

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Abderahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire fin de cycle

En vue d'obtention du Diplôme de
MASTER en Recherche Opérationnelle

Option : Modélisation Mathématique et Évaluation des Performances des Réseaux

Thème :

Modélisation et analyse de réseau de files d'attente fermé [M/M/1 → M/M/1] via les RdPSG

Présenté par : MELLAL Djamel & MOUSSAOUI Belaid

Devant le jury composé de :

Président	Dr Asli Larbi	M.C.B.	Univ. de Béjaïa
Rapporteur	Dr Lekadir Ouiza	M.C.A.	Univ. de Béjaïa
Co-rapporteur	Dr Hakmi Sedda	M.C.B.	Univ. de Béjaïa
Examinatrice	Dr Bernine Nassima	M.C.B.	Univ. de Béjaïa
Examineur	Dr Ikhlef Lyes	M.C.B.	Univ. Alger 3

Année universitaire 2019/2020.

Remerciements

Au terme de ce travail, nous tenons à remercier le bon Dieu tout puissant de nous avoir donné la force et le courage pour réaliser ce modeste travail. Nous tenons également à adresser nos vifs remerciements à Madame O. Lekadir et Madame S. Hakmi qui ont accepté de diriger ce travail.

Nous remercions vivement les membres du jury d'avoir accepté d'examiner ce travail. Leurs remarques et leurs conseils nous permettront sûrement de corriger et d'améliorer ce document.

Le plus fort de nos remerciements vont à nos parents. Ce sont leur présence, leur amour et leur soutien qui alimentent notre inspiration.

Que tous ceux qui de près ou de loin ont contribué à l'aboutissement de ce travail trouvent ici l'expression de notre vive reconnaissance.

Dedicaces

Je dédie ce modeste travail

**A* mes très chers parents,

**A* mes frères et ma chère soeur ,

**A* toute ma famille,

**A* tout(e) s mes ami(e) s

**A* tous ceux qui m'aiment et croient en moi.

Mellal Djamel

Dedicaces

Je dédie ce modeste travail :

- ✠ A mes chers parents et beaux parents.
- ✠ A ma très chère femme qui est toujours à mes cotés et mon petit ange Aksil qui est mon espoir de vivre et mes plus beaux cadeaux de la vie.
- ✠ A ma scœr Soraya qui me réclame toujours son aide sans cesse.
- ✠ A mon frère de m'avoir encouragé de sa part aussi.
- ✠ A mes beaux frères Abdelkrim - Mouhou-Lakhdar et Omar pour leurs aides.
- ✠ A mes amis.
- ✠ tous les intervenants de loin ou de près.

MOUSSAOUI BELAID.

Table des matières

Table des matières	i
Table des figures	i
Liste des tableaux	ii
Introduction générale	1
1 Les Réseaux de Petri	4
1.1 Introduction	4
1.2 Concepts de base	4
1.2.1 Structure d'un Réseau de Petri	4
1.2.2 Un réseau marqué	5
1.2.3 Graphe associé à un RdP et notations matricielles	6
1.2.4 Comportement dynamique	7
1.3 RdP généralisés	9
1.4 Propriétés des <i>RdP</i>	9
1.4.1 Les propriétés dynamiques	10
1.4.2 Les propriétés structurelles	12
1.5 Extensions des réseaux de Petri	12
1.5.1 Les réseaux de Petri étendus	13
1.5.2 RdP T -temporels	14
1.5.3 <i>RdP</i> P-temporel :	14
1.5.4 Les Réseaux de Petri temporisés	14
1.5.5 Les Réseaux de Petri colorés	15
1.5.6 Les Réseaux de Petri interprétés	15
1.5.7 Les Réseaux de Petri stochastiques	16
1.5.8 Réseaux de Petri stochastiques Généralisés	17

1.6	Conclusion	21
2	Systèmes et réseaux de files d'attente	22
2.1	Introduction	22
2.2	Chaîne de Markov à temps continu	22
2.3	Systèmes de Files d'attente	23
2.3.1	Description du File d'attente	23
2.3.2	Classification des systèmes d'attente	24
2.3.3	Mesures de performance d'une file d'attente	24
2.3.4	Les différentes disciplines de service	25
2.3.5	Notation de Kendall	26
2.4	Analyse mathématique d'un SFA	27
2.5	Les files d'attente markoviennes	28
2.5.1	La file d'attente $M/M/1$:	28
2.5.2	La file d'attente $M/M/m$	29
2.5.3	La file d'attente $M/M/m/K$	30
2.6	Les files d'attente non markoviennes	31
2.6.1	La file d'attente $M/G/1$	32
2.6.2	La file d'attente $G/M/1$	33
2.7	Réseaux de files d'attente	33
2.7.1	RFA à forme produit	35
2.7.2	Les réseaux markoviens ouverts	35
2.7.3	Les réseaux markoviens fermés	36
2.8	Algorithme MVA pour les réseaux purement fermés	40
2.9	Conclusion	41
3	Modélisation et Analyse d'un RFA par les RdPSG	42
3.1	Introduction	42
3.2	Description du réseau $[M/M/1 \rightarrow M/M/1]$	42
3.2.1	Description du modèle	42
3.2.2	Modélisation du système $[M/M/1 \rightarrow M/M/1]$ via les RdPSG	43
3.2.3	Analyse des performances de notre réseau	44
3.2.4	Application sur le réseau $[M/M/1 \rightarrow M/M/1]$ avec $N=3$	46
3.3	Application numérique sous le logiciel GRIF	52
3.4	Conclusion	62

Table des matières	iii
Conclusion générale	63
A Annexe	65
Annexe	65
Bibliographie	70

Table des figures

1.1	Représentation graphique d'un RdP.	6
1.2	Franchissement d'une transition.	8
1.3	Exemple d'un RdP avec le graphe des marquages associé.	9
1.4	Exemple d'un réseau de Petri généralisé.	10
1.5	Franchissement de transition avec arc inhibiteur.	13
2.1	La trajectoire d'une chaîne de Markov à temps continu.	23
2.2	Structure générale d'un système de file d'attente.	24
2.3	Réseau ouvert de files d'attente.	36
2.4	Exemple de réseau fermé de FA.	37
3.1	Présentation d'un réseau de files d'attente à deux stations	43
3.2	Modélisation du réseau $[M/M/1 \rightarrow M/M/1]$ via les RdPSG.	43
3.3	Graphe des marquages du réseau $[M/M/1 \rightarrow M/M/1]$ fermé avec $n=3$	47
3.4	La chaîne de Markov associée du réseau fermé $[M/M/1 \rightarrow M/M/1]$ pour $N=3$.	48
3.5	Le RDPSG modélisant le réseau $(M/M/1 \rightarrow M/M/1)$ sous le logiciel GRIF. . .	52
3.6	L'influence de N sur le nombre moyen de clients et le temps d'attente. . . .	55
3.7	L'influence de ρ_1 sur le nombre moyen de clients et le temps d'attente. . . .	58
3.8	L'influence de ρ_2 sur le nombre moyen de clients et le temps d'attente. . . .	61

Liste des tableaux

3.1	L'influence de N sur le temps d'attente.	53
3.2	L'influence de N sur le nombre moyen de clients.	54
3.3	Temps d'attente en fonction de ρ_1	56
3.4	Nombre moyen de clients en fonction de ρ_1	57
3.5	Temps d'attente en fonction de ρ_2	59
3.6	Nombre moyen de clients en fonction de ρ_2	60

Introduction générale

Dans la réalité des phénomènes d'attente se manifestent souvent sous différentes formes, on peut citer comme exemple : l'arrivée des voitures à une station de service, la vente de billets auprès d'un guichet, l'exécution des tâches dans un centre de calcul, ... etc. Pour étudier ou modéliser ces phénomènes on fait appel à la théorie des files d'attente. Cette théorie utilise des outils probabilistes pour étudier et modéliser le comportement des systèmes réels où l'attente intervient.

La théorie des files d'attente, qui est relativement ancienne, connaît actuellement un regain d'intérêt dû à l'extraordinaire développement des réseaux de communication. Cette théorie est un domaine de la recherche opérationnelle qui permet de modéliser un système d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions.

Les premiers travaux sur la théorie des files d'attente ont été initiés par l'ingénieur danois A. A. Erlang dans le but de modéliser les réseaux téléphoniques[12, 13]. De nos jours, cette théorie intervient dans de nombreux autres domaines d'application et particulier dans les domaines industriels, économiques, systèmes informatiques, etc. Toutefois, la majorité des systèmes réels sont assez complexes pour qu'ils soient modélisés par une seule file d'attente, dans ce cas on fait appel aux réseaux de files d'attente (RFA)[31, 6].

Un réseau de files d'attente est un ensemble de station de file d'attentes interconnecter. Le séjour d'un client dans ces réseaux consiste à parcourir une partie ou l'ensemble de toutes ces files d'attente. En effet, le besoin de l'outil des réseaux de files d'attente est apparu dans les années 70 pour étudier les performances des systèmes informatiques. Les réseaux de files d'attente sont classés en deux catégories principales : réseaux de files d'attente mono-classes et réseaux de files d'attente multi classes. Dans le cas de réseaux de files d'attente mono-classes, on fait également la distinction entre : réseaux de files d'attente ouverts et réseaux de files d'attente fermés. Des solutions exactes existent pour

une classe bien particulière de RFA connus sous le nom de réseaux séparables où à forme produit, cette classe de réseaux à forme produit est très restreinte mais de extensions lui sont établies depuis. Néanmoins, des difficultés sont rencontrées pour élargissement de ces extensions et cela suite à la difficulté de la nature des flux inter-stations de ces réseaux. Bien sûr de nombreuses méthodes approximatives d'évaluation des performances des *RFA* non à forme produit ou ayant une forme produit difficile à exploiter ont été développées, cependant la complexité algorithmique de ces méthodes ce complique au fur et à mesure que le nombre de stations ou le nombre de clients dans ces réseaux s'agrandissent. Pour pallier à toutes ces difficultés d'analyse des RFA, dans ce mémoire on va proposer une approche de modélisation, d'analyse et d'évaluation de performances des RFA fermés basée sur l'un des formalismes des réseaux de Petri Stochastiques Généralisés (RdPSG).

Les réseaux de Petri (RdP) constituent un outil mathématique de modélisation développés au début des années soixante par l'ingénieur allemand Carl Adam Petri dans sa thèse "Communication avec des automates" [29].

Dans ce mémoire nous allons analysé un réseau de files d'attente fermé $[M/M/1 \rightarrow M/M/1]$ en utilisant le formalisme RdPSG (Réseaux de Petri stochastique généralisés). En effet, cette approche proposée, va nous permettre d'avoir facilement la chaîne de Markov à partir du graphe d'accessibilité par conséquent de calculer efficacement les mesures de performances. En outre, nous allons étudier l'effet des paramètres du réseau sur les indices de performance à travers quelques exemples numériques et ce en faisant appeler au simulateur GRIF et les résultats obtenus seront interprétés.

Organisation du document

Ce mémoire est composé :

- d'une introduction générale où le sujet à traiter est bien exposé ;
- de trois chapitres :
 - Dans le chapitre 1 nous présentons l'outil de modélisation mathématique qui est les réseaux de Petri. Nous donnons quelques notions de bases ainsi que quelques extensions des RdP.
 - Dans le chapitre 2, nous exposons quelques notions de base des files d'attente et des réseaux de files d'attente qui nous seront nécessaires dans les chapitres suivants.
 - Le chapitre trois est consacré à la présentation de notre modèle obtenu via les RdP et à l'étude des principaux indices de performances de notre réseau en faisant usage de logiciel GRIF.

- et d'une conclusion générale qui donne un bilan de notre travail et une idée sur les perspectives envisagées.

Chapitre 1

Les Réseaux de Petri

1.1 Introduction

Les réseaux de Petri(RdP) représentent un ensemble d'outils très puissants, permettant de faire à la fois la modélisation, l'analyse qualitative et l'analyse quantitative de plusieurs classes de systèmes. Ils ont le double avantage de fournir un support graphique naturel qui est d'une aide précieuse dans l'analyse, et de posséder des propriétés analytiques qui permettent une évaluation simple du comportement du système étudié. Nous sommes donc face à un outil complet et efficace. Le modèle réseau de Petri est bien adapté pour la modélisation des systèmes de coopération. Il a l'avantage d'être rigoureux et lisible et de ne posséder que deux primitives : les places et les transitions. Un *RdP* permet la représentation de l'état d'un système et les évolutions possibles de cet état. De plus ce modèle peut être étendu par la prise en compte du temps et de l'aléatoire par son extension stochastique.

1.2 Concepts de base

Un réseau de Petri est un graphe biparti orienté qui a deux types de noeuds : un ensemble de places représentant les ressources et un ensemble de transitions symbolisant les événements ou les opérations.

1.2.1 Structure d'un Réseau de Petri

Une structure de *RdP* est donnée par un quadruplet :

$$R = (P, T, Pre, Post).$$

tels que :

- $P = p_1, p_2, \dots, p_n$ est un ensemble fini de places,
- $T = t_1, t_2, \dots, t_m$ est un ensemble fini de transitions,
- $Pre : P \times T \rightarrow \mathbb{N}$ c'est l'application d'incidence avant (places précédentes),
- $Post : T \times P \rightarrow \mathbb{N}$ c'est l'application d'incidence arrière (places suivantes).

On note par C la matrice d'incidence du réseau de Petri qui est définie par :

$$C = Post - Pre.$$

Les places sont représentées par un cercle et les transitions en général par un rectangle. Ainsi, un réseau de Petri est représenté par un graphe avec deux types de sommets : les transitions et les places. Ces différents sommets sont reliés entre eux par des arcs qui joignent les places aux transitions (les preconditions) et des transitions aux places (les postconditions). Ces arcs possèdent des poids qui sont donnés par des entiers, mais si les poids sont pas mentionnés sur le réseau ça sous-entend que leurs poids valent 1.

1.2.2 Un réseau marqué

Un réseau marqué est un couple $\langle R; M \rangle$, où R est le réseau et M est l'application définie de P dans \mathbb{N} , telle que $M(p)$ est égale au nombre de marques dans la place $p \in P$. Dans ce cadre M pourra être représenté par un vecteur de dimension $m = |P|$, Pre et $Post$ par des matrices de $m = |P|$ lignes et $n = |T|$ colonnes. $Pre(\bullet, t)$ et $Post(\bullet, t)$ représenteront les colonnes des matrices Pre et $Post$ relatives à la transition t . $Pre(p, \bullet)$ et $Post(p, \bullet)$ représenteront les lignes des matrices Pre et $Post$ relatives à la place $p \in P$. On utilise également la notation : $C = Post - Pre$; qui est appelée matrice d'incidence du réseau de Petri.

Exemple :

Soit le RdP marqué $N_1 = \langle R_1; M \rangle$, défini avec les matrices Pre et $Post$ ainsi que par le marquage initial M_0 suivants :

$$Pre = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}; Post = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \text{ et le marquage } M_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

1.2.3 Graphe associé à un RdP et notations matricielles

Un RdP est décrit par un graphe qui possède deux types de noeuds : les places notées graphiquement par des cercles et les transitions notées graphiquement par des traits ou des rectangles. Un arc relie une place p à une transition t si et seulement si $Pre(p, t) \neq 0$. Un arc relie une transition t à une place p si et seulement si $Post(p, t) \neq 0$. Les valeurs non nulles des matrices Pre et $Post$ sont associées aux arcs comme étiquettes. Le marquage M peut être représenté par un vecteur ayant pour dimension le nombre de places. Les fonctions Pre , $Post$ et C seront présentées par des matrices dont le nombre de lignes est égal aux nombres de places et le nombre de colonnes est égal aux nombres de transitions.

Proposition 1.1.

- Chaque place d'un réseau de Petri peut avoir deux états : elle est marquée ou non marquée.
- L'état du système est représenté par une distribution de marques dans l'ensemble des places, cette distribution est appelée le marquage du réseau de Petri.
- On appelle places d'entrée ou ensemble d'entrée d'une transition le sous-ensemble des places du RdP qui participent à l'expression de sa précondition.
- Une transition d'un RdP marqué est dite validée ou sensibilisée si et seulement si chaque place de son ensemble d'entrée est marquée. Lorsqu'une transition est validée, l'événement qui lui est associé peut avoir lieu.
- Le poids d'un arc est donné par les fonctions Pre et $Post$.

La FIGURE1.1 suivante illustre la représentation graphique d'un RdP.

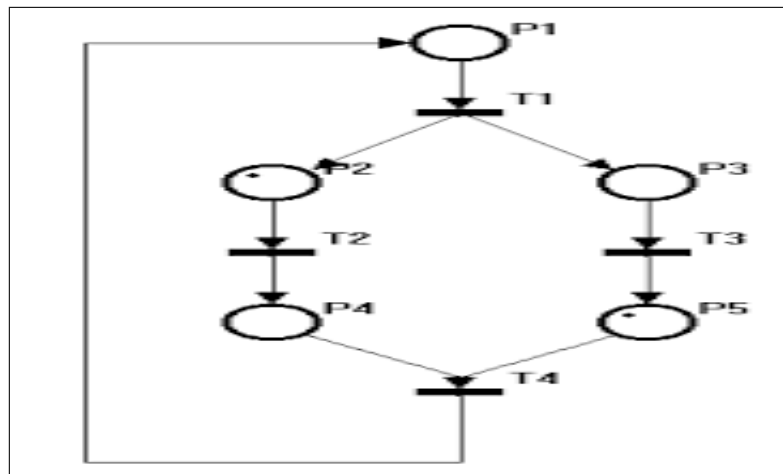


FIGURE 1.1 – Représentation graphique d'un RdP.

1.2.4 Comportement dynamique

Le comportement dynamique d'un réseau de Petri représente son évolution au cours du temps après l'occurrence des événements (on dit sa sémantique opérationnelle). les définitions suivantes spécifie les différents aspects du comportement dynamique d'un réseau de Petri [16].

Définition 1.1. (Règle de franchissement) : Une transition t est dite franchissable ou tirable ou validée pour un marquage M si et seulement si :

$$\forall p \in t, \quad M(p) \geq \text{Pre}(p, t). \quad (1.1)$$

et on note $M(t >)$, avec t représente l'ensemble de places en amont de t . Le franchissement d'une transition validée modifie le marquage des places en amont et en aval de cette transition. Le franchissement d'une transition a pour conséquences le retrait d'un nombre de marques égal aux poids des arcs pour les places en amont de cette transition et l'ajout d'un nombre de marques égal aux poids des arcs pour les places en aval.

Si le *RdP* considéré est ordinaire, alors le nombre de marques à ajouter ou à retirer est égal à 1.

Le tirage de la transition t à partir du marquage M induit un marquage M' tel que :

$$\forall p \in P, \quad M'(p) = M(p) + \text{Post}(p, t) - \text{Pre}(p, t). \quad (1.2)$$

Ce passage on le note $M(t > M')$, ce qui signifie que t est franchissable pour le marquage M et atteint le marquage M' .

La FIGURE1.2 suivante illustre le franchissement d'une transition :

Remarque 1.1. A partir de cette règle de franchissement, il est possible d'obtenir l'ensemble des suites finies ou infinies d'évolution du système à partir du marquage initial. Cette suite d'ensembles définit l'ensemble des marquages accessibles et peut être représenté sous forme de graphe dit graphe des marquages accessibles.

Définition 1.2. (Séquence de franchissement) [5] : Soit R un *RdP*, $\sigma = t_1, t_2, \dots, t_n \in T$ une séquence de transition et M un marquage. La séquence σ est franchissable depuis M si et seulement s'il existe des marquages $M_1, M_2 \dots M_n$ tels que : $M(t_1 > M_1(t_2 > M_2(\dots M_{n-1}(t_n > M_n$. Si la séquence de franchissement σ est réalisable à partir de M_i , le marquage obtenu M_k est donné par l'équation fondamentale :

$$M_k = M_i + C \cdot \sigma; \quad (1.3)$$

où C est la matrice d'incidence du *RdP* et $k = \overline{1, n}$. Le tir de σ conduit alors au marquage M_n , ainsi on note $M(\sigma > M_n$.

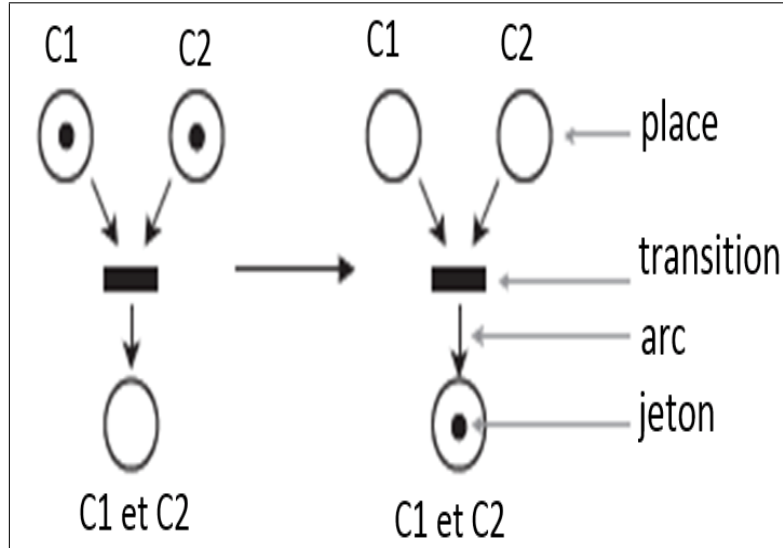


FIGURE 1.2 – Franchissement d’une transition.

