation (BC) qui sont l'élément de base. Il existe plusieurs types de BC. Dans notre approche nous avons considéré la BC de 2×2 , c'est à dire 2 entrées / 2 sorties, tel que le montre la figure 4.1. Ce type de BC est utilisé aussi bien dans les RIP que dans celui des RIT. Nous commencerons d'abord par donner un ensemble de définitions standard qui régissent les deux types de RI, ainsi que quelques hypothèses considérées dans notre approche. Ces hypothèses sont utilisées par la majeure partie des chercheurs; nous présenterons également la généralisation des modèles des RI dynamiques. Enfin, dans le chapitre V, nous exposerons l'application pour un réseau à 3 étages [BEN 95].

4.2 Modélisation d'un réseau d'interconnexions à plusieurs étages

Dans un RIP, on utilise des boîtes de commutation de 2×2 pour lier les entrées aux sorties. Nous supposons que les entrées sont des processeurs et que les sorties sont des mémoires. Un RI est un réseau capable de connecter n'importe quelle entrée à n'importe quelle sortie.

Un RIP est un réseau qui possède un chemin unique entre l'entrée et la sortie. Ceci nous amène à dire que le chemin entre les BC du réseau forme un arbre de lien. Un RI à n -étages est un réseau dont les BC peuvent être arrangé en étage. Les processeurs sont liés à l'entrée de l'étage 1 du réseau, toutes les sorties de l'étage i sont connectées à l'entrée de l'étage $i+1$, et toutes les mémoires sont connectées à la sortie de l'étage n . Nous pouvons étendre cette définition, et dire que les processeurs sont à l'étage 0 et les mémoires sont à l'étage $n+1$. Un réseau uniforme est un réseau où les BC du même étage ont le même nombre de liens d'entrée et le même nombre de liens de sortie. Un réseau carré de degré m est un réseau avec $m \times m$ BC. L'interconnectibilité entre étage détermine la topologie du réseau. Chaque étage est connecté à l'étage suivant par au moins N chemins avec ($N = m^n$).

Une classe de RI [Wu 92] est conçue pour connecter $N=m^n$ entrées à N sorties. Ce réseau contient n étages de N/m BC de dimension $m \times m$. Les BC des étages adjacents sont connectées de façon à pouvoir lier les entrées et sorties du réseau. La figure 2.13 montre un RIP avec $N=2^3=8$ entrées ($m=2$ et $n=3$), $N=8$ sorties, et trois étages de $8/2=4$ BC de 2×2 chacun. Ce réseau est appelé réseau baseline de 8×8 .

Dans le cas où nous considérons une BC de 2×2 , une requête peut être connectée de trois manières différentes telles que le montre la figure 4.1.

Dans l'état de la figure 4.1 (a), l'entrée basse de la BC est disponible c'est-à-dire qu'il n'y a aucune requête qui occupe cette entrée. Pendant cette phase, s'il y a une arrivée de requête dans l'autre entrée, cette requête aura une probabilité de passage égale à 1.

Dans l'état de la figure 4.1 (b), nous supposons l'entrée basse de la BC occupée par une commutation, c'est-à-dire qu'il existe une connexion entre l'entrée et la sortie de la BC; pendant cette phase, s'il y a une arrivée d'une requête à l'autre entrée, elle aura une probabilité de passage égale à $1/2$.

Dans l'état de la figure 4.1 (c), nous supposons que les deux entrées de la BC sont disponibles. Nous considérons l'arrivée de deux requêtes dans le même cycle aux deux entrées. Si les deux requêtes demandent la même sortie de la BC, alors une seule requête va accéder à cette sortie, l'autre sera bloquée. Le choix entre les deux requêtes en conflit sera aléatoire et dépendra de la priorité de passage donnée par la BC. La probabilité de passage d'une requête dans l'état (c) sera égale à $3/4$. Dans la modélisation des RIP, nous avons déjà considéré cette forme d'hypothèse qui gère la BC de 2×2 . Nous pouvons avoir d'autres formes d'hypothèses concernant la gestion de la BC. Une autre forme d'hypothèse est présentée et appliquée dans la modélisation des RIT dans le chapitre VI.

Grâce à cette première forme d'hypothèse qui gère la BC, nous pouvons dire quand un processeur à l'entrée du réseau génère une requête; cette requête établira un lien entre le processeur demandeur et l'entrée de la BC du premier étage. Cette requête se connectera à l'étage 1 du réseau avec une probabilité de passage égale à 1, 0.5 ou 0.75; cela dépendra de l'état de la BC.

Les requêtes qui passeront l'étage 1, vont progresser vers l'étage 2 du réseau de la même manière et ainsi de suite. Quand une requête passera l'étage n ($=\text{Log}_2 N$; où N est le nombre d'entrées ou de sorties du réseau) nous dirons qu'un chemin est établi entre le processeur

(entrée du réseau) et la mémoire (sortie du réseau). Une requête peut passer un étage dans 1 cycle de temps, donc une connexion d'une requête au sein du réseau nécessite au moins n cycles (n étant le nombre d'étages du réseau).

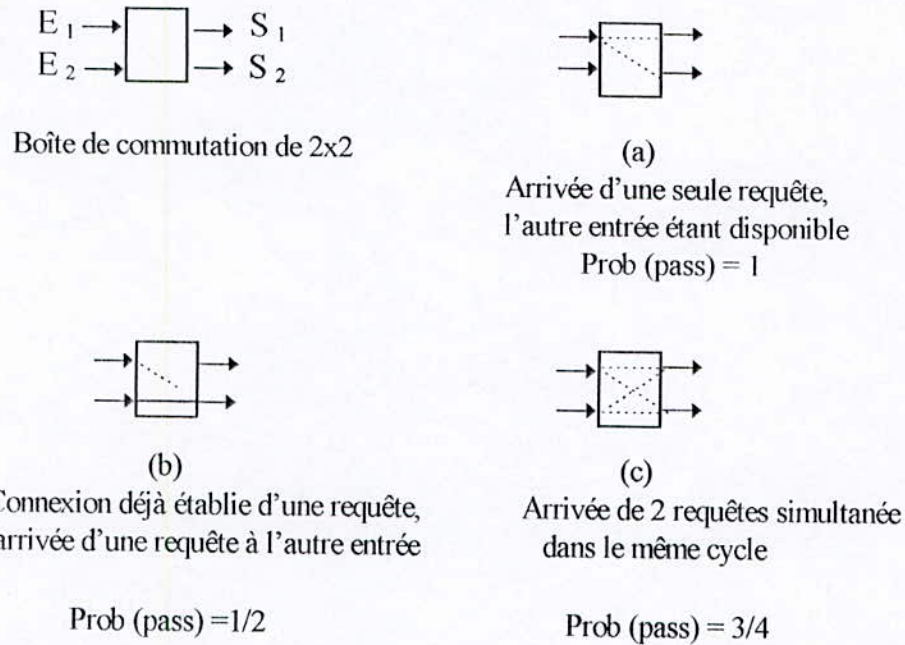


Fig. 4.1 Boîte de commutation de 2x2 et les probabilités d'établir une connexion

Avant de présenter le modèle, nous donnerons les hypothèses considérées, qui sont généralement prises par l'ensemble des chercheurs [DIN 91], [WU 92], [YOO 90]:

• **Hypothèses du modèle**

- a - Le franchissement d'une transition suit une loi de distribution exponentielle.
- b - L'état d'une transition prend un cycle de temps.
- c - Quand un chemin est établi entre l'entrée et la sortie du réseau, ce chemin est maintenu pour transférer les données pendant un nombre de cycles fixe. Le temps de transfert de données est noté ' d '. Les requêtes arrivent au réseau avec une probabilité ' r '. Du moment qu'on utilise le cycle comme unité de temps, donc le taux et la probabilité d'arrivée de requête sont identiques.
- d - Le nombre maximum d'étages dans le cas de RIP (ou tour dans le cas de RIT) dans le réseau est n .
- e - Si deux requêtes arrivent simultanément aux entrées d'une même BC et demandent l'accès à une même sortie, la commutation se fera pour une requête, l'autre requête régénèrera

une nouvelle demande d'accès au prochain cycle. Le choix de passage de la requête dépendra de la priorité donnée par la BC.

f - La probabilité de destination des requêtes est identique à travers les N sorties du réseau.

L'hypothèse (c) est utilisée pour faire la différence entre la distribution du temps de service et la distribution des requêtes générées au réseau. L'hypothèse (f) est généralement acceptée dans la plus part des modèles pour simplifier la construction du modèle. L'hypothèse (e) est reliée à la conception de la BC; elle est aussi généralement adoptée dans l'évaluation des performances.

4.2.1 Modèle 1 sans mémoire

Dans le modèle de base, nous nous proposons d'adopter la stratégie sans mémoire. Cette stratégie est basée sur l'hypothèse (e), où deux requêtes demandent la même sortie de la BC dans le même cycle. Cela veut dire qu'une requête va être connectée, l'autre sera bloquée et devra régénérer une nouvelle demande d'accès au prochain cycle. L'idée de cette stratégie est que, d'une part, cela permet de réduire le trafic des requêtes dans le réseau, et d'autre part, elle permet d'augmenter la probabilité de passage dans les étages près de la sortie du réseau.

Dans l'article de référence [WU 92], on prend en considération les probabilités de passages ou de blocages des requêtes. Cet article suppose qu'il n'y a aucune défaillance de la BC, c'est à dire qu'on dispose d'un réseau idéal. Afin que l'étude soit plus réaliste, nous avons pris en considération une probabilité de défaillance de la BC. Ainsi nous aurons une approche d'évaluation des performances des RI plus objective.

Dans notre approche, nous avons ainsi considéré les mêmes hypothèses pour des raisons pratiques pour valider la démarche et les résultats ainsi obtenus. Par la suite, nous avons pris en compte la probabilité de défaillance de la BC. Cette probabilité suit une loi de distribution exponentielle. Cet ajout de probabilité nous donne une approche réelle du modèle. Cela nous permettra d'avoir une meilleure appréciation d'évaluation des performances des RI. Le modèle de base présenté dans la figure 4.2, montre que l'étage 1 est entièrement disponible pour tout arrivée de requête. Nous proposerons un deuxième modèle dans la section suivante pour prendre en considération la non-disponibilité du premier étage pour des raisons déjà invoquées.

4.2.1.1 Description du modèle 1 sans mémoire

La figure 4.2 montre le cas général d'une modélisation sans mémoire d'un RIP [BEN 98].

La place S_0 est identifiée comme générateur de requête du réseau, sachant que la requête précédente générée par le processeur source est complète, et il n'y a pas d'arrivée de nouvelle requête. Les requêtes arrivent au réseau avec une probabilité r .

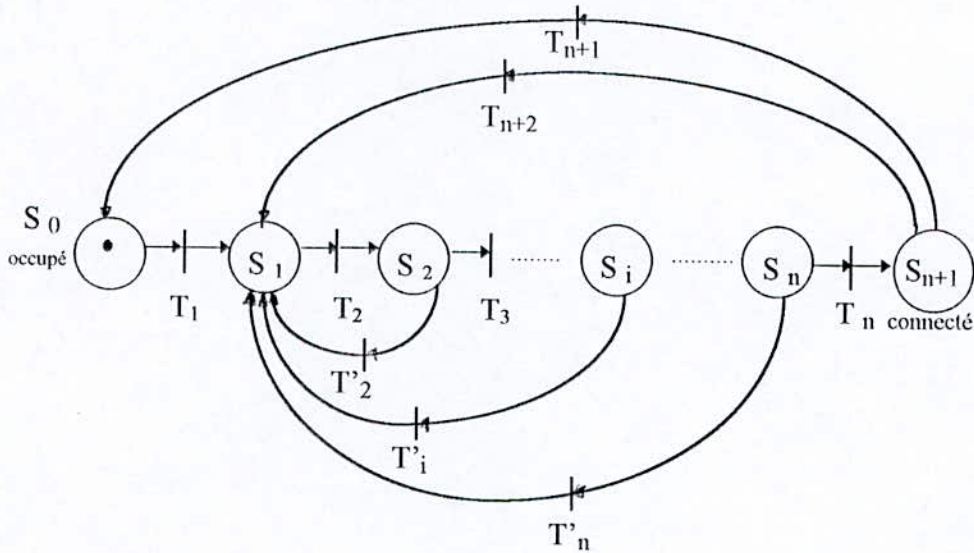


Fig. 4.2 Modèle 1 sans mémoire d'un RIP par les RdPS

Les places de S_1 à S_n sont identifiées comme les états du réseau à partir de l'étage 1 jusqu'à l'étage n respectivement. Dans ces états, la requête traverse le réseau à partir du premier étage, vers le deuxième étage en validant la transition T_2 . Si cet étage est bloqué, la requête reviendra au premier étage en validant la transition T'_2 , et essaiera de se reconnecter de nouveau. Dans le cas où elle n'est pas bloquée, elle poursuivra son chemin vers l'étage 3, et ainsi de suite jusqu'à l'étage n .

La place S_{n+1} est utilisée pour informer le processeur source qu'un chemin est établi entre l'entrée et la sortie du réseau, c'est à dire qu'une liaison est faite entre le processeur demandeur et la mémoire demandée. Dans cet état, la requête a établi un chemin pour transférer les données, et ce chemin est maintenu pendant d cycles. Le processeur source sera informé de la connexion par la validation de la transition T_{n+1} .

Dans cette thèse, nous utiliserons la convention suivante pour désigner les probabilités des états. La probabilité de la transition de l'état S_1 à l'état S_2 est écrite comme suit: $P(S_2/S_1)$ ou

encore $P(2/1)$. La probabilité absolue qu'une requête est dans l'état S_1 est écrite $P(S_1)$ ou P_1 .

Dans la stratégie sans mémoire, quand une requête est bloquée, elle est automatiquement écartée du réseau et sera réinjectée de nouveau pour demander accès. Nous exprimerons l'ensemble des probabilités avec les trois paramètres suivants: la dimension du réseau (n), la probabilité d'arrivée des requêtes (r) et le temps de transfert des données (d). Nous pourrions déduire quelques probabilités apparentes;

$P(S_1/S_0) = r$; probabilité d'arrivée des requêtes au réseau.

$P(S_2/S_1) = \lambda_1$; probabilité de passage de la requête de la place S_1 à la place S_2 .

$P(S_1/S_2) = \mu_1$; probabilité de blocage de la requête entre la place S_1 et la place S_2 .

$P(S_0/S_{n+1}) = (1-r)/d$; Quand une requête est dans l'état de connexion; cette requête sera complète avec une probabilité de $1/d$. Le nouvel état du réseau sera aussi bien dans l'état occupé avec la probabilité $(1-r)$ que dans celui où on aura une arrivée d'une nouvelle requête à l'étage 1 avec une probabilité r . Pour $r=0$, cela veut dire qu'il n'y a pas de demande d'accès au réseau, donc il n'y a pas de commutation. Les probabilités des états sont nulles.

4.2.2 Modèle 2 sans mémoire

Nous adoptons la même démarche avec les hypothèses du modèle 1. La seule différence entre les deux modèles est que le modèle 1 considère une disponibilité totale du premier étage, alors que dans le modèle 2, nous considérons que l'étage 1 peut ne pas être disponible pour recevoir une requête [BEN 98]. Donc, quand un processeur à l'entrée du réseau génère une requête, cette requête arrive à l'étage 1 avec une probabilité r , mais si l'étage 1 n'est pas disponible à recevoir cette requête, il faut informer le processeur demandeur de la non-réceptivité de cette requête, pour pouvoir régénérer une nouvelle demande. Ce retour est validé par la transition T'_1 de la figure 4.3 qui représente le modèle 2 sans mémoire. La description de ce modèle reste la même que précédemment.

4.2.3 Modèle avec mémoire

Dans ce modèle, nous adoptons la stratégie avec mémoire. Dans ce cas nous modifierons l'hypothèse (e) comme suit:

Hypothèse (e) : Si deux requêtes arrivent dans le même cycle à l'entrée de la BC et demandent la même sortie: l'une sera commutée vers la sortie désirée, quant à l'autre, elle sera emmagasinée dans une mémoire.

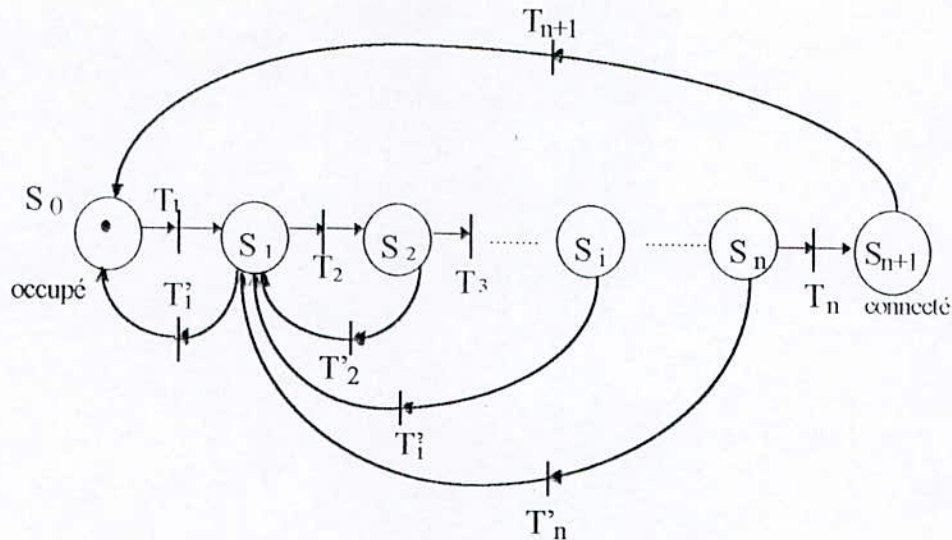


Fig. 4.3 Modèle 2 sans mémoire d'un RIP par les RdPS

Cette mémoire peut prendre un seul message à la fois. De cette manière, si une requête est bloquée, nous conserverons une partie du chemin préétabli dans le réseau. Quand la BC sera libérée, la requête bloquée emmagasinée dans la mémoire pourra continuer son chemin sans toute fois recommencer à zéro la connexion. La figure 4.4 montre le cas général d'une modélisation avec mémoire d'un RIP par les RdPS.

4.2.3.1 Description du modèle avec mémoire

La place S_0 est identifiée comme générateur de requête du réseau, sachant que la requête précédente générée par le processeur source est complète, et il n'y a pas d'arrivée de nouvelle requête. Les requêtes arrivent au réseau avec une probabilité r .

Les places de S_1 à S_n sont identifiées comme les états du réseau à partir de l'étage 1 jusqu'à l'étage n respectivement. Dans ces états, la requête traverse le réseau à partir du premier étage, vers le deuxième étage en validant la transition T_2 . Si le deuxième étage est bloqué, la requête passera à la place S_{b1} en validant la transition T_{b1} . La requête bloquée restera ainsi dans la mémoire de la longueur du message le temps qu'il faudra jusqu'au déblocage de la place S_2 . Quand la place S_2 est libérée, la requête qui était dans la place S_{b1} passera à la place S_2 par la validation de la transition T'_1 . De cette manière, le chemin préétabli par la requête bloquée sera maintenu. Cette requête continuera son chemin de la même

manière, jusqu'à l'établissement de la connexion.

La place S_{n+1} est utilisée pour informer le processeur source qu'un chemin est établi entre l'entrée et la sortie du réseau, c'est à dire qu'une liaison est faite entre le processeur demandeur et la mémoire demandée. Dans cet état, la requête a établi un chemin pour transférer les données, et ce chemin est maintenu pendant d cycles. Le processeur source sera informé de la connexion par la validation de la transition T_{n+1} .

De la même manière que précédemment, nous pourrions déduire quelques probabilités comme suit:

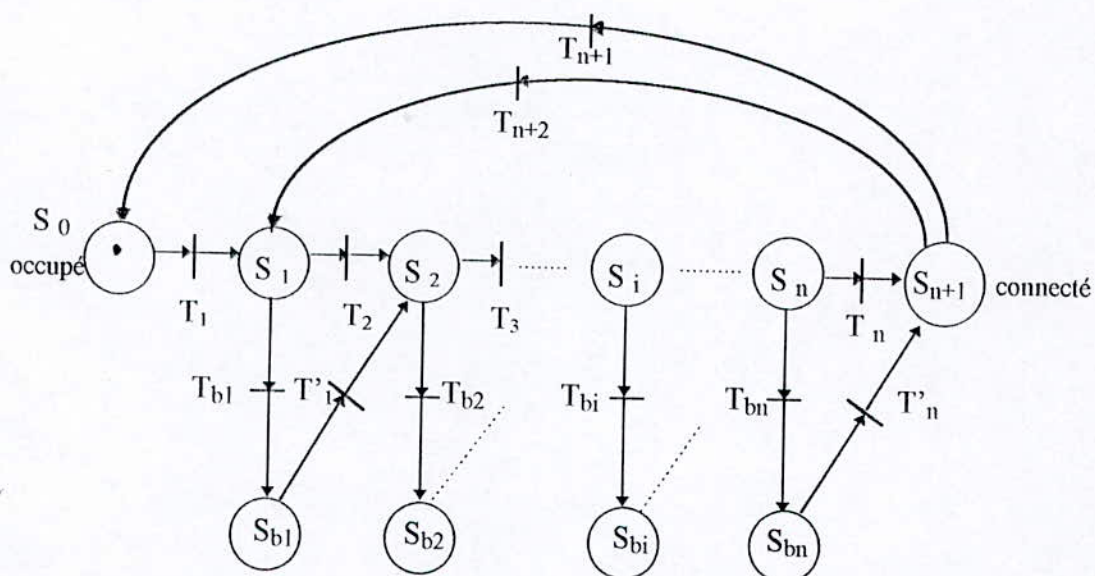


Fig. 4. 4 Modèle général avec mémoire d'un RIP par les RdPS

$P(S_1/S_0) = r$; probabilité d'arrivée des requêtes au réseau.

$P(S_2/S_1) = \lambda_1$; probabilité de passage de la requête de la place S_1 à la place S_2 .

$P(S_{b1}/S_1) = \mu_1$; probabilité de blocage de la requête entre la place S_1 et la place S_2 . La requête sera emmagasinée momentanément dans la place S_{b1} .