Remarque 1.2. Notons que, l’équation fondamentale ne garantit pas que σ soit franchissable mais dans le cas où elle est franchissable cette équation nous permet simplement de trouver le nouveau marquage lorsqu’on connaît le marquage M_I .

Définition 1.3. (Marquage accessible) : Soit (R, M_0) un *RdP* marqué. Un marquage M est accessible si et seulement s’il existe une séquence de franchissement σ telle que $M_0(\sigma > M)$.

Définition 1.4. (Ensemble d’accessibilité) [19] : Soit (R, M_0) un *RdP* marqué. L’ensemble des marquages accessibles ou l’ensemble d’accessibilité de cet *RdP* est noté $Acc(R, M_0)$ ou Acc est l’ensemble des marquage atteints par une séquence de franchissement :

$$Acc(R, M_0) = \{M \in \mathbb{N} / \exists \sigma \in t^* \text{ tel que } M_0(\sigma > M)\}. \quad (1.4)$$

Définition 1.5. (Graphe des marquages accessibles)[19] : Soit (R, M_0) un *RdP* marqué. Le graphe des marquages accessibles (ou graphe d’accessibilité) de ce réseau, noté $G(R, M_0)$ est défini comme le graphe dont les noeuds (ou sommets) sont les marquages accessibles (i.e. les marquages appartenant à $Acc(R, M_0)$) et dont les arcs sont étiquetés par les noms des transitions, qui définissent la relation de tir entre les marquages. Un arc étiqueté par t joint M à M' si et seulement si $M(t > M')$.

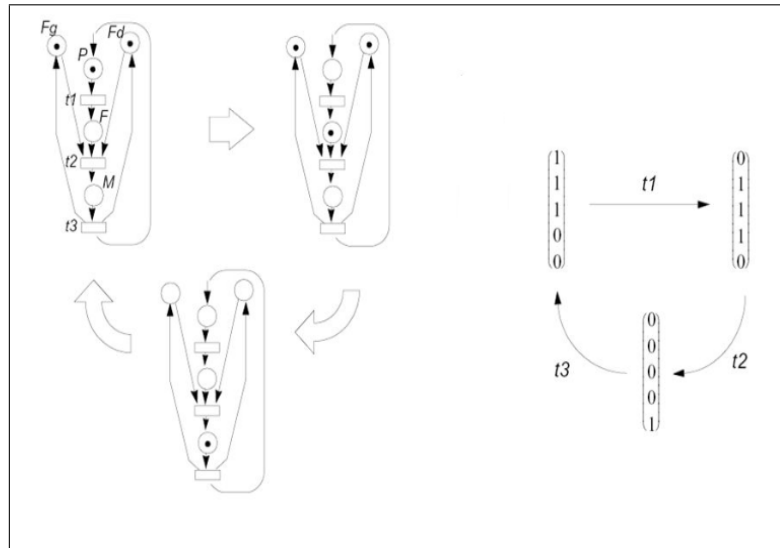


FIGURE 1.3 – Exemple d'un RdP avec le graphe des marquages associé.

Définition 1.6. (Graphe de couverture) : On peut avoir une situation où l'ensemble d'accessibilité n'est pas fini, le graphe des marquages est ainsi infini et donc il n'est pas possible de le construire dans sa globalité. Dans le cas où le graphe des marquages est infini on remplace le graphe des marquages par le graphe de couverture, qui est une approximation.

1.3 RdP généralisés

Un *RdP* est dit généralisé si on associe à ses arcs un poids supérieur strictement à 1. Une transition est franchissable dans un réseau de Petri généralisé, si toutes les places p_i en amont des transitions t_j contiennent au moins le nombre de jetons associés aux arcs. Lors du franchissement de la transition t_j , le nombre de jetons dans les places en aval de cette transition sera augmenté par le poids p , voir la figure 1.4. Cette extension des *RdP* permet la réduction de la taille du *RdP* ordinaire. Toutefois, la transformation est toujours possible du *RdP* généralisé vers le *RdP* ordinaire.

1.4 Propriétés des *RdP*

Les *RdP* permettent de modéliser des systèmes complexes [10]. Ceux-ci se doivent de répondre à un certain nombre d'exigence (absence de blocage, préservation des fonctionnalités, existence d'un fonctionnement en régime permanent, ...etc.). Nous définissons maintenant les propriétés élémentaires des *RdP*.

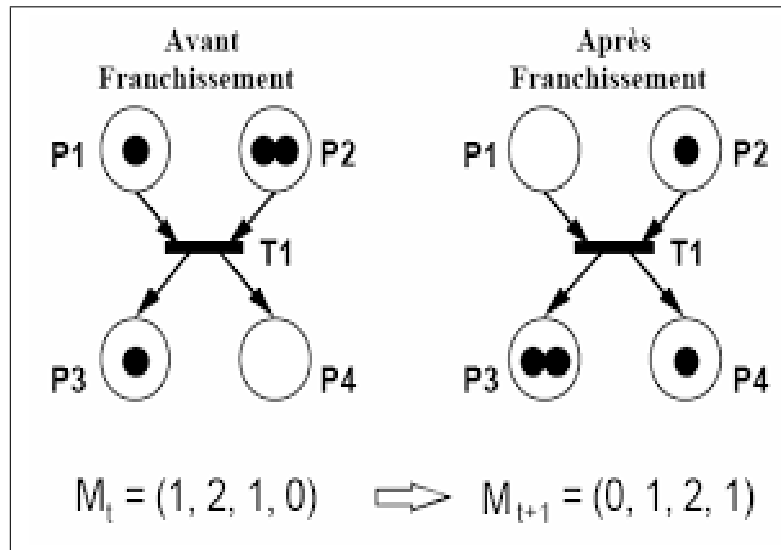


FIGURE 1.4 – Exemple d’un réseau de Petri généralisé.

1.4.1 Les propriétés dynamiques

Définition 1.7. (Bornitude) : Soit un réseau R avec un marquage initial M_0 , une place p_j du réseau marqué (R, M_0) est k -bornée si pour tout marquage M_i accessible depuis M_0 (i.e. $M_i(p_j) \leq k$). Un *RdP* marqué est borné si toutes ses places sont bornées.

Définition 1.8. (*RdP* sauf) : Un *RdP* marqué est sauf ou binaire pour un marquage initial M_0 s’il est 1-borné.

► Vivacité et blocage

L’évolution du marquage d’un *RdP* se fait par franchissement de transitions. Lorsqu’au cours de son évolution, certaines transitions ne sont jamais franchies, cela indique que l’événement associé à la transition ne se produit pas et que le marquage d’une partie du *RdP* n’évolue pas. Cela indique que le sous système modélisé par cette partie-là ne fonctionnera pas. Il y a donc un problème au niveau de la conception du système. L’idée est d’être capable de détecter systématiquement ce phénomène par l’analyse de propriétés du modèle *RdP* du système afin de disposer d’un outil d’aide à la conception des systèmes.

Définition 1.9. (Vivacité) : Une transition t_j est vivante pour un marquage initial M_0 si pour tout marquage accessible M_k , il existe une séquence de franchissements à partir de M_k contenant t_j . Si une transition t_j est vivante alors, à tout instant, on sait que t_j peut être franchie dans le futur. Dans le cas d’un *RdP* modélisant un système fonctionnant en permanence, si une transition n’est pas vivante et si une fonction du système est associée

au franchissement de cette transition, cela veut dire qu'à partir d'un certain instant, cette fonction ne sera plus disponible dans le futur, ce qui peut traduire une erreur ou une panne.

Définition 1.10. : Un *RdP* marqué est vivant pour un marquage initial M_0 si toutes ses transitions sont vivantes pour ce marquage initial. Un *RdP* est dit conforme s'il est sauf et vivant.

Définition 1.11. (Blocage) : Un blocage est un marquage pour lequel aucune transition est validée. Un *RdP* marqué est dit sans blocage pour un marquage initial M_0 si aucun marquage accessible n'induit un blocage.

► Conflits et parallélisme

Définition 1.12. (Conflits structurel et effectif) [19] : Deux transition t_1 et t_2 sont en conflit structurel si et seulement si elles ont au moins une place commune en entrée, i.e. :

$$\exists p \in P, \text{Pre}(p, t_1) \times \text{Pre}(p, t_2) \neq 0. \quad (1.5)$$

Elles sont en conflit effectif pour un marquage M si de plus :

$$M(t_1 > \text{ et } M(t_2 > \quad (1.6)$$

et

$$\exists p \in P, M(p) < \text{Pre}(p, t_1) + \text{Pre}(p, t_2). \quad (1.7)$$

Définition 1.13. (Parallélisme) Deux transitions t_1 et t_2 sont en parallèles structurellement si :

$$\left(\text{Pre}(\cdot, t_1)^T \times (\text{Pre}(\cdot, t_2) = 0 \right). \quad (1.8)$$

Elles sont en parallèles effectifs pour un marquage M si de plus :

$$M(t_1 > \text{ et } M(t_2 > . \quad (1.9)$$

► La réversibilité

Définition 1.14. : Un *RdP* est dit réversible si, à partir de n'importe quel marquage atteignable M_i , il existe une séquence de transitions franchissables σ qui permet de revenir au marquage initial M_0 .

► Réinitialisabilité et état d'accueil

Définition 1.15. (État d'accueil) [5] : Un *RdP* a un état d'accueil M_a pour un marquage initial M_0 si pour tout marquage M_i accessible depuis M_0 , il existe une séquence de franchissement σ tel que M_a est accessible depuis M_i en franchissant σ .

Définition 1.16. (Réinitialisabilité) : Le *RdP* est dit réinitialisable pour un marquage initial M_0 si M_0 est un état d'accueil.

► **Graphes d'événement, la connexité et la forte connexité**

Définition 1.17. (La connexité) [5] : Un graphe d'événement est un *RdP* dans lequel chaque place a exactement une transition d'entrée et une transition de sortie. Il est caractérisé par l'absence de conflits structurels.

Définition 1.18. (La connexité) : Un *RdP* est dit connexe si et seulement si il existe un chemin (non forcément orienté) depuis n'importe quel noeud vers un autre noeud. Un noeud peut être une place ou une transition.

Définition 1.19. (La forte connexité) : Un *RdP* est dit fortement connexe si et seulement si il existe un chemin orienté entre chaque paire de noeuds de types différents.

1.4.2 Les propriétés structurelles

► **Représentation matricielle d'un RdP**

La matrice correspondante à *Pre* et celle correspondante à *Post* sont construites de la manière suivante. Les lignes identifient les places et les colonnes identifient les transitions. $(a_{i,j})$ de la matrice *Pre* correspond au coût associé à l'arc menant de la place p_i à la transition t_j , tandis que $(a_{i,j})$ dans la matrice *Post* correspond au coût associé à l'arc menant de la transition t_j à la place p_i . La matrice d'incidence d'un *RdP* est la matrice entière $C \in \text{Mat}(P, T)$ définie par :

$$\forall (p, t) \in P \times T, C(p, t) = \text{Post}(p, t) - \text{Pre}(p, t). \quad (1.10)$$

1.5 Extensions des réseaux de Petri

Les réseaux de Petri décrits dans les paragraphes précédents présentent certaines limitations dans leurs pouvoirs de modélisation. En particulier, il est apparu pour les *RdP* ordinaire, le besoin d'introduire la notion d'arc inhibiteur. D'autre part l'aspect temporel n'intervient pas dans la définition d'un *RdP*, limite ainsi son pouvoir de description aux seules relations de causalité, d'où la création des réseaux de Petri temporels. Une dernière extension, justifiée par l'imprécision des temps et la prise en compte du phénomène aléatoire, consiste à associer aux transitions des variables aléatoires modélisant le délai de franchissement.

1.5.1 Les réseaux de Petri étendus

La notion des réseaux de Petri sera étendue, par l'introduction des arcs inhibiteurs, d'une relation de priorité et la dépendance du poids des arcs en fonction du marquage.

► Arcs inhibiteurs

Un arc inhibiteur est un arc orienté qui part d'une place p pour aboutir à une transition t . Son extrémité est marquée par un petit cercle. L'arc inhibiteur entre la place P et la transition t signifie que la transition t n'est validée que si la place p ne contient aucune marque. Le franchissement consiste à retirer une marque dans chaque place d'entrée de t à l'exception de p , et à ajouter une marque dans chaque place de sortie de t . On utilise aussi les expressions $\ll R dP \text{ étendus} \gg, \ll \text{test à zéro} \gg$

Définition 1.20. (RdP à arc inhibiteur) [38] : Un RdP à arc inhibiteur est défini par un couple $\langle R, IN \rangle$ tel que :

- R est un Réseau de Petri marqué ;
- $IN : P \times T \longrightarrow \mathbb{N} \setminus \{0\}$ est la fonction d'inhibition ;

pour un marquage M donné, la condition du tirage d'une transition t , devient alors :

$$\forall p \in P, \quad M(p) \geq \text{Pre}(p, t) \wedge (M(p) < IN(p, t) \vee IN(p, t) = 0). \quad (1.11)$$

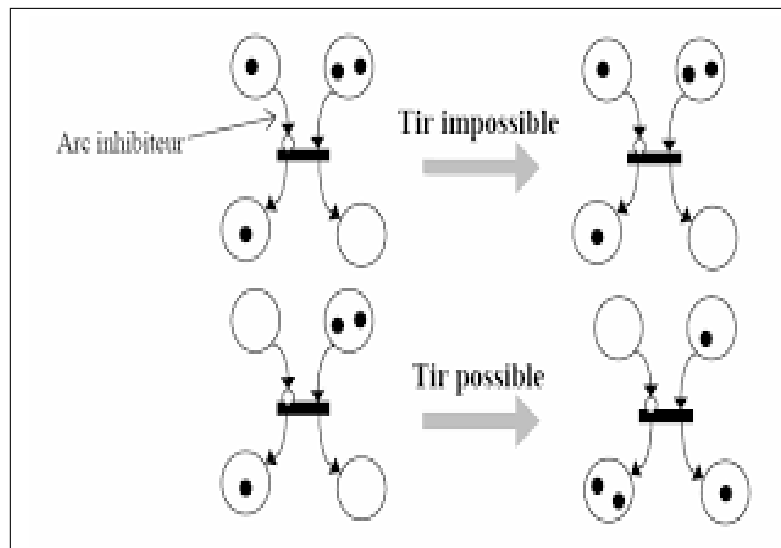


FIGURE 1.5 – Franchissement de transition avec arc inhibiteur.

► Priorité

Lors d'un conflit effectif, il convient de préciser un choix pour la transition à tirer . Ce choix peut être aléatoire ou défini par une certaine notion sur ces transitions. Une de ces possibilités consisterait à associer à chaque transition du RdP , un nombre entier croissant avec la priorité, établissant ainsi un ordre total.

1.5.2 RdP T-temporels

Les réseaux de Petri T -temporels ($RdPT$)[25] étendent les réseaux de Petri avec des intervalles de temps $[a(x), b(x)]$ associés à chaque transition t du réseau. Pour être tirée, une transition t doit non seulement être sensibilisée mais aussi l'avoir été continûment pendant une durée comprise entre $a(x)$ et $b(x)$. Donc une fois la date $b(x)$ atteinte, la transition t devra obligatoirement être franchie, à moins d'avoir été désensibiliser puis nouvellement sensibilisée par le tir d'une autre transition.

Définition 1.21. (RdPT-temporel) : Un RdP temporel est un couple $\langle R, I_s \rangle$ où

- R est un RdP marqué;
- $I_s : T \rightarrow I^+$ est une fonction appelée Intervalle statique; avec I^+ l'ensemble des intervalles réels non vides à bornes rationnelles non négatives.

1.5.3 RdP P-temporel :

Ce modèle a été élaboré par Sifakis [37] tel que, une temporisation (valeur rationnelle positive) est associée à chaque place qui représente la durée de séjour minimale d'une marque dans cette place. On notera par d_i la temporisation de la place p_i .

Définition 1.22. (RdPP-temporel) : Le RdP P-temporisé est défini par le couple $((R, M_0), d)$ tel que : (R, M_0) : Réseau marqué. $d : P \rightarrow Q^+$ est la fonction de temporisation ($d_i =$ temps que passent les marques dans la place p_i). L'état de disponibilité d'un jeton permet de spécifier la durée nécessaire pour effectuer une opération sur un produit donné.

1.5.4 Les Réseaux de Petri temporisés

Dans ce genre de réseau, une durée est associée à chaque transition (T -temporisé) ou à chaque place (P -temporisé). On peut faire une analyse quantitative qui permet l'évaluation des performances du système ainsi modélisé.

Définition 1.23. (Les Réseaux de Petri T-temporisés) [36]

Un réseau de Petri T -temporisé est un couple $\langle R, \text{Tempo} \rangle$:

- R est un RdP marqué $\langle P, T, Pre, Post, M_0 \rangle$;
- $Tempo$ est une application qui associe à chaque transition une durée de franchissement.

1.5.5 Les Réseaux de Petri colorés

Les Réseaux de Petri Colorés ($RdPC$), sont des Réseaux de Petri dans lesquels les jetons portent des couleurs [1], on associe ainsi à chaque jeton une valeur appelée la couleur du jeton et permet de distinguer les jetons entre eux. Chaque couleur correspond à une information bien définie associée au jeton. Deux principales raisons justifient l'appel aux RdP colorés :

- La possibilité de transporter une information structurée car les places ne contiennent pas que des jetons uniformes ;
- Ce type de réseaux s'utilise pour éviter le problème de taille importante des RdP ordinaires dans le cas où des entités différentes présentent des comportements similaires ce qui condense le modèle.

L'intérêt des réseaux colorés est de pouvoir décrire des dynamiques relativement complexes avec un petit nombre de places et de transitions. En effet la notion de couleur et les expressions d'arcs permettent de rendre plus compacte la structure du réseau en réduisant de manière importante le nombre de places et de transitions nécessaires à la modélisation du système.

1.5.6 Les Réseaux de Petri interprétés

Les réseaux de Petri interprétés aussi appelés RdP Prédicat/Transition présentent la première classe de réseaux de Petri à haut niveau introduit par Genrich et Lautenbach en 1981 [40]. Dans un RdP Interprété, le contrôle et les données sont séparés :

Le contrôle est modélisé par un RdP place-transition, tant dis qu'un environnement externe fait d'un ensemble fini de variable est associé à l'état du réseau pour traiter les données. Un prédicat sur cet environnement et une action qui puisse modifier cet environnement, sont associés à chaque transition [27]. Dans la description d'une transition, il est possible de spécifier une liste d'expressions booléennes qui doivent être nécessairement vérifiées pour que la transition soit valide. Ces expressions sont séparées par des virgules. Il s'agit des conditions de validation.

Définition 1.24. (Réseau de Petri Interprété) : Un RdP Interprété est un quadruplet $\langle R, X, Pred, Act \rangle$ où

- R est un RdP marqué ;
- X est un ensemble fini de variables ;
- $Pred$ est une application qui associe à chaque transition un prédicat, qui est une expression booléenne, construite sur les variables de l'ensemble X . Par défaut, la valeur du prédicat associée à une transition est vraie ;
- Act est une séquence d'instructions mettant à jour quelques variable de X , par défaut, cette séquence est vide et rien n'est exécuté.

L'état d'un RdP interprété est représenté par un couple (M, val) , où M est le marquage de R et val une application qui donne des valeurs à toutes les variables de X . L'état initial est (M_0, val_0) .

Une transition t est franchissable pour un état (M, val) est t est franchissable pour le marquage M dans R et si l'évaluation de $Pred(t)$ dans l'état où chaque variable x de valeur $val(x)$ est vraie. Si t est tirées pour un marquage M dans R et val , le nouvel état atteint est (M', val') où

- M' est le nouveau marquage ;
- val' nouvelle valeur des variables de X , résultant des séquences d'instructions $Act(t)$.

Le formalisme des RdP interprétés, est principalement utilisé pour la simulation, le développement de prototype et pour les algorithmes distribués.

1.5.7 Les Réseaux de Petri stochastiques

Les réseaux de Petri stochastiques ($RdPS$) sont des modèles formels qui permettent à la fois des analyses qualitatives, par la vérification de propriétés cité précédemment, et quantitatives par l'évaluation des performances.

Définition 1.25. ($RdPS$) : Dans un RdP temporisé, une durée fixe est associée à chaque place ou transition du réseau. On obtient des modèles qui sont bien adaptés pour étudier des systèmes où les durées opératoires sont fixes. En revanche, certains phénomènes sont imprévisibles et ne peuvent plus être traités avec ce type de modèles. Dans ces cas, chaque durée est modélisée par une variable aléatoire. Nous obtenons alors un $RdPS$, dans lequel un temps aléatoire est associé au franchissement d'une transition. La méthode d'analyse couramment employée consiste à construire le graphe des marquages du RdP sous-jacent au $RdPS$ et à étiqueter chaque arc par un taux de franchissement. Le comportement aléatoire du $RdPS$ est alors identique à celui d'une chaîne de Markov.

Définition 1.26. (Réseau de Petri stochastique) [35], [28] : Un Réseau de Petri stochastique ($RdPS$) est un couple $\langle R, Taux \rangle$ tel que :

- R est un RdP marqué ;
- $Taux$ est une fonction qui à chaque transition t associe un taux de transition λ_t .

Remarque 1.3. : Cela revient à associer à chaque transition un intervalle de sensibilisation continu $[0, \infty[$ avec une distribution exponentielle. Ceci explique que l'ensemble des marquages accessibles est le même que celui du réseau sous-jacent.

► Durée de sensibilisation stochastique

Au lieu de prendre, de façon égale, toutes les valeurs d'un intervalle $([a(x), b(x)])$ comme dans le réseau de Petri temporels, on peut supposer que la durée de sensibilisation est une variable stochastique θ avec une distribution de probabilité exponentielle.

$$\theta = \int_0^{\infty} (1 - \Pr_{\theta}(x)) \cdot dx = \int_0^{\infty} e^{-\lambda x} \cdot dx = \frac{1}{\lambda}. \quad (1.12)$$

où λ est appelé taux de transaction.

1.5.8 Réseaux de Petri stochastiques Généralisés

Les réseaux de Petri stochastiques généralisés $RdPSG$ ont été introduits par Ajmone [2, 15], afin de limiter certaines restrictions. Les réseaux de Petri stochastiques généralisés sont une extension des réseaux de Petri stochastiques autorisant deux classes de transitions :

- des transitions instantanées à temporisation nulle (transition immédiate) qui sont franchies immédiatement dès qu'elles sont sensibilisées ;
- des transitions temporisées à qui correspondent des variables aléatoires déterminant la durée de franchissement.

Les taux de transition peuvent dépendre du marquage. S'il existe plusieurs transitions franchissables, les transitions instantanées sont franchies avant les autres (car elles sont prioritaires par rapport aux transitions temporisées) .i.e. elles sont franchissables immédiatement à partir du moment où elles sont sensibilisées. Quand il existe plusieurs transitions instantanées franchissables, il faut donner la distribution de probabilité régissant le choix de la transition à franchir. Les transitions instantanées représentent des activités du système qui s'achèvent en un temps négligeable par rapport aux durées précédentes. Les états qui ne sont pas associés à une transition instantanée sont appelés des états tangibles (qui sensibilisent seulement les transitions temporisées). Sinon, si au moins une transition immédiate est sensibilisée on parle d'un état évanescent, donc un réseau de Petri stochastique généralisé présente deux types d'états :

- Les états tangibles, pour lesquels l'ensemble des transitions sensibilisées sont des transitions temporisées.

- Les états évanescents, pour lesquels au moins une transition sensibilisée est une transition immédiate.

Définition 1.27. Un *RdPSG* est un 8-uplet $\langle P, T, Pre, Post, Inh, pri, W, M_0 \rangle$ où :

- P est l'ensemble fini des places ;
- T est l'ensemble fini des transitions temporisées et des transitions immédiates ;
- $Pre, Post, et Inh : P \times T \rightarrow \mathbb{N}$ sont les fonctions d'incidence avant, d'incidence arrière et d'inhibition respectivement ;
- $pri : T \rightarrow \{0, 1\}$ est la fonction de priorité qui associe à chaque transition temporisée la valeur 0 et à chaque transition immédiate la valeur 1 ;
- $W : T \rightarrow \mathbb{R}^+$ est une fonction qui associe à chaque transition temporisée un taux de franchissement et à chaque transition immédiate un poids ;
- $M_0 : P \rightarrow \mathbb{N}$ est le marquage initial du réseau.

À chaque application d'incidence est associée une matrice $|P| \times |T|$.

1.5.8.1 Sémantique des *RdPSG* :

La politique d'exécution d'un *RdPSG* est composée de trois éléments :

1. La politique de sélection (politique de choix, "présélection Policy") :

Elle consiste à la définition de la prochaine transition à tirer quand plusieurs transitions sont simultanément sensibilisées. Pour le choix de la transition à tirer, on distingue deux techniques :

- La compétition (modèle concurrentiel, politique de course, "race Policy") :
Elle consiste à tirer, parmi les transitions franchissables, celle qui a le plus petit délai de franchissement. Généralement, on applique cette technique si on a des transitions temporisées. Elle représente un inconvénient, dans le cas où les transitions ont des délais identiques.

- **La présélection :**

Elle consiste à associer aux transitions des poids au moment de la description du réseau. Généralement, on applique cette technique si on a des transitions immédiates.

2. La politique de la mémoire temporelle :

La prise en compte, dès l'instant d'entrée dans un marquage, du passé (histoire) du système. On distingue trois mémoires :

- **La réinitialisation ("resampling")** : Elle consiste en une remise à zéro de la mémoire temporelle (après le tir d'une transition) de toutes les transitions simultanément sensibilisées (ne prend pas en compte le passé du système).
 - **La mémoire de la dernière sensibilisation ("enabling memory")** : Pour cette règle, la durée totale de sensibilisation mémorisée est remise à zéro à chaque désensibilisation de la transition (seule la dernière période de sensibilisation est prise en compte).
 - **La mémoire de toutes les sensibilisations (age memory)** : Le comportement le plus général d'un RdPS correspond à cette règle, qui consiste à mémoriser pour chaque transition la durée totale pendant laquelle elle a été sensibilisée jusqu'à son tir (toutes les périodes de sensibilisation de la transition sont prises en compte).
3. **La politique de service** : Dans la littérature trois politiques de service sont appliquées :
- **Serveur unique (single server)** : A chaque instant, un client est servi à la fois c'est la sémantique par défaut des *RdPSG*.
 - **Serveurs multiples (multiple servers)** : Au plus k clients peuvent être servis simultanément.
 - **Serveurs infinis (infinite servers)** : Autant de clients que possible, peuvent être servis en même temps.

1.5.8.2 Processus stochastique associé à un *RdPSG* :

Le temps de séjour dans un marquage est le temps passé par le *RdPSG* dans un état. Il correspond à la durée de franchissement de la transition qui a provoqué le changement d'état. Nous pouvons constater que la distribution du temps de séjour dans un marquage arbitraire peut être exprimée comme une composition de distributions exponentielles et de distributions déterministes nulles. Nous pouvons donc déduire que le processus stochastique engendré par un *RdPSG* est un processus semi-markovien. Ce processus comprend deux types de marquages :

- **Marquages tangibles ("tangible markings")** : Générés suite au franchissement des transitions temporisées.
- **Marquages évanescents ("vanishing markings")** : Générés lors du franchissement des transitions immédiates.

1.5.8.3 Analyse des *RdPSG*

L'analyse des *RdPSG* est basée sur deux aspects essentiels :

► **Analyse qualitative** : consiste à vérifier les propriétés de la vivacité, la bornitude, les états d'accueil, ...etc.

► **Analyse quantitative** : Cette partie d'analyse consiste à calculer les probabilités stationnaires et les indices de performances. Elle est basée sur la chaîne de Markov associée au *RdPSG*. Cette chaîne peut être construite à partir du graphe des marquages accessibles de la manière suivante : la première étape consiste à éliminer tous les états évanescents. À partir de ce nouveau réseau, on cherche le graphe des marquages relatif aux états tangibles. Le graphe de Markov possède alors la même structure que le graphe des marquages. Il ne reste plus qu'à étiqueter les arcs par les taux de franchissement dépendants des taux de transition du réseau de Petri et du marquage. On obtient le générateur infinitésimal Q de cette *CTMC* qui est alors une matrice carrée de dimension $(r \times r)$ (r le nombre fini de marquages tangibles du *RdPSG*) qui regroupe l'ensemble des taux de transition d'un marquage vers un autre.

La distribution de probabilité $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ à l'état stationnaire sur les marquages tangibles peut alors être obtenue par la résolution de système d'équation linéaire suivant :

$$\begin{cases} \pi Q = 0. \\ \sum_{i=1}^r \pi_i = 1. \end{cases}$$

Proposition 1.2. *Le graphe de marquage d'un *RdPSG* est isomorphe à une chaîne de Markov à temps continue.*

Théorème 1.1. *Un *RdPSG* bornée et tel que son graphe des marquages accessibles est fortement connexe est ergodique.*

Théorème 1.2. *Un *RdPSG* bornée et ergodique s'il admet le marquage initial comme état d'accueil.*

Remarque 1.4. : Le processus stochastique engendré par un *RdPSG* borné avec le marquage initial comme état d'accueil, peut être classé comme un processus semi-markovien à temps continu, à espace d'états fini, stationnaire et irréductible.

1.5.8.4 Evaluation des indices de performances

Une fois le modèle est obtenu, on vérifie ses propriétés qualitative pour déduire son ergodicité pour faire de l'analyse quantitative. Si le modèle est ergodique, alors la distribution

de probabilité des marquages à l'état stationnaire existe et elle est unique [22]. Plusieurs indices de performances peuvent être calculés. Parmi ces indices les plus importants on cite :

- **Fréquence moyenne de franchissement d'une transition** : On appelle fréquence moyenne (débit moyen) de franchissement d'une transition t_i , le nombre moyen de tirs de t_i en une unité du temps. Elle est calculée par :

$$\bar{\lambda}(t_j) = \sum_{M_j \in E(t_i)} \lambda_i(M_j) \pi_j. \quad (1.13)$$

où : $E(t_i)$ est l'ensemble des marquages où la transition t_i est franchissable. $\lambda(M_j)$ est le taux de franchissement de t_i en M_j

- **Nombre moyen de marques dans une place** : Le nombre moyen de marques dans une place p est calculé en appliquant la formule :

$$E[T] = \frac{E[N]}{E[\beta]}. \quad (1.14)$$

où :

$E[T]$: est le nombre moyen de jetons dans S .

$E[\beta]$: est le taux d'arrivée effectif des jetons dans S .

1.6 Conclusion

Dans ce chapitre, nous avons présenté les différents formalismes de l'outil de modélisation *RdP*. Les *RdP* sont particulièrement bien adaptés à la description des aspects dynamiques ou comportementaux d'un système. Des concepts tels que la concurrence ou la synchronisation entre interactions s'expriment aisément dans le cadre de ces formalismes. L'avantage des *RdP* par rapport à la grande majorité des autres outils de modélisation tels que les chaînes de Markov, les systèmes de files ou réseaux de files d'attente, est leur aptitude à donner des informations concernant l'exécution du système. De la théorie mathématique, on peut déduire un certain nombre de propriétés (invariants, vivacité, existence d'un état d'accueil, ...etc.) qui sont autant d'informations à la disposition du concepteur pour se faire une idée du comportement de son système.

Chapitre 2

Systemes et reseaux de files d'attente

2.1 Introduction

Les origines de la théorie des files d'attente (FA) remontent à 1909, à l'époque où Agner Krarup Erlang a posé les bases dans ses recherches sur le trafic téléphonique. Par la suite, ses travaux ont été intégrés dans la Recherche Opérationnelle [14]. En considérant un phénomène d'attente, la théorie des files d'attente permet de modéliser ce phénomène par un système de files d'attente (SFA), de calculer ses performances et de déterminer ses caractéristiques, chose qui facilite la prise de décision pour les gestionnaires ou les décideurs [11]. Dans ce chapitre, nous allons présenter le formalisme des FA et RFA. En effet, premièrement nous allons présenter les éléments essentiels de quelques systèmes classiques de files d'attente dont l'étude nous sera nécessaire pour la compréhension des prochains chapitres.

2.2 Chaîne de Markov à temps continu

Un processus stochastique $\{X(t), t \geq 0\}$ est une chaîne de Markov à temps continu si et seulement s'il vérifie conditions suivantes :

- L'espace d'états S est dénombrable ;
- Le temps d'observation est de nature continue ;
- Le processus vérifie la propriété de Markov : $\forall n$ et $\forall t_0 < t_1, \dots, t_n$, on a :

$$P[X(t_n) = j / X(t_{n-1}) = i_{n-1}, \dots, X(t_0) = i_0] = P[X(t_n) = j / X(t_{n-1}) = i_{n-1}]$$

Une chaîne de Markov à temps continu peut être décrite soit par un diagramme de transition d'état ou bien par une matrice des taux de transition dite générateur infinitésimal.

Le diagramme de transition est un graphe orienté, les sommets correspondent aux états de la chaîne de Markov et les arcs sont étiquetés par les taux de distribution associés à la transition d'un état à un autre.

Le générateur infinitésimal Q est une matrice carrée d'ordre égal au nombre d'états de la chaîne. Q est défini par :

$$q_{ij} = \begin{cases} \mu_{ij} & \text{si } i \neq j \\ -\sum_{k=1, k \neq i} \mu_{ik} & \text{si } i = j \end{cases}$$

q_{ij} désigne le taux de transition μ_{ij} de l'état j , $i \neq j$. Les éléments diagonaux q_{ii} sont choisis, par définition, égaux à l'opposé de la somme des autres éléments de la ligne. L'espace des états est un ensemble fini ou dénombrable et le temps passé dans chaque état est une variable aléatoire réelle positive, suivant une loi exponentielle.

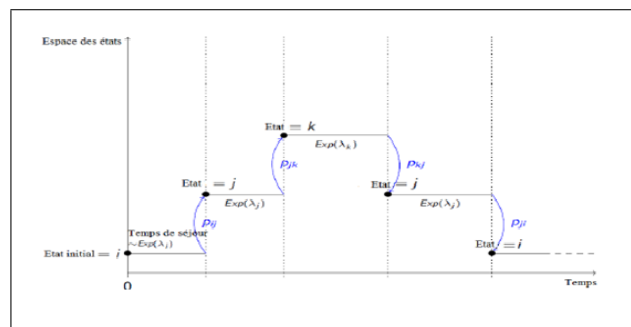


FIGURE 2.1 – La trajectoire d'une chaîne de Markov à temps continu.

2.3 Systèmes de Files d'attente

La théorie de files d'attente est une technique de la Recherche Opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les praticiens dans leurs prises de décisions. Des résultats et formulations théoriques sont bien établis pour les modèles de files d'attente avec arrivées poissonniennes et les durées de services exponentielles [3].

2.3.1 Description du File d'attente

Une file d'attente est un système stochastique composé d'une salle d'attente ou plusieurs places d'attente et d'un ou plusieurs serveurs comme illustré par la FIGURE 2.3. Lorsque plusieurs clients tentent simultanément d'obtenir un service, certains doivent patienter et attendre leurs tours dans la salle [18].

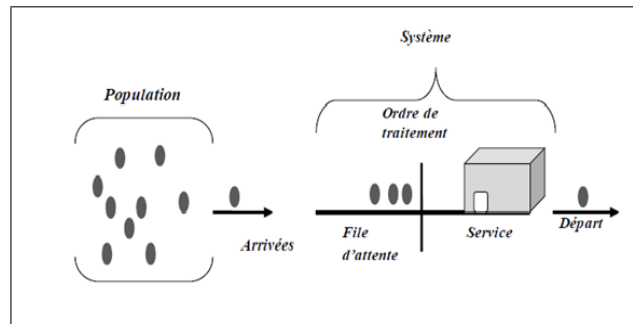


FIGURE 2.2 – Structure générale d'un système de file d'attente.

2.3.2 Classification des systèmes d'attente

Pour identifier un système d'attente, on a besoin des spécifications suivantes [37] :

- La nature stochastique du processus des arrivées, qui est défini par la distribution des intervalles séparant deux arrivées consécutives ;
- La distribution du temps aléatoire de service ;
- Le nombre m de serveurs (stations de service) qui sont montées en parallèle. On admet généralement que les temps de service correspondants suivent la même distribution et que les clients qui arrivent forment une seule file d'attente (dans le cas homogène) ;
- La capacité N du système. Si $N < \infty$, la file d'attente ne peut dépasser une longueur de $N - m$ unités. Dans ce cas, certains clients arrivant vers le système n'ont pas la possibilité d'y entrer ;
- La source des clients potentiels.

2.3.3 Mesures de performance d'une file d'attente

L'étude d'une FA ou RFA a pour but de calculer ou d'estimer les performances du système dans des conditions de fonctionnement données. Ce calcul se fait le plus souvent pour le régime stationnaire uniquement, et les mesures les plus fréquemment utilisées sont :

- \bar{N} : le nombre moyen de clients dans le système ;
- \bar{Q} : le nombre moyen de clients dans la file d'attente ;
- \bar{T} : le temps moyen de séjour d'un client dans le système ;
- \bar{W} : le temps moyen d'attente d'un client dans la file ;
- \bar{U} : le taux d'utilisation de chaque serveur ;
- \bar{S} : le temps moyen de service ;
- \bar{A} : le temps moyen entre deux arrivées.

Remarque 2.1. De manière générale, une file est stable si et seulement si le nombre moyen d'arrivées de clients par unité de temps, noté λ , est inférieur au nombre moyen des clients pouvant être servis par unité de temps. Si chaque serveur peut traiter μ clients par unité de temps et si le nombre de serveurs est m , une file est stable si et seulement si :

$$\lambda < m\mu \Leftrightarrow \rho = \frac{\lambda}{m\mu} < 1;$$

où ρ : est appelé l'intensité du trafic.

Formule de Little

Soient λ le taux des arrivées, λ_e le taux réel des arrivées (taux d'entrée), \bar{T} le temps moyen de séjour et \bar{N} le nombre moyen de clients présents dans le système. De plus, si λ , \bar{N} et \bar{T} existent, ils sont reliés l'un à l'autre par l'équation suivante :

$$\bar{N} = \lambda_e \bar{T}. \quad (2.1)$$

Cette expression appelée formule de Little [30] est l'un des résultats les plus généraux et utile dans la théorie des files d'attente. En utilisant cette formule, on obtient également :

- $\bar{Q} = \lambda \bar{W}$;
- $\bar{T} = \bar{W} + \frac{1}{\mu}$; où μ représente le taux de service ;
- $\bar{N} = \bar{Q} + \rho$.

2.3.4 Les différentes disciplines de service

La discipline de service, est la règle de priorité déterminant l'ordre dans lequel les clients vont accéder à la ressource modélisée par le serveur. Les disciplines d'attente classiques, ainsi que leurs acronymes, sont :

- **FIFO** : (first in, first out) ou *FCFS* (first come first served) ou *PAPS* (premier arrivé, premier servi) : c'est la file standard dans laquelle les clients sont servis dans leur ordre d'arrivée. Notons que les disciplines *FIFO* et *FCFS* ne sont pas équivalentes lorsque la file contient plusieurs serveurs. Dans la première, le premier client arrivé sera le premier à quitter la file alors que dans la deuxième, il sera le premier à commencer son service. Rien n'empêche alors qu'un client qui commence son service après lui, dans un autre serveur, termine avant lui. En français, le terme *PAPS* comporte une ambiguïté, puisqu'il ne peut différencier une file "premier arrivé, premier servi" d'une file "premier arrivé, premier sorti".

- **LIFO** : (last in, first out) où *LCFS* (last come, first served) où DAPS (dernier arrivé, premier servi). Cela correspond à une pile, dans laquelle le dernier client arrivé (donc posé sur la pile) sera le premier traité (retiré de la pile). A nouveau, les disciplines *LIFO* et *LCFS* ne sont pas équivalentes que pour une file mono-serveur.
- **RANDOM (aléatoire)** : Le prochain client qui sera servi est choisi aléatoirement dans la file d'attente.
- **Round-Robin (cyclique)** : Tous les clients de la file d'attente entrent en service à tour de rôle, effectuant un quantum Q de leur temps de service et sont replacés dans la file, jusqu'à ce que leur service soit totalement accompli. Cette discipline de service a été introduite afin de modéliser des systèmes informatiques.
- **Ps (Processor Sharing)** : C'est le cas limite de la distribution Round-Robin lorsque le quantum de temps Q tend vers 0. Tous les clients sont servis en même temps, mais avec une vitesse inversement proportionnelle au nombre de clients simultanément présents. Si le taux du service est égal à μ et qu'à un instant donné il y a n clients dans la station, tous les clients sont donc servis simultanément avec un taux $\frac{\mu}{n}$.
- **Avec priorité** : Chaque client a une priorité (statique ou dynamique, absolue ou relative), le serveur sélectionne le client de haute priorité.
 1. **Priorité relative** : Un client accède au service selon sa priorité. La file est gérée par ordre de priorité de la plus forte à la plus faible.
 2. **Priorité absolue** : Le service d'un client est interrompu lorsqu'un client de priorité supérieure se présente devant la file d'attente. Le client dont ce service est interrompu est remis en tête de la file.