$P(S_2/S_{b1}) = \sigma_1$; probabilité de passage après blocage de la requête entre la place S_1 et la place S_2 , appelée aussi probabilité de déblocage ou de repassage.

$P(S_0/S_{n+1}) = (1-r)/d$; Quand une requête est dans l'état de connexion, cette requête sera complète avec une probabilité de $1/d$. Le nouvel état du réseau sera aussi bien dans l'état occupé avec la probabilité $(1-r)$ que dans celui où on aura une arrivée d'une nouvelle requête à

l'étage 1 avec une probabilité r .

Comme nous l'avons défini précédemment, l'analyse d'un RdPS se ramène à celle de processus de Markov homogène à espace d'état discret à temps continu. Donnons d'abord quelques relations qui découlent des définitions introduites par Florin [FLO 91].

4.3 Processus de Markov

Soit M un marquage accessible pour un RdPS:

$\lambda_i(M)$ ou λ_i est la probabilité de franchissement associée à la transition T_i pour le marquage M . Si cette transition est q validée pour le marquage M , alors;

$$\lambda_i(M) = q \cdot \lambda_i$$

Soit $T(M)$, l'ensemble des transitions validées par le marquage M . Si $T_k \in T(M)$, alors la probabilité conditionnelle de franchissement de cette transition à partir de M est:

$$P_r(T_k/M) = \lambda_k(M) / \sum \lambda_i(M); \quad i \text{ telle que } T_i \in T(M)$$

et la durée de séjour dans l'état M suit une loi exponentielle de paramètre;

$$\lambda(M) = \sum \lambda_i(M); \quad i \text{ telle que } T_i \in T(M)$$

cela donne un temps de séjour moyen dans le marquage M égal à $1/\lambda(M)$.

4.3.1 Propriété de conservation

Soit M_0 le marquage initial et $N(i)$ le nombre aléatoire de franchissement de la transition T_i dans l'intervalle $[0, t[$. Le vecteur $N(t)$ dont la $i^{\text{ème}}$ composante est $N_i(t)$ est appelé vecteur aléatoire de franchissement.

A tout instant t , le marquage $M(t)$ et le franchissement $N(t)$ sont reliés par la relation temporelle de franchissement:

$$M(t) = M_0 + W \cdot N(t)$$

où W désigne la matrice d'incidence.

Si l'on appelle $E(x)$ l'espérance mathématique sur l'espace de probabilité considéré:

$$E\{M(t)\} = E(M_0) + W \cdot E\{N(t)\}$$

$E\{M(t)\}$ est le marquage moyen du RdPS à l'instant t .

4.3.2 Conservation du flux en régime stationnaire

Nous supposons que le processus de marquage et de franchissement sont ergodique.

Nous notons F^* le vecteur de fréquences moyennes de franchissement des transitions en régime stationnaire. Ce vecteur vérifie la relation :

$$W \cdot F^* = 0 \quad (4.1)$$

C'est l'équation de la conservation de flux dans un RdPS. Elle exprime l'égalité du flux d'entrée et du flux de sortie pour toute place du réseau. En effet l'Eq. 4.1 s'écrit:

$$W^+ \cdot F^* = W^- \cdot F^*$$

Soit encore;

$$(W^+)_i \cdot F^* = (W^-)_i \cdot F^*$$

Or $(W^+)_i \cdot F^*$ est le flux entrant de marques dans la place P_i et

$(W^-)_i \cdot F^*$ est le flux sortant de marques dans la place P_i

Pour calculer les composantes de ce vecteur, il faut résoudre la chaîne de Markov associée au RdPS.

4.3.3 Formule de Little pour les RdPS

Avec les mêmes hypothèses d'ergodicité pour les processus de marquage $M(t)$ et de franchissement $F(t)$; la formule de Little donne le temps moyen de séjour d'une marque dans une place P_i noté $D^*(P_i)$:

$$D^*(P_i) = M^*(P_i) / (W_i^+ \cdot F^*) \quad (4.2)$$

ou W_i^+ est la $i^{\text{ème}}$ ligne de la matrice W^+

$M^*(P_i)$ est la marque moyenne en régime stationnaire de la place P_i

$W_i^+ \cdot F^*$ est la somme des fréquences réelles de franchissement des transitions antécédentes de la place P_i

Dans la suite, nous allons rappeler quelques définitions relatives aux processus markovien.

4.3.4 Chaîne de Markov

Nous commençons par présenter le principe général des chaînes de Markov, puis nous les utiliserons dans les modèles proposés en présentant l'algorithme d'évaluation des indices.

Considérons un système pouvant prendre $(n+1)$ états possibles E_1, E_2, \dots, E_{n+1} , $(n+1)$ étant fini ou non, mais les états étant toujours dénombrables. Donc, la chaîne de Markov est définie par le repère $E = \{ E_1, E_2, \dots, E_{n+1} \}$. La probabilité que le système soit dans l'état E_i à la date t

sera appelée $P_i(t)$; les $P_i(t)$ avec $i = 1, 2, 3, \dots, (n+1)$, constitue l'ensemble des vecteurs d'états décrivant le système pour toutes les dates considérées, on désignera par

$P(t) = [P_1(t), P_2(t), \dots, P_{n+1}(t)]$ le vecteur d'état à la date t .

Le passage d'un état E_i à un autre état E_j (ou transition), ne dépend que de ces deux états. Plus rigoureusement, supposons qu'à toute paire (E_i, E_j) , il soit possible d'associer une probabilité que le système se trouve dans l'état E_j à la date $(t+1)$, sachant qu'il était dans l'état E_i à la date t . Alors, ce passage s'effectue selon la probabilité conditionnelle :

$$P(E_i / E_j) = P_{ij} = \lambda_k$$

sachant que les éléments de la matrice du vecteur d'état P vérifient la condition suivante:

$$\sum P_{ij} = 1 \quad \text{avec } i = 1, 2, 3, \dots, (n+1)$$

Nous nous limitons au cas où le processus de Markov possède un espace d'état fini. Cet espace qu'on appelle chaîne de Markov, s'obtient par la construction du graphe des marquages accessibles en associant à chaque arc une probabilité de franchissement qui dépend de la probabilité de la transition que l'on franchit et des marquages des places en amont de cette transition.

Le générateur de processus markovien associé au RdPS est une matrice carrée, notée Q , de dimension $L \times L$, ou L est le nombre fini d'états de la chaîne de Markov correspondante.

Algorithme de construction du générateur du processus markovien Q associé au RdPS:

1- L'élément hors de la diagonale Q_{rs} ($r \neq s$) est égal au poids de l'arc allant de l'état r vers l'état s et représente la probabilité de transition d'un état vers un autre.

2- L'élément diagonal $Q_{rr} = -\sum \lambda_{rk}$; λ_{rk} représente les probabilités dans la ligne r .

Après avoir construit le générateur Q du processus markovien, la résolution du système linéaire homogène permet d'obtenir les probabilités d'états en régime stationnaire grâce à la relation suivante: (P : vecteur ligne des probabilités des états)

$$P \cdot Q = 0 \tag{4.3}$$

avec la condition que $\sum P_i = 1$ pour $i = 1, L$. Les P_i représentent les probabilités d'états.

A partir des probabilités P_i , nous pouvons déduire trois résultats importants:

1- Les fréquences moyennes de franchissement des transitions $T_i \in T$;

$$F_i^* = \sum_k \lambda_i(k) \cdot P_k; \text{ tels que } T_i \text{ est franchissable à partir de } M_k \tag{4.4}$$

$\lambda_i(k)$ sont les probabilités de franchissements de la matrice Q .

2- Les marquages moyens des places $P_i \in P$ en régime stationnaire;

$$M^*(P_i) = \sum_k M_k(P_i) \cdot P_k \quad (4.5)$$

3- Les temps moyens de séjour des marques dans les places $P_i \in P$ en régime stationnaire, c'est à dire $D^*(P_i)$ dont l'expression est donnée par la formule de Little.

Nous avons rappelé, dans cette section, l'essentiel de la théorie des RdPS qui est riche en propriétés mathématiques. Mais la difficulté majeure de l'approche markovienne réside dans l'explosion très rapide de graphes de marquage (ensemble de tous les états du processus de marquage). Cela nous a conduit à réfléchir sur un modèle qui permette de ne pas s'engouffrer dans une telle difficulté. L'idée retenue, était de présenter chaque étage du RIP par une place quel que soit la dimension du réseau. Pour le réseau à un seul étage ce sera un tour l'équivalent à l'étage du RIP et qui sera représenté par une place. Une telle démarche permettra de limiter considérablement le nombre de places dans la modélisation.

Dans le cadre du principe général du processus markovien associé au RdPS, nous avons présenté la généralisation des deux modèles: sans mémoire et avec mémoire [BEN 98], ou a été associé, à chacune des transitions, la probabilité correspondante. C'est ainsi que les figures 4.2 et 4.3 revêtiront la forme de chaîne de Markov, comme le montrent ci-après, les figures 4.5 et 4.6.

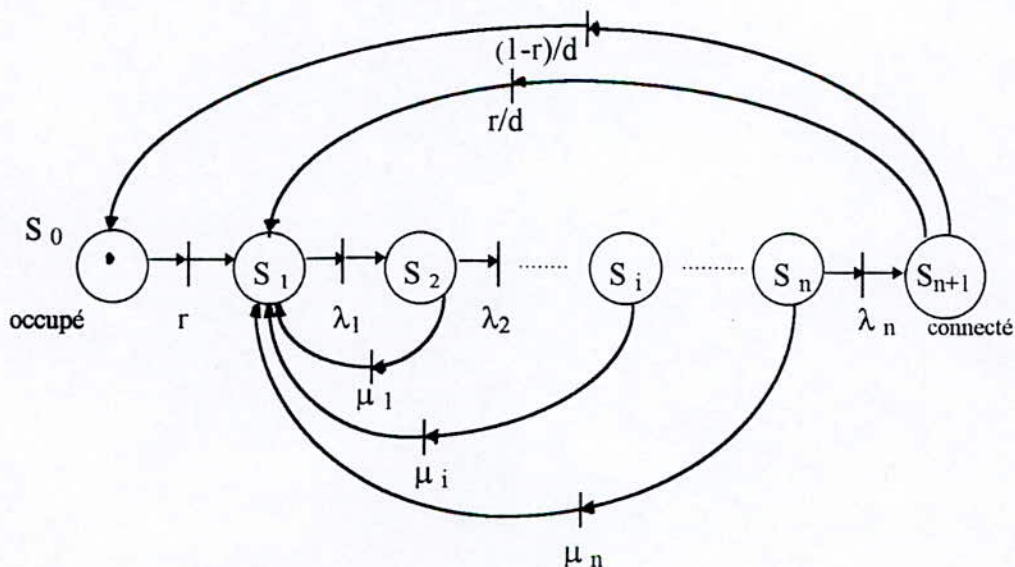


Fig. 4.5 Chaînes de Markov du modèle 1 sans mémoire.

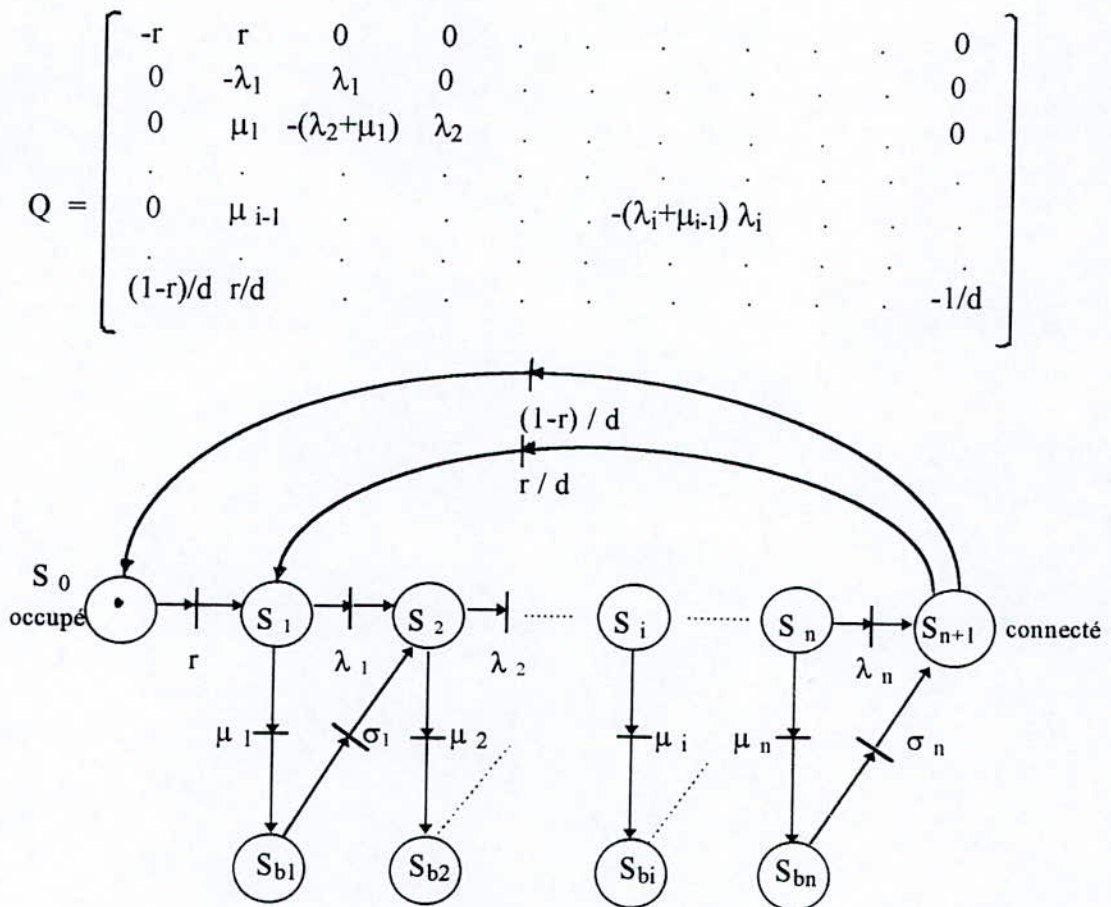


Fig. 4.6 Chaînes de Markov du modèle avec mémoire

Grâce à l'algorithme ci-dessus exposé, qui permet de construire le générateur de processus markovien associé au RdPS, nous pourrions réaliser la matrice associée aux chaînes de Markov du modèle sans mémoire laquelle est représentée par la variable Q , ainsi que pour le modèle avec mémoire qui est représenté par la variable Q_m .

$$Q_m = \begin{bmatrix} -r & r & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & -(\lambda_1 + \mu_1) & \mu_1 & \lambda_1 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & -\sigma_1 & \sigma_1 & \dots & \dots & \dots & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & -(\lambda_i + \mu_i) & \mu_i & \lambda_i & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & -\sigma_i & \sigma_i & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & -\sigma_n & \sigma_n \\ (1-r)/d & r/d & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & -1/d \end{bmatrix}$$

L'approche de l'évaluation des performances des réseaux d'interconnexions à plusieurs étages est présentée dans le chapitre suivant.



**EVALUATION DES
PERFORMANCES DES RESEAUX
D'INTERCONNEXIONS
A PLUSIEURS ETAGES**

CHAPITRE V

Evaluation des performances des réseaux d'interconnexions à plusieurs étages

L'application des modèles de notre approche est présentée pour un RI à trois étages ($n=3$) selon deux types de stratégies: la stratégie dite sans mémoire et la stratégie dite avec mémoire. Nous avons étendu l'application à des RI de 4 à 6 étages pour les deux stratégies.

5.1 Evaluation du modèle 1 sans mémoire

Dans cette application, nous avons pris en considération le réseau baseline qui est constitué de BC de 2×2 . Cette BC est munie de commandes qui permettent de gérer les commutations de manière à relier les entrées du réseau avec les sorties selon la demande.

Le réseau baseline à trois étages est présenté dans la figure 2.13. On dispose de 8 entrées/ 8 sorties, le réseau comporte 4 BC par étage, qui nous donne un total de 12 BC. Chaque sortie d'une BC en amont est connectée à une seule entrée de la BC adjacente en aval.

Comme nous l'avons défini dans l'hypothèse qui gère la BC, qui est l'élément fondamental d'un RI, tel que représenté dans la figure 4.1, où nous avons considéré trois possibilités de connexion de la BC. Lorsque le processeur demandeur génère une requête, celle-ci est transférée à l'entrée de la BC du premier étage avec une probabilité de passage égale à 1, 0.5, ou 0.75.

Dans la stratégie sans mémoire, La BC qui décide le blocage d'une requête, informe automatiquement le processeur source du blocage à travers le chemin préétabli. A la fin du cycle, ce chemin est effacé. La requête bloquée sera réinjectée de nouveau pour demander accès au réseau.

Se basant sur les probabilités de passages tels que défini dans la figure 4.1, nous pourrions dire que le passage d'une requête dépend des états de connexion des autres requêtes dans le réseau. Si une requête arrive à la BC à l'étage i , la probabilité que l'une des sorties soit déjà occupée par d'autres requêtes constitue la somme des probabilités de $P(S_{i+1})$ à $P(S_n)$ et de $P(S_{n+1})$. C'est le cas (b) représenté dans la figure 4.1. La probabilité qu'une requête arrive dans l'autre entrée de la BC dans le même cycle est $P(S_i)$. C'est le cas (c) de la figure 4.1. Enfin,

une requête peut ne pas rencontrer d'autres requêtes qui peuvent la mettre en conflit, c'est le cas (a) de la figure 4.1 qui nous donnera le reste de la probabilité. Les requêtes qui se bloquent en cours de connexion changent leurs états à la place S_1 pour essayer de se reconnecter de nouveau. Nous pourrions donc déterminer les équations des probabilités des trois cas de la figure 4.1 comme suit;

$$\begin{aligned} \text{Prob (cas (a) Fig.4.1)} &= 1 - P(S_{n+1}) - \sum_{j=i}^n P(S_j) \\ \text{Prob (cas (b) Fig.4.1)} &= \sum_{j=i+1}^n P(S_j) + P(S_{n+1}) \\ \text{Prob (cas (c) Fig.4.1)} &= P(S_i) \end{aligned} \quad (5.1)$$

Grâce à ces probabilités nous pourrions déduire la probabilité de passage de l'état S_i à l'état S_{i+1} comme suit;

$$P(S_{i+1}/S_i) = \lambda_i = (1 - P(S_{n+1}) - \sum_{j=i}^n P(S_j)) + 0.5 (\sum_{j=i+1}^n P(S_j) + P(S_{n+1})) + 0.75 P(S_i)$$

$$P(S_{i+1}/S_i) = \lambda_i = 1 - 0.25 P(S_i) - 0.5 \sum_{j=i+1}^n P(S_j) - 0.5 P(S_{n+1}) \quad (5.2)$$

$$P(S_1/S_i) = 1 - P(S_{i+1}/S_i) = \mu_i = 0.25 P(S_i) + 0.5 \sum_{j=i+1}^n P(S_j) + 0.5 P(S_{n+1}) \quad (5.3)$$

Avec les Eq. 5.2 et 5.3 nous pouvons déterminer toutes les probabilités de passages ou de blocages du modèle. Ces probabilités seront augmentées par une probabilité due à la défaillance de la BC du réseau.

5.2 Evaluation du modèle 1 avec mémoire

Dans la stratégie avec mémoire, la requête bloquée maintient le chemin préétabli, et restera en attente jusqu'à la libération de la BC qui a provoquée le blocage. Une fois la BC libérée, la requête continuera son chemin à partir du point de conflit jusqu'à sa destination.

La probabilité de passage de l'état S_i à l'état S_{i+1} est similaire à celle de la stratégie sans mémoire, sauf que, au lieu de considérer S_i , nous considérerons les deux états S_i et S_{bi} .

Les probabilités de passages, de blocages et de repassages seront définies comme suit;

$$\text{Prob (cas (a) Fig.4.1)} = 1 - P(S_{n+1}) - \sum_{k=i}^n (P(S_k) + P(S_{bk}))$$

$$P(S_{i+1}/S_i) = \lambda_i = 1 - 0.25 [P(S_i) + P(S_{bi})] - 0.5 \sum_{k=i+1}^n [P(S_k) + P(S_{bk})] - 0.5P(S_{n+1}) \quad (5.4)$$

$$P(S_{i+1}/S_{bi}) = \sigma_i = B_i \cdot P(S_{bi}) / [P(S_{bi}) + P(S_i)] \quad (5.5)$$

$$B_i = (1/d) \cdot P(S_{n+1}) / (P(S_{n+1}) + \sum_{k=i+1}^n [P(S_k) + P(S_{bk})]) ; \quad k=1, n$$

Lorsque la requête est à l'étage S_{bi} , elle ne peut pas passer cet étage jusqu'à l'achèvement des transferts de données de la requête courante. Quand la BC est libérée, la requête bloquée reprendra son chemin à partir de S_{bi} vers S_{i+1} .

5.3 Algorithme de résolution

Etape 1 : Introduire les données n, r, d

Etape 2 : Construire la matrice associée à la chaîne de Markov

Etape 3 : Résolution du système Eq.4.2 avec

$$P \cdot Q = 0 \quad \text{et} \quad \sum_{i=0}^{n+1} P_i = 1^*$$

3.1 calcul des probabilités de passages et de blocages tel que donné par les Eq. 5.2 et Eq. 5.3 pour le modèle sans mémoire.

3.2 calcul des probabilités de passages et de blocages ou de repassages tel que donné par les Eq. 5.4 et Eq. 5.5 pour le modèle avec mémoire.

Sachant que $\lambda_i + \mu_i = 1$

Nous ne retiendrons que les probabilités qui vérifient le test (*).

Etape 4 : Calcul des critères d'évaluation du réseau d'interconnexion tel que donné par les Eq. 5.6 et Eq. 5.7.

Dans l'étape 1; nous introduisons tous les paramètres qui conditionnent le réseau. Ces paramètres sont: n qui spécifie la dimension du réseau; d qui détermine le temps moyen de transfert des données dans le réseau; et enfin r qui donne la probabilité d'arrivée des requêtes.

Dans l'étape 2; nous déclarons la matrice associée au chaîne de Markov selon le modèle

Dans l'étape 3; nous calculons les probabilités des états du réseau en utilisant un algorithme de résolution du système d'équation du processus markovien qui permet de calculer les probabilités de passages ou de blocages par la méthode itérative. Les probabilités retenues doivent satisfaire le test (*).