2.3.5 Notation de Kendall

Un modèle de file d'attente est totalement décrit selon la notation de Kendall. Dans sa version étendue, un SFA est spécifié par une suite de six symboles [37] :

$$A/B/s/N/K/D.$$

La signification de chacun de ces symboles est :

- A : Nature du processus des arrivées ;
- B : Nature du processus de service ;
- s : Nombre de serveurs en parallèle ;
- N : Capacité du système (serveurs + file d'attente) ;

- K : Taille de la population.

Dans la description des processus d'arrivée et de service, les symboles les plus courants sont :

- M : Distribution sans mémoire (qui vérifie donc la propriété de Markov) ;
- E : Distribution d'Erlang ;
- G : Distribution générale ;
- D : loi Déterministe (temps d'inter-arrivées ou de service constant).

La forme abrégée : $A/B/s$ signifie que N et K sont infinies.

2.4 Analyse mathématique d'un SFA

L'étude mathématique d'un SFA se fait généralement par l'introduction d'un processus stochastique, défini de façon appropriée. On s'intéresse principalement au processus $\{X(t)\}_{t \geq 0}$, qui décrit le nombre de clients se trouvant dans le système à l'instant $t (t \geq 0)$. En fonction des quantités qui définissent le système, on cherche à déterminer :

- Les probabilités d'état $P_n(t) = P(X(t) = n)$, qui définissent le régime transitoire du processus stochastique $\{X(t), t \geq 0\}$. Il est évident que les fonctions $P_n(t)$ dépendent de l'état initial ou de la distribution initiale du processus.

Le régime stationnaire du processus stochastique est défini par :

$$\pi_n = \lim_{n \rightarrow \infty} p_n(t) = P(X(+\infty) = n), n = 1, 2, 3, \dots \quad (2.2)$$

où, $\{\pi_n\}_{n \geq 0}$ est appelée distribution stationnaire du processus $\{X(t), t \geq 0\}$. A partir de cette distribution on pourra obtenir d'autres caractéristiques du système telles que :

- Le nombre moyen de clients dans le système $L = E(X)$;
- La durée d'attente d'un client ;
- La durée moyen de séjour dans le système qui est composée de la durée moyen d'attente et la durée moyen de service ;
- Le taux d'occupation des postes de service ;
- Le pourcentage de clients n'ayant pu être servi ;
- La durée moyen d'une période d'activité, c'est-à-dire l'intervalle de temps pendant lequel il y a toujours au moins un client dans le système.

Remarque 2.2. Il faut toutefois, constater que le calcul explicite du régime transitoire s'avère pénible, voire impossible, pour la plupart des modèles considérés, mis à part certains modèles particulièrement faciles à traiter. Nous nous contenterons donc dans la suite de déterminer le régime stationnaire d'un système d'attente. Notons qu'il existe des systèmes d'attente dont l'évolution temporelle n'est plus déterminée par le processus $\{X(t), t \geq 0\}$.

2.5 Les files d'attente markoviennes

Les modèles markoviens caractérisent les systèmes dans lesquels les deux quantités stochastiques principales, qui sont le temps inter-arrivées et la durée de service, sont des variables aléatoires indépendantes et exponentiellement distribuées. La propriété d'absence de mémoire de la loi exponentielle facilite l'étude de ces modèles. L'étude mathématique de tels systèmes se fait par l'introduction d'un processus stochastique approprié. Ce processus est souvent le processus $\{X(t), t \geq 0\}$ défini comme étant le nombre de clients dans le système à l'instant t . L'évolution temporelle du processus markovien est complètement définie grâce à la propriété d'absence de mémoire.

2.5.1 La file d'attente $M/M/1$:

Le SFA $M/M/1$ est le système d'attente markovien le plus simple. Dans ce système on a qu'un seul serveur, le processus d'arrivées est un processus de Poisson de paramètre λ , et le temps de service pour chaque client suit une loi exponentielle de paramètre μ .

► **Régime transitoire :**

Soit le processus $\{X(t)\}_{t \geq 0}$ le nombre de clients présents dans le système à l'instant $t (t > 0)$. Grâce aux propriétés fondamentales du processus de Poisson et de la loi exponentielle, $X(t)$ est un processus markovien homogène. Les probabilités d'état $P_n(t) = P(X(t) = n)$ peuvent être calculées par les équations différentielles de Kolmogorov ci-dessous, connaissant les conditions initiales du processus :

$$[p'_n(t) = -(\lambda + \mu)p_n(t) + \lambda p_{n-1}(t) + \mu p_{n+1}(t)].$$

et

$$[p_0(t) = -\lambda p_0(t) + \mu p_1(t)].$$

► **Régime stationnaire :**

Sous la condition d'ergodicité du système $\rho = \frac{\lambda}{\mu}$, pour laquelle le régime stationnaire existe, il est aisé d'obtenir les probabilités stationnaires :

$$\pi_n(t) = \lim_{t \rightarrow \infty} p_n(t) = (1 - \rho)\rho^n, \forall n \in \mathbb{N}.$$

$\pi = \{\pi_n\}_{n \geq 0}$ est appelé distribution stationnaire, elle suit une loi géométrique. Les caractéristiques de ce système sont données par les propositions suivantes :

Proposition 2.1. *Le nombre moyen de clients dans le système est :*

$$\bar{N} = E(X) = \sum_{n=0}^{+\infty} n\pi_n = (1 - \rho) \sum_{n=0}^{+\infty} n\rho^n. \quad (2.3)$$

D'où pour $\rho < 1$:

$$\bar{N} = \frac{\rho}{1 - \rho}. \quad (2.4)$$

Proposition 2.2. *Le nombre moyen de clients dans la file est :*

$$\bar{Q} = \sum_{n \geq 1}^{+\infty} (n - 1)\pi_n = \frac{\rho^2}{1 - \rho}. \quad (2.5)$$

Remarque 2.3. Le temps moyen de séjour dans le système \bar{T} et le temps moyen d'attente dans la file \bar{W} sont obtenus à partir de la formule de Little par :

- Le temps moyen de séjour dans le système est :

$$\bar{T} = \frac{\bar{N}}{\lambda} = \frac{1}{\mu(1 - \rho)}. \quad (2.6)$$

- Le temps moyen d'attente dans la file est :

$$\bar{W} = \frac{\bar{Q}}{\lambda} = \frac{\rho}{\mu - \lambda} = \frac{\rho}{\mu(1 - \rho)}. \quad (2.7)$$

Théorème 2.1. [8] *Sous la condition de stabilité $\lambda < \mu$, à l'équilibre, le processus de départ de la file $M/M/1$ a la même loi que le processus des arrivées et ce d'après le théorème de Burk. De plus, le nombre de clients dans la file à l'instant $t = t_0$ est indépendant du processus des départs avant $t = t_0$.*

2.5.2 La file d'attente $M/M/m$

Dans ce modèle, m serveurs identiques et indépendants partagent les mêmes places d'attente. Les arrivées suivent un processus de Poisson de paramètre λ et la durée de chaque service est une variable exponentielle de paramètre μ . Les caractéristiques de ce système sont données par les relations suivantes :

Proposition 2.3. *La probabilité qu'il y ait n clients dans le système à l'instant d'entrée est :*

$$p(n) = \begin{cases} \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} p_0, & \text{si } n \leq m; \\ \frac{\left(\frac{\lambda}{\mu}\right)^n}{m!m^{n-m}} p_0, & \text{si } n \geq m; \end{cases} \quad \text{où : } p_0 = \left[\sum_{n \geq 0}^m \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^m + 1}{m! \left(m - \frac{\lambda}{\mu}\right)} \right]^{-1}.$$

Par ailleurs la probabilité qu'il y est d'attente dans ce système est donnée par :

$$\zeta = P(\text{attente}) = P(X \geq m) = \frac{p_m}{1 - \rho} \quad (2.8)$$

Remarque 2.4. Voici quelques caractéristiques de ce système :

- Le taux d'utilisation de chaque serveur :

$$U = \rho = \frac{\lambda}{m\mu}.$$

- Le nombre moyen de clients présents dans le système et en attente sont respectivement :

$$\begin{aligned}\bar{N} &= m\rho + \frac{\rho\zeta}{1+\rho}; \\ \bar{Q} &= \frac{\rho\zeta}{1-\rho}.\end{aligned}$$

- Le temps moyen de réponse et d'attente sont respectivement :

$$\begin{aligned}\bar{T} &= \frac{1}{\mu} \left(1 + \frac{\zeta}{m(1-\rho)} \right); \\ \bar{W} &= \frac{\zeta}{m\mu(1-\rho)}.\end{aligned}$$

2.5.3 La file d'attente $M/M/m/K$

La file $M/M/m/K$ est une file markovienne de capacité K et de m serveurs. Le nombre maximal de clients en attente est donc $K - m$. Si un client arrive alors que le système est plein, il ne peut pas entrer au système donc il sera rejeté, donc ce système est toujours stable quel que soit l'intensité du trafic $\rho = \frac{\lambda}{\mu} < 1$. Le taux de service de cette file est :

$$\mu_k = \begin{cases} k\mu, & k = 1, 2, \dots, m-1; \\ m\mu, & k = m, m+1, \dots, K. \end{cases}$$

Comme tout client arrivant alors que le système est plein doit repartir, le taux effectif d'arrivées dans le système n'est pas λ mais $\lambda_e = \sum_{k=0}^{K-1} \lambda p_k = \lambda(1 - p_K)$ où p_k est la probabilité qu'il y ait k clients dans le système. La distribution stationnaire est donnée d'après la distribution du Processus de Naissance et de Mort par :

$$\begin{cases} \frac{\rho^n}{n!} p_0, & \text{si } 0 \leq n \leq m; \\ \frac{\rho^n}{m!m^{n-m}} p_0, & \text{si } m \leq n \leq K; \end{cases} \quad (2.9)$$

avec :

$$p_0 = \frac{1}{\sum_{n=0}^m \frac{\rho^n}{n!} + \sum_{n=m+1}^K \frac{\rho^n}{m!m^{n-m}}}. \quad (2.10)$$

Proposition 2.4. [6] On a :

$$\begin{aligned} p(\text{perte}) &= P(\text{tous les serveurs sont occupés}) = p_m; \\ &= \frac{\rho^n}{m!m^{n-m}}p_0. \end{aligned}$$

avec p_0 est donnée par (2.10).

En ce qui concerne les caractéristiques du système, on a :

$$\bar{N} = \sum_{k=0}^K k p_k; \quad (2.11)$$

$$\bar{Q} = \sum_{k=m+1}^K (k - m) p_k. \quad (2.12)$$

C'est ce taux effectif λ_e qu'il faut utiliser pour calculer \bar{T} et \bar{W} à l'aide des formules de Little.

$$\lambda_e = \sum_{k=0}^{K-1} \lambda \pi_k = \lambda (1 - \pi_K)$$

2.6 Les files d'attente non markoviennes

En l'absence des processus sans mémoire dans les SFA à étudier, par exemple on s'écartant de l'hypothèse d'exponentialité de l'une des deux quantités stochastiques : le temps des inter-arrivées et/ou la durée de service, ou en prenant en compte certaines spécificités des problèmes par introduction de paramètres tels que la priorité, les rappels, ..., on aboutit à un modèle non markovien. La combinaison de tous ces facteurs rend l'étude mathématique du modèle très délicate, voire impossible, on essaye alors de se ramener à un processus de Markov judicieusement choisi à l'aide de l'une des méthodes d'analyse suivantes :

1. **Méthode des étapes d'Erlang** : Son principe est d'approximer toute loi de probabilité ayant une transformation de Laplace rationnelle par une loi de Cox (mélange de lois exponentielles), cette dernière possède la propriété d'absence de mémoire par étape.
2. **Méthode de la chaîne de Markov induite** : Élaborée par Kendall, cette méthode est souvent utilisée. Elle consiste à choisir une suite d'instantanés $1, 2, 3, \dots, n$ (déterministes ou aléatoires) tels que la chaîne induite $\{X_n, n \geq 0\}$, où $X_n = X(n)$, soit markovienne et homogène.

3. **Méthode des variables supplémentaires** : Elle consiste à compléter l'information sur le processus $\{X_t, t \geq 0\}$ de telle manière à lui donner le caractère markovien. Ainsi, on se ramène à l'étude du processus $\{X(t), A(t_1), A(t_2), A(t_3), \dots, A(t_n), t \geq 0\}$. Les variables $A(t_k), k \in \{1, 2, \dots, n\}$ sont dites supplémentaires.
4. **Méthode des événements fictifs** : Le principe est d'introduire des événements fictifs qui permettent de donner une interprétation probabiliste aux transformées de Laplace et aux variables aléatoires décrivant le système étudié.
5. **Simulation** : C'est un procédé d'imitation artificielle d'un processus réel effectué sur ordinateur. Elle nous permet d'étudier les systèmes les plus complexes, de prévoir leurs comportements et de calculer leurs caractéristiques. Les résultats obtenus ne sont qu'approximatifs, mais peuvent être utilisés avec une bonne précision. Cette technique se base sur la génération de variables aléatoires suivant les lois gouvernant le système.

2.6.1 La file d'attente $M/G/1$

La file $M/G/1$ est une file d'attente à serveur unique, le processus d'arrivée est Poisson avec le taux λ . Le service est assuré par un seul serveur, le temps de service suit une loi non négative quelconque d'espérance \bar{S} et de variance σ^2 finies. A l'arrivée d'un client, si le serveur est libre, le client sera pris en charge immédiatement, dans le cas contraire, il rejoint la file d'attente (de capacité illimitée et de discipline *FIFO*), les durées de service sont des variables aléatoires indépendantes et sont identiquement distribuées et de loi générale.

La formule de Pollaczek-Khinchin :

La formule de Pollaczek-Khinchin est un résultat très élégant montrant que les différences de performances entre une file $M/G/1$ et une file $M/M/1$ se résument à un facteur multiplicatif.

Théorème 2.2. Formule de Pollaczek-Khinchin [22]

Le nombre moyen de clients en attente dans une file d'attente $M/G/1$ sous la condition de stabilité $\rho = \lambda \bar{S} < 1$, est donné par :

$$\bar{Q} = \left(\frac{1 + C_2^S}{2} \right) \frac{\rho^2}{1 + \rho}; \quad (2.13)$$

où C_2^S est le coefficient de variation au carré du temps de service $\left(C_2^S = \frac{\sigma_S^2}{\bar{S}^2} \right)$.

En caractérisant les performances par le type de file en question, l'expression (2.13) de-

vient :

$$\bar{Q}_{M/G/1} = \left(\frac{1 + C_2^S}{2} \right) \bar{Q}_{M/M/1}. \quad (2.14)$$

D'après la formule de Little, le temps moyen d'attente est donné par :

$$\bar{W}_{M/G/1} = \left(\frac{1 + C_2^S}{2} \right) \bar{W}_{M/M/1} = \left(\frac{1 + C_2^S}{2} \right) \frac{\rho \bar{S}}{1 + \rho}. \quad (2.15)$$

Pour le calcul de \bar{N} et \bar{T} on utilise les relations suivantes :

$$\bar{N} = \bar{Q} + U = \left(\frac{1 + C_2^S}{2} \right) \frac{\rho^2}{1 - \rho} + \rho; \quad (2.16)$$

$$T = W + \bar{S} = \left(\frac{1 + C_2^S}{2} \right) \frac{\rho^2 \bar{S}}{1 - \rho} + \bar{S}; \quad (2.17)$$

le taux d'utilisation du serveur étant toujours $U = \rho$.

2.6.2 La file d'attente $G/M/1$

Ce SFA peut être considéré comme un système symétrique au système précédent i.e. $M/G/1$. Il possède un processus d'arrivées caractérisé par les intervalles de temps entre chaque deux arrivées successives indépendantes et identiquement distribuées et une distribution de service exponentielle de taux $\mu = \frac{1}{s}$ [23].

◇ Le temps moyen d'attente dans la file est donné par :

$$\bar{W} = \frac{\sigma \bar{S}}{1 - \sigma}. \quad (2.18)$$

◇ Le nombre moyen de clients dans la file d'attente est donné par :

$$\bar{N} = \frac{\rho \sigma}{1 - \sigma}. \quad (2.19)$$

2.7 Réseaux de files d'attente

Les réseaux de files d'attente (RFA) sont un ensemble des stations de systèmes d'attente inter-connectées. Le séjour d'un client dans un réseau de files d'attente consiste à parcourir une partie ou l'ensemble de toutes ces SFA. Le besoin des RFA est apparu dans les années 70 pour étudier les performances des systèmes informatiques et télématiques. La définition d'un RFA requiert d'une part la définition de toutes les files qui le constituent et d'autre part la définition du routage entre ses files. En effet, lorsqu'un client arrive dans le RFA où termine son service à une station i , il faut préciser où ce client va se rendre : soit à une autre station j , soit à l'extérieur (le client quitte alors le Réseau, c'est à dire il faut préciser la matrice de routage des clients dans ce réseau. En effet, plusieurs types de routages existent :

- **Routage probabiliste** : soit r_{0i} la probabilité pour qu'un client qui arrive se rende à la station i , r_{ij} la probabilité pour qu'un client qui quitte la station i se rende à la station j et r_{i0} la probabilité pour qu'un client qui quitte la station i quitte le système. Les r_{ij} sont telles que $\sum_{j=0}^M r_{ij} = 1, i = \overline{0, M}, M$ est le nombre de stations du réseau. L'ensemble des probabilités de routage de toutes les stations sont regroupées dans une matrice de routage carrée de dimension $M \times M$.
- **Routage dynamique** : est un routage qui se fait suivant l'état du système au moment du routage. Par exemple, un client sortant d'une station i peut choisir la station comportant moins de clients parmi ces destinations possibles.
- **Routage cyclique** : un client sortant d'une station choisira à tour de role une de ces destinations possibles.

Les RFA sont classés en deux catégories principales : RFA mono-classes et RFA multi-classes. Dans ces classes on fait également la distinction entre RFA ouverts et RFA fermés. Dans un RFA ouvert, les clients arrivent de l'extérieur, circulent à travers des noeuds et quittent le RFA. Dans un RFA fermé, il y a un nombre fixe de clients circulant à travers les noeuds sans qu'il n'y ait ni des arrivées ni des départs de clients du réseau.

Dans le cas de RFA multi-classes, il faut préciser pour chaque classe de clients s'il s'agit d'une classe ouverte ou d'une classe fermée. Si toutes les classes de clients sont des classes ouvertes, on parlera de RFA purement ouverts et si toutes les classes de clients sont des classes fermées, on parlera de RFA purement fermé. Un RFA parcouru à la fois par des classes ouvertes et des classes fermées sera qualifié de RFA mixte. Dans ce qui suivra on s'intéressera au cas de la catégorie de RFA mono-classe fermé qui fait l'objet de ce mémoire.

Des solutions exactes existent pour une classe bien particulière de RFA connus sous le nom des réseaux séparables où à forme produit (Jackson 1963, Basket et al 1975, Gordon et Newell 1967, ...). Cette classe de réseaux à forme produit est très restreinte cependant des extensions fait actuellement l'objet d'axes de recherche. La difficulté des extensions provient de la difficulté de la nature des flux inter-stations de ces réseaux. L'analyse des RFA non à forme produit est très difficile. Cependant, de nombreuses méthodes approximatives d'évaluation des performances des *RFA* non à forme produit ou ayant une forme produit difficile à exploiter ont été développées, telles que :

- La méthode d'isolation ;
- La méthode par décomposition ;
- Les méthodes numériques ;
- La méthode d'agrégation ; ...

2.7.1 RFA à forme produit

La solution à forme produit, introduite par Jackson [21] en 1975, permet d'exprimer la solution stationnaire du nombre de clients dans un RFA comme le produit des probabilités marginales de la longueur de chaque file de ce réseau, ce qui simplifie le calcul explicite ou numérique de sa distribution. A cause de la simplicité de sa structure, la distribution stationnaire à forme produit constitue un outil analytique puissant.

Les RFA à forme produit ont une expression simple de la distribution stationnaire d'état qui permettent de définir des algorithmes efficaces pour évaluer les mesures de performances moyennes. Ainsi, des modèles de RFA ont été intensivement appliqués pour représenter et analyser différents systèmes tels que les systèmes de communication et des systèmes informatiques et ils se sont avérés être un outil puissant et souple pour l'évaluation et la prévision des performances de ces systèmes. Malheureusement, peu de RFA ont une solution simple de cette sorte. Ceci provient de la difficulté d'étudier les propriétés des flux, à l'intérieur des réseaux. Ces dernières années, les chercheurs se sont beaucoup investis pour étendre l'ensemble des RFA admettant une solution à forme produit. Dans les sections suivantes, nous allons présenter les RFA à forme produit classiques principaux à savoir les RFA markoviens.

2.7.2 Les réseaux markoviens ouverts

Un réseau de files d'attente est un ensemble de files d'attente inter-connectées. Dans un réseau de files d'attente ouvert, les clients arrivent de l'extérieur, circulent dans le réseau à travers les différentes stations, puis quittent le réseau. Le nombre de clients pouvant se trouver à un instant donné dans un réseau ouvert n'est pas limité. La FIGURE 2.3 ci-dessous illustre un exemple de réseau mono-classe ouvert [20] :

► **Processus d'arrivées** : Le processus d'arrivée des clients dans le réseau sera décrit, comme pour une file simple, à l'aide d'un processus de renouvellement (et sera donc caractérisé par la distribution du temps d'inter-arrivées).

Si l'arrivée des clients suit un processus de Poisson, les inter-arrivées sont exponentielles et sont caractérisées par un unique paramètre : le taux d'arrivée λ . Il faut préciser, lorsqu'un client arrive dans le réseau, à quelle file il se rend. On caractérisera la plupart du temps le routage d'entrée de façon probabiliste : soit p_{0i} la probabilité pour qu'un client qui arrive, se rende à la station i . Les probabilités p_{0i} sont bien sûr telles que $\sum_{i=1}^M p_{0i} = 1$; où : M est le nombre de stations.

► **Routage des clients** : Lorsqu'un client termine son service à une station, il faut préciser

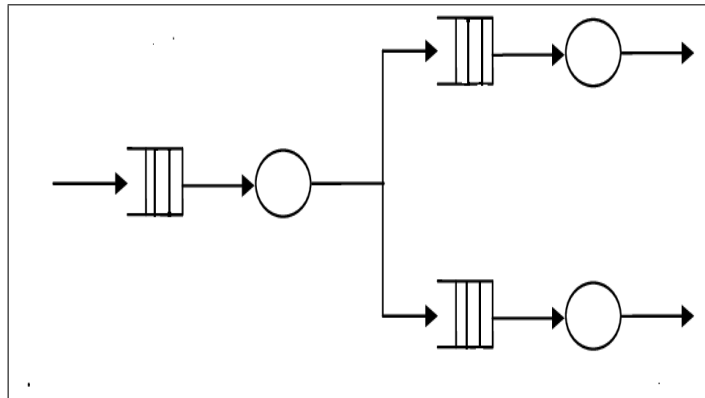


FIGURE 2.3 – Réseau ouvert de files d'attente.

où ce client va se rendre :

soit à une autre station, soit à l'extérieur (le client quitte alors le réseau). A nouveau, le routage des clients est très souvent caractérisé de façon probabiliste : soit p_{ij} la probabilité pour qu'un client qui quitte la station i se rende à la station j et soit p_{i0} la probabilité pour qu'un client qui quitte la station i quitte le système. Les p_{ij} sont telles que $\sum_{j=0}^M p_{ij} = 1$.

2.7.3 Les réseaux markoviens fermés

Les réseaux markoviens fermés sont des RFA dans lesquels N clients circulent, mais pas de clients n'y accèdent de l'extérieur à ce réseau et pas de clients ne quittent ce réseau vers l'extérieur, bien sur les processus décrivant le comportement de ces réseaux ils sont markoviens La spécification d'un réseau fermé se réduit donc à celle des différentes stations et à celle du routage des clients. Pour un mécanisme de routage probabiliste, on définit p_{ij} la probabilité qu'un client qui quitte la station i se rende à la station j . Les p_{ij} sont telles que :

$$\sum_{j=1}^M p_{ij} = 1.$$

Les réseaux fermés ont beaucoup d'applications réelles importantes, par exemple, ils ont été utilisés pour modéliser un système informatique multi-programmation dans lequel seul, un nombre fixe de processus peut être traité à la fois [39]. Ces réseaux sont connus sous le nom de réseaux de Jackson fermés ou encore réseaux de Gordon et Newell (voir l'exemple illustré dans la FIGURE 2.4) suivante :

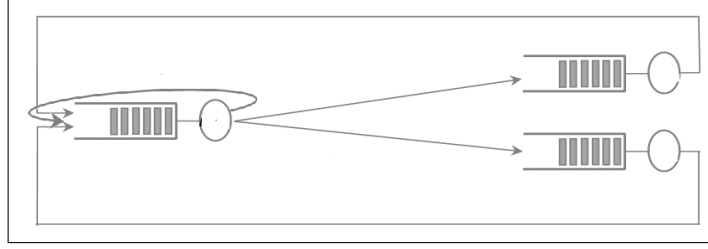


FIGURE 2.4 – Exemple de réseau fermé de FA.

Dans ces réseaux, l'ergodicité est vérifiée et ce suite à la seule contrainte de population. Les équations d'états se déduisent plus simplement que dans le cas ouvert et la probabilité stationnaire est donnée par le théorème de Gordon-Newell suivant :

Théorème 2.3. [17] *La probabilité stationnaire du réseau est sous la forme produit suivante :*

$$p(n) = \frac{1}{G(M, N)} \prod_{i=1}^M f_i(n_i); \text{ où } f_i(n_i) = \left(\frac{e_i}{\mu_i} \right)^{n_i}, \quad (2.20)$$

$$\text{avec } e_i = \sum_{j=1}^M e_j p_{ji} \text{ est le taux de visite de la station } i;$$

$G(M, N)$ est la constante de normalisation.

Parmi les méthodes numériques conçues pour l'analyse des indices de performance des RFA à forme produit on cite l'algorithme de convolution et l'algorithme MVA (Mean Value Analysis) suivant :

1.7.3. Algorithme de convolution

Le calcul de la constante de normalisation $G(M, N)$ conduirait à des sommations multiples dont la complexité est exponentielle. Heureusement, des algorithmes efficaces dits de "convolution" permettent de l'obtenir d'une manière simple [7, 9] En effet, de la condition de normalisation de la distribution stationnaire $\sum_{n \in E(M, N)} p(n) = 1$, on déduit l'expression :

$$G(M, N) = \sum_{E(M, N)} \prod_{i=1}^M f_i(n_i) = \sum_{E(M, N)} \prod_{i=1}^M \left(\frac{e_i}{\mu_i} \right)^{n_i} = \sum_{E(M, N)} \prod_{i=1}^M \rho_i^{n_i}.$$

Ceci peut s'exprimer d'une manière équivalente :

$$G(M, N) = \rho_1^N + \rho_1^{N-1} \rho_2 + \dots + \rho_1^{N-1} \rho_M + \rho_1^{N-2} \rho_2^2 + \dots + \rho_M^N.$$

On peut alors obtenir la formule de récurrence suivante :

$$G(M, N) = G(M - 1, N) + \rho_M G(M, N - 1); \text{ avec } \rho_M = \frac{e_M}{\mu_M}. \quad (2.21)$$

Cette formule de récurrence permet de calculer toute les constantes de normalisation $G(m, n)$, pour $m = \overline{1, M}$ et $n = \overline{0, N}$ en partant des conditions initiales :

$$G(1, n) = \rho_1^n, n = \overline{0, N} \quad \text{et} \quad G(m, 0) = 1, \quad m = \overline{1, M}.$$

Contrairement au cas des réseaux ouverts, les paramètres de performances de chaque station ne peuvent plus se déduire de l'analyse d'une file simple en isolation mais plutôt de la manipulation de l'expression des probabilités stationnaires. En effet, par sommation sur les probabilités stationnaires, on obtient les probabilités marginales :

$$p_i(k) = \sum_{n \in E(M, N) \setminus n_i = k} p(n). \quad (2.22)$$

Les paramètres de performances de chaque station s'en déduisent immédiatement :

$$U_i = 1 - p_i(0), \quad X_i = \sum_{k=1}^N p_i(k) \mu_i = (1 - p_i(0)) \mu_i, \quad Q_i = \sum_{k=1}^N k p_i(k). \text{ et } R_i = \frac{Q_i}{X_i}.$$

Un calcul des probabilités marginales par sommation sur les probabilités stationnaires nécessite d'effectuer des sommations multiples qui se compliquent au fur et à mesure que le nombre de stations du réseau s'agrandi. Néanmoins, on peut éviter ces sommations en remplaçant dans la formule (2.22) l'expression des probabilités stationnaires. On peut alors exprimer les paramètres de performances de la station i en fonction des constantes de normalisation. Ainsi, on obtient :

$$U_i = \rho_i \frac{G(M, N - 1)}{G(M, N)}, \quad X_i = e_i \frac{G(M, N - 1)}{G(M, N)}. \quad (2.23)$$

où $G_i(M - 1, n)$ est la constante de normalisation du réseau complémentaire, i.e. la constante de normalisation du réseau constitué des M stations du réseau initial privé de la station i avec : $G_i(M - 1, n) = G(M, N) - \rho_i G(M, n - 1), i = \overline{1, M}$. On aboutit ainsi à l'algorithme de convolution dont la complexité est en $O(MN)$.

Théorème de Gordon-Newell a été étendu au cas de stations multi-serveurs. Toutes les hypothèses étant conservées, la probabilité stationnaire possédera la forme produit suivante :

$$p(n) = \frac{1}{G(M, N)} \prod_{i=1}^M f_i(n_i).$$

$$\text{où } : f_i \begin{cases} \frac{1}{n_i!} \left(\frac{e_i}{\mu_i} \right)^{n_i} & \text{si } n_i < s_i \\ \frac{1}{s_i! s_i^{n_i - s_i}} \left(\frac{e_i}{\mu_i} \right)^{n_i} & \text{si } n_i \geq s_i. \end{cases} \quad (2.24)$$

et $G(M, N)$ est une constante de normalisation.

1.7.4. Algorithme MVA

La difficulté de l'algorithme de convolution réside dans le calcul des constantes de normalisation et des constantes de normalisation complémentaires nécessaires à l'obtention des probabilités marginales et des paramètres de performances moyens. Cependant, si seuls les paramètres de performances sont requis, il existe un algorithme récursif performant, développé par Reiser [32, 34] appelé algorithme *MVA* "Mean Value Analysis". Le principe de cet algorithme est d'exprimer les paramètres de performances moyens du réseau contenant N clients en fonction des paramètres de performances du même réseau contenant un client en moins. L'algorithme repose sur la relation recursive :

$$R_i(N) = \frac{1}{\mu_i} (1 + Q_i(N - 1)).$$

où $R_i(N)$ est le temps moyen de séjour d'un client dans la station i , dans le réseau contenant N clients et $Q_i(N - 1)$ est le nombre moyen de clients de la station i dans le réseau contenant $N - 1$ clients. La seconde relation-clé de cet algorithme est tout simplement la loi de Little [24] appliquée sur l'ensemble du réseau qui s'exprime par :

$$N = R(N)X(N) = \left(\sum_{i=1}^M e_i R_i(N) \right) X(N);$$

où $X(N)$: est le débit moyen.

Ainsi, en partant des conditions initiales évidentes $Q_i(0) = 0; \forall i = \overline{1, M}$, et grâce à l'algorithme MVA, on obtient les paramètres de performances suivants :

$$X(n) = \frac{n}{\sum_{i=1}^M e_i R_i(n)}; \quad X_i(n) = e_i X(n); \quad Q_i(n) = R_i(n) X_i(n); \quad \forall i = 1, M.$$

Remarque 2.5. L'algorithme *MVA* s'étend également au cas à stations multiserveurs et au cas de stations à taux de service dépendant de l'état [4, 33]. Sa complexité est la même que celle de l'algorithme de convolution, soit $O(MN^2)$.

2.8 Algorithme MVA pour les réseaux purement fermés

peut être généralisé au cas d'un réseau multi-classe purement fermé. Le principe de base reste le même puisque l'algorithme consiste à exprimer les paramètres de performances du réseau ayant une population $N = (N_1, \dots, N_C)$, en fonction de ceux du même réseau mais contenant un client de classe c en moins (et donc ayant une population $N - 1_c = (N_1, \dots, N_{c-1}, N_{c+1}, \dots, N_C)$) Les relations donnant les paramètres de performances doivent être réécrites, en y faisant clairement apparaître la population du réseau :

$$p_i(k, N) = f_i(k) \frac{G_i(M-1, N-k)}{G(M, N)}. \quad (2.25)$$

$$X_{ic}(N) = e_{ic} \frac{G(M, N-1_c)}{G(M, N)}. \quad (2.26)$$

$$Q_{ic}(N) = \sum_{k=(0,0,\dots,0)}^N k_c p_i(k, N). \quad (2.27)$$

$$R_{ic}(N) = \frac{Q_{ic}(N)}{X_{ic}(N)}. \quad (2.28)$$

La relation la plus importante de l'algorithme *MVA* multi-classes est celle exprimant le temps moyen de séjour d'un client de classe c à la station i (de type 1, 2, ou 4), dans le réseau ayant une population N , en fonction du nombre moyen de clients présents à la station i , toutes classes confondues, dans le réseau ayant une population $N - 1_c$:

$$R_{ic}(N) = \frac{1}{\mu_{ic}} \quad (2.29)$$

Le MVA a été développé par Reiser et Lavenberg pour l'analyse des RFA fermés avec une solution sous forme de produit. L'avantage de cet algorithme est que les mesures de performance peuvent être calculées sans calculer explicitement la constante de normalisation [3]. Cependant la complexité de cet algorithme se complique au fur et à mesure que le nombre de stations ou le nombre de clients dans ces réseaux s'agrandissent.

Pour pallier à toutes ces difficultés d'analyse des RFA fermés que ce soit l'analyse analytique ou algorithmique, nous envisageons dans le chapitre suivant de proposer une approche formelle de modélisation, d'analyse et d'évaluation de performances des RFA fermés basée sur les réseaux de Petri Stochastiques Généralisés (RdPSG). La proposition de cette approche est due à l'efficacité et la facilité d'usage des RdP tel qu'il a été explicité dans le chapitre précédent.

Algorithme MVA

Initialisation :

$$Q_i((0 \dots 0)) = 0 \quad i = 1, \dots, M.$$

Pour n variant de $(0 \dots 0)$ à N ($n \neq (0, \dots, 0)$) fairePour toutes les stations $i = 1 \dots M$ de type 1, 2 ou 4

$$R_{ic}(n) = \frac{1}{\mu_{ic}}(1 + Q_i(n - 1_c)) \quad c = 1, \dots, C$$

Pour toutes les stations $i = 1 \dots M$ de type 3

$$R_{ic}(n) = \frac{1}{\mu_{ic}} \quad c = 1, \dots, C$$

$$X_c(n) = \frac{n_c}{\sum_{i=1}^M e_{ic} R_{ic}(n)} \quad c = 1, \dots, C.$$

$$X_{ic}(n) = e_{ic} X_c(n) \quad i = 1, \dots, M, \text{ et } c = 1, \dots, C.$$

$$Q_{ic}(n) = R_{ic}(n) X_{ic}(n) \quad i = 1, \dots, M, \text{ et } c = 1, \dots, C.$$

$$Q_i(n) = \sum_{c=1}^C Q_{ic}(n) \quad i = 1, \dots, M.$$

2.9 Conclusion

Au cours de ce chapitre, nous avons présenté des notions mathématiques nécessaires à la compréhension des modèles de FA et des RFA. Nous avons abordé la théorie des files d'attente classiques, puis celle des RFA, un intérêt particulier a été donné aux RFA fermés puisque le chapitre suivant fera l'objet d'une application sur l'analyse et le calcul des indices de performance du RFA fermé $[M/M/1 \rightarrow M/M/1]$.

Chapitre 3

Modélisation et Analyse d'un RFA par les RdPSG

3.1 Introduction

Dans ce chapitre, nous analysons un réseau de files d'attente (RFA) fermé de deux stations en tandem $[M/M/1 \rightarrow M/M/1]$ via les RdPSG. Les RdPSG constituent un important modèle graphique et mathématique qui est très adapté à la description des systèmes parallèles en présence des contraintes de synchronisation. Une modélisation et l'approche proposée nous donne une représentation graphique détaillée et nous permet d'obtenir les principaux indices de performances. Des résultats numériques exacts sont obtenus en utilisant le logiciel *GRIF*.

3.2 Description du réseau $[M/M/1 \longrightarrow M/M/1]$

Considérons un réseau de files d'attente à deux stations comme est illustré dans la FIGURE 3.1.

3.2.1 Description du modèle

Les clients arrivent à la File 1 selon un processus poissonnien de paramètre λ , si le serveur est libre le client est servi avec un taux μ_1 sinon il rejoint la file d'attente. Chaque client reçoit une partie de son service à la première station. A la fin du service du client dans la première station il rejoint la file d'attente de la deuxième station avec un processus

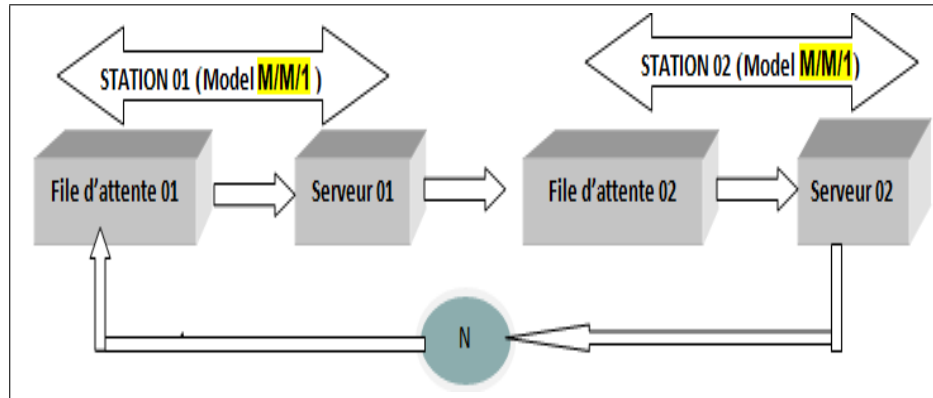
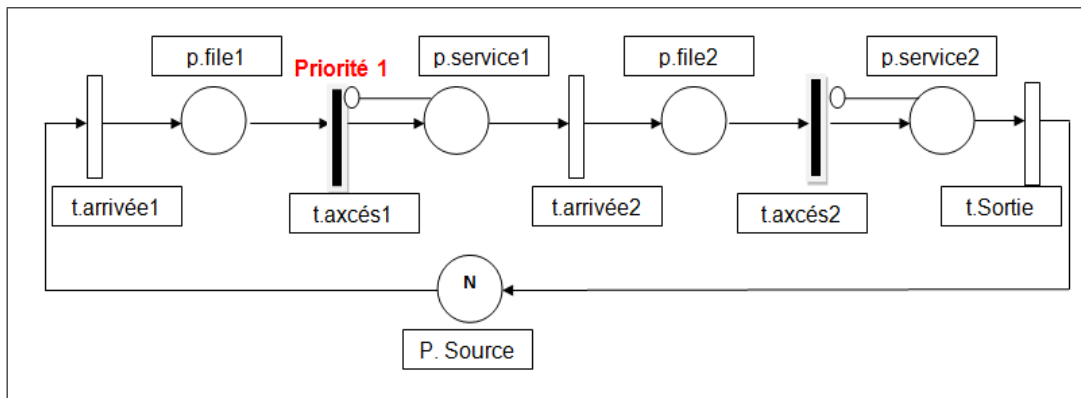


FIGURE 3.1 – Présentation d'un réseau de files d'attente à deux stations .

poissonnien de paramètre λ . Si le serveur de la File 2 est libre, ce dernier sera servi sinon il patiente jusqu'à ce que le serveur 2 se libère.

3.2.2 Modélisation du système $[M/M/1 \rightarrow M/M/1]$ via les RdPSG

Dans cette section nous modélisons le système $[M/M/1 \rightarrow M/M/1]$ par les *RdPSG* voir la figure 3.2 :

FIGURE 3.2 – Modélisation du réseau $[M/M/1 \rightarrow M/M/1]$ via les RdPSG.

- $p.file1$: Représente la file d'attente 01 (sera abrégées en graphe de marquage par $PF1$);
- $p.file2$: Représente la file d'attente 02 (sera abrégées en graphe de marquage par $PF2$);
- $p.service1$: Représente le service du client émanant la file d'attente 01 (sera abrégées en graphe de marquage par $PS1$);

- p.service2 : Représente le service du client émanant la file d'attente 02 (sera abrégées en graphe de marquage par PS2) ;
- t.arrivée1 : Représente la transition temporisée dans la file d'attente 01 (sera abrégées en graphe de marquage par t.arr1) ;
- t.arrivée2 : Représente la transition temporisée dans la file d'attente 02 (sera abrégées en graphe de marquage par *t.arr2*) ;
- t.axes1 : Représente la transition immédiates indiquant le début de service du client émanant de la file d'attente 01 (sera abrégées en graphe de marquage par *t.axes1*) ;
- t.axes2 : Représente la transition immédiates indiquant le début de service du client émanant de la file d'attente 02 (sera abrégées en graphe de marquage par *t.axes2*) ;
- P. Source : Contient les clients de la source (sera abrégées en graphe de marquage par PS) ;
- t.Sortie : Représente la transition temporisée indiquant la fin de service du client dans la station 2 .