Dans l'étape 4; nous prenons pour base les critères retenus qui permettent d'évaluer les performances des réseaux d'interconnexions à plusieurs étages. Ces critères sont: la bande passante, noté **BW**, le temps moyen de service d'une requête, noté **Avt**. Les fréquences moyennes de franchissements, noté **F*moy** et les temps de séjour des marques dans les places, noté **D*moy**.

Dans la plupart des articles liés à l'évaluation des performances, les chercheurs emploient deux paramètres principaux qui sont la bande passante et le temps moyen de service d'une requête. La BW représente le nombre d'accès de requête que le réseau satisfera par une unité de temps. L'Avt. représente le temps entre la génération d'une requête à l'entrée du réseau et l'achèvement de cette requête. Les équations de la BW et du Avt sont données par Wu [WU 92] pour un cas général. Ces équations sont adaptées au modèle proposé et sont comme suit;

$$BW = 2^n \cdot d \cdot P_4 \quad (5.6)$$

$$Avt = d \cdot (1 - P_0) / P_4 \quad (5.7)$$

2^n désigne le nombre d'entrée/sortie du réseau.

d désigne le temps de transfert des données (en cycle).

P_4 désigne la probabilité de connexion de la requête.

P_0 désigne la probabilité d'occupation du réseau.

5.4 Applications

5.4.1 Modèle 1 sans mémoire

Nous présenterons les modèles sans mémoires puis avec mémoire pour $n=3$.

La figure 5.1 présente la modélisation 1 selon la stratégie sans mémoire en considérant une disponibilité totale du premier étage.

La place S_0 est identifiée comme générateur de requête du réseau, sachant que la requête précédente générée par le processeur source est complète, et qu'il n'y a pas d'arrivée de nouvelle requête. Les requêtes arrivent au réseau avec une probabilité r .

Les places de S_1 à S_3 sont identifiées comme les états du réseau à partir de l'étage 1 jusqu'à l'étage 3 respectivement. Dans ces états, la requête passe du premier étage vers le deuxième

étage avec une probabilité de passage λ_1 . Si cet étage est bloqué, la requête reviendra au premier étage avec une probabilité de blocage μ_1 , et essaiera de se reconnecter de nouveau. Dans le cas où elle n'est pas bloquée, elle poursuivra son chemin jusqu'à l'étage 3.

La place S_4 est utilisée pour l'état de connexion entre le processeur source et la mémoire de destination. Dans cet état, la requête a établi un chemin pour transférer les données, et ce chemin est maintenu pendant d cycles.

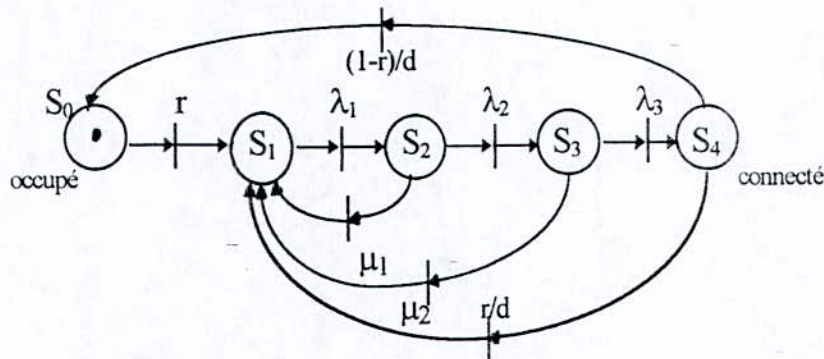


Fig. 5.1 Modélisation 1 sans mémoire du réseau baseline à 3 étages par les RdPS

Grâce à l'algorithme présenté au chapitre 4, qui permet de construire le générateur de processus markovien associé au RdPS, nous pourrions construire la matrice associée aux chaînes de Markov du modèle comme suit;

$$Q = \begin{bmatrix} -r & r & 0 & 0 & 0 \\ 0 & -\lambda_1 & \lambda_1 & 0 & 0 \\ 0 & \mu_1 & -(\lambda_2 + \mu_1) & \lambda_2 & 0 \\ 0 & \mu_2 & 0 & -(\lambda_3 + \mu_2) & \lambda_3 \\ (1-r)/d & r/d & 0 & 0 & -1/d \end{bmatrix}$$

En utilisant l' Eq. 4.3, nous pourrions déduire les probabilités de passages et de blocages comme suit;

$$\begin{aligned} -r P(S_0) + ((1-r)/d) \cdot P(S_4) &= 0 \\ \lambda_1 \cdot P(S_1) - (\lambda_2 + \mu_1) \cdot P(S_2) &= 0 \\ \lambda_2 \cdot P(S_2) - (\lambda_3 + \mu_2) \cdot P(S_3) &= 0 \\ \lambda_3 \cdot P(S_3) - (1/d) \cdot P(S_4) &= 0 \end{aligned}$$

En exprimant ces probabilités d'états en fonction de $P(S_4)$ nous obtenons les expressions suivantes;

$$P(S_3) = 1/(d \cdot \lambda_3) \cdot P(S_4)$$

$$P(S_2) = (\lambda_3 + \mu_2) \cdot P(S_4)/(d \cdot \lambda_2 \cdot \lambda_3)$$

$$P(S_1) = (\lambda_3 + \mu_2) \cdot (\lambda_2 + \mu_1) \cdot P(S_4)/(d \cdot \lambda_1 \cdot \lambda_2 \cdot \lambda_3)$$

$$P(S_0) = (1-r) \cdot P(S_4)/(d \cdot r)$$

En tenant compte de la condition (*) que la somme des probabilités est égale à 1, cela nous conduit à l'expression suivante;

$$P(S_4) = 1 / [1 + (1-r)/(d \cdot r) + 1/(d \lambda_3) \cdot [1 + ((\lambda_3 + \mu_2)/\lambda_2)(1 + (\lambda_2 + \mu_1)/\lambda_1)]]$$

Grâce à l'algorithme de résolution, nous pourrions déterminer les probabilités de passages, de blocages et des états. Ces probabilités sont données dans la Table 1. Ces probabilités sont utilisées dans les Eq. 5.6 et 5.7 pour la détermination de la BW et du Avt.

r	d = 5						d = 10						d = 20					
	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0
λ_1	.77	.68	.62	.60	.59	.58	.71	.64	.59	.58	.56	.56	.65	.60	.57	.56	.55	.55
λ_2	.81	.76	.72	.71	.71	.70	.76	.70	.67	.66	.65	.70	.69	.65	.63	.62	.62	.61
λ_3	.81	.81	.79	.78	.77	.76	.78	.74	.72	.71	.71	.71	.71	.77	.66	.65	.65	.65
P_0	.48	.27	.12	.05	.02	0.0	.37	.19	.09	.04	.013	0.0	.25	.12	.05	.02	.008	0.0
P_1	.11	.18	.26	.25	.31	.31	.10	.16	.21	.22	.24	.25	.09	.13	.16	.17	.17	.17
P_2	.08	.12	.14	.16	.16	.17	.07	.10	.11	.12	.12	.12	.06	.07	.03	.09	.082	.09
P_3	.06	.08	.10	.10	.11	.11	.05	.07	.07	.08	.077	.08	.04	.05	.05	.05	.05	.06
P_4	.27	.35	.38	.40	.40	.41	.41	.48	.52	.54	.55	.55	.56	.63	.66	.67	.69	.68

Table 1. Probabilité de passage et des états du modèle 1 sans mémoire pour $n=3$, $d=5, 10$ et 20 cycles.

Nous avons résolu ces équations en faisant varier quelques paramètres pour pouvoir mieux apprécier les performances du réseau. Nous avons procédé à la sélection des valeurs de la manière suivante: probabilité d'arrivée des requêtes = 0.1, 0.2, 0.4, 0.6, 0.8, 1.0 par cycle; temps de transfert des données = 5, 10 et 20 cycles. Ce temps est choisi suffisamment grand par rapport au temps de commutation.

Pour chacune des valeurs de n , r et d , nous avons calculé la BW et l'Avt qui sont présentés dans la Table 1.1, et illustrés par des graphes tels que le montre la figure 5.2.

Pour la détermination des fréquences de franchissement et en utilisant l'Eq. 4.4, nous obtenons les équations suivantes;

$$\begin{aligned}
 f_1^* &= r \cdot P(S_0) & f_5^* &= \mu_1 \cdot P(S_2) \\
 f_2^* &= \lambda_1 \cdot P(S_1) & f_6^* &= \mu_2 \cdot P(S_3) \\
 f_3^* &= \lambda_2 \cdot P(S_2) & f_7^* &= r/d \cdot P(S_4) \\
 f_4^* &= \lambda_3 \cdot P(S_3) & f_8^* &= (1-r)/d \cdot P(S_4)
 \end{aligned}
 \tag{5.8}$$

d	r	avt	BW	d	r	avt	BW	d	r	avt	BW
5	.1	09.64	10.70	10	.1	15.55	32.56	20	.1	26.84	89.21
	.2	10.62	13.68		.2	16.73	38.60		.2	27.99	99.93
	.4	11.50	15.38		.4	17.52	42.00		.4	28.79	105.75
	.6	11.79	16.03		.6	17.85	43.14		.6	28.03	107.66
	.8	12.02	16.30		.8	18.05	43.71		.8	29.21	108.62
	1.0	12.12	16.48		1	18.11	44.12		1	29.28	109.14

Table 1.1 Probabilité d'arrivée des requête de service, Avt et BW dans la stratégie sans mémoire du modèle 1 pour n = 3, d = 5, 10 et 20 cycles.

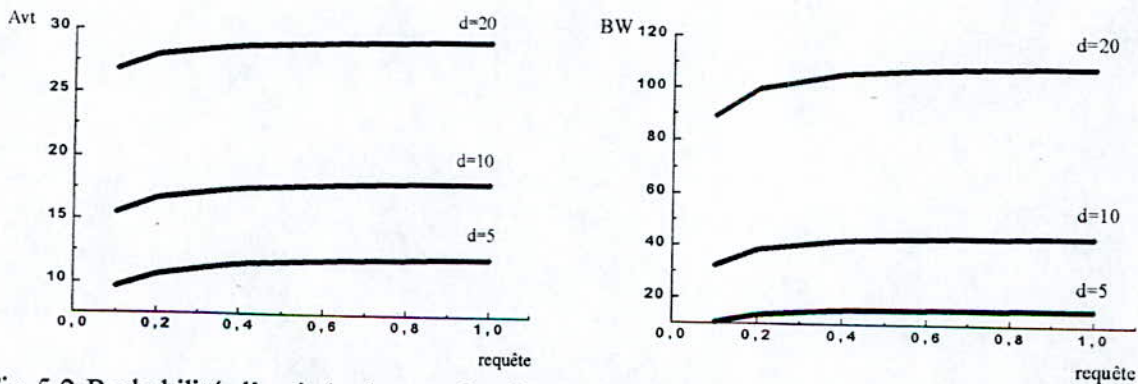


Fig 5.2 Probabilité d'arrivée de requête de service, Avt et BW dans la stratégie sans mémoire du modèle 1 pour n=3.

Se basant sur l'Eq. 4.5, nous déduisons le marquage moyen dans les places comme suit;

$$\begin{aligned}
 M^*(P_0) &= P(S_0) \\
 M^*(P_1) &= P(S_1) \\
 M^*(P_2) &= P(S_2) \\
 M^*(P_3) &= P(S_3) \\
 M^*(P_4) &= P(S_4)
 \end{aligned}
 \tag{5.9}$$

La conservation du flux de marquage donnée est vérifiée car nous avons;

$$\sum_{i=0,4} M^*(P_i) = P(S_0) + \sum_{i=1,3} P(S_i) + P(S_4)$$

En ce qui concerne le temps moyen de séjour des marques dans les places, avec l'Eq. 4.2, nous obtenons;

$$D^*(P_0) = M^*(P_0) / f_8^*$$

$$D^*(P_1) = M^*(P_1) / (f_1^* + f_5^* + f_6^* + f_7^*) \quad (5.10)$$

$$D^*(P_i) = M^*(P_i) / f_i^* \quad \text{pour } i=2,..4$$

Grâce aux Eq. 5.8 et 5.10 nous avons déterminé les fréquences moyennes de franchissements et les temps de séjour des marques dans les places et qui sont donnés dans la Table 2.

d	r	f*moy	D*moy	d	r	f*moy	D*moy	d	r	f*moy	D*moy
5	.1	0.335	18.22	10	.1	0.280	23.29	20	.1	0.214	33.45
	.2	0.462	13.36		.2	0.361	18.44		.2	0.258	28.73
	.4	0.547	10.94		.4	0.408	16.08		.4	0.282	26.24
	.6	0.567	10.15		.6	0.418	15.26		.6	0.284	25.37
	.8	0.569	9.78		.8	0.415	14.94		.8	0.283	25.05
	1.0	0.607	7.82		1	0.400	13.95		1	0.273	24.92

Table 2. Probabilité d'arrivée de requête de service, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie sans mémoire du modèle 1 pour d = 5, 10 et 20 cycles.

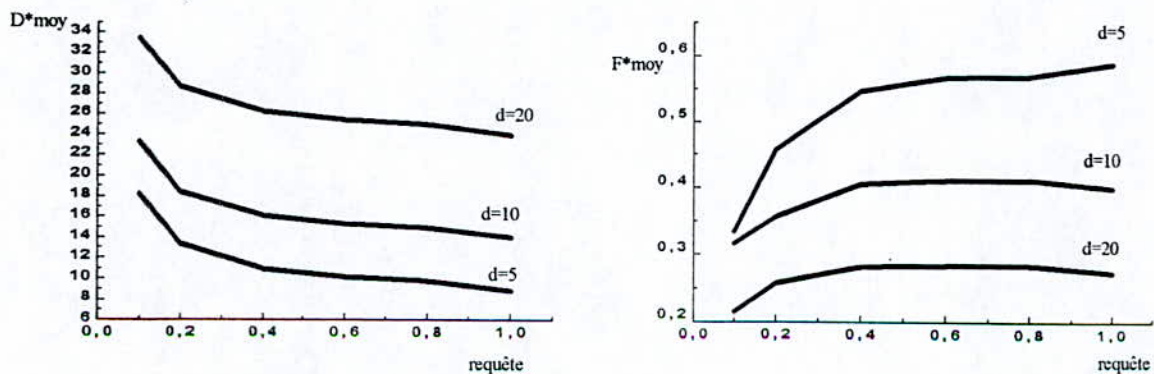


Fig. 5.3 Probabilité d'arrivée de requête de service, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie sans mémoire du modèle 1 pour n=3.

Afin de pouvoir apprécier nos résultats, nous avons calculé le temps moyen de service pour $n=4$ et fait une comparaison des résultats obtenus avec ceux de l'article de référence [WU 92]. Nous avons choisi le cas de $n=4$ et $d=10$, car ce sont les paramètres qui sont donnés dans l'article. La figure 5.4 montre l'évolution des deux courbes qui sont très proche. La Table 3 donne les résultats des deux approches avec la différence en %. La plus grande différence est d'environ 7.5%. Cette différence s'explique par l'ajout de probabilité que nous avons effectué sur notre modèle et qui à notre sens donne satisfaction à l'approche que nous avons développé.

d	n	r	Avt.(notre approche)	Avt. (Wu)	Diff.%
10	4	.1	18.45	17.75	3.9
		.2	20.16	18.75	7.5
		.4	21.28	20.00	6.4
		.6	21.57	20.25	6.5
		.8	21.94	20.75	5.7
		1.0	21.97	21.25	3.4

Table 3. Comparaison des temps moyens de services entre nos résultats et ceux de [Wu 92]

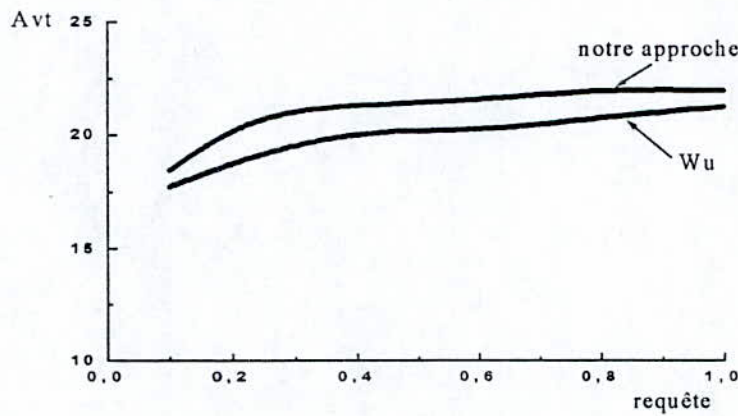


Fig. 5.4 Comparaison du temps moyen de service entre les deux modèles dans la stratégie sans mémoire.

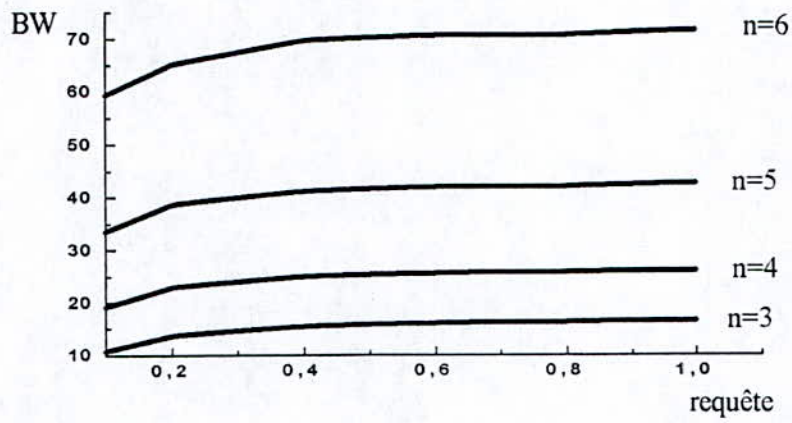


Fig. 5.5 a) BW du modèle 1 sans mémoire pour $d=5$ cycles.

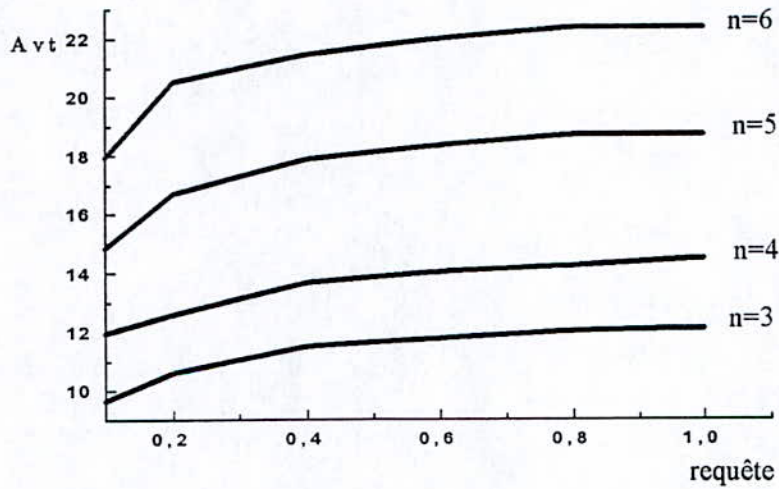


Fig. 5.5 b) Avt du modèle 1 sans mémoire pour $d=5$ cycles.

5.4.2 Modèle 2 sans mémoire

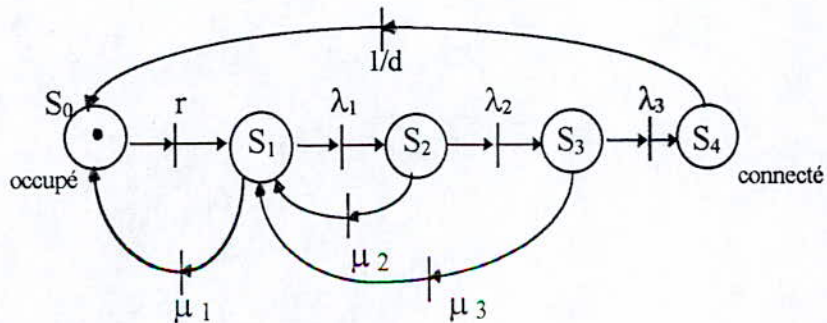


Fig. 5.6 Modélisation 2 sans mémoire du réseau baseline à 3 étages par les RdPS

La figure 5.6 présente la modélisation 2 selon la stratégie sans mémoire[BEN 98]. A l'inverse du modèle 1, nous considérons que l'étage 1 peut ne pas être disponible pour recevoir une requête. Cette requête revient au processeur source pour l'informer de la non réceptivité de la requête à cause d'une défaillance probable de l'étage 1. La suite du modèle demeure inchangée.

La place S_0 est identifiée comme générateur de requête du réseau, sachant que la requête précédente générée par le processeur source est complète, et il n'y a pas d'arrivée de nouvelle requête. Les requêtes arrivent au réseau avec une probabilité r . Quand une requête ne peut pas accéder à l'étage 1, elle revient à la source en validant la transition associée à la probabilité de blocage μ_1 . Les places de S_1 à S_4 restent inchangées.