Pour l'accès des clients qui se trouvent dans la place P. Source la transition temporisée t.arrivée1 doit être franchie ,donc a son franchissement un jeton est retiré dans la place P. Source et puis déposé dans la place p.file1, suite a la non disponibilité d'un jeton au niveau de la place P.service1 qui est contrôlé par l'arc inhibiteur ; la transition immédiate t.axes1 peut-être franchie .Au franchissement de cette dernière un jeton est retiré de la place p.file1 et puis déposé dans la place p.service1.

Pour l'accès à la file 2, la transition temporisée t .arrivée2 doit être franchie .Au franchissement de la transition t .arrivée2 un jeton est retiré de la place p.service1 et puis déposé dans la place p.file2 ,et du fait qu'aussi a la non disponibilité de jeton dans la place p.service2 qui est aussi contrôlé par l'arc inhibiteur la transition immédiate t.service2 est franchie et a son franchissement un jeton est retiré de la place P.file2 est puis sera déposé dans la place P.service2 et a la fin ,la transition temporisée t.Sortie sera franchie avec le retrait d'un jeton depuis la place p.service2 et puis le déposé dans la place P.source ainsi que le processus sera répété par le même déroulement.

Remarque 3.1. Lorsque les transitions immédiates (t.axes1 et t.axes2) sont sensibilisées en même temps on tir la transition (t.axes1).

3.2.3 Analyse des performances de notre réseau

Les indices de performances de notre modèle sont donnés somme suit :

- **Le taux moyen effectif des arrivées des clients a la file 1 :**

$$\bar{\lambda} = \sum_{j \in (SM_j)} \lambda(M_j) \pi_j; \quad (3.1)$$

avec :

- SM_j est l'ensemble des marquages ou la transition t.arrivée1 est franchissable;
- $\lambda(M_j)$ est le taux de franchissement associe à la transition t.arrivée1 dans M_j .

- **Taux moyen effectif des arrivées des clients a la file 2 :**

$$\bar{\mu}_1 = \sum_{j \in (SM_j)} \mu_1(M_j) \pi_j; \quad (3.2)$$

avec :

- SM_j est l'ensemble des marquages ou la transition t.arrivée2 est franchissable;
- $\mu_1(M_j)$ est le taux de franchissement associe à la transition t.arrivée2 dans M_j .

- **Nombre moyen de clients dans la file 1 :**

$$\bar{\eta}_{\text{file1}} = \sum_j M_j (P_{\text{file1}}) \pi_j; \quad (3.3)$$

avec :

- $M_j (P_{\text{file1}})$ est le nombre de jetons dans la place P_{file1} dans le marquage;
- M_j La somme dans cette formule se fait sur tous les marquages accessibles.

- **Nombre moyen de clients dans la file 2 :**

$$\bar{\eta}_{\text{file2}} = \sum_j M_j (P_{\text{file2}}) \pi_j; \quad (3.4)$$

avec :

- $M_j (P_{\text{file2}})$ est le nombre de jetons dans la place P_{file2} dans le marquage M_j . La somme dans cette formule se fait sur tous les marquages accessibles.

- **Nombre moyen de clients dans le service 2 :**

$$\bar{\eta}_{\text{service2}} = \sum_j M_j (P_{\text{service2}}) \pi_j; \quad (3.5)$$

avec :

- $M_j (P_{\text{service2}})$ est le nombre de jetons dans la place P_{service2} dans le marquage M_j . La somme dans cette formule se fait sur tous les marquages accessibles.

- **Temps moyen d'attente dans la file 1 :**

On application de la formule de little on aura le résultat suivant :

$$\bar{Q}_{\text{file1}} = \frac{\bar{\eta}_{\text{file1}}}{\bar{\lambda}}; \quad (3.6)$$

- **Temps moyen d'attente dans la file 2 :**

On application de la formule de little on aura le résultat suivant :

$$\bar{Q}_{\text{service1}} = \bar{Q}_{\text{file2}} = \frac{\bar{\eta}_{\text{file2}}}{\bar{\mu}_1}. \quad (3.7)$$

- **Nombre moyen de clients dans le système :**

$$\bar{\eta}_{\text{système}} = \bar{\eta}_{\text{file1}} + \bar{\eta}_{\text{file2}} + \bar{\eta}_{\text{service1}} + \bar{\eta}_{\text{service2}}. \quad (3.8)$$

- **Temps moyen d'attente dans le service 2 :**

On application de la formule de little on aura le résultat suivant :

$$\bar{Q}_{\text{service2}} = \frac{\bar{\eta}_{\text{service2}}}{\bar{\mu}_2} \quad (3.9)$$

- **Nombre moyen d'attente dans le système :**

$$\bar{Q}_{\text{système}} = \bar{Q}_{\text{file1}} + \bar{Q}_{\text{file2}} + \bar{Q}_{\text{service1}} + \bar{Q}_{\text{service2}}. \quad (3.10)$$

3.2.4 Application sur le réseau $[M/M/1 \rightarrow M/M/1]$ avec $N=3$

Le graphe des marquages associé pour $N = 3$ est illustré par la FIGURE 3.3, le marquage de notre modèle est comme suit :

$$M_i = (M(PS), M(PF1), M(PS1), M(PF2), M(PS2)); M_0 = (3, 0, 0, 0, 0).$$

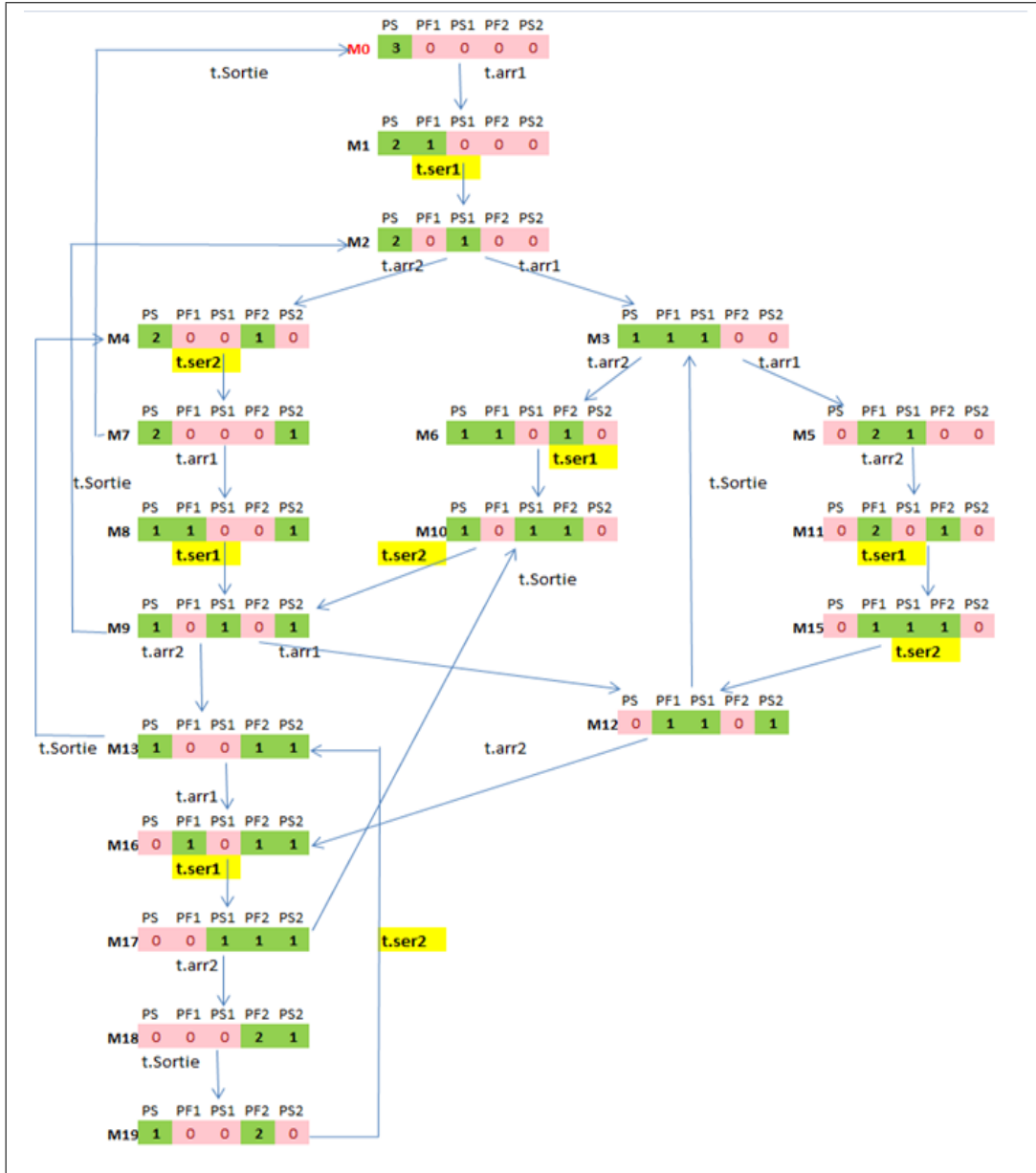


FIGURE 3.3 – Graphe des marquages du réseau $[M/M/1 \rightarrow M/M/1]$ fermé avec $n=3$.

D'après le graphe des marquages FIGURE 3.3 , on remarque que notre modèle est borné(3 -borné), vivant, sans blocage ainsi que le marquage initial est un état d'accueil donc il est ergodique. Par conséquent la distribution stationnaire π existe et elle est unique. A partir du graphe des marquages, on peut construire la chaîne de Markov continue associée à notre modèle. Pour la construction de la chaîne de Markov il faudrait bien déterminer les marquages évanescents (au moins une transition immédiate est sensibilisée) et les marquages tangibles (les transitions temporisées sont sensibilisées) . On remarque que les marquages $M1$, $M4$, $M6$, $M10$, $M8$, $M11$, $M15$, $M16$, $M19$ seront fusionnés respectivement avec leurs successeurs qui sont $M2$, $M7$, $M10$, $M9$, $M9$, $M15$, $M12$, $M17$, $M13$.

Les marquages évanescents sont $M1, M4, M6, M10, M8, M11, M15, M16, M19$.

La chaîne de Markov est construite à partir des marquages tangibles. Voir FIGURE 3.4

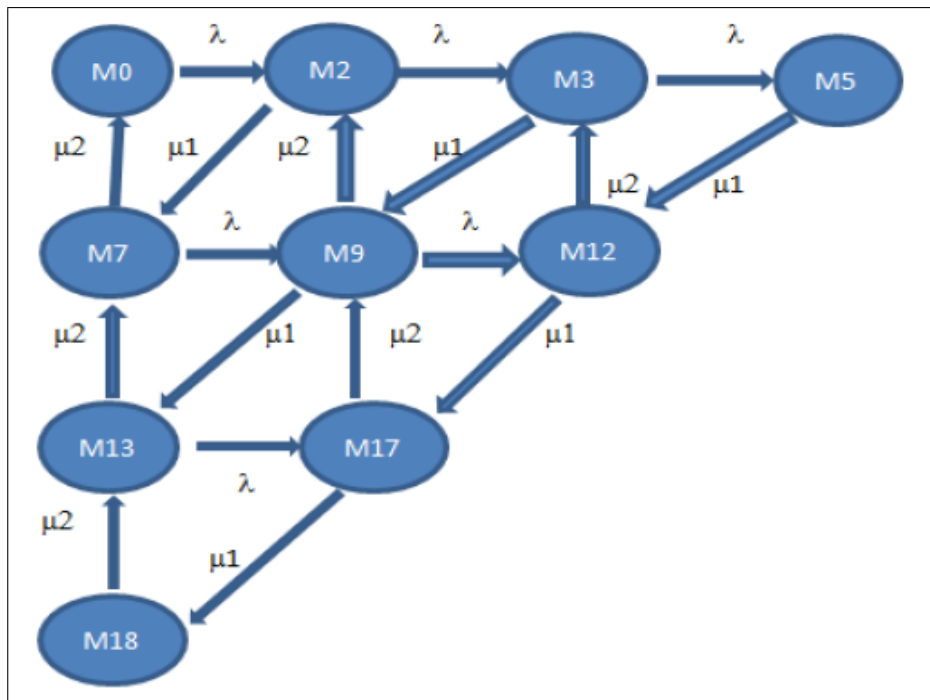


FIGURE 3.4 – La chaîne de Markov associée du réseau fermé $[M/M/1 \rightarrow M/M/1]$ pour $N=3$

Le générateur infinitésimal "Q" associée à cette chaine de Markov est donné dans la matrice suivante :

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -(\lambda + \mu_1) & \lambda & 0 & \mu_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -(\lambda + \mu_1) & \lambda & 0 & \mu_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\mu_1 & 0 & 0 & \mu_1 & 0 & 0 & 0 \\ \mu_2 & 0 & 0 & 0 & -(\lambda + \mu_2) & \lambda & 0 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 & 0 & -(\lambda + \mu_1 + \mu_2) & \lambda & \mu_1 & 0 & 0 \\ 0 & 0 & \mu_2 & 0 & 0 & 0 & (\mu_1 + \mu_2) & 0 & \mu_1 & 0 \\ 0 & 0 & 0 & 0 & \mu_2 & 0 & 0 & -(\lambda + \mu_2) & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu_2 & 0 & 0 & (\mu_1 + \mu_2) & \mu_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_2 & 0 & -\mu_2 \end{bmatrix}$$

Cette matrice est carrée de dimension (10×10) , le vecteur des probabilités stationnaires noté $\pi = (\pi_0, \pi_2, \pi_3, \pi_5, \pi_7, \pi_9, \pi_{12}, \pi_{13}, \pi_{17}, \pi_{18})$ est la solution du système :

$$\begin{cases} \pi * Q = 0 \\ \sum_{i=0}^n \pi_i = 1 \end{cases}$$

Donc le système à résoudre est définie ci-dessous :

$$\begin{cases} -\lambda\pi_0 + \mu_2\pi_7 & = 0 \\ \lambda\pi_0 - (\lambda + \mu_1)\pi_2 + \mu_2\pi_9 & = 0 \\ \lambda\pi_2 - (\lambda + \mu_1)\pi_3 + \mu_2\pi_{12} & = 0 \\ \lambda\pi_3 - \mu_1\pi_5 & = 0 \\ \mu_1\pi_2 - (\lambda + \mu_2)\pi_7 + \mu_2\pi_{13} & = 0 \\ \mu_1\pi_3 + \lambda\pi_7 - (\lambda + \mu_1 + \mu_2)\pi_9 + \mu_2\pi_{17} & = 0 \\ \mu_1\pi_5 + \lambda\pi_9 + (\mu_2 + \mu_1)\pi_{12} & = 0 \\ \mu_1\pi_9 - (\lambda + \mu_2)\pi_{13} + \mu_2\pi_{18} & = 0 \\ \mu_1\pi_{12} + \lambda\pi_{13} + (\mu_1 + \mu_2)\pi_{17} & = 0 \\ \mu_1\pi_{17} - \mu_2\pi_{18} & = 0 \\ \pi_0 + \pi_2 + \pi_3 + \pi_5 + \pi_7 + \pi_9 + \pi_{12} + \pi_{13} + \pi_{17} + \pi_{18} & = 1 \end{cases}$$

► **Exemple d'illustration** : Dans cette partie ,nous obtenant les indices de performances en remplaçant $\lambda = 2, \mu_1 = 7, \mu_2 = 6$.

En effet, nous avons calculé la matrice Q .

$$Q = \begin{bmatrix} -2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -9 & 2 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -9 & 2 & 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -7 & 0 & 0 & 7 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & -8 & 2 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & -15 & 2 & 7 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & -13 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & -8 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & -13 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & -6 \end{bmatrix}$$

On utilisant le logiciel Matlab on a obtenu les résultats suivant :

π_i	π_0	π_2	π_3	π_5	π_7	π_9	π_{12}	π_{13}	π_{17}	π_{18}
Valeurs	0.2468	0.1001	0.0726	0.1208	0.0823	0.0678	0.0755	0.1097	0.0575	0.0671

Après avoir calculés les π , plusieurs d'indices de performances peuvent être calculés tels que :

- **Le taux moyen effectif des arrivées des clients à la file 1 :**

$$\begin{aligned} \bar{\lambda} &= \lambda (\pi_0 + \pi_2 + \pi_3 + \pi_7 + \pi_9 + \pi_{13}) \\ &= 2 * (0,2468 + 0,1001 + 0,0726 + 0,0823 + 0,0678 + 0,1097) \\ &= 1,3586. \end{aligned}$$

- **Taux moyen effectif des arrivées des clients à la file 2 :**

$$\begin{aligned} \bar{\mu}_1 &= \mu_1 (\pi_2 + \pi_3 + \pi_5 + \pi_9 + \pi_{17}) \\ &= 7 * (0,1001 + 0,0726 + 0,1208 + 0,0678 + 0,0575) \\ &= 2,9316 \end{aligned}$$

- **Taux moyen effectif des arrivées des clients à la source :**

$$\begin{aligned} \bar{\mu}_2 &= \mu_2 (\pi_7 + \pi_9 + \pi_{12} + \pi_{13} + \pi_{17} + \pi_{18}) \\ &= 6 * (0,0823 + 0,0678 + 0,0755 + 0,1097 + 0,0575 + 0,0671) \\ &= 2,7594 \end{aligned}$$

- **Nombre moyen de clients dans la file 1 :**

$$\begin{aligned} \bar{n}_{\text{file1}} &= (\pi_0 + \pi_2 + \pi_3 + \pi_7 + \pi_{12} + \pi_{13}) \\ &= (0,2468 + 0,1001 + 0,0726 + 0,0823 + 0,0755 + 0,1097) \\ &= 0,6870. \end{aligned}$$

- Nombre moyen de clients dans la file 2 :

$$\begin{aligned}\bar{\eta}_{\text{file2}} &= (\pi_2 + \pi_3 + \pi_5 + \pi_9 + \pi_{12} + \pi_{17}) \\ &= (0,1001 + 0,0726 + 0,1208 + 0,0678 + 0,0755 + 0,0575) | \\ &= 0,4943;\end{aligned}$$

Ça nous donne les mêmes résultats pour le nombre de clients dans le service1

$$\bar{\eta}_{\text{service1}} = 0,4943.$$

- Nombre moyen de clients dans le service 2 :

$$\begin{aligned}\bar{\eta}_{\text{service2}} &= (\pi_7 + \pi_9 + \pi_{12} + \pi_{13} + \pi_{18}) \\ &= (0,0823 + 0,0678 + 0,0755 + 0,1097 + 0,0671) \\ &= 0,4024\end{aligned}$$

- Temps moyen d'attente dans la file 1 :

On application de la formule de little on aura le résultat suivant :

$$\begin{aligned}\bar{Q}_{\text{file1}} &= \frac{\bar{\eta}_{\text{file1}}}{\bar{\lambda}} \\ &= \frac{0,6870}{1,3586} = 0,5056.\end{aligned}$$

- Temps moyen d'attente dans la file 2 :

On application de la formule de little on aura le résultat suivant :

$$\begin{aligned}\bar{Q}_{\text{service1}} &= \bar{Q}_{\text{file2}} = \frac{\bar{\eta}_{\text{file2}}}{\bar{\mu}_1} \\ &= \frac{0,4943}{2,9316} = 0,1686.\end{aligned}$$

- Nombre moyen de clients dans le système :

$$\begin{aligned}\bar{\eta}_{\text{système}} &= \bar{\eta}_{\text{file1}} + \bar{\eta}_{\text{file2}} + \bar{\eta}_{\text{service1}} + \bar{\eta}_{\text{service2}} \\ &= 0,6870 + 0,4943 + 0,4943 + 0,4024 \\ &= 2,078.\end{aligned}$$

- Temps moyen d'attente dans le service 2 :

On application de la formule de little on aura le résultat suivant :

$$\begin{aligned}\bar{Q}_{\text{service2}} &= \frac{\bar{\eta}_{\text{service2}}}{\bar{\mu}_2} \\ &= \frac{0,4024}{2,7594} = 0,1458.\end{aligned}$$

- Nombre moyen d'attente dans le système :

$$\begin{aligned}\bar{Q}_{\text{systeme}} &= \bar{Q}_{\text{file1}} + \bar{Q}_{\text{file2}} + \bar{Q}_{\text{service 1}} + \bar{Q}_{\text{service2}} \\ &= 0,5056 + 0,1686 + 0,1686 + 0,1458 \\ &= 0,9886\end{aligned}$$

3.3 Application numérique sous le logiciel GRIF

Dans cette section nous allons évaluer les principaux indices de performances pour N quelconque et pour plusieurs valeurs de λ, μ_1, μ_2 . Pour cela nous avons fait appel au logiciel GRIF, voir LA FIGURE 3.5.