La matrice associée aux chaînes de Markov du modèle 2 sans mémoire est comme suit;

$$Q = \begin{bmatrix} -r & r & 0 & 0 & 0 \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & 0 & 0 \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & 0 \\ 0 & \mu_3 & 0 & -(\lambda_3 + \mu_3) & \lambda_3 \\ 1/d & 0 & 0 & 0 & -1/d \end{bmatrix}$$

L'Eq. 4.3, nous permet de déduire les probabilités de passages et de blocages comme suit;

$$\begin{aligned} -r.P(S_0) + \mu_1.P(S_1) + (1/d).P(S_4) &= 0 \\ r.P(S_0) - (\lambda_1 + \mu_1).P(S_1) + \mu_2.P(S_2) + \mu_3.P(S_3) &= 0 \\ \lambda_1.P(S_1) - (\lambda_2 + \mu_2).P(S_2) &= 0 \\ \lambda_2.P(S_2) - (\lambda_3 + \mu_3).P(S_3) &= 0 \\ \lambda_3.P(S_3) - (1/d).P(S_4) &= 0 \end{aligned}$$

En exprimant ces probabilités d'états en fonction de $P(S_4)$ nous obtenons les expressions suivantes;

$$\begin{aligned} P(S_3) &= P(S_4) / (d.\lambda_3) \\ P(S_2) &= (\lambda_3 + \mu_3) / (d.\lambda_2\lambda_3).P(S_4) \\ P(S_1) &= (\lambda_3 + \mu_3).(\lambda_2 + \mu_2) / (d.\lambda_1\lambda_2.\lambda_3).P(S_4) \\ P(S_0) &= (1/(r.d)).[1 + \mu_1 (\lambda_3 + \mu_3)(\lambda_2 + \mu_2) / (\lambda_1\lambda_2\lambda_3)].P(S_4) \end{aligned}$$

En tenant compte de la condition (*) que la somme des probabilités est égale à 1, cela nous conduit à l'expression suivante;

$$P(S_4) = 1 / [1 + 1/(d\lambda_3) \cdot ((1 + (\lambda_3 + \mu_3)/\lambda_2) \cdot (1 + (\lambda_2 + \mu_2)/\lambda_1) \cdot (1 + \mu_1/r)) + 1(d.r)]$$

Les probabilités de passages sont exprimées à partir de l'Eq. 5.2 comme suit;

$$\lambda_1 = 1 - 0.25 P(S_1) - 0.5 (P(S_2) + P(S_3)) - 0.5 P(S_4)$$

$$\lambda_2 = 1 - 0.25 P(S_2) - 0.5 P(S_3) - 0.5 P(S_4)$$

$$\lambda_3 = 1 - 0.25 P(S_3) - 0.5 P(S_4)$$

avec $\mu_1 = 1 - r$

De même que dans le modèle 1, nous avons appliqué l'algorithme de résolution pour déterminer les probabilités de passages et probabilités des états. Ces probabilités sont données par la Table 4. Pour chacune des valeurs de n, r et d, on a calculé la bande passante et le temps moyen de service qui sont présentés dans la Table 4.1 et représentés par les graphes de la figure 5.7.

r	d = 5						d = 10						d = 20					
	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0
λ_1	.87	.81	.72	.67	.64	.61	.83	.76	.68	.64	.61	.58	.78	.70	.64	.60	.58	.56
λ_2	.90	.84	.79	.75	.73	.72	.85	.79	.73	.70	.68	.67	.80	.74	.68	.65	.63	.62
λ_3	.92	.87	.83	.80	.79	.78	.87	.82	.77	.74	.73	.72	.81	.75	.70	.68	.67	.65
P_0	.72	.56	.37	.25	.15	.08	.64	.48	.30	.19	.11	.05	.53	.38	.22	.14	.08	.03
P_1	.05	.09	.15	.20	.24	.27	.04	.08	.12	.15	.19	.22	.04	.07	.10	.13	.14	.16
P_2	.04	.07	.10	.12	.14	.15	.04	.05	.08	.10	.11	.12	.03	.04	.06	.07	.08	.09
P_3	.03	.05	.07	.08	.09	.10	.03	.04	.06	.07	.07	.07	.02	.03	.04	.04	.05	.05
P_4	.016	.23	.31	.35	.38	.40	.25	.35	.44	.49	.52	.54	.38	.38	.58	.62	.65	.67

Table 4.1. Probabilité de passage et des états du modèle 2 sans mémoire pour n=3, d=5,10 et 20 cycles.

d	r	avt	BW	d	r	avt	BW	d	r	avt	BW
5	.1	8.78	6.27	10	.1	14.22	20.26	20	.1	24.83	60.18
	.2	9.35	9.29		.2	14.96	27.99		.2	25.86	76.79
	.4	10.15	12.36		.4	16.01	35.10		.4	27.05	92.02
	.6	10.82	13.92		.6	16.68	38.90		.6	27.88	99.20
	.8	11.23	15.08		.8	17.20	41.29		.8	28.42	103.96
	1.0	11.60	15.87		1	17.66	42.88		1	28.93	106.91

Table 4.1. Probabilité d'arrivée de requête, Avt et BW dans la stratégie sans mémoire du modèle 2 pour n=3, d=5,10 et 20 cycles.

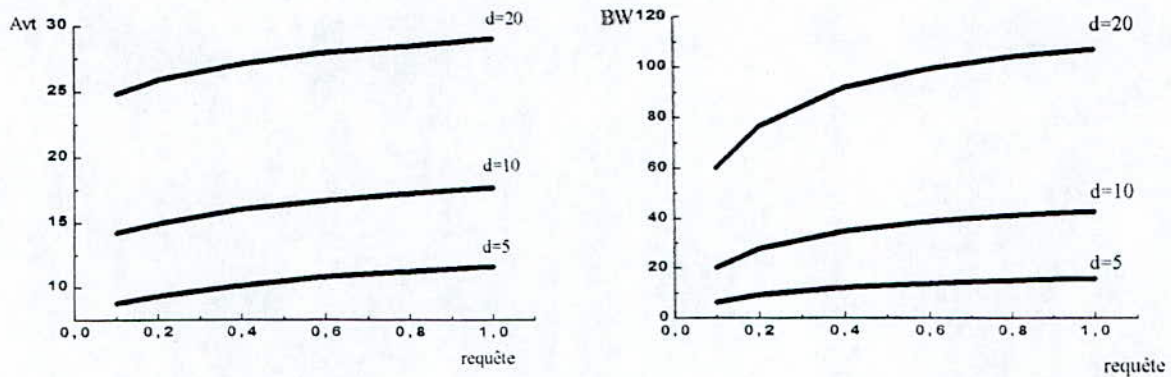


Fig. 5.7 Probabilité d'arrivée de requête de service, Avt et BW dans la stratégie sans mémoire du modèle 2 pour $n=3$.

Pour la détermination des fréquences de franchissement, nous utilisons l'Eq. 4.4 et nous obtenons les équations suivantes;

$$\begin{aligned}
 f_1^* &= r \cdot P(S_0) & f_5^* &= \mu_1 \cdot P(S_1) \\
 f_2^* &= \lambda_1 \cdot P(S_1) & f_6^* &= \mu_2 \cdot P(S_2) \\
 f_3^* &= \lambda_2 \cdot P(S_2) & f_7^* &= \mu_3 \cdot P(S_3) \\
 f_4^* &= \lambda_3 \cdot P(S_3) & f_8^* &= 1/d \cdot P(S_4)
 \end{aligned}
 \tag{5.11}$$

Se basant sur l'Eq. 5.5, nous déduirons le marquage moyen dans les places comme suit;

$$\begin{aligned}
 M^*(P_0) &= P(S_0) \\
 M^*(P_1) &= P(S_1) \\
 M^*(P_2) &= P(S_2) \\
 M^*(P_3) &= P(S_3) \\
 M^*(P_4) &= P(S_4)
 \end{aligned}
 \tag{5.12}$$

La conservation du flux de marquage donnée est vérifiée car nous avons;

$$\sum_{i=0,4} M^*(P_i) = P(S_0) + \sum_{i=1,3} P(S_i) + P(S_4)$$

En ce qui concerne le temps moyen de séjour des marques dans les places, avec l'Eq. 4.2, nous obtenons;

$$\begin{aligned}
 D^*(P_0) &= M^*(P_0) / (f_5^* f_8^*) \\
 D^*(P_1) &= M^*(P_1) / (f_1^* + f_6^* + f_7^*)
 \end{aligned}
 \tag{5.13}$$

$$D^*(P_i) = M^*(P_i) / f_i^* \quad \text{pour } i=2, \dots, 4$$

Grâce aux Eq. 5.11 et 5.13 nous avons déterminé les fréquences moyennes de franchissements et les temps de séjour des marques dans les places et qui sont donnés dans la Table 5 et représentés par les graphes de la figure 5.8.

d	r	f*moy	D*moy	d	r	f*moy	D*moy	d	r	f*moy	D*moy
5	.1	0.263	17.05	10	.1	0.229	22.24	20	.1	0.194	32.22
	.2	0.417	12.54		.2	0.349	17.54		.2	0.275	28.02
	.4	0.587	10.16		.4	0.468	15.24		.4	0.348	25.19
	.6	0.651	9.45		.6	0.502	14.53		.6	0.362	24.63
	.8	0.648	9.32		.8	0.489	14.43		.8	0.342	24.35
	1.0	0.600	9.49		1	0.440	14.55		1	0.300	24.61

Table 5. Probabilité d'arrivée de requête de service, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie sans mémoire du modèle 2 pour $d=5, 10$ et 20 cycles.

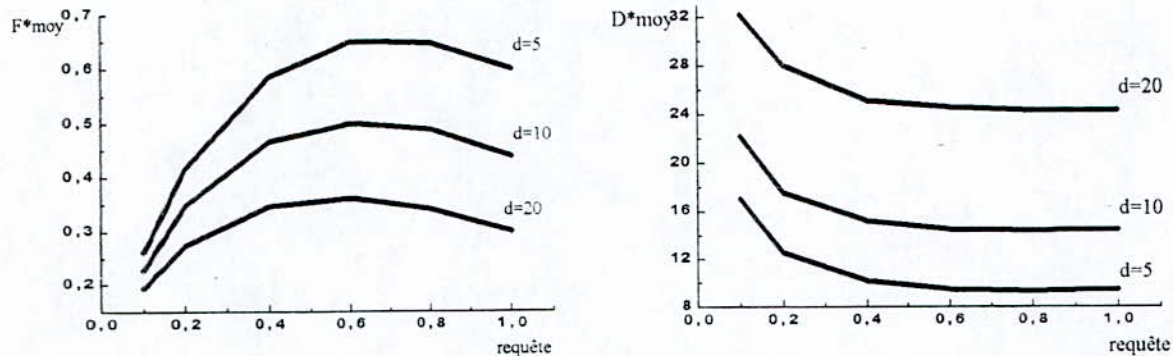


Fig. 5.8 Probabilité d'arrivée de requête de service, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie sans mémoire du modèle 2 pour $n=3$.

5.4.3 Modèle avec mémoire

La figure 5.9 présente la modélisation d'un RIP selon la stratégie avec mémoire pour $n = 3$.

La place S_0 est identifiée comme générateur de requête du réseau, sachant que la requête précédente générée par le processeur source est complète, et il n'y a pas d'arrivée de nouvelle requête. Les requêtes arrivent au réseau avec une probabilité r .

Les places de S_1 à S_3 sont identifiées comme les états du réseau à partir de l'étage 1 jusqu'à l'étage 3 respectivement. Dans ces états, la requête passe du premier étage vers le deuxième

étage avec une probabilité de passage λ_1 . Si cet étage est bloqué, la requête passera dans une mémoire temporaire, symbolisée par la place S_{b1} , avec une probabilité de blocage μ_1 . Cette mémoire ne peut recevoir qu'une requête à la fois. Quand le deuxième étage est libéré, la requête dans la mémoire temporaire se connectera de nouveau avec une probabilité de repassage σ_1 . Cette requête poursuivra son chemin jusqu'à l'étage 3 de la même manière.

La place S_4 est utilisée pour l'état de connexion entre le processeur source et la mémoire de destination. Dans cet état, la requête a établi un chemin pour transférer les données, et ce chemin est maintenu pendant d cycles.

Grâce à l'algorithme présenté au chapitre IV, qui permet de construire le générateur de processus markovien associé au RdPS, nous pourrions construire la matrice associée aux chaînes de Markov du modèle notée Q .

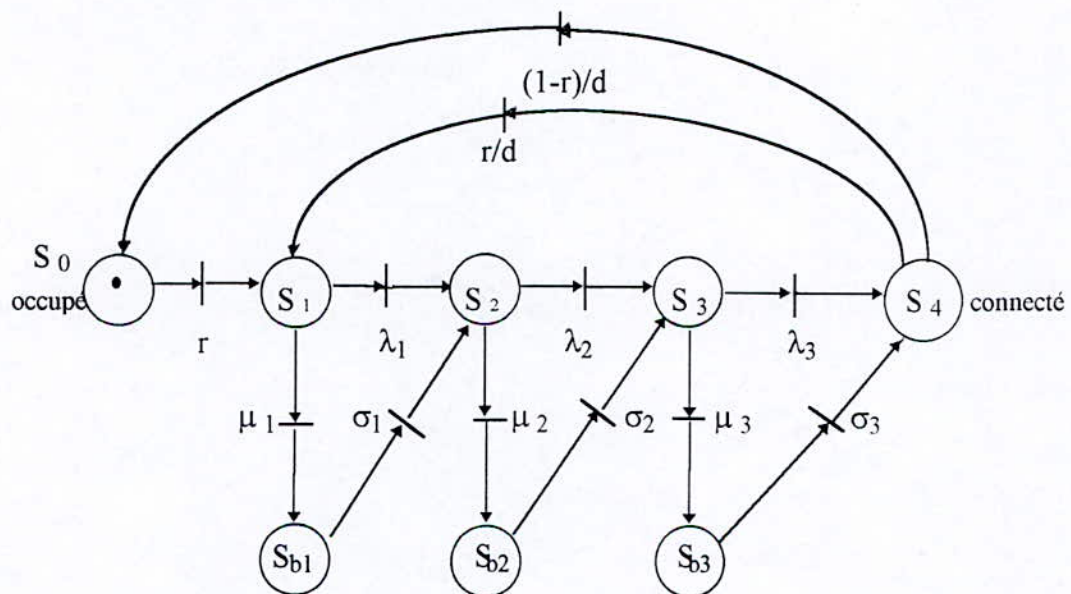


Fig. 5.9 Modélisation avec mémoire du réseau baseline à 3 étages par les RdPS

$$Q = \begin{bmatrix} -r & r & 0 & 0 & 0 & 0 & 0 \\ 0 & -(\lambda_1 + \mu_1) & \mu_1 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & -\sigma_1 & \sigma_1 & 0 & 0 & 0 \\ 0 & 0 & -(\lambda_2 + \mu_2) & \mu_2 & \lambda_2 & 0 & 0 \\ 0 & 0 & 0 & -\sigma_2 & \sigma_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -(\lambda_3 + \mu_3) & \mu_3 & \lambda_3 \\ 0 & 0 & 0 & 0 & 0 & -\sigma_3 & \sigma_3 \\ (1-r)/d & r/d & 0 & 0 & 0 & 0 & -1/d \end{bmatrix}$$

En utilisant l' Eq. 4.3, nous pourrions déduire les probabilités de passages, de blocages et de repassages comme suit;

$$\begin{aligned}
 -r P(S_0) + ((1-r)/d) \cdot P(S_4) &= 0 \\
 r P(S_0) - (\lambda_1 + \mu_1) \cdot P(S_1) + (r/d) \cdot P(S_4) &= 0 \\
 \mu_1 \cdot P(S_1) - \sigma_1 \cdot P(S_{b1}) - (\lambda_2 + \mu_2) \cdot P(S_2) &= 0 \\
 \lambda_1 \cdot P(S_1) + \sigma_1 \cdot P(S_{b1}) + \mu_2 \cdot P(S_2) - \sigma_2 \cdot P(S_{b2}) &= 0 \\
 \lambda_2 \cdot P(S_2) + \sigma_2 \cdot P(S_{b2}) - (\lambda_3 + \mu_3) \cdot P(S_3) &= 0 \\
 \mu_3 \cdot P(S_3) - \sigma_3 \cdot P(S_{b3}) &= 0 \\
 \lambda_3 \cdot P(S_3) + \sigma_3 \cdot P(S_{b3}) - (1/d) \cdot P(S_4) &= 0
 \end{aligned}$$

En exprimant ces probabilités d'états en fonction de $P(S_4)$ nous obtenons les expressions suivantes en tenant compte de la condition (*) que la somme des probabilités est égale à 1; cela nous conduit à l'expression suivante;

$$P(S_4) = 1 / [(1/d) [(1-r)/d + 1/(\mu_1 + \lambda_1)(1 + \mu_1/\sigma_1) + 1/(\lambda_2 + \mu_2)(1 + \mu_2/\sigma_2) + 1/(\mu_3/\lambda_3)(1 + \mu_3/\sigma_3) + 1]]$$

Grâce à l'algorithme de résolution nous avons déterminé les probabilités des états. Ces probabilités sont utilisées dans les Eq. 5.6 et 5.7 pour la détermination de la BW et du Avt.

	d = 5						d = 10						d = 20					
r	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0
λ_1	.74	.67	.62	.61	.60	.59	.68	.63	.60	.59	.58	.58	.63	.60	.59	.58	.57	.57
λ_2	.82	.78	.75	.74	.74	.74	.77	.74	.72	.72	.71	.71	.73	.71	.71	.70	.70	.70
λ_3	.87	.85	.83	.83	.82	.82	.83	.81	.80	.80	.80	.79	.80	.79	.78	.78	.78	.78
σ_1	.08	.08	.07	.08	.08	.08	.05	.05	.05	.05	.05	.05	.03	.03	.03	.02	.02	.02
σ_2	.09	.10	.09	.09	.10	.10	.06	.06	.06	.06	.06	.06	.03	.03	.03	.03	.03	.03
σ_3	.11	.11	.12	.12	.12	.12	.07	.07	.07	.07	.07	.08	.04	.04	.04	.04	.04	.04
P_0	.38	.20	.08	.04	.01	0.0	.26	.13	.05	.03	.009	0.0	.15	.07	.03	.03	.005	0.0
P_1	.04	.05	.05	.06	.06	.06	.03	.03	.03	.03	.03	.03	.02	.02	.02	.02	.02	.02
P_2	.04	.05	.05	.06	.06	.06	.03	.03	.03	.03	.03	.03	.02	.02	.02	.02	.02	.02
P_3	.04	.05	.05	.06	.06	.06	.03	.03	.03	.03	.03	.03	.02	.02	.02	.02	.02	.02
P_4	.21	.25	.27	.28	.28	.28	.29	.32	.33	.34	.34	.34	.34	.36	.38	.37	.37	.39
P_{b1}	.15	.21	.27	.29	.30	.31	.18	.24	.27	.28	.29	.29	.21	.24	.28	.25	.27	.27
P_{b2}	.09	.12	.15	.14	.15	.15	.11	.14	.16	.16	.16	.17	.15	.17	.16	.19	.19	.19
P_{b3}	.05	.07	.08	.07	.08	.08	.07	.08	.10	.10	.091	.10	.09	.10	.10	.10	.105	.10

Table 6. Probabilité de passage et des états du modèle avec mémoire pour $n=3$, $d=5,10$ et 20 cycles.

Nous avons résolu ces équations en faisant varier quelques paramètres pour pouvoir mieux apprécier les performances du réseau. Nous avons procédé à la sélection des valeurs de la manière suivante: probabilité d'arrivée des requêtes = 0.1, 0.2, 0.4, 0.6, 0.8, 1.0 par cycle; temps de transfert des données = 5, 10 et 20 cycles. La probabilité de passage, de déblocages et probabilités des états sont données dans la Table 6. Pour chacune des valeurs de n, r et d, nous avons calculé la BW et l'Avt qui sont présentés dans la Table 6.1, représentés par des graphes tels que le montre la figure 5.10.

d	r	avt	BW	d	r	avt	BW	d	r	avt	BW
5	.1	14.63	08.46	10	.1	25.66	23.08	20	.1	49.33	54.86
	.2	16.02	09.99		.2	27.45	25.44		.2	51.25	57.92
	.4	17.11	10.74		.4	28.52	26.65		.4	52.50	59.26
	.6	17.22	11.18		.6	28.72	27.22		.6	52.50	60.19
	.8	17.43	11.30		.8	29.09	27.27		.8	52.83	90.28
	1.0	17.70	11.36		1	29.23	27.36		1	52.83	60.57

Table 6.1 Probabilité d'arrivée de requête de service, Avt et BW dans la stratégie avec mémoire pour n=3, d=5,10 et 20 cycles.

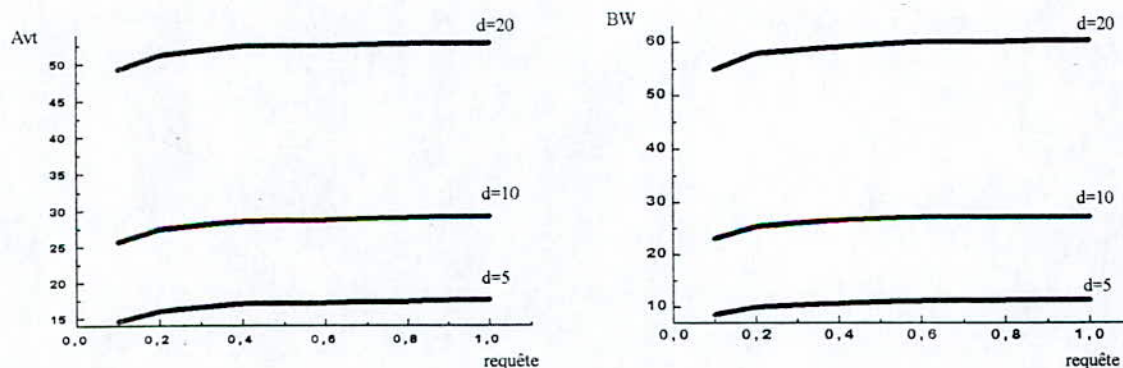


Fig 5.10 Probabilité d'arrivée de requête de service, Avt et BW dans la stratégie avec mémoire pour n = 3.

Pour la détermination des fréquences de franchissement, en utilisant l'Eq. 4.4 nous obtenons les équations suivantes;

$$\begin{aligned}
 f_1^* &= r \cdot P(S_0) & f_7^* &= \mu_3 \cdot P(S_3) \\
 f_2^* &= \lambda_1 \cdot P(S_1) & f_8^* &= \sigma_1 \cdot P(S_{b1})
 \end{aligned}
 \tag{5.14}$$

$$\begin{aligned}
 f_3^* &= \lambda_2 \cdot P(S_2) & f_9^* &= \sigma_2 \cdot P(S_{b2}) \\
 f_4^* &= \lambda_3 \cdot P(S_3) & f_{10}^* &= \sigma_3 \cdot P(S_{b3}) \\
 f_5^* &= \mu_1 \cdot P(S_1) & f_{11}^* &= r/d \cdot P(S_4) \\
 f_6^* &= \mu_2 \cdot P(S_2) & f_{12}^* &= (1-r)/d \cdot P(S_4)
 \end{aligned}$$

Se basant sur l'Eq. 5.5, nous pourrions déduire le marquage moyen dans les places comme suit;

$$\begin{aligned}
 M^*(P_0) &= P(S_0) \\
 M^*(P_1) &= P(S_1) \\
 M^*(P_2) &= P(S_2) \\
 M^*(P_3) &= P(S_3) \\
 M^*(P_4) &= P(S_4) & (5.15) \\
 M^*(P_{b1}) &= P(S_{b1}) \\
 M^*(P_{b2}) &= P(S_{b2}) \\
 M^*(P_{b3}) &= P(S_{b3})
 \end{aligned}$$

La conservation du flux de marquage donnée est vérifiée car nous avons;

$$\sum_{i=0,4} M^*(P_i) = P(S_0) + \sum_{i=1,3} (P(S_i) + P(S_{bi})) + P(S_4)$$

En ce qui concerne le temps moyen de séjour des marques dans les places, avec l'Eq. 4.2, nous obtenons;

$$\begin{aligned}
 D^*(P_0) &= M^*(P_0) / f_{12}^* \\
 D^*(P_1) &= M^*(P_1) / (f_1^* + f_{11}^*) \\
 D^*(P_2) &= M^*(P_2) / (f_2^* + f_8^*) \\
 D^*(P_3) &= M^*(P_3) / (f_3^* + f_9^*) & (5.16) \\
 D^*(P_4) &= M^*(P_4) / (f_4^* + f_{10}^*) \\
 D^*(P_i) &= M^*(P_i) / f_i^* \quad \text{pour } i=5,..7
 \end{aligned}$$

Grâce aux Eq. 5.8 et 5.10 nous avons déterminé les fréquences moyennes de franchissements et les temps de séjour des marques dans les places; ils sont donnés dans la Table 7 et représentés dans la figure 5.11.

d	r	f*moy	D*moy	d	r	f*moy	D*moy	d	r	f*moy	D*moy
5	.1	0.234	51.29	10	.1	0.163	73.30	20	.1	0.098	130.94
	.2	0.274	47.83		.2	0.178	68.03		.2	0.102	122.44
	.4	0.285	44.09		.4	0.182	66.60		.4	0.118	111.25
	.6	0.291	41.10		.6	0.182	64.17		.6	0.109	114.90
	.8	0.285	40.62		.8	0.175	65.02		.8	0.098	104.75
	1.0	0.275	39.83		1	0.168	65.07		1	0.094	114.30

Table 7. Probabilité d'arrivée de requête de service, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie avec mémoire pour $n=3$, $d=5, 10$ et 20 cycles.

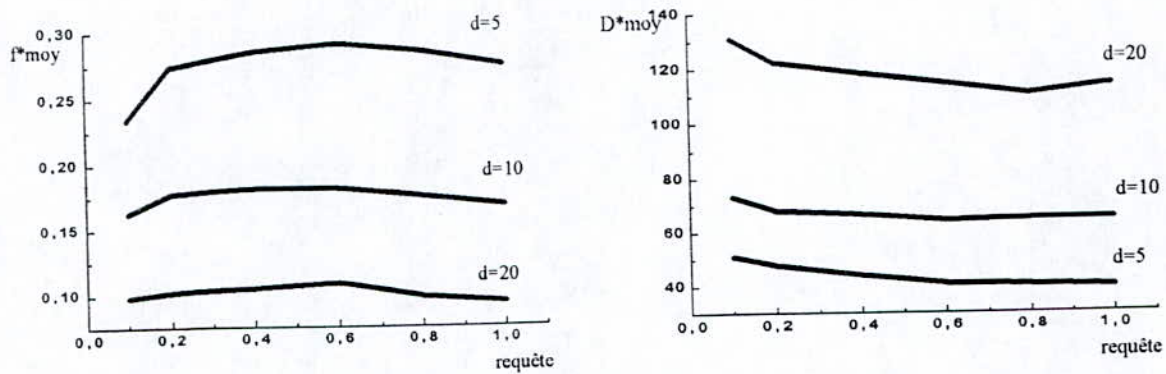


Fig. 5.11 Probabilité d'arrivée de requête de service, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie avec mémoire pour $n=3$.

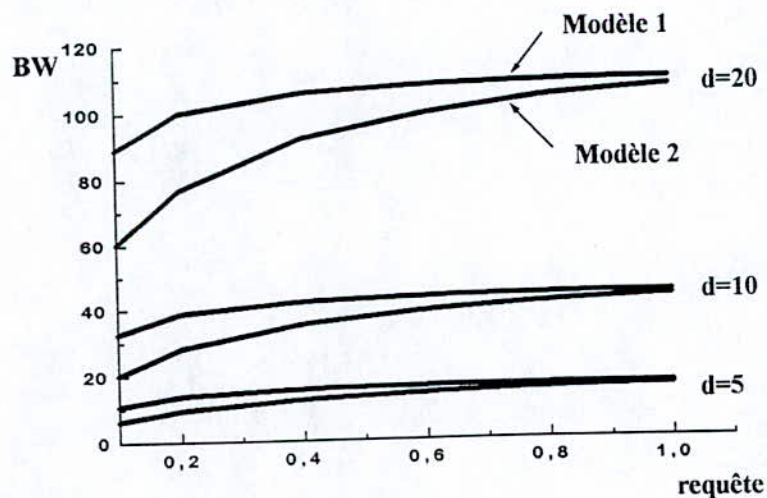


Fig. 5.12 Comparaison de la bande passante entre le modèle 1 et le modèle 2 sans mémoire pour $n=3$.

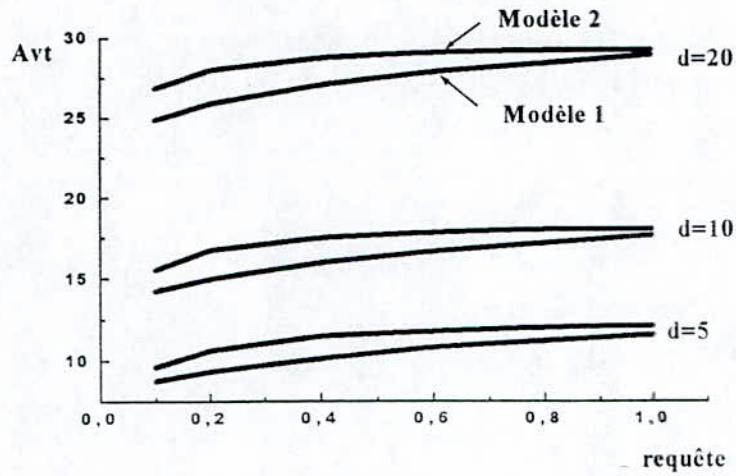


Fig. 5.13 Comparaison du temps moyen de service entre le modèle 1 et modèle 2 sans mémoire pour $n=3$.

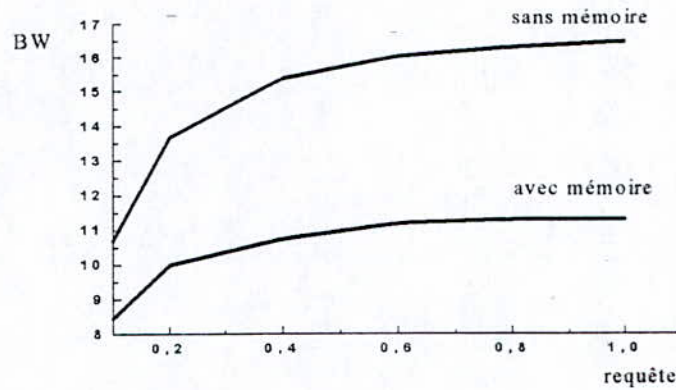


Fig. 5.14 Comparaison de la BW entre le modèle 1 sans mémoire et le modèle avec mémoire pour $n=3$ et $d=5$ cycles.

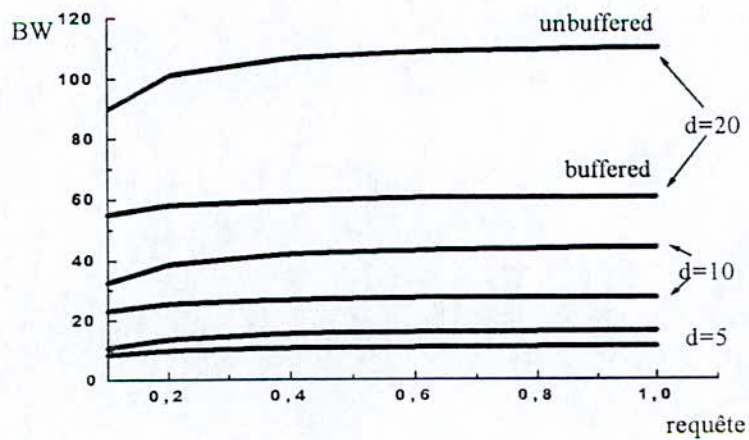


Fig. 5.15 Comparaison de la BW entre le modèle 1 sans mémoire et le modèle avec mémoire ($n=3$).

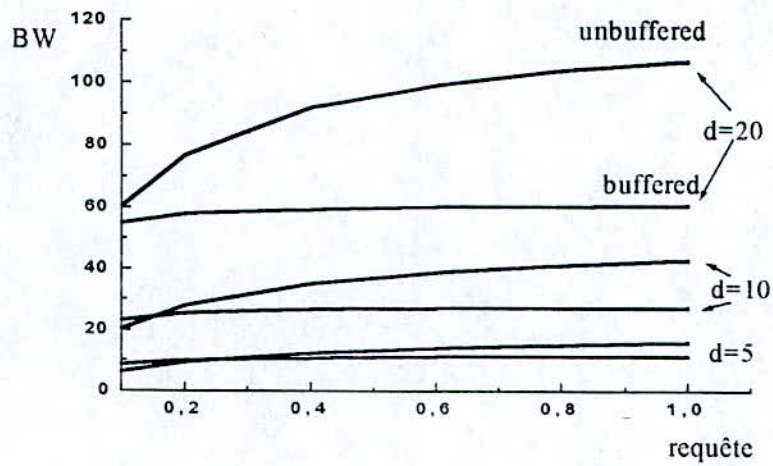


Fig. 5.16 Comparaison de la BW entre le modèle 2 sans mémoire et le modèle avec mémoire (n=3).

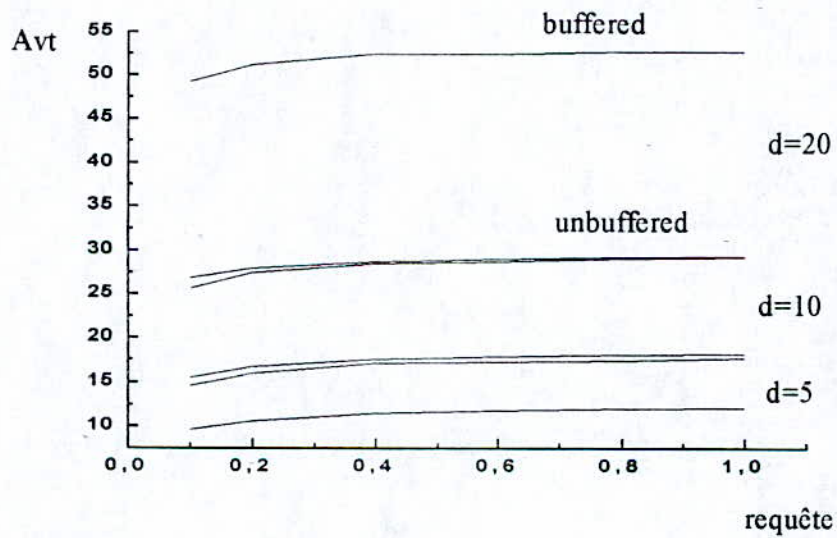


Fig. 5.17 Comparaison du Avt entre le modèle 1 sans mémoire et le modèle avec mémoire (n=3).

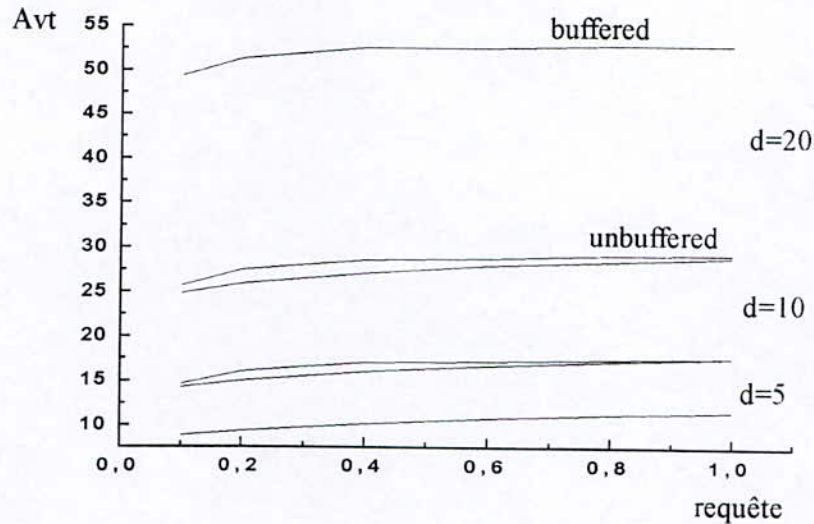


Fig. 5.18 Comparaison du Avt entre le modèle 2 sans mémoire et le modèle avec mémoire ($n=3$).

5.5 Discussion

L'application de cette modélisation 1 selon la stratégie sans mémoire nous a permis de reconstituer le modèle de référence [WU 92] en utilisant les RdPS qui sont un autre outil mathématique différent de celui utilisé dans l'article. Nous avons établi une sélection des paramètres suivants: la probabilité d'arrivée de requête; $r = 0.1, 0.2, 0.4, 0.6, 0.8, 1.0$ par cycle; le temps de transfert des données; $d = 5, 10, 20$ cycles. Ce choix des paramètres permet de mieux apprécier l'évaluation des performances des réseaux d'interconnexions.

La figure 5.2 montre l'évolution de la bande passante du modèle 1 sans mémoire. Nous constatons que l'écart d'évolution des courbes de la BW n'est pas linéaire par rapport au temps de transfert des données. C'est à dire la BW pour $d=10$ cycles n'est pas le double pour $d=5$ cycles. Cette observation est d'ailleurs valable pour tous les modèles présentés. La BW est faible pour $d=5$ cycles, par contre, elle est très significative pour $d=20$ cycles. Cela veut dire que le réseau offre une meilleure disponibilité de commutation pour des longs messages. L'accroissement de la BW est important pour $d \leq 0.2$ par cycle et reste à peu près constante pour $d > 0.2$ par cycle. Ces mêmes constatations se confirment pour le temps moyen de service, les fréquences moyennes de franchissement et les temps moyens de séjours. Plus la BW et la fréquence moyenne de franchissement sont larges, plus l'Avt et le temps moyen de séjour sont faibles car il y a une meilleure fluidité d'information dans le réseau. La Table 3 et la figure 5.4 montrent une comparaison des temps moyens de service entre le modèle 1 sans mémoire et l'article de référence [WU 92]. La Table 3 donne les résultats des deux approches avec la

différence en %. La plus petite et la plus grande différence sont d'environ 3.4% et 7.5% respectivement. Cette différence s'explique par l'ajout de probabilité que nous avons effectué sur notre modèle et le choix de méthode de résolution des probabilités.

La figure 5.5 montre l'allure de la BW et de l'Avt. en fonction de n pour d fixe (=5 cycles). Nous constatons qu'il y a toujours cette non-linéarité entre les écarts de n . Plus la taille du réseau est grande plus la BW est grande.

Concernant le modèle 2 sans mémoire et le modèle avec mémoire, l'allure des courbes de la BW et de l'Avt. revêt la même forme que celle du modèle 1. Leurs graphes respectifs sont donnés par les figures 5.7, 5.8 et les figures 5.10, 5.11.

Les figures 5.12 et 5.13 montrent une comparaison entre le modèle 1 et 2 sans mémoire. D'une part, le modèle 1 offre une meilleure BW que le modèle 2. Ceci est valable pour une probabilité d'arrivée des requêtes au réseau inférieure à 0.8 par cycle. Car par la suite les deux modèles deviennent presque identiques. Cela veut dire que la probabilité de défaillance des boîtes de commutations du premier étage est très faible. D'autre part, nous remarquons que l'écart est très significatif entre les deux modèles pour une probabilité d'arrivée des requêtes égale à 0.1 par cycle. Cet écart est plus important quand le temps de transfert des données est grand, particulièrement pour $d = 20$ cycles. L'allure des courbes devient presque identique pour une probabilité d'arrivée des requêtes égale à 1 par cycle. Ceci est valable pour les temps de transferts des données choisis, tels que le montrent les figures 5.12 et 5.13. Cette dernière observation nous permet de dire que les modèles 1 et 2 présentent les mêmes performances pour une probabilité d'arrivée des requêtes $r \geq 0.8$ par cycle.

Concernant la comparaison entre le modèle sans mémoire et le modèle avec mémoire, la figure 5.14 montre que la BW du modèle sans mémoire est meilleure que le modèle avec mémoire pour $n=3$ et $d=5$. Nous pouvons même généraliser cette conclusion pour $d=10$ et 20 cycles, tel que le montre la figure 5.15.

La figure 5.16 montre que la BW du modèle 2 sans mémoire est meilleure que le modèle avec mémoire uniquement pour $r > 0.2$ par cycle et pour $d \leq 10$ cycles. Cela veut dire

quand la probabilité d'arrivée des requêtes est grande, le premier étage est moins disponible pour acheminer les requêtes à leurs destinations. Par contre pour une probabilité d'arrivée égale à 0.2 par cycle, les deux modèles présentent les mêmes performances, sauf pour $d > 10$ cycles.

Aussi, nous observons que l'accroissement de la bande passante dans la stratégie sans mémoire est plus significatif que dans la stratégie avec mémoire. Par exemple, pour des valeurs de $n=3$ et $d=20$, la $BW_{UNBUFFERED}$ change de 60.18 à 106.91. Cela correspond à un accroissement de 51.92. De la même façon, si nous observons la $BW_{BUFFERED}$ dans les mêmes conditions, ces changements de la bande passante sont de 54.86 à 60.57. Cela correspond à un accroissement de 6.34. Donc le modèle sans mémoire est plus intéressant si nous choisissons une probabilité d'arrivée des requêtes qui varie de 0.4 à 1 cycle.

Donc le choix d'un réseau d'interconnexions fiable dépendra d'abord du type de stratégie utilisé, la taille du message (d), la dimension du réseau et enfin la probabilité d'arrivée des requêtes. Grâce aux figures comparatives entre les différents modèles, nous pourrions choisir les paramètres les mieux appropriés à notre réseau. Ces paramètres doivent donner un minimum de probabilité de blocage dans le réseau.

Les graphes des figures de la bande passante et du temps moyen de service ne montrent pas qu'il y a une linéarité entre la BW ou l'Avt. et la probabilité d'arrivée des requêtes dans le réseau. Nous concluons que les probabilités de connexions $P_{(n+1)}$ sont fonction implicite du temps de transfert d ce qui explique cette non linéarité obtenue dans ces figures.



**MODELISATION
DES RESEAUX
D'INTERCONNEXIONS
A UN SEUL ETAGE**

CHAPITRE VI

Modélisation des réseaux d'interconnexions à un seul étage

6.1 Introduction

Pour la modélisation des réseaux d'interconnexions à un seul étage nous avons utilisé l'autre forme d'hypothèse qui gère la BC. Nous savons que dans les RIP, les requêtes doivent obligatoirement traverser tous les étages du réseau pour établir la connexion entre le processeur source et la mémoire destination. Dans le cas des RIT, on parle de nombre de passe ou de tour, c'est à dire que les requêtes peuvent se connecter à partir du 1^{er} tour pour établir une liaison entre le processeur source et la mémoire destination. Cela veut dire que la requête n'est pas obligée de traverser toutes les boîtes de commutations pour établir la liaison.

Comme nous l'avons déjà défini préalablement, dans les réseaux d'interconnexions à plusieurs étages, si nous considérons N-processeurs reliés à l'entrée du RI et N-mémoires reliées à la sortie du RI, nous dirons que le nombre de BC d'un RIP réarrangeable avec N entrées et N sorties est de $O(N \log_2 N)$.

Plusieurs types de RI notamment les RIT [BUR 91], [HUA 87] et les RIP [AGR 85], [FEN 81] ont été proposés pour les réseaux d'interconnexions à systèmes informatiques distribués.

Un RIT est un RIP avec seulement un seul étage tel que le montre la figure 2.6 (page 21). Le message peut recirculer à travers les processeurs aussi souvent que nécessaire jusqu'à atteindre la destination désirée, parce que les sorties de chaque BC sont reliées en boucle fermée aux processeurs sources et certains/tous peuvent recirculer comme entrées si les requêtes n'ont pas encore atteint les destinations finales. Les requêtes arrivent dans le réseau à travers les chemins reliant les processeurs sources aux BC, tels que le montre la figure 2.6 (à gauche des BC du réseau). Les requêtes suivent leurs chemins vers la droite des BC. Chaque processeur source génère une requête qui établit un chemin unique entre l'entrée et la sortie. A chaque cycle les requêtes se déplacent du tour (i) au tour ($i+1$). Par exemple, quand le processeur source 1 veut envoyer un message vers la mémoire 7 (voir Fig. 2.6), deux cycles sont nécessaires pour établir la liaison: dans le cycle 1, les données sont envoyées de l'entrée 1 à la sortie 6 en commandant le commutateur dans la position 'croisée'; dans le cycle 2, les données sont envoyées de l'entrée 6 à la sortie 7, en commandant le commutateur dans la

position 'directe'. Alors, nous pouvons dire que la requête a établi deux passes ou deux tours pour atteindre sa destination.

Un RIT possède des performances comparables au réseau multiétage, et son coût est de seulement $O(N)$. En plus de la réduction du coût du hardware, les RIT possèdent un plus court chemin de liaison (shorter average path length) [BUR 91]. La longueur du path dans un RIT est définie comme le nombre de tour à travers le réseau nécessaire pour atteindre la destination. Logiquement, un tour dans un RIT est équivalent à un étage dans un RIP. Chaque requête doit aller à travers Log_2N étages pour atteindre sa destination dans un RIP. Par contre dans un RIT le nombre maximum de tours pour établir une connexion entre l'entrée et la sortie est inférieur ou, au plus, égale à Log_2N . Ceci est un avantage considérable pour ce type de réseau.