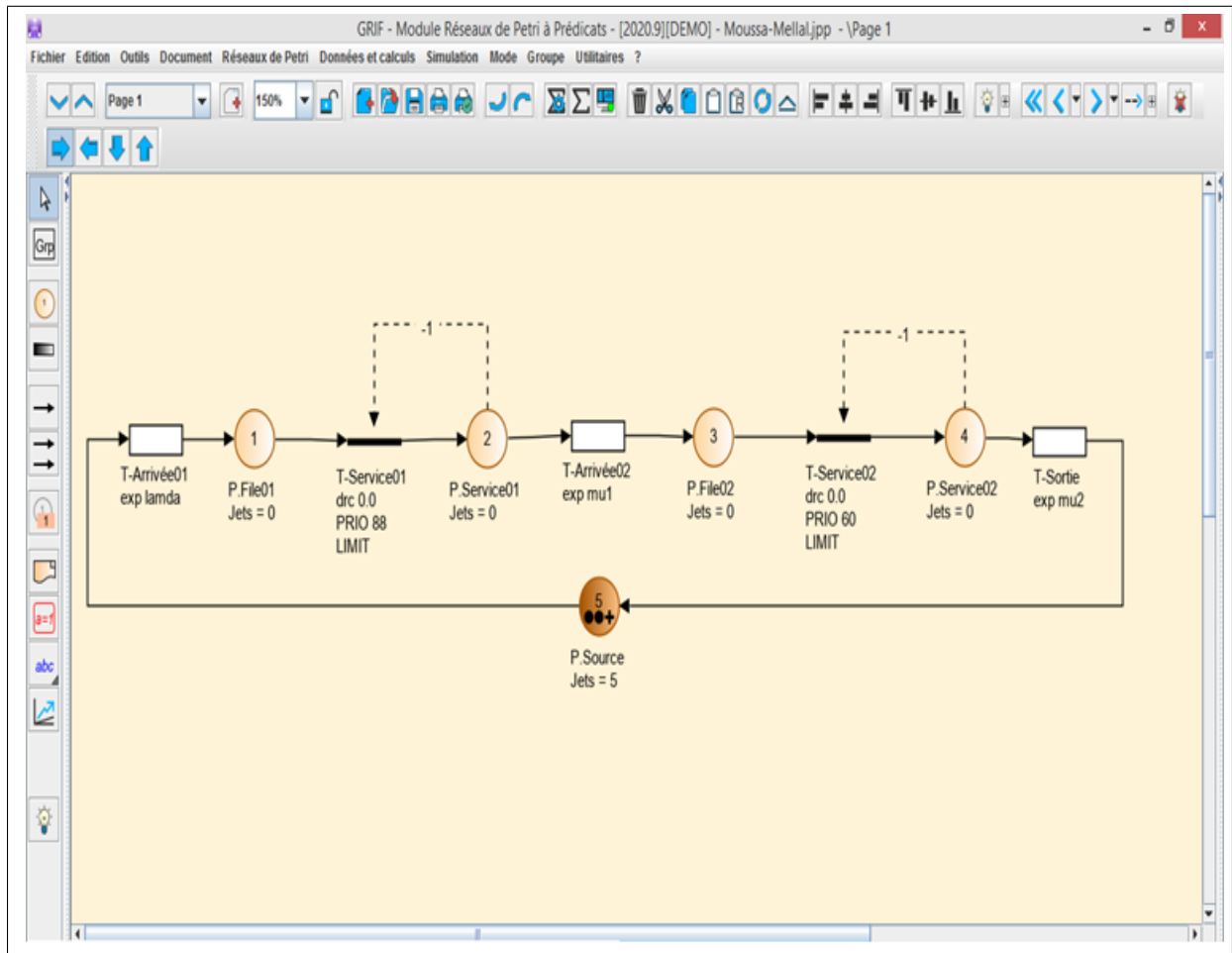


FIGURE 3.5 – Le RDPSG modélisant le réseau ($M/M/1 \rightarrow M/M/1$) sous le logiciel GRIF.

► **Influence de N sur les principaux indices de performances :**

Dans cette partie nous avons étudié l'influence de la taille de la source N sur les indices de performances pour $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 5$, les résultats sont illustrés dans la TABLE 3.1 et la TABLE 3.2 :

Variation de N	Temps d'attente dans la P.File1	Temps d'attente dans la P.Service1	Temps d'attente dans la P.File2	Temps d'attente dans la P.Service2
5	3674, 52503	6195, 021785	1322, 232208	3708, 722463
10	4359, 653882	6606, 853379	1577, 417771	3966, 90000
15	4447, 787134	6658, 608594	1617, 496918	3999, 453448
20	4436, 516862	6649, 380778	1612, 177113	3995, 175994
25	4439, 680205	6650, 907022	1612, 382038	3995, 11335
30	4439, 95628	6651, 31637	1610, 521752	3993, 591474
35	4439, 95628	6651, 31637	1610, 521752	3993, 591474
40	4439, 95628	6651, 31637	1610, 521752	3993, 591474
45	4439, 95628	6651, 31637	1610, 521752	3993, 591474
50	4439, 95628	6651, 31637	1610, 521752	3993, 591474
55	4439, 95628	6651, 31637	1610, 521752	3993, 591474
60	4439, 95628	6651, 31637	1610, 521752	3993, 591474
65	4439, 95628	6651, 31637	1610, 521752	3993, 591474
70	4439, 95628	6651, 31637	1610, 521752	3993, 591474
75	4439, 95628	6651, 31637	1610, 521752	3993, 591474
80	4439, 95628	6651, 31637	1610, 521752	3993, 591474
85	4439, 95628	6651, 31637	1610, 521752	3993, 591474
90	4439, 95628	6651, 31637	1610, 521752	3993, 591474
95	4439, 95628	6651, 31637	1610, 521752	3993, 591474
100	4439, 95628	6651, 31637	1610, 521752	3993, 591474

TABLE 3.1 – L'influence de N sur le temps d'attente.

Variation de N	Nombre moyen de clients dans la p.file1	Nombre moyen de clients dans la p.service1	Nombre moyen de clients dans la p.file2	Nombre moyen de clients dans la p.service2
5	0,6966759492	0,6195021785	0,1911762196	0,3708722463
10	1,1700638883	0,6606853379	0,2609215440	0,3966900000
15	1,3135108755	0,665860859	0,275698161	0,399945345
20	1,3099070416	0,664938078	0,272463932	0,399517599
25	1,3157577128	0,665090702	0,272804776	0,399511335
30	1,3182015221	0,665131637	0,272053141	0,399359147
35	1,3182015221	0,665131637	0,272053141	0,399359147
40	1,3182015221	0,665131637	0,272053141	0,399359147
45	1,3182015221	0,665131637	0,272053141	0,399359147
50	1,3182015221	0,665131637	0,272053141	0,399359147
55	1,3182015221	0,665131637	0,272053141	0,399359147
60	1,3182015221	0,665131637	0,272053141	0,399359147
65	1,3182015221	0,665131637	0,272053141	0,399359147
70	1,3182015221	0,665131637	0,272053141	0,399359147
75	1,3182015221	0,665131637	0,272053141	0,399359147
80	1,3182015221	0,665131637	0,272053141	0,399359147
85	1,3182015221	0,665131637	0,272053141	0,399359147
90	1,3182015221	0,665131637	0,272053141	0,399359147
95	1,3182015221	0,665131637	0,272053141	0,399359147
100	1,3182015221	0,665131637	0,272053141	0,399359147

TABLE 3.2 – L'influence de N sur le nombre moyen de clients.

Leurs graphes associés respectivement sont :

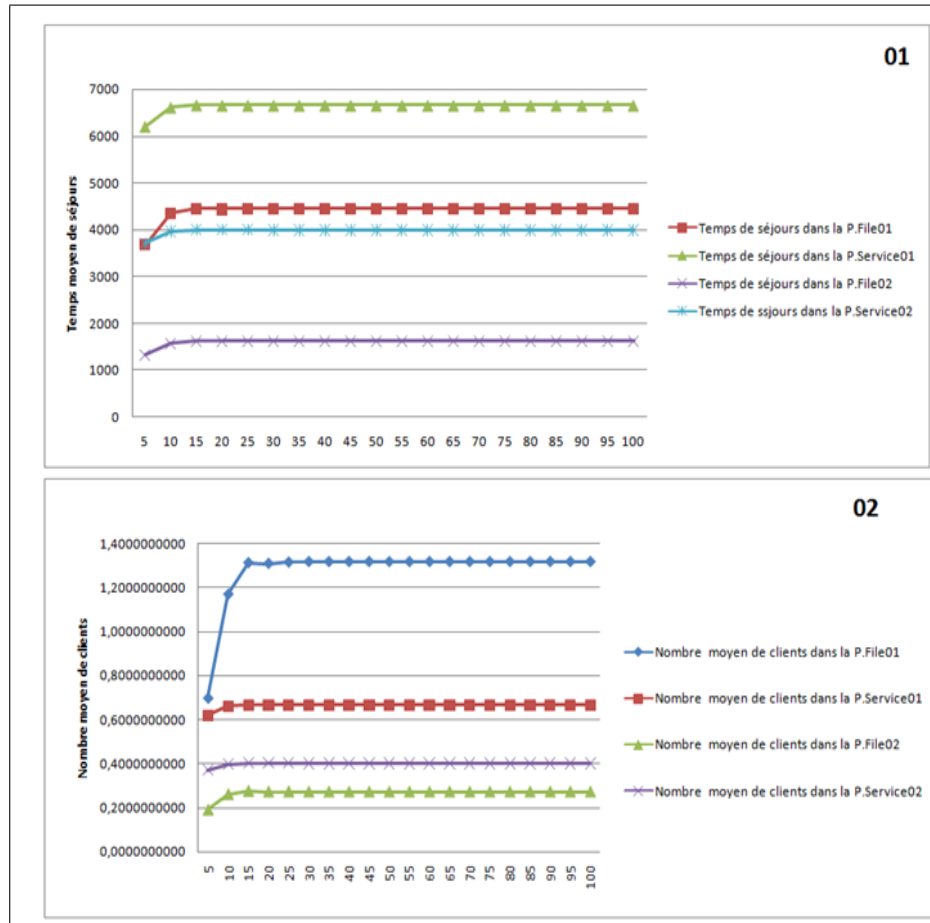


FIGURE 3.6 – L'influence de N sur le nombre moyen de clients et le temps d'attente.

► **Interprétation des résultats :**

Dans la FIGURE 3.6(01), on remarque que le nombre de clients dans la première station augmente plus rapidement jusqu'à $N = 30$ contrairement à la deuxième station où le nombre de clients augmente de manière progressive. De même pour le temps de séjour dans la première station augmente. Ceci est expliqué par le fait que la transition $t.service1$ est prioritaire par rapport à la transition $t.service2$.

► **Influence de ρ_1 sur les principaux indices de performances :**

Dans cette partie nous avons étudié l'influence de la charge $\rho_1 = \lambda/\mu_1$ sur les indices de performances pour $\lambda = 1$, $N = 5$ et $\mu_2 = 5$, les résultats sont illustrés dans la TABLE 3.3 et la TABLE 3.4 :

La charge ρ_1	Temps d'attente dans la p.file1	Temps d'attente dans la p.service1	Temps de d'attente dans la p.file2	Temps d'attente dans la p.service2
0,03333333	43,19531418	660,0722627	1553,876169	3959,921033
0,04166667	66,9777166	823,4188691	1550,566512	3957,128589
0,05555556	119,8567665	1096,607529	1551,079443	3956,525676
0,06666667	173,4793306	1317,764089	1558,499231	3963,402312
0,08333333	267,5720683	1646,847679	1553,96174	3964,457863
0,16666667	1056,427172	3278,943886	1517,969226	3931,339933
0,33333333	3674,52503	6195,021785	1322,232208	3708,722463
0,55555556	6932,136574	8532,267666	891,3811302	3061,920155
0,58823529	7265,378217	8733,500319	836,422393	2960,223164
0,66666667	7980,414807	9123,353807	709,8377111	2727,369641
0,71428571	8284,190365	9286,398406	644,2482166	2581,312941
0,83333333	8882,277944	9577,008086	506,8513939	2290,826085
0,9009009	9116,345692	9682,802735	441,1037238	2144,97813
0,95238095	9238,356737	9739,465451	406,8482747	2050,688336
0,999001	9334,655602	9778,680218	377,757591	1976,375665

TABLE 3.3 – Temps d'attente en fonction de ρ_1 .

La charge ρ_1	Nombre moyen de clients dans la P.File01	Nombre moyen de clients dans la P.Service01	Nombre moyen de clients dans la P.File02	Nombre moyen de clients dans la P.Service02
0,033333333	0,004612319	0,066007226	0,241388991	0,395992103
0,041666667	0,007275288	0,082341887	0,240539343	0,395712859
0,055555556	0,013367014	0,109660753	0,23973744	0,395652568
0,066666667	0,019863221	0,131776409	0,241122153	0,396340231
0,083333333	0,031747959	0,164684768	0,239921799	0,396445786
0,166666667	0,148597756	0,327894389	0,229452414	0,393133993
0,333333333	0,696675949	0,619502178	0,19117622	0,370872246
0,555555556	1,697684978	0,853226767	0,119029858	0,306192015
0,588235294	1,822718499	0,873350032	0,110332632	0,296022316
0,666666667	2,126529194	0,912335381	0,092226961	0,272736964
0,714285714	2,272973863	0,928639841	0,082995806	0,258131294
0,833333333	2,579759572	0,957700809	0,064119181	0,229082609
0,900900901	2,720900195	0,968280274	0,054688116	0,214497813
0,952380952	2,806256546	0,973946545	0,049916333	0,205068834
0,999000999	2,874342801	0,977868022	0,045970962	0,197637566

TABLE 3.4 – Nombre moyen de clients en fonction de ρ_1 .

Leurs graphes associés respectivement sont :

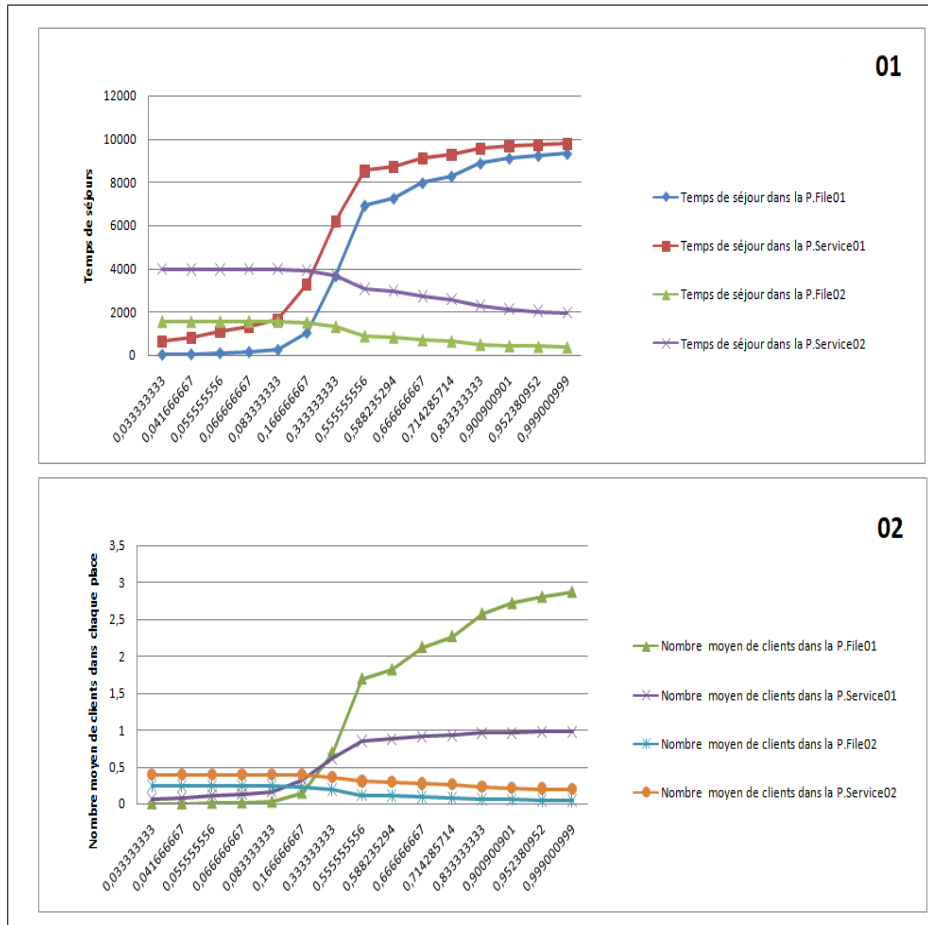


FIGURE 3.7 – L'influence de ρ_1 sur le nombre moyen de clients et le temps d'attente.

► **Interprétation des résultats :**

D'après la FIGURE 3.7(02), on remarque qu'avec l'augmentation de la charge ρ_1 de la première station, le nombre de clients augmente dans la première et diminue par rapport à la deuxième station jusqu'à ($\rho_1 > 0.33$). L'augmentation est très rapide dans la première station, ceci est dus a sa surcharge.

► **Influence de ρ_2 sur les principaux indices de performances :** Dans cette partie nous étudions l'influence de la charge ρ_2 sur les indices de performances pour $\lambda = 1$, $N = 5$ et $\mu_1 = 3$, les résultats sont illustrés dans les TABLE 3.5 et TABLE 3.6 :

La charge ρ_2	Temps de d'attente dans la p.file1	Temps d'attente dans la p.service1	Temps d'attente dans la p.file2	Temps d'attente dans la p.service2
0,033333333	3939,522804	6366,95682	39,8113173	635,8254173
0,041666667	3944,324159	6370,512804	61,7057081	793,5320079
0,055555556	3927,609601	6361,211170	109,453265	1056,557642
0,066666667	3934,432299	6362,391935	156,869706	1264,920676
0,083333333	3915,591702	6344,926239	241,625838	1575,015625
0,166666667	3744,446978	6236,809116	946,19409	3126,12647
0,333333333	3260,483094	5884,992887	3271,66003	5873,163597
0,555555556	2302,786467	4935,69322	6429,73324	8243,051568
0,588235294	2180,619326	4793,395181	6792,05888	8470,88682
0,666666667	1875,33136	4450,264873	7505,67885	8890,214194
0,714285714	1718,665587	4255,120076	7852,84745	9075,545274
0,833333333	1374,798648	3772,437034	8526,10241	9416,882868
0,900900901	1208,561917	3542,888931	8787,23577	9533,574913
0,952380952	1103,409549	3380,035303	8970,69745	9621,89398
0,999000999	1023,906442	3245,272332	9102,53629	9675,991756

TABLE 3.5 – Temps d'attente en fonction de ρ_2 .

La charge ρ_2	Nombre moyen de clients dans la p.file1	Nombre moyen de clients dans la p.service1	Nombre moyen de clients dans la p.file2	Nombre moyen de clients dans la p.service2
0,033333333	0,792976159	0,636695682	0,00422631	0,063582542
0,041666667	0,794408635	0,63705128	0,0066493	0,079353201
0,055555556	0,786910384	0,636121117	0,01212309	0,105655764
0,066666667	0,787495322	0,636239193	0,01781522	0,12649207
0,083333333	0,779551912	0,634492624	0,02830145	0,15750156
0,166666667	0,720540871	0,623680912	0,12951757	0,31261265
0,333333333	0,583077229	0,588499289	0,58451756	0,58731636
0,555555556	0,372260899	0,493569322	1,48113473	0,82430516
0,588235294	0,346923948	0,479339518	1,60292583	0,84708868
0,666666667	0,28888531	0,445026487	1,87627098	0,88902142
0,714285714	0,26255333	0,425512008	2,03008077	0,907554527
0,833333333	0,200535001	0,377243703	2,3541721	0,941688287
0,900900901	0,172654624	0,354288893	2,49702553	0,953357491
0,952380952	0,155633537	0,33800353	2,60351986	0,962189398
0,999000999	0,143943787	0,324527233	2,6898896	0,967599176

TABLE 3.6 – Nombre moyen de clients en fonction de ρ_2 .