Dans notre approche, nous nous sommes intéressés au réseau beta (voir Fig. 2.6. e) proposé par Huang [HUA 87]. Dans cette modélisation des RIT [BEN 99], nous avons considéré une BC de 2×2 . Cette BC est l'élément essentiel des RI. Chaque entrée ou sortie de la BC possède une commande. Cette commande permet l'aiguillage en directe ou croisée de la commutation. En analysant de près cette BC, nous pouvons avoir 16 combinaisons possibles de connexions. Ces combinaisons sont représentées dans la figure 6.1.

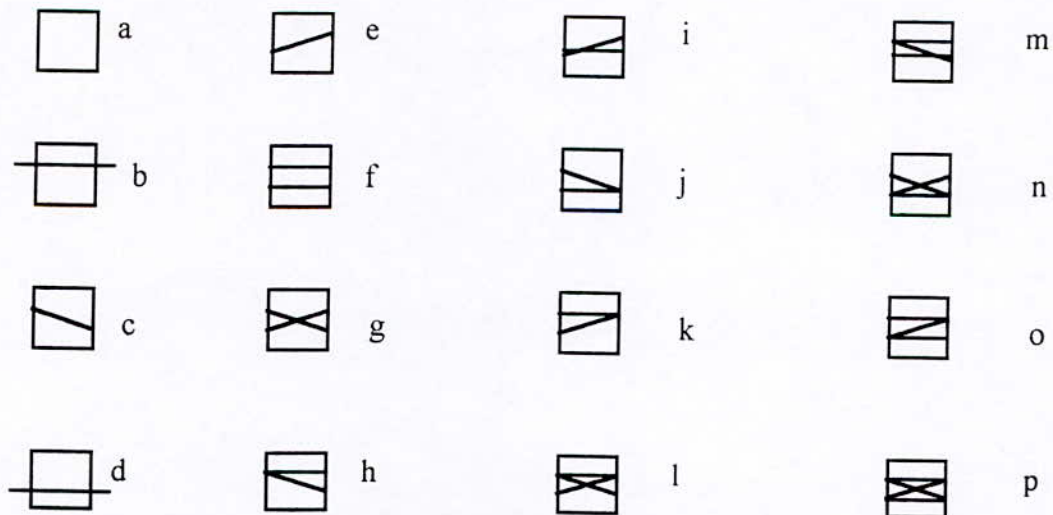


Fig. 6.1 L'ensemble des 16 états et leurs représentations symboliques dans une BC de 2×2 .

Nous pouvons observer que les 16 états qui sont représentés dans La figure 6.1, montre que les BC des cas de (a) à (i) ne représentent pas de conflit, cela veut dire que nous avons une

probabilité de $(9/16)$ qu'un conflit ne surviendra pas lors d'une commutation ainsi qu'une probabilité de $(7/16)$ où un conflit surviendra lors d'une commutation, lequel ensemble est représenté par les cas de (j) à (p).

Dans le domaine des RIT, nous avons développé des modélisations pour la stratégie sans mémoire et la stratégie avec mémoire.

6.2 Modélisation du RIT selon la stratégie sans mémoire

Dans cette section de l'approche, nous avons considéré les mêmes hypothèses que dans la modélisation des RIP (section 4.2 du chapitre IV). Dans le modèle de base, nous nous proposons d'adopter la stratégie sans mémoire. Cette stratégie est basée sur l'hypothèse (e), de la même manière que dans les RIP. Afin de mieux apprécier l'approche d'évaluation des performances des réseaux d'interconnexions, nous avons considéré les défaillances des BC, qui se traduisent par l'ajout d'une probabilité de disponibilité de la BC. Cette probabilité suit une loi de distribution exponentielle. Nous avons procédé exactement de la même manière que dans la modélisation des RIP.

Comme nous l'avons montré dans la modélisation des RIP, la faisabilité de la généralisation du modèle a fait l'objet d'une présentation dans le chapitre IV. Pour les RIT, le principe de la généralisation du modèle reste le même. Nous pourrions donc présenter directement l'application pour un RI à 3 tours. La figure 6.2 présente la modélisation des RIT pour $n=3$ (n représente cette fois-ci le nombre de tour) par les RdPS en adoptant la stratégie sans mémoire [BEN 98].

6.2.1 Description du modèle sans mémoire

La place S_0 est identifiée comme générateur de requête du réseau, sachant que la requête précédente générée par le processeur source est complète, et il n'y a pas d'arrivée de nouvelle requête. Les requêtes arrivent au réseau avec une probabilité r .

Les places de S_1 à S_3 sont identifiées comme les états du réseau à partir du 1^{er} tour jusqu'au 3^{ème} tour respectivement. Dans ces états, la requête arrive au réseau avec une probabilité r . Si cette requête se connecte à sa destination à partir du 1^{er} tour on validera la transition T_5 pour que la requête arrive à la place S_4 qui représente l'état de connexion. Si cette requête n'est pas connectée, elle passera au 2^{ème} tour qui est représenté par la place S_2 en validant la transition T_2 . Si cette place n'est pas disponible pour recevoir cette requête, on

reviendra à l'entrée du réseau, représenté par la place S_1 en validant la transition T_7 pour réinjecter cette requête au réseau, et ainsi de suite jusqu'au 3^{ème} tour où la requête doit trouver obligatoirement sa destination.

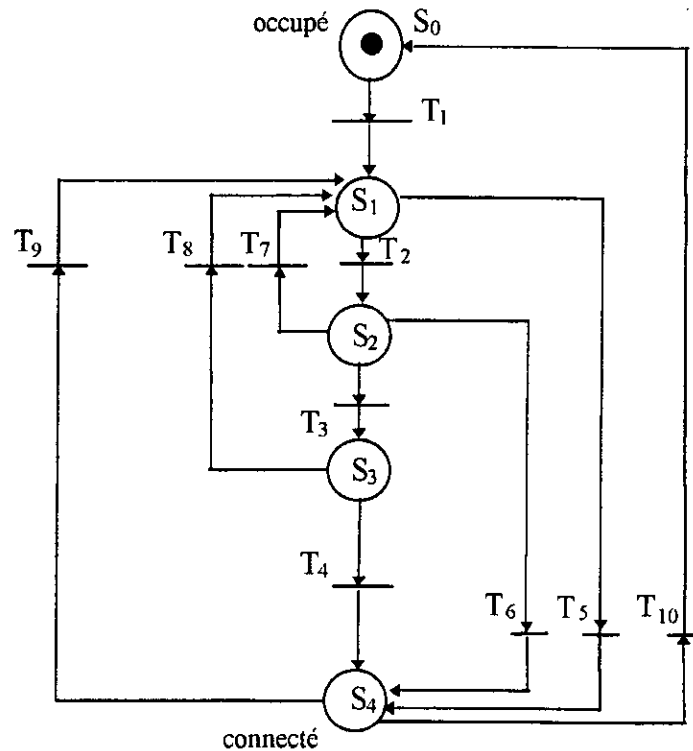


Fig. 6.2 Modélisation d'un RIT sans mémoire par les RdPS pour $n=3$

La place S_4 est utilisée pour informer le processeur source qu'un chemin est établi entre l'entrée et la sortie du réseau, c'est à dire qu'une liaison est faite entre le processeur demandeur et la mémoire demandée. Dans cet état, la requête a établi un chemin pour transférer les données, et ce chemin est maintenu pendant d cycles. Le processeur source sera informé de la connexion par la validation de la transition T_{10} .

Nous associerons à chacune des transitions une probabilité. Ces probabilités seront définies comme probabilité de passage, de blocage ou de connexion. Ces probabilités seront désignées comme suit;

λ_i : probabilité de passage d'un tour.

μ_i : probabilité de blocage.

ϕ_i : probabilité de connexion.

Nous associerons aux transitions de T_1 à T_{10} du modèle représenté dans la figure 6.2 les probabilités correspondantes de la manière suivante:

Le premier paramètre représente la transition et le deuxième paramètre représente la probabilité associée à la transition, nous obtenons les couples suivants; (T_1, r) ; (T_2, λ_1) ; (T_3, λ_2) ; (T_4, λ_3) ; (T_5, φ_1) ; (T_6, φ_2) ; (T_7, μ_1) ; (T_8, μ_2) ; $(T_9, r/d)$; $(T_{10}, (1-r)/d)$.

Nous exprimerons l'ensemble des probabilités avec les trois paramètres suivants: la dimension du réseau (n), la probabilité d'arrivée des requêtes (r) et le temps de transfert des données (d).

Nous pourrions déduire quelques probabilités apparentes;

$P(S_1/S_0) = r$; probabilité d'arrivée des requêtes au réseau.

$P(S_2/S_1) = \lambda_1$; probabilité de passage de la requête de la place S_1 à la place S_2 .

$P(S_1/S_2) = \mu_1$; probabilité de blocage de la requête entre la place S_1 et la place S_2 .

$P(S_4/S_1) = \varphi_1$; probabilité de connexion.

$P(S_0/S_4) = (1-r)/d$; Quand une requête est dans l'état de connexion; cette requête sera complète avec une probabilité de $1/d$. Le nouvel état du réseau sera aussi bien dans l'état occupé avec la probabilité $(1-r)$ que dans celui où on aura une arrivée d'une nouvelle requête à l'étage 1 avec une probabilité r .

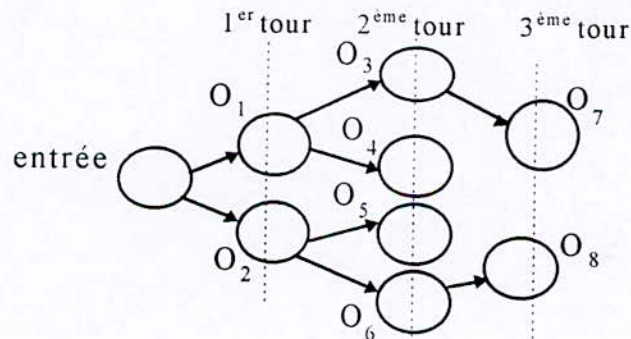


Fig. 6.3 Construction d'un arbre binaire d'une entrée et ses possibilités de sorties.

Grâce aux différentes possibilités de commutation d'une BC représentée dans la figure 6.1 et en tenant compte du modèle sans mémoire représenté pour $n=3$ de la figure 6.2, nous analyserons les différentes possibilités de connexion d'une requête à travers le réseau. Ce qui nous ramène à construire un arbre binaire qui permet de mieux comprendre la possibilité de

connexion d'une requête à l'entrée du réseau. Cet arbre binaire qui est représenté dans la figure 6.3, nous permettra de calculer les probabilités de passages, de blocages et de connexions.

Essayons d'analyser une connexion d'une requête qui arrive à l'entrée du réseau à 3 tours. Cette requête peut se connecter directement au 1^{er} tour en contrôlant la BC en directe ou en croisé. Cette connexion est représentée par les sorties O_1 et O_2 de la figure 6.3. Si cette requête n'est pas connectée au 1^{er} tour, elle poursuivra son chemin pour essayer de se connecter au 2^{ème} tour. Dans le cas où cette requête provient de la sortie O_1 , la BC commutera en directe ou en croisé vers les sorties O_3 et O_4 respectivement. Dans le cas où cette requête provient de la sortie O_2 , la BC commutera en directe ou en croisé vers les sorties O_5 et O_6 respectivement. Donc, les sorties O_3 , O_4 , O_5 et O_6 représentent les états de connexion au 2^{ème} tour. Si cette requête n'est pas connectée au 2^{ème} tour, elle poursuivra son chemin pour essayer de se connecter au 3^{ème} tour. Logiquement la requête doit trouver sa connexion dans ce dernier tour. Si la requête arrive par la sortie O_3 elle se connectera à la sortie O_7 , mais si elle arrive par la sortie O_6 elle se connectera à la sortie O_8 . Pour alléger la représentation de l'arbre binaire, nous n'avons pas représenté les possibilités de sorties des états O_4 et O_5 . Donc, les sorties O_7 et O_8 représentent les états de connexion au 3^{ème} tour.

Nous constaterons que nous ne pouvons avoir que 8 états tels que représentés par l'arbre binaire d'une entrée pour un réseau à 3 tours. Sur ces 8 possibilités d'états, nous aurons une probabilité de connexion au 1^{er} tour égale à $2/8$. Ce sont les sorties représentées par O_1 et O_2 de la figure 6.3. La probabilité de se connecter au 2^{ème} tour sera égale à $4/8$. Ce sont les sorties représentées par O_3 , O_4 , O_5 et O_6 . La probabilité de se connecter au 3^{ème} tour sera égale à $2/8$. Ce sont les sorties représentées par O_7 et O_8 .

Grâce à ces probabilités, nous déduirons les probabilités de passages, de blocages et de connexions du modèle comme suit:

λ_1 : la probabilité de passage d'une requête de la première BC à la seconde BC; cela signifie que le message est dans l'état S_1 et essaie de se connecter à l'état S_2 .

$$\lambda_1 = (6/8) \times (9/16).$$

λ_2 : la probabilité de passage d'une requête de la seconde BC à la troisième BC; cela signifie que le message est dans l'état S_2 et essaie de se connecter à l'état S_3 .

$$\lambda_2 = (2/6) \times (9/16).$$

λ_3 : la probabilité de passage d'une requête de la troisième BC à la quatrième BC; cela signifie que le message est dans l'état S_3 et essaie de se connecter à l'état S_4 .

$$\lambda_3 = (9/16).$$

φ_1 : la probabilité de connexion en un tour à partir de la première BC.

$$\varphi_1 = (2/8) \times (9/16).$$

φ_2 : la probabilité de connexion en deux tours à partir de la seconde BC.

$$\varphi_2 = (2/8) \times (9/16).$$

μ_1 : la probabilité de blocage dans la seconde BC.

$$\mu_1 = 7/16.$$

μ_2 : la probabilité de blocage dans la troisième BC.

$$\mu_2 = 7/16.$$

Grâce à l'algorithme présenté au chapitre IV (voir page 45), qui permet de construire le générateur de processus markovien associé au RdPS, nous construirons la matrice associée aux chaînes de Markov du modèle, notée Q , comme suit:

$$Q = \begin{bmatrix} -r & r & 0 & 0 & 0 \\ 0 & -(\lambda_1 + \varphi_1) & \lambda_1 & 0 & \varphi_1 \\ 0 & \mu_1 & -(\lambda_2 + \mu_1 + \varphi_2) & \lambda_2 & \varphi_2 \\ 0 & \mu_2 & 0 & -(\lambda_3 + \mu_2) & \lambda_3 \\ (1-r)/d & r/d & 0 & 0 & -1/d \end{bmatrix}$$

6.3 Modélisation du RIT selon la stratégie avec mémoire

Dans cette section de l'approche, nous maintiendrons les mêmes hypothèses que dans la modélisation des RIT selon la stratégie sans mémoire en modifiant uniquement l'hypothèse (e), de la même manière que dans la modélisation avec mémoire des RIP.

hypothèse (e) : Si deux requêtes arrivent dans le même cycle à l'entrée de la BC et demandent la même sortie, alors une requête sera commutée vers la sortie désirée par contre l'autre requête sera emmagasinée dans une mémoire. Cette mémoire peut prendre un seul message à la fois. De cette manière, si une requête est bloquée, nous conserverons une partie du chemin

préétabli dans le réseau. Quand la BC sera libérée, la requête bloquée dans la mémoire pourra continuer son chemin sans toute fois recommencer la connexion au début du réseau.

Pour les RIT, le principe de la généralisation du modèle reste la même que dans le cas des RIP. Nous pourrions donc présenter directement l'application pour un RI à 3 tours. La figure 6.4 présente la modélisation des RIT pour $n=3$ (n représente cette fois ci le nombre de tour) par les RdPS en adoptant la stratégie avec mémoire.

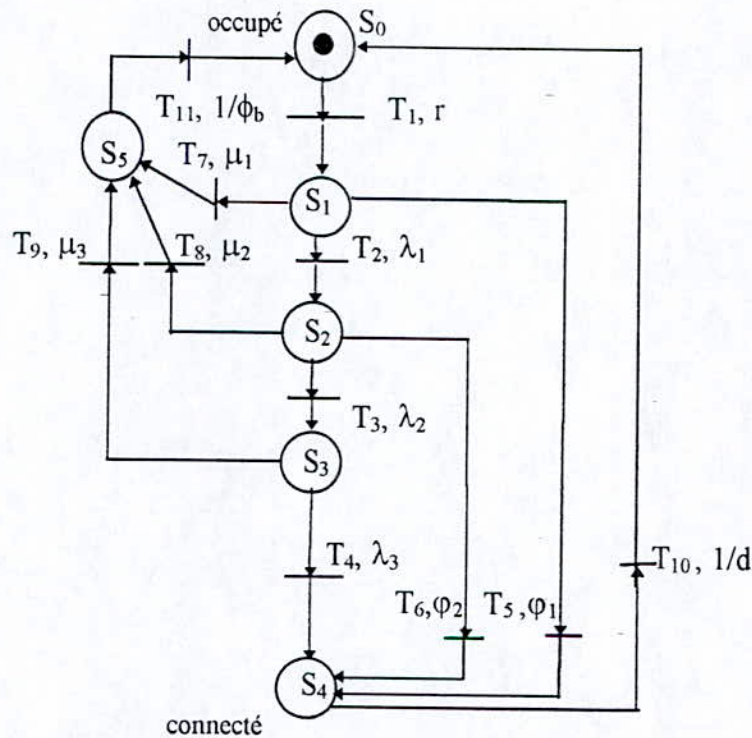


Fig. 6.4 Modélisation du RIT selon la stratégie avec mémoire par les RdPS pour $n=3$.

6.3.1 Description du modèle avec mémoire

Les places de S_0 à S_4 sont identifiées de la même manière que dans le modèle sans mémoire. Dans cette stratégie, la requête arrive au réseau avec une probabilité r . Si cette requête se connecte à sa destination à partir du 1^{er} tour on validera la transition T_5 pour que la requête arrive à la place S_4 qui représente l'état de connexion. Si cette requête n'est pas connectée, elle passera au 2^{ème} tour qui est représenté par la place S_2 en validant la transition T_2 . Si cette place n'est pas disponible pour recevoir cette requête elle sera mémorisée dans place S_5 en validant la transition T_7 . Cette place est identifiée comme une place d'attente

(waiting place). La requête sera emmagasinée dans cette place jusqu'à la libération de la place S_2 . Quand celle-ci est libérée, la requête poursuivra son chemin jusqu'à l'établissement de la connexion.

La place S_4 est utilisée pour informer le processeur source qu'un chemin est établi entre l'entrée et la sortie du réseau, c'est à dire qu'une liaison est faite entre le processeur demandeur et la mémoire demandée. Dans cet état, la requête a établi un chemin pour transférer les données, et ce chemin est maintenu pendant d cycles. Le processeur source sera informé de la connexion par la validation de la transition T_{10} .

De la même manière que dans la modélisation précédente, nous associerons à chacune des transitions une probabilité. Ces probabilités seront définies comme probabilité de passage, de blocage ou de connexion. Ces probabilités seront désignées comme suit;

λ_i : probabilité de passage d'un tour.

μ_i : probabilité de blocage.

φ_i : probabilité de connexion.

Nous associerons aux transitions de T_1 à T_{11} du modèle représenté dans la figure 6.4 les probabilités correspondantes de la manière suivante:

Le premier paramètre représente la transition et le deuxième paramètre représente la probabilité associée à la transition, nous obtenons les couples suivants: (T_1, r) ; (T_2, λ_1) ; (T_3, λ_2) ; (T_4, λ_3) ; (T_5, φ_1) ; (T_6, φ_2) ; (T_7, μ_1) ; (T_8, μ_2) ; (T_9, μ_3) ; $(T_{10}, 1/d)$, $(T_{11}, 1/\phi_b)$.

Nous exprimerons l'ensemble des probabilités avec les trois paramètres suivants: la dimension du réseau (n), la probabilité d'arrivée des requêtes (r) et le temps de transfert des données (d).

Nous déduirons quelques probabilités apparentes;

$P(S_1/S_0) = r$; probabilité d'arrivée des requêtes au réseau.

$P(S_2/S_1) = \lambda_1$; probabilité de passage de la requête de la place S_1 à la place S_2 .

$P(S_5/S_1) = \mu_1$; probabilité de blocage de la requête entre la place S_1 et la place S_2 .

$P(S_4/S_1) = \varphi_1$; probabilité de connexion.

$P(S_0/S_5) = 1/\phi_b$; probabilité moyenne d'attente.

$P(S_0/S_4) = 1/d$; Quand une requête est dans l'état de connexion; cette requête sera complète avec une probabilité de $1/d$.

Grâce aux différentes possibilités de commutation d'une BC représentée dans la figure 6.1, en se basant sur l'arbre binaire qui représente les possibilités de connexion d'une requête à l'entrée du réseau et en tenant compte du modèle avec mémoire représenté pour $n=3$ de la figure 6.4, nous calculerons les probabilités de passages, de blocages et de connexions comme suit;

λ_1 : la probabilité de passage d'une requête de la première BC à la seconde BC, cela signifie que le message est dans l'état S_1 et essaie de se connecter à l'état S_2 .

$$\lambda_1 = (6/8) \times (9/16).$$

λ_2 : la probabilité de passage d'une requête de la seconde BC à la troisième BC, cela signifie que le message est dans l'état S_2 et essaie de se connecter à l'état S_3 .

$$\lambda_2 = (2/6) \times (9/16).$$

λ_3 : la probabilité de passage d'une requête de la troisième BC à la quatrième BC, cela signifie que le message est dans l'état S_3 et essaie de se connecter à l'état S_4 .

$$\lambda_3 = (9/16).$$

φ_1 : la probabilité de connexion en un tour à partir de la première BC.

$$\varphi_1 = (2/8) \times (9/16).$$

φ_2 : la probabilité de connexion en deux tours à partir de la seconde BC.

$$\varphi_2 = (2/8) \times (9/16).$$

μ_1 : la probabilité de blocage dans la seconde BC.

$$\mu_1 = 7/16.$$

μ_2 : la probabilité de blocage dans la troisième BC.

$$\mu_2 = 7/16.$$

$1/\phi_b$: probabilité moyenne d'attente, cette probabilité est fixée à 1. Nous avons choisi cette valeur car la capacité de la mémoire est égale à la longueur du message.

Grâce à l'algorithme présenté au chapitre IV, qui permet de construire le générateur de processus markovien associé au RdPS, nous construirons la matrice associée aux chaînes de Markov du modèle, notée Q_m , comme suit;

$$P_j \cdot B = C$$

tel que P_j est le vecteur ligne défini par l'Eq. (n+1). C'est un vecteur ligne dont les éléments sont nuls sauf l'élément $C_j = 1$. Pour résoudre ce système, nous déterminons B^{-1} où la ligne j représente directement la solution du système. Pour l'inversion de la matrice B , l'algorithme de Jordan a été utilisé.

Algorithme de résolution

Etape 1:

Initialisation

n, d, r, ϕ_b introduction des paramètres taille du réseau,
longueur du message, probabilité d'arrivée des requêtes
et probabilité moyenne d'attente

$Q(i, j)$ construction de la matrice associée aux chaînes de Markov

Etape 2:

Changement de l'une des équations de (1) à (n)
avec l'équation (n+1)

Etape 3:

Construction du système

$$P_j \cdot B = C$$

ou P_j représente le vecteur ligne définie par l'équation (n+1)

Etape 4:

Détermination de B^{-1}
 en utilisant l'algorithme de Jordan

Etape 5:

La solution du système est donnée directement par
 la ligne j des éléments de la matrice B^{-1}

6.5 Evaluation du modèle sans mémoire

Grâce à l'algorithme de résolution, nous avons déterminé les probabilités de passages, de connexions, de blocages et des états. Les probabilités des états sont données dans la Table 7.

r	$d = 5$						$d = 10$						$d = 20$					
	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0
P_0	.53	.33	.16	.076	.029	0.0	.49	.23	.103	.048	.019	0.0	.28	.001	.061	.028	.017	0.0
P_1	.096	.14	.17	.19	.196	.26	.074	.096	.112	.12	.123	.125	.051	.06	.066	.069	.070	.071
P_2	.072	.10	.13	.14	.147	.15	.056	.072	.084	.089	.092	.094	.038	.045	.050	.052	.053	.053
P_3	.013	.19	.24	.026	.028	.028	.010	.013	.16	.017	.017	.018	.007	.009	.009	.010	.010	.010
P_4	.29	.41	.53	.57	.599	.62	.45	.58	.69	.73	.75	.76	.623	.738	.813	.842	.857	.866

Table 7. Probabilité des états du modèle sans mémoire des RIT pour $n=3$, $d=5,10$ et 20 cycles.

Nous utiliserons l'Eq. 5.6 pour la détermination de la bande passante. En ce qui concerne le temps moyen de service d'une requête celle-ci est donnée par l'Eq. 5.7. La signification de cette équation est que $Avt = d \cdot \{ \text{la probabilité que la requête soit à l'intérieure du réseau} \} / \{ \text{la probabilité de connexion} \}$. Dans un RIP, la requête doit traverser tous les étages du réseau pour pouvoir établir une connexion. Par contre dans un RIT, la requête peut se connecter dès le 1^{er} tour, 2^{ème} tour où le $n^{\text{ème}}$ tour. Si nous considérons un RIT à 8 entrées/ 8 sorties, la requête établira sa connexion au plus au 3^{ème} tour. Nous désignerons donc les

temps moyens de service pour 1 tour, 2 tours ou 3 tours par Avt_1 , Avt_2 ou Avt_3 respectivement [BEN 99]. Ainsi, nous dirons que le temps moyen de service d'une requête dans le modèle d'un RIT sans mémoire est de:

$$Avt = (Avt_1 + Avt_2 + Avt_3)/3 \quad (6.1)$$

avec $Avt_1 = d \cdot [1 - P(S_0) - P(S_2) - P(S_3)] / P(S_4)$ (6.2)

$$Avt_2 = d \cdot [1 - P(S_0) - P(S_3)] / P(S_4) \quad (6.3)$$

$$Avt_3 = d \cdot [1 - P(S_0)] / P(S_4) \quad (6.4)$$

Pour le modèle d'un RIT avec mémoire nous aurons :

$$Avt_1 = d \cdot [1 - P(S_0) - P(S_5) - P(S_2) - P(S_3)] / P(S_4) \quad (6.5)$$

$$Avt_2 = d \cdot [1 - P(S_0) - P(S_5) - P(S_3)] / P(S_4) \quad (6.6)$$

$$Avt_3 = d \cdot [1 - P(S_0) - P(S_5)] / P(S_4) \quad (6.7)$$

Nous avons résolu ces équations en faisant varier quelques paramètres pour pouvoir mieux apprécier les performances du réseau. Nous avons procédé à la sélection des valeurs de la manière suivante: probabilité d'arrivée des requêtes = 0.1, 0.2, 0.4, 0.6, 0.8, 1.0 par cycle; temps de transfert des données = 5, 10 et 20 cycles.

Pour chacune des valeurs de n , r et d , nous avons calculé les bandes passantes et les temps moyens de service qui sont présentés dans la Table 7.1, représentés par des graphes tels que le montre la figure 6.5.

d	r	avt	BW	d	r	avt	BW	d	r	avt	BW
5	.1		11.70	10	.1		36.20	20	.1		99.70
	.2	$Avt_1=6.64$	16.53		.2	$Avt_1=11.64$	47.79		.2	$Avt_1=21.64$	118.10
	.4	$Avt_2=7.87$	20.84		.4	$Avt_2=12.87$	54.80		.4	$Avt_2=22.87$	130.09
	.6	$Avt_3=8.10$	27.82		.6	$Avt_3=13.10$	58.12		.6	$Avt_3=23.10$	134.66
	.8		23.96		.8		59.94		.8		137.06
	1.0	$Avt=7.53$	24.70		1	$Avt=12.53$	61.08		1	$Avt=22.53$	138.54

Table 7.1. Probabilité d'arrivée de requête, temps moyen de service et bande passante dans la stratégie sans mémoire de la modélisation des RIT pour $n=3$, $d=5, 10$ et 20 cycles.

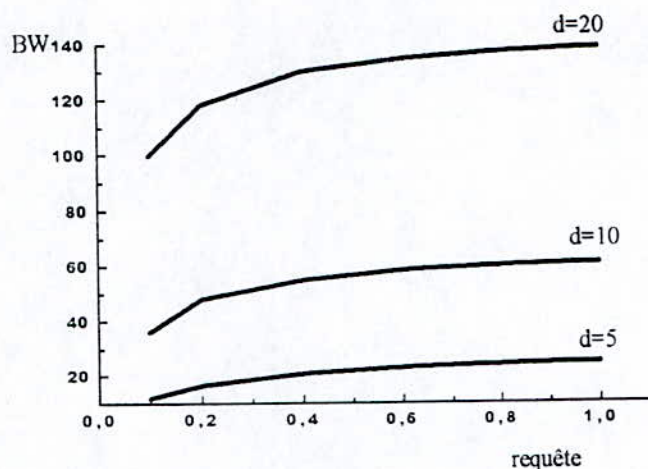


Fig 6.5 Probabilité d'arrivée de requête et bande passante dans la stratégie sans mémoire de la modélisation des RIT pour $n=3$.

Pour la détermination des fréquences de franchissement, avec l'Eq. 4.4, nous obtenons les équations suivantes;

$$\begin{aligned}
 f_1^* &= r \cdot P(S_0) & f_6^* &= \varphi_2 \cdot P(S_2) \\
 f_2^* &= \lambda_1 \cdot P(S_1) & f_7^* &= \mu_1 \cdot P(S_2) \\
 f_3^* &= \lambda_2 \cdot P(S_2) & f_8^* &= \mu_2 \cdot P(S_3) \\
 f_4^* &= \lambda_3 \cdot P(S_3) & f_9^* &= r/d \cdot P(S_4) \\
 f_5^* &= \varphi_1 \cdot P(S_1) & f_{10}^* &= (1-r)/d \cdot P(S_4)
 \end{aligned} \tag{6.8}$$

La conservation du flux de marquage donnée est vérifiée car nous avons;

$$\sum_{i=0,4} M^*(P_i) = P(S_0) + \sum_{i=1,3} P(S_i) + P(S_4)$$

En ce qui concerne le temps moyen de séjour des marques dans les places, avec l'Eq. 4.2, nous obtenons;

$$\begin{aligned}
 D^*(P_0) &= M^*(P_0) / f_{10}^* \\
 D^*(P_1) &= M^*(P_1) / (f_1^* + f_7^* + f_8^* + f_9^*) \\
 D^*(P_4) &= M^*(P_4) / (f_4^* + f_5^* + f_6^*) \\
 D^*(P_i) &= M^*(P_i) / f_i^* \quad \text{pour } i=2, 3
 \end{aligned} \tag{6.9}$$

Grâce aux Eq. 6.8 et 6.9 nous avons déterminé les fréquences moyennes de franchissements et les temps de séjour des marques dans les places et qui sont donnés dans la Table 8.

d	r	f*moy	D*moy	d	r	f*moy	D*moy	d	r	f*moy	D*moy
5	.1	0.346	32.86	10	.1	0.191	34.36	20	.1	0.176	51.41
	.2	0.459	25.413		.2	0.227	27.71		.2	0.119	41.39
	.4	0.571	20.308		.4	0.395	20.62		.4	0.148	45.10
	.6	0.376	16.414		.6	0.254	24.08		.6	0.145	43.77
	.8	0.384	15.50		.8	0.247	23.77		.8	0.146	41.17
	1.0	0.479	11.773		1	0.237	22.38		1	0.132	41.51

Table 8. Probabilité d'arrivée de requête, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie sans mémoire de la modélisation des RIT pour d=5,10 et 20 cycles.

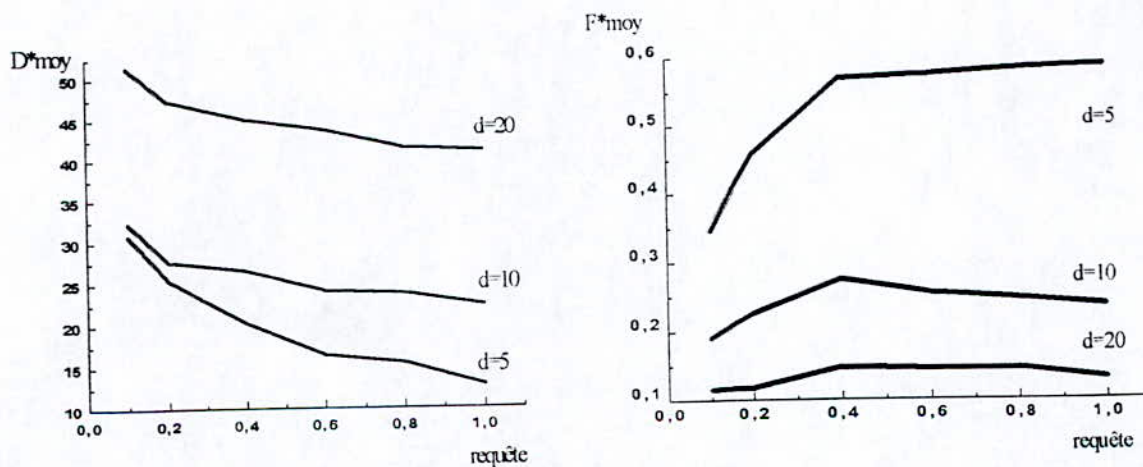


Fig. 6.6 Probabilité d'arrivée de requête, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie sans mémoire de la modélisation des RIT pour n=3.

6.6 Evaluation du modèle avec mémoire

Grâce à l'algorithme de résolution, nous avons déterminé les probabilités de passages, de connexions, de blocages et des états. Les probabilités des états sont données dans la Table 9.

	d = 5						d = 10						d = 20					
	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0	0.1	0.2	0.4	0.6	0.8	1.0
P_{00}	.72	.563	.392	.30	.244	.205	.641	.472	.31	.30	.183	.152	.53	.357	.217	.156	.122	.098
P_{10}	.072	.113	.157	.18	.195	.205	.064	.094	.124	.138	.146	.152	.053	.071	.087	.094	.97	.099
P_{20}	.030	.048	.066	.076	.082	.087	.007	.040	.055	.058	.062	.064	.022	.03	.037	.039	.041	.042
P_{30}	.006	.009	.012	.014	.015	.016	.005	.007	.010	.011	.012	.012	.004	.006	.007	.007	.077	.079
P_{40}	.24	.193	.269	.31	.335	.35	.22	.324	.424	.473	.502	.52	.361	.490	.596	.642	.668	.685
P_{50}	.47	.074	.103	.12	.13	.135	.042	.062	.081	.090	.096	.10	.035	.047	.057	.061	.064	.066

 Table 9 Probabilité des états du modèle avec mémoire des RIT pour $n=3$, $d=5, 10$ et 20 cycles.

Pour chacune des valeurs de n , r et d , nous avons calculé les bandes passantes et les temps moyens de service qui sont présentés dans la Table 9.1, et représentés par des graphes que le montre la figure 6.7.

n	r	avt	BW	d	r	avt	BW	d	r	avt	BW
3	.1		04.95	10	.1		17.61	20	.1		57.75
	.2	Avt ₁ =7.91	07.74		.2	Avt ₁ =13.91	25.93		.2	Avt ₁ =22.91	78.34
	.4	Avt ₂ =9.14	10.67		.4	Avt ₂ =14.14	33.95		.4	Avt ₂ =24.14	95.33
	.6	Avt ₃ =9.37	12.39		.6	Avt ₃ =14.37	37.84		.6	Avt ₃ =24.37	102.76
	.8		13.40		.8		40.15		.8		106.93
	1.0	Avt=8.81	14.09		1	Avt=13.81	41.67		1	Avt=23.81	109.59

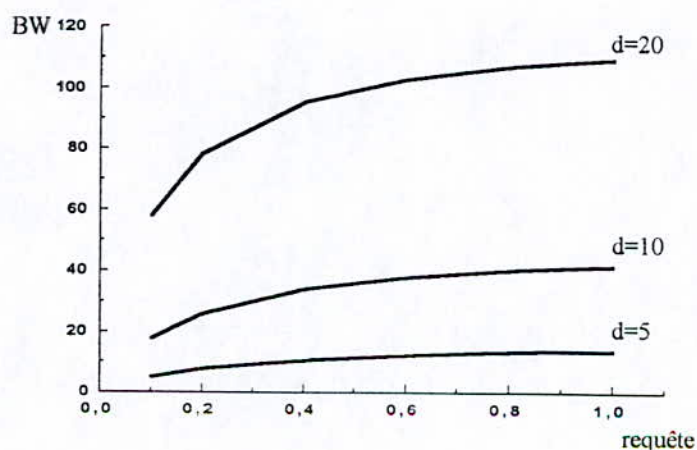
 Table 9.1 Probabilité d'arrivée de requête, temps moyen de service et bande passante dans la stratégie avec mémoire pour $n=3$, $d=5, 10$ et 20 cycles.


Figure 6.7 Probabilité d'arrivée de requête et bande passante dans la stratégie avec mémoire pour

Pour la détermination des fréquences de franchissement, avec l'Eq. 4.4, nous obtenons les équations suivantes;

$$\begin{aligned}
 f_1^* &= r \cdot P(S_0) & f_7^* &= \mu_1 \cdot P(S_1) \\
 f_2^* &= \lambda_1 \cdot P(S_1) & f_8^* &= \mu_2 \cdot P(S_2) \\
 f_3^* &= \lambda_2 \cdot P(S_2) & f_9^* &= \mu_3 \cdot P(S_3) \\
 f_4^* &= \lambda_3 \cdot P(S_3) & f_{10}^* &= 1/d \cdot P(S_4) \\
 f_5^* &= \varphi_1 \cdot P(S_1) & f_{11}^* &= 1/\phi_b \cdot P(S_5) \\
 f_6^* &= \varphi_2 \cdot P(S_2) & &
 \end{aligned} \tag{6.10}$$

La conservation du flux de marquage donnée est vérifiée car nous avons;

$$\sum_{i=0,4} M^*(P_i) = P(S_0) + \sum_{i=1,3} (P(S_i) + P(S_{bi})) + P(S_4)$$

En ce qui concerne le temps moyen de séjour des marques dans les places, avec l'Eq. 4.2, nous obtenons;

$$\begin{aligned}
 D^*(P_0) &= M^*(P_0) / (f_{10}^* + f_{11}^*) \\
 D^*(P_4) &= M^*(P_4) / (f_4^* + f_5^* + f_6^*) \\
 D^*(P_5) &= M^*(P_5) / (f_7^* + f_8^* + f_9^*) \\
 D^*(P_i) &= M^*(P_i) / f_i^* \quad \text{pour } i = 1, 2, 3
 \end{aligned} \tag{6.11}$$

Grâce aux Eq. 6.10 et 6.11 nous avons déterminé les fréquences moyennes de franchissements et les temps de séjour des marques dans les places et qui sont donnés dans la Table 10.

d	r	f*moy	D*moy	d	r	f*moy	D*moy	d	r	f*moy	D*moy
5	.1	0.244	21.30	10	.1	0.202	31.21	20	.1	0.179	44.01
	.2	0.385	15.87		.2	0.319	23.12		.2	0.241	37.87
	.4	0.545	13.58		.4	0.421	20.64		.4	0.296	34.87
	.6	0.613	12.71		.6	0.468	19.59		.6	0.318	33.59
	.8	0.663	12.23		.8	0.499	18.82		.8	0.377	48.93
	1.0	0.698	12.04		1	0.512	16.41		1	0.409	24.24

Table 10. Probabilité d'arrivée de requête, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie avec mémoire pour n=3, d=5, 10 et 20 cycles.

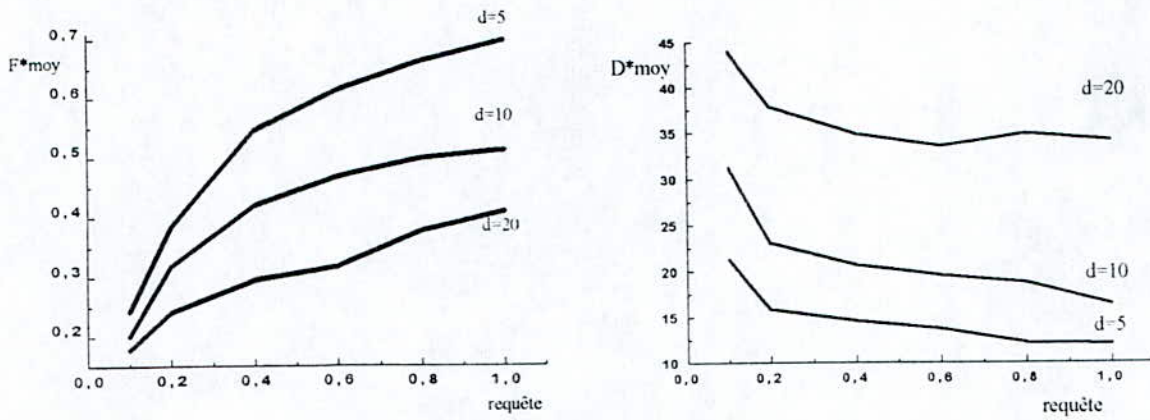


Fig. 6.8 Probabilité d'arrivée de requête, fréquence moyenne de franchissement et temps moyen de séjour dans la stratégie avec mémoire pour $n=3$.

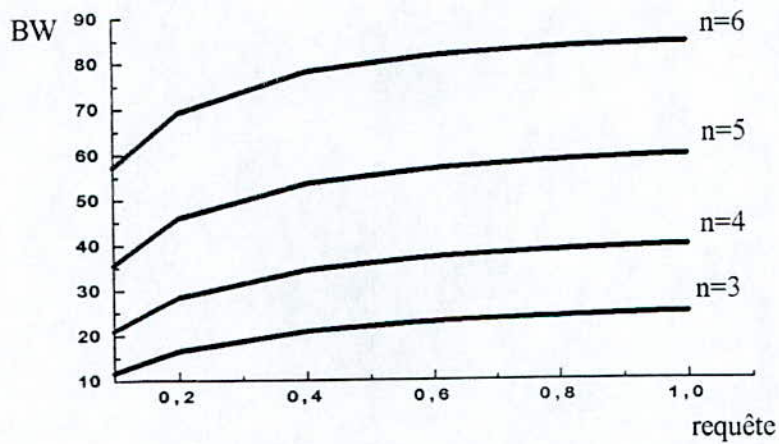


Fig. 6.9 Probabilité d'arrivée, bande passante du modèle sans mémoire pour $d=5$ cycles.