Leurs graphes associés respectivement sont :

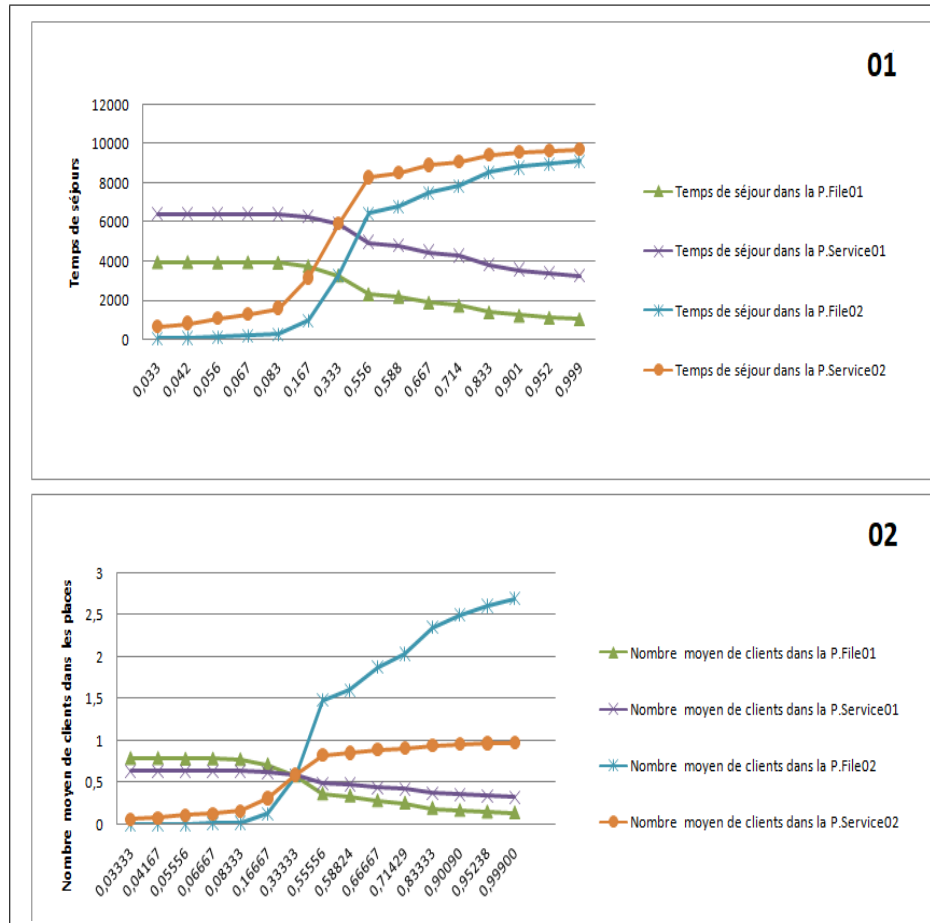


FIGURE 3.8 – L'influence de ρ_2 sur le nombre moyen de clients et le temps d'attente.

► **Interprétation des résultats :**

D'après la FIGURE 3.8(02) , on remarque qu'avec l'augmentation de la charge ρ_2 de la deuxième station, le nombre de clients augmente faiblement dans la deuxième et diminue faiblement par rapport à la première station jusqu'à l'atteinte de la valeur de la charge $\rho_2 = 0.33$. Après cette dernière on remarque que l'augmentation est très rapide dans la deuxième station surtout dans la file 2 autant que dans le service 2 ceci est dus a sa surcharge .

3.4 Conclusion

Dans ce chapitre, nous avons modélisé un réseau de files d'attente (RFA) fermé de deux stations en tandem $[M/M/1 \rightarrow M/M/1]$ via les RdPSG. Après l'obtention du modèle, nous avons construit la chaîne de Markov à partir du graphe des marquages et calculer les indices de performances en utilisant le simulateur GRIF.

Conclusion générale

La théorie analytique des modèles de RFA s'avère d'une portée limitée en raison de la complexité des résultats analytiques connus. En effet, dans la majorité des cas, on se retrouve confronté à des RFA décrits par des modèles analysés par des équations dont la résolution est complexe ou possédant des solutions qui ne sont pas facilement interprétables afin que le praticien puisse en bénéficier.

En effet, des solutions exactes existent pour une classe bien particulière de RFA connus sous le nom de réseaux séparables où à forme produit, cette classe de réseaux à forme produit est très restreinte même avec toutes les extensions faites jusque là. La difficulté d'élargissement de ces extensions provient de la difficulté de la nature des flux inter-stations de ces réseaux. Bien sûr de nombreuses méthodes approximatives d'évaluation des performances des *RFA* non à forme produit ou ayant une forme produit difficile à exploiter ont été développées, cependant la complexité algorithmique de ces méthodes se complique au fur et à mesure que le nombre de stations ou le nombre de clients dans ces réseaux s'agrandissent.

Pour pallier à toutes ces difficultés d'analyse des RFA, nous avons proposé dans ce mémoire une approche de modélisation, d'analyse et d'évaluation de performances des RFA fermés basée sur les réseaux de Petri Stochastiques Généralisés (RdPSG). La proposition de cette approche est due à l'efficacité et la facilité d'usage des RdP démontrées depuis leur introduction en 62 par C. A. Petri.

Pour la réalisation de notre approche proposée, on a dû nous intéresser dans le premier chapitre de ce mémoire aux notions générales des RdP, leurs définitions, leurs propriétés et leurs extensions nécessaires pour la compréhension des chapitres suivants. Dans le deuxième chapitre, on a donné des généralités sur les systèmes et réseaux de files d'attente en particuliers sur des RFA fermés.

Après ces chapitres introductifs nous sommes passés au chapitre application où on a modéliser le réseau fermé choisi pour étude à savoir le réseaux fermé $[M/M/1 \rightarrow M/M/1]$. La modélisation réalisée est basée sur le formalisme des RdPSG, ce qui nous a permis de construire la chaîne de Markov associée à partir de laquelle qui nous avons défini les

différents indices de performance. Enfin, en utilisant le simulateur GRIF des résultats numériques pour des indices de performances de ce SFA ont été obtenus et leurs interprétations sont réalisées.

Le travail exécuté dans ce mémoire montre que les RdPSG sont des formalismes très puissants en terme d'expressivité et qu'ils constituent une approche fiable pour le calcul des paramètres de performances des SFA complexes.

Perspectives du travail

Le travail réalisé dans ce mémoire et les résultats obtenus ouvrent nous permet d'envisager un ensemble de perspectives, à savoir :

- Modéliser d'autres RFA fermés markoviens ;
- Modéliser les RFA fermés non markoviens ;
- Modéliser les RFA fermés markoviens ou non markoviens où des critères tels que les rappels ou la priorité, ... interviennent.

Annexe A

Annexe

Fenêtre principale du module Réseaux de Pétri à prédicats

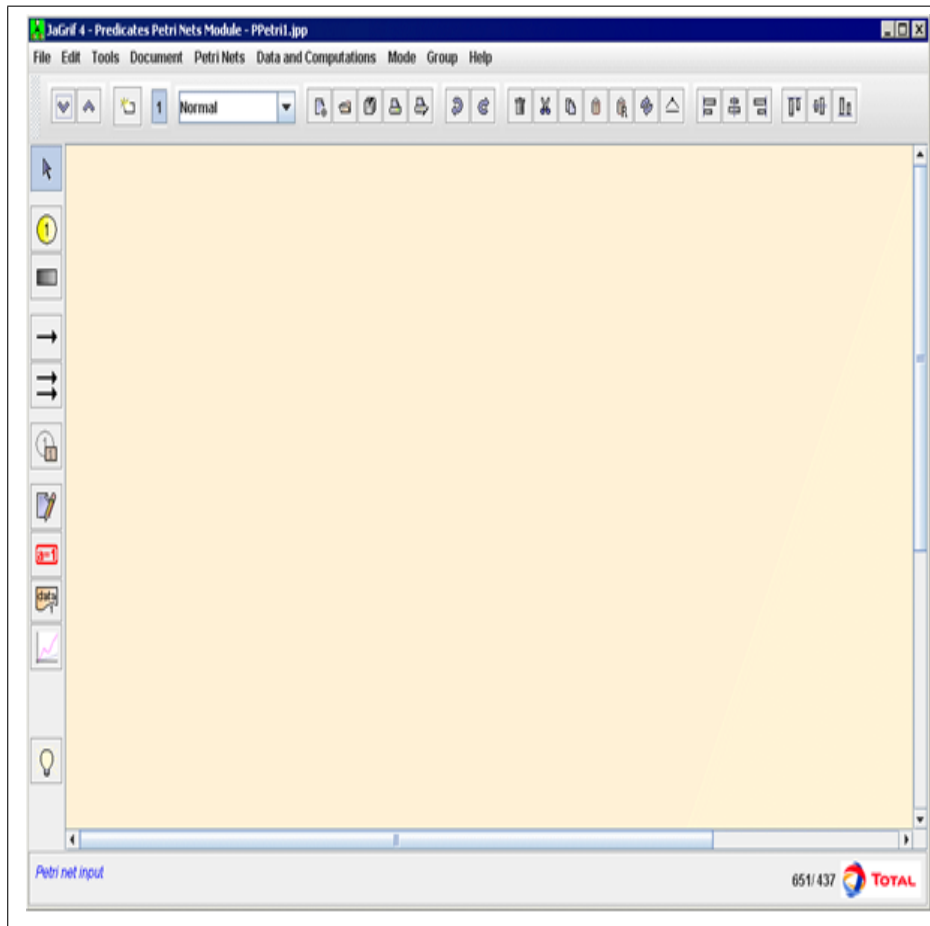
La fenêtre principale est décomposée en plusieurs parties :

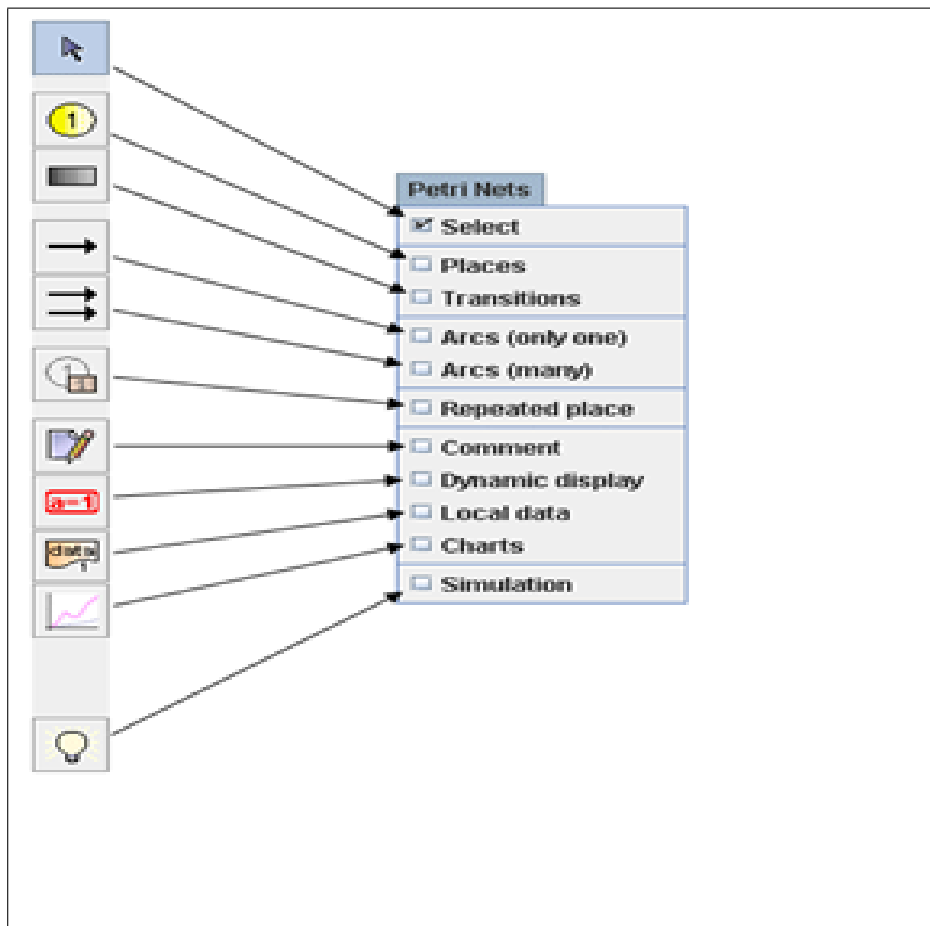
- **Barre de titre** : La barre de titre indique le nom du module et le nom du fichier en cours d'édition ;
- **Barre de menu** : La barre de menu permet d'accéder à toutes les fonctions de l'application ;
- **Barre d'icônes (raccourcis)** : La barre de raccourcis est une barre (horizontale) d'icônes permettant d'accéder plus rapidement aux fonctions usuelles.
- **Barre d'outils** : La barre d'outils (verticale) permet de sélectionner les éléments pour modéliser.
- **Zone de saisie** : Un maximum de place a été laissée à la zone de saisie graphique pour permettre de réaliser le modèle.
- **Arborescence** : L'arborescence est "cachée" entre la zone de saisie et la barre d'outil. Elle permet de naviguer dans les pages et groupes du document ;
- **Ensemble des tableaux** : Les tableaux sont regroupés dans des onglets "cachés" à droite de la zone de saisie.

Barre d'outils verticale

Chaque modèle utilisé en sûreté de fonctionnement possède sa propre iconographie. L'ensemble de symboles graphiques relatifs aux réseaux de Pétri est représenté sur la barre d'icônes placée verticalement à gauche de la fenêtre de saisie. La barre d'outils verticale comporte les éléments suivants :

- Places représentées par des cercles ;

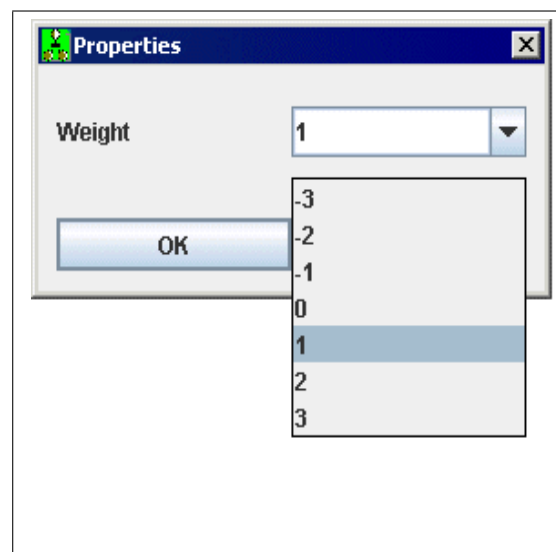




- Transitions représentées par des rectangles ;
- Arcs amont et aval représentés par des flèches ;
- Place répétée (ou Renvoi) pour réaliser des liaisons entre plusieurs parties du même modèle (sur des pages ou dans des groupes différents) ;
- Commentaire pour ajouter du texte directement sur le graphique ;
- Affichage dynamique pour afficher une valeur d'un élément du modèle. ;
- Variables locales pour créer des variables liées uniquement à une partie du modèle. ;
- Courbe pour tracer des courbes représentant des calculs sur le modèle ;
- Simulation permet de passer en mode simulation.

Paramétrage des arcs

Par défaut le Poids de tous les arcs (amont et aval) est de "1". Cependant, il est possible de le modifier. Il suffit pour cela de cliquer sur l'arc concerné avec le bouton droit de la souris afin de faire apparaître l'éditeur ci dessous. Un clic sur le petit triangle noir



fait apparaître quelques valeurs possibles sélectionnables à la souris. Pour les arcs avals les poids sont toujours positifs et correspondent au nombre de jetons qui seront ajoutés dans la place aval lors du tir de la transition correspondante. Pour les arcs amont il faut distinguer trois cas :

- Poids strictement positif : il s'agit des arcs "normaux" qui valident la transition lorsque le nombre de jetons dans la place amont est supérieur ou égal au poids de cet arc. Lors du tir de la transition, un nombre de jetons égal au poids de l'arc sera retiré de la place amont correspondante.

- Poids strictement négatif : il s'agit des arcs "inhibiteurs" qui inhibent la transition lorsque le nombre de jeton dans la place amont correspondante dépasse la valeur absolue du poids de l'arc (3 jetons par exemple pour un poids de -3). Ce type d'arc est représenté graphiquement tirée.
- Poids "0" il s'agit d'un arc qui vide la place amont correspondante lorsque la transition est tirée (ceci quel que soit le marquage de la place amont avant le tir).

Bibliographie

- [1] H. Alla. *Réseaux de Petri colorés et réseaux de Petri continus : Application à l'étude des systèmes à événements discrets*. PhD thesis, INPG Grenoble, France, 1987.
- [2] Y. Atamna. *Réseaux de Petri Temporisés Stochastiques Classiques et Bien Formés. Définition, Analyse et Application aux Systèmes Distribuée Temps Réel*. Thèse Doctorat en Informatique Industrielle, Université de Paul SABATIER, 1994.
- [3] A.Willing. *A short introduction to queueing theory*. Technical University Berlin, Telecommunication Networks Group, 1999.
- [4] B. Baynat. *Réseaux de Files d'Attente : Des chaînes de Markov aux Réseaux a Forme Produit*. Eyrolles Edition, 2000.
- [5] R. Bergoute Bekrar. *Identification des systèmes à événements discrets par réseaux de Petri*. These Doctorat, Université de Reims Champagne Ardenne, 2009.
- [6] L. Bouraine. *Support de cours sur les files d'attente*. Université A. Mira de Bejaia, 2015.
- [7] S. Bruel and G. Balbo. *Computational Algorithms for Closed Networks*. Operating and Programming Systems Series, New York, Oxford : North-Holland, 1980.
- [8] P. J. Burke. *The output of a queuning system*. Operations Research, Vol. 4 : 699-704, 1956.
- [9] J. P. Buzen. *Computational Algorithms for Closed Queueing Networks with Exponential Servers*. Comm. of the ACM, 16(9) :527–531, 1973.
- [10] S. Collart-Dutilleul. *Les Réseaux de Petri P-temporels. Modélisation et validation d'exigences temporelles*. Mémoire d'habilitation à diriger des recherches en automatique et informatique, Université des Sciences et Technologies de Lille, 2008.
- [11] Dad. Djouhra. *Modelisation et simulation du flex dans un reseau pour la regulation du trafic*. Magister Dissertation. Universite Ahmed Ben Bella d'Oran 1 Es Senia, Algeria, 2011.

- [12] A.K. Erlang. *Probability and telephone calls*. Nyt. Tidsskr Krarup Mat. Ser, 20 : 33-39,.
- [13] A.K. Erlang. *Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges*. Elektroteknikeren, 13 : 5-13, 1917.
- [14] B. Filipowicz and J. Kwicien. *Queueing systems and networks. Models and applications*. Bulletin of the Polish Academy of Sciences. Technical Sciences, 56(4), 2008.
- [15] J. Marsan A. M. Balbo. G and G. Conte. *A class of Generalized Stochastic Petri Nets for the Performances of multiprocessor Systems*. In ACM TOCS, vol.2, May 1984.
- [16] A.C. Geniet. *Les réseaux de Petri. Un outil de modélisation*. Springer-Verlag Berlin. Edition 2, 2006.
- [17] W. J. Gordon and F. Newell. *Closed Queueing Systems with Exponential Servers*. Operational Research, 15 : 254-265, 1967.
- [18] Dao Thi Thu Ha. *Les files et les réseaux zero-automatiques. Réseaux et télécommunications*. Thèse de doctorat. Université Paris-Diderot - Paris VII, 137 pages, 2007.
- [19] S. Hakmi. *Evaluation des performances des systèmes prioritaires à l'aide des Réseaux de Petri Stochastiques Généralisées*. Mémoire magistère en Recherche Opérationnelle, Université A/Mira de Béjaia, 2011.
- [20] A. Harbaoui. *Vers une modelisation et un dimensionnement automatiques des applications réparties*. thèse de Doctorat, Université de Grenoble, 2011.
- [21] J. R. Jackson. *Networks of Waiting Lines*. Operational Research, 5, 518-521, 1957.
- [22] T. M. Liebling, De Werra. D, Heche, and J-F. Recherche. *Opérationnelle pour ingénieurs*. Presses Polytechniques et Universitaires Romandes, Tome 2, 2003.
- [23] T. M. Liebling, De Werra. D, and J-F. Heche. *Recherche Opérationnelle pour ingénieurs*. Presses Polytechniques et Universitaires Romandes, Tome 2, 2003.
- [24] J. D. C. Little. *A Proof for the Queueing Formula $l = \lambda w$* . Oper. Res., 9 :383-387, 1961.
- [25] M. Magnin. *Réseaux de Petri à chronomètres. Temps dense et temps discret*. Thèse de Doctorat, Automatique et Informatique Appliquée, Université de NANTES, 2007.
- [26] E. Sanchis P. Azema, G. Juanole and M. Mombemard. *Specification and Veriication of Disrribwed Syslems Using PROLOG Interpreted Petri Nets*. in Proc. Coof. '84 IEEE Software Engineering, pp.510-518, 1984.

-
- [27] E. Sanchis P. Azema, G. Juanole and M. Mombemard. *Specification and Veriication of Disrribwed Syslems Using PROLOG Interpreted Petri Nets*. in Proc. Coof. ' 84 IEEE Software Engineering, 