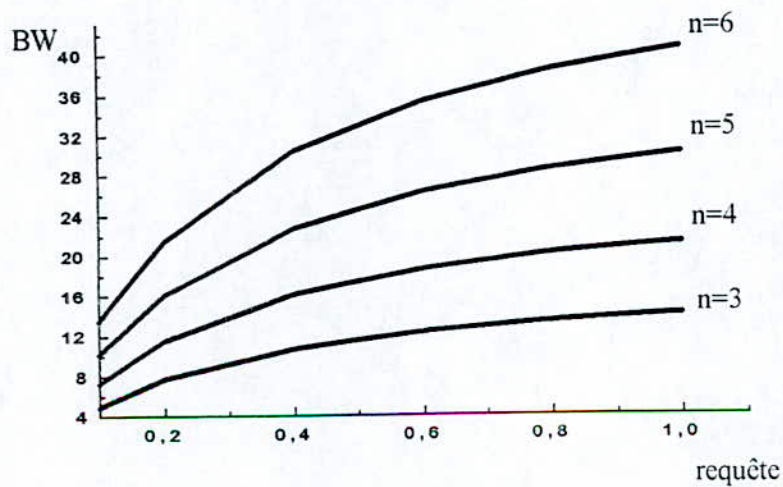


Fig. 6.10 Probabilité d'arrivée, bande passante du modèle avec mémoire pour $d=5$ cycles

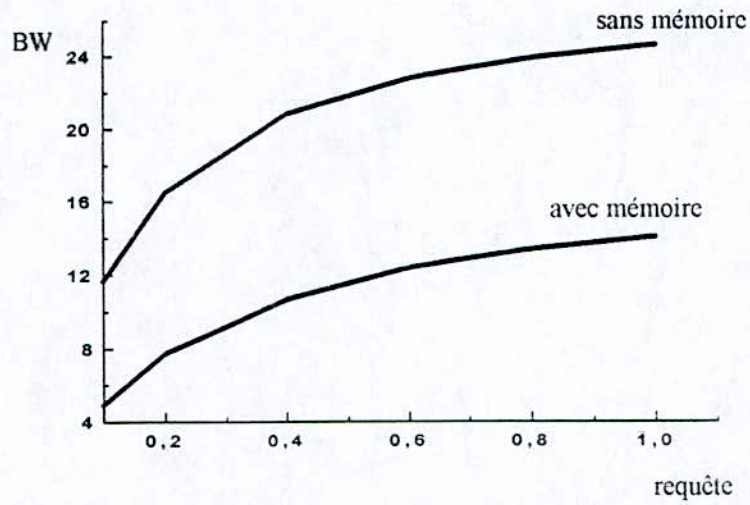
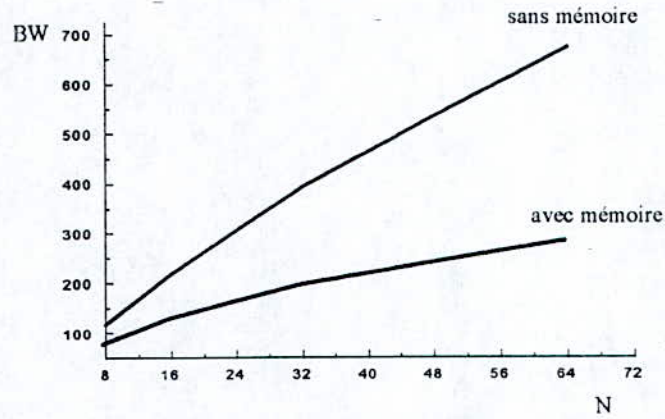
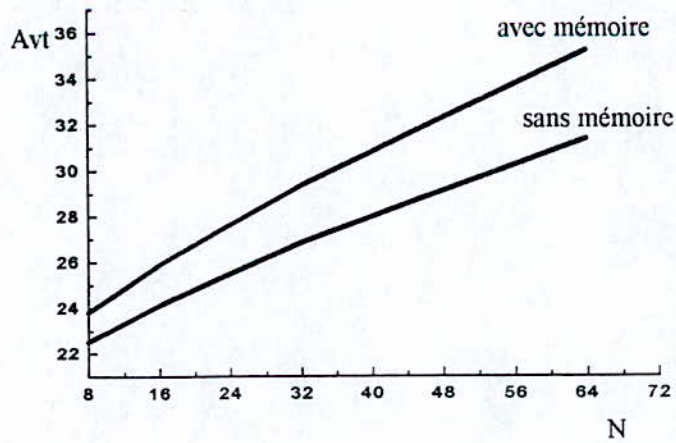


Fig. 6.11 Comparaison entre BW sans mémoire et avec mémoire pour $n=3$ et $d=5$ cycles



a) Comparaison des BW entre les deux stratégies $d=20$ cycles et $r=0.2$ par cycle



b) Comparaison des Avt entre les deux stratégies $d=20$ cycles et $r=0.2$ par cycle

Fig. 6.12 Comparaison des BW et Avt des 2 stratégies pour $d=20$ cycles et $r=0.2$ par cycle.

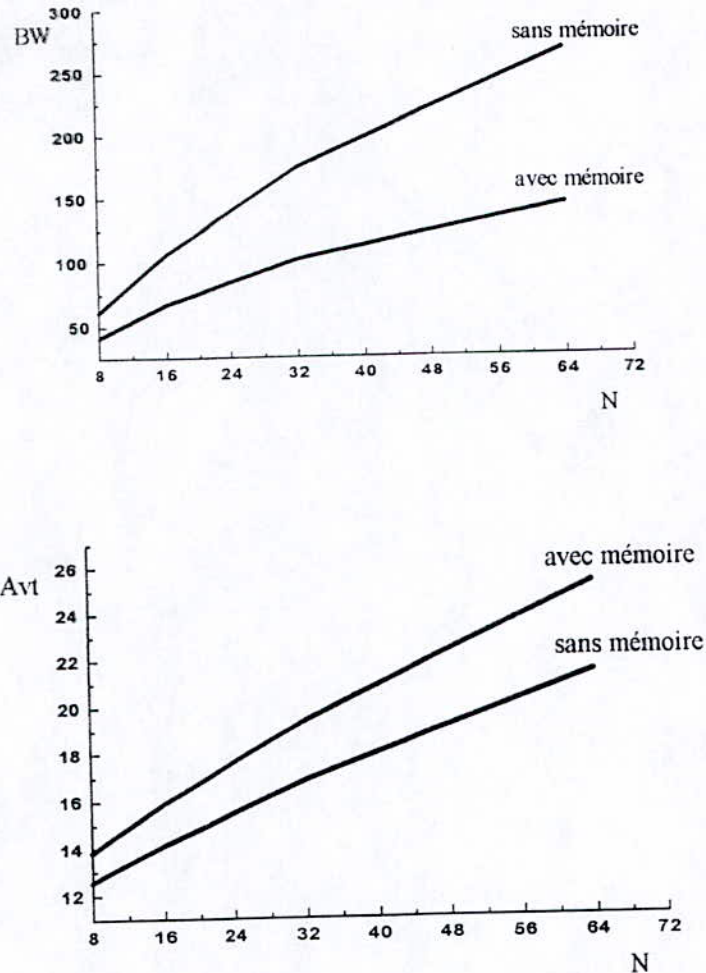


Fig. 6.13 Comparaison entre les 2 stratégies pour $d=10$ cycles et $r=1$ par cycle

6.7 Discussion

Nous avons procédé à la sélection suivante: la taille du réseau; $N = 8, 16, 32, 64$; probabilité d'arrivée des requêtes; $r = 0.1, 0.2, 0.4, 0.6, 0.8, 1.0$ par cycle; le temps de transfert des données; $d = 5, 10, 20$ cycles. La figure 6.5 montre l'évolution de la bande passante du modèle sans mémoire pour $N=8$ entrées. Nous observons que l'allure des courbes est de la même forme que dans la modélisation des réseaux d'interconnexions à plusieurs étages. La même observation faite sur les RIP reste valable sur les RIT, c'est à dire que l'écart de la BW n'est pas linéaire par rapport au temps de transfert des données. Par exemple; la BW pour $d=10$ cycles n'est pas le double pour $d=5$ cycles. Les fréquences moyennes de franchissements et les temps moyens de séjour du modèle sans mémoire sont représentés dans la figure 6.6. Les courbes confirment l'écart qui existe entre les divers temps de transferts.

Les figures 6.7 et 6.8 montrent l'évolution de la BW, fréquences moyennes de franchissements et les temps moyens de séjour du modèle avec mémoire pour $n=3$. Nous retrouvons la même forme des courbes avec un écart de la BW expressif de la même manière que dans le modèle sans mémoire.

La figure 6.10 montre la BW du modèle avec mémoire pour $d=5$ cycles des diverses tailles du réseau, allant de $N=8$ à $N=64$ entrées. Ces courbes montrent une très faible BW pour $r=0.1$ par cycle pour tout N . L'allure de la courbe croît d'une manière tangentielle en fonction de la probabilité d'arrivée des requêtes. Dans ce cas nous n'avons pas une sorte de saturation, telle que dans le cas des RIP.

La figure 6.11 montre une comparaison entre les deux stratégies pour $N=8$ entrées et $d=5$ cycles. Nous observons que la stratégie sans mémoire est meilleure que la stratégie avec mémoire. D'ailleurs, cette observation se généralise pour toute taille du réseau telle que le montre la figure 6.12; où nous avons pris $d = 20$ cycles et $r = 0.2$ par cycle. Afin de confirmer cette observation et montrer que le temps de transfert des données et la probabilité d'arrivée des requêtes ne peuvent pas inverser les performances, nous avons tracé d'autres courbes telles que le montre la figure 6.13; où nous avons pris $d = 10$ cycles et $r = 1$ par cycle.

Si nous considérons le cas de $N = 8$ et $d = 5$ cycles, la variation des bandes passantes correspondantes est de 11.70 à 24.70 telle que le montre la Table 7.1. Ce changement correspond à un accroissement de 14.4. De la même façon, si nous considérons $N = 64$ et $d = 5$ cycles, l'accroissement de la bande passante serait de 30.19. Nous dirons que la bande passante d'un réseau de taille 64 est à peu près deux fois plus grande que celle du réseau de taille 8. Donc, l'accroissement de la bande passante n'est pas significatif pour des transferts de temps courts, du fait que nous avons multiplié la taille du réseau par 8, nous obtenons ainsi uniquement le double de la bande passante. En revanche, si nous faisons la même comparaison avec des temps de transfert plus longs; $d = 20$ cycles, nous obtenons, pour $N = 8$, $BW_{\min} = 99.70$, $BW_{\max} = 138.54$, un accroissement de la bande passante de 43.16, tel qu'exposé par la Table 7.1. Pour $N = 64$, nous avons $BW_{\min} = 595.92$, $BW_{\max} = 753.86$, l'accroissement de la bande passante est de 175.49. Donc, cet accroissement est beaucoup plus important que pour $d = 5$ cycles. Nous pourrions dire que le réseau à un seul étage s'adapte mieux pour des temps de transferts longs. D'autre part nous observons que l'accroissement de la bande passante est rapide pour une probabilité d'arrivée des requêtes $r \leq 0.4$ par cycle, par la suite elle devient presque constante.



CONCLUSION

CONCLUSION

Les réseaux de Petri stochastiques peuvent être employés dans la modélisation et l'évaluation des réseaux d'interconnexions dynamiques. L'évaluation des performances est fonction de certains paramètres du réseau tels que la topologie, la taille du réseau (n), le temps de transfert des données (d) et la probabilité d'arrivée des requêtes (r). Notre approche permet l'évaluation des performances des RIP et des RIT en présence de conflit. Ce conflit a été géré selon deux stratégies définies comme stratégie sans mémoire et stratégie avec mémoire. Un modèle de base est proposé. Ce modèle peut être facilement généralisé. Le modèle **beta** (pour les RIT) ou le modèle **baseline** (pour les RIP) sont choisis à titre d'exemple dans notre approche. Le nombre de place et de transition est minutieusement bien étudié afin de minimiser le plus possible les combinaisons des états.

L'analyse des RdPS se ramène à celle d'un processus de Markov homogène à espace d'état discret à temps continu. Nous nous sommes limités au processus de Markov à espace d'état fini. Cet espace s'appelle chaîne de Markov, qui s'obtient par la construction du graphe de marquage accessible en associant une probabilité de franchissement. Les modèles proposés sont des modèles qui simplifient et limitent considérablement le nombre d'état de la chaîne de Markov tout en permettant une prise en charge du problème d'interconnexion. Grâce à la représentation d'un étage ou tour par une place, l'explosion combinatoire des états de la chaîne de Markov se trouve réduite. Les modèles s'adaptent facilement à l'extensibilité du réseau tout en gardant les propriétés de fonctionnement du réseau.

Les résultats obtenus montrent qu'il n'y a pas de linéarité, d'une part, entre la BW ou le Avt et la taille du réseau, et d'autre part, entre la BW ou le Avt et la longueur du message. L'analyse des modèles 1 et 2 des RIP selon la stratégie sans mémoire montre qu'ils sont presque identiques. Nous retenons que la stratégie sans mémoire est mieux adaptée aux RI dynamiques, du fait qu'elle offre une meilleure fluidité de passage de messages à travers le réseau. Nous montrons que les probabilités de passages sont meilleures au début du réseau qu'à la fin du réseau, pour toutes les valeurs de r ou d . Cela s'explique par le fait qu'il y a une meilleure accessibilité au réseau et que les conflits surviennent intensivement vers la fin du réseau. Concernant les modèles des RIP et les modèles des RIT, nous ne pouvons pas faire directement une comparaison du fait que nous avons considéré des hypothèses

différentes qui gèrent la BC. Mais nous retrouvons la même forme des courbes dans l'approche d'évaluation des performances des deux types de RI.

Notre approche nous a permis de mieux apprécier l'évaluation des performances des RI dynamiques. Cette évaluation prend en considération, d'une part, la défaillance de la BC, et d'autre part, la probabilité de passage ou de blocage des requêtes à travers le réseau selon la stratégie qui gère la BC.

BIBLIOGRAPHIE

- [AGR 85] D. P. Agrawal and J. S. Leu, 'Dynamic accessibility testing and path length optimization of multistage interconnection network', IEEE Trans. Comput. Vol. C-34, n° 3, pp. 255-266, (1985).
- [BEN 94] D. Benazzouz and A. Farah, 'Petri net model for a baseline multistage interconnection network', in Proc. International Conference, ICEEE, Oran, Algeria, pp. 146-152, (1994).
- [BEN 95] D. Benazzouz, 'Performance analysis of the baseline class of a MIN', in Proc. 17th IFIP-TC, International Conference on System Modelling and Optimization, Prague, Czech Republic, pp. 284-287, (1995).
- [BEN 96] D. Benazzouz and A. Farah, 'Petri model for baseline multistage interconnection network', AJOT, International Publication, serie B, vol. 12, n°1, pp.15-26, (1996).
- [BEN 98] D. Benazzouz and A. Farah, 'Performance evaluation of baseline MIN', in Proc. IMACS-IEEE on Computational Engineering in Systems Application', Nabeul-Hammamet, Tunisia, pp.350-356, (1998).
- [BEN 98] D. Benazzouz and A. Farah, 'The use of Petri nets in the performance evaluation of baseline MIN in uniform traffic environments', in Proc. CNP'98, 1^{er} colloque national sur la productique, pp.309-315, (1998).
- [BEN 98] D. Benazzouz and A. Farah, 'The use of Petri nets in the performance evaluation of shuffle-exchange network under uniform traffic distribution', the Arabian Journal for Science and Engineering, AJSE, vol. 23, n° 2B, pp. 253-263, (1998).
- [BEN 99] D. Benazzouz and A. Farah, 'Performance of an unbuffered beta model using stochastic Petri nets', International Journal of Modelling and Simulation, IASTED, vol. 9, n° 4, (1999) (in print).
- [BHU 87] L. N. Bhuyan, 'Interconnection networks for parallel and distributed processing', IEEE on Computer, vol. 20, n° 6, pp. 9-12, (1987).
- [BRA 83] G. W. Brams, '*Réseaux de Petri: théorie et pratique*', Ed. Masson, Paris, (1983).

- [BUR 91] I. R. Burke, C. Chen, T. Y. Lee and D. P. Agrawal, 'Performance analysis of single stage interconnection networks', IEEE Trans on computer, vol. 40, n° 3, pp 357-365, (1991).
- [DIN 91] J. Ding and L. N. Bhuyan, 'Performance evaluation of MINs with finite buffers', in Proc., Int. Conf. Parallel Processing, pp. 592-595, (1991).
- [DUN 90] R. Duncan, 'A survey of parallel computer architectures', IEEE Computer, pp. 5-16, Février (1990).
- [ETU 95] D. Etueimble, '*L'évolution des machines parallèles et massivement parallèles*', Thèse de Doctorat, université de Paris-Sud, (1995).
- [FEN 81] T. Feng, 'A survey of interconnection networks', IEEE on Computer vol. 14, n°12, pp. 12-27, (1981).
- [FLO 90] G. Florin, C. Fraise, and S. Natkin, 'Stochastic Petri nets : properties, application and tools. Rapport de recherche 05, CEPRIC-CNAM, (1990)
- [FLO 91] G. Florin, C. Fraize and S. Natkin, 'Stochastic Petri Net: Properties, Applications and Tools', Microelectron. reliability, 31, n° 4, pp. 669-697, (1991).
- [FLY 72] M. T. Flynn, 'Some computer organizations and their effectiveness', IEEE Trans. comp., vol. C-21, n°9, pp. 948-960, (1972).
- [FOR 85] J. M. Forneau, '*Modélisation des réseaux multitétages*', Thèse de PhD, Université Paris-sud, France, (1985).
- [FRA 95] P. Fraigniaud, 'Vers un principe de localité pour les communications dans les réseaux d'interconnexions', Thèse d'habilitation, Université de Lyon, laboratoire d'informatique du parallélisme, CNRS, Ecole Normale Supérieure de Lyon, (1995).
- [HUA 87] C. F. Huang & W. T. Chen, 'Fault tolerance single stage interconnection network', IEEE Trans. Computers, vol. C-36, n°5, pp 637-640, (1987).
- [HWA 93] K. Hwang, '*Advanced computer architecture*', Ed. Mc. Graw Hill, (1993).
- [LEI 85] C.E. Leiserson, 'Fat-Tree: Universal networks for hardware efficient supercomputing, in ICPP', IEEE computer, pp. 393-402, (1985).
- [MOL 81] M. K. Molloy, 'On the integration of delay and throughput measures in distributed processing models', PhD thesis, University of California, Los Angeles, (1981).
- [MUR 89] T. Murata, 'Petri nets; properties, analysis and applications', Proceedings of the IEEE Comp., vol. 77, n°4, pp. 541-580, (1989).

- [NAT 80] S. Natkin, '*Les réseaux de Petri stochastiques et leur application à l'évaluation des systèmes*', thèse de Docteur-Ingénieur, CNAM, Paris, (1980).
- [PET 62] C. A. Petri, '*Communication with automata*', PhD. dissertation, tech. rep. RADC-TR 65-377, Rome Air Development Center, NY, (1966).
- [RAM 93] G. Ramathan & J. Oren, 'Survey of commercial parallel machines' ACM SIGARCH computer architecture news, vol. 21, pp. 13-33, (1991).
- [REW 97] H. El-Rewini & T. G. Lewis, '*Distributed and parallel computing*', Ed. Manning, pp. 102-103, (1997).
- [RUM 94] J. de Rumeur, '*Communication dans les réseaux de processeurs*', collection études et recherches en informatique, Ed. Masson, Paris (1994).
- [SAA 89] Y. Saad and M. H. Schultz, 'Data communication in parallel architectures', *Journal of parallel computing*, 11, pp. 579-590, (1989).
- [SAN 91] J. P. Sansonnet, 'Concepts d'architectures avancées', LRI, Archi, (1991).
- [TMC 91] 'The connection machine CM-200 series technical summary', Thinking Machines Corporation, Combridge, Massachussets, June (1991).
- [TMC 92] 'The connection machine CM-5 series technical summary', Thinking Machines Corporation, Combridge, Massachussets, (1992).
- [UPL 94] E. Uplal, 'On the theory of interconnection networks for parallel computers', Research report CS94-06 Weizmann institute of science, department of applied mathematics and computer science, April (1994).
- [Wu 92] C. L. Wu and M. Lee, 'Performance Analysis of Multistage Interconnection Network, Configurations and Operations', *IEEE Trans. Comput.*, Vol. 41, n° 1, pp. 18-27, (1992).
- [YOO 90] H. Yoon, K. Y. Lee and M. T. Liu, 'Performance analysis of multibuffered packet switching networks in multiprocessor systems', *IEEE Trans. comput.*, vol.39, n°3, pp. 319-327, (1990).
- [ZUB 85] W. M. Zuberek, 'M-timed Petri nets. Modelling and performance evaluation of systems', Technical report 92163, department of computer science, memorial university of New foundland, St. John's, Canada, (1985).

Résumé

Cette thèse présente la modélisation des réseaux d'interconnexions dynamiques (RID) par les réseaux de Petri stochastiques (RdPS). Notre approche permet l'évaluation des performances des RID en présence de conflit. Ce conflit a été géré selon deux stratégies définies comme stratégie sans mémoire et stratégie avec mémoire. L'analyse des RdPS se ramène à celle d'un processus de Markov homogène à espace d'état discret à temps continu. Les modèles proposés sont des modèles simplifiés qui limitent considérablement le nombre d'état de la chaîne de Markov. Grâce à la représentation d'un étage ou tour par une place réduit l'explosion des combinaisons des états de la chaîne de Markov. Les modèles s'adaptent facilement à l'extensibilité du réseau tout en gardant les propriétés de fonctionnement du réseau. Les résultats obtenus montrent qu'il n'y a pas de linéarité, d'une part, entre la bande passante (BW) ou le temps moyen de service (Avt) et la taille du réseau, et d'autre part, entre la BW ou le Avt et la longueur du message. Nous retenons que la stratégie sans mémoire est mieux adaptée aux RID, du fait qu'elle offre une meilleure fluidité de passage de messages à travers le réseau. Notre approche nous a permis de montrer que les RdPS constituent un puissant outil mathématique d'évaluation des performances des RID. Cette évaluation prend en considération, d'une part, la défaillance de la boîte de commutation, et d'autre part, la probabilité de passage ou de blocage des requêtes à travers le réseau.

Mots clés: Réseau de Petri stochastique - Processus de Markov - Réseau d'interconnexion dynamique - Système parallèle - Commutation de circuit.

Abstract

This thesis presents an evaluation method of dynamic interconnection networks (DIN) using stochastic Petri nets (SPN). Our approach allows the performance evaluation of a DIN in presence of conflict. This conflict has been managed into two strategies; defined as unbuffered strategy and buffered strategy. The analysis of SPN is a homogeneous markovian process, and therefore to all SPN one could associate a homogeneous markovian chain (MC). Proposed models are simplified models that limit considerably the number of state of the MC. With the representation of a stage or a pass by a place reduce the state explosion combinations of the MC. Models are adapted easily to the scalability of the network. The obtained results show that there is no linearity, on one hand, between the bandwidth (BW) or the average service time (Avt) and the size of the network, and on the other hand, between the BW or the Avt and the length of the message. We retain that the unbuffered strategy is better adapted to DIN, because it offers a low blocking probabilities of the network. Our approach allows to show that SPN constitute a powerful mathematical performance evaluation tool of DIN. This evaluation takes in consideration, on one hand, the failure of the switching element, and on the other hand, the passing or blocking probability of requests through the network.

Keywords: Stochastic Petri net - Markov process - Dynamic interconnection network - Parallel processing - Circuit switching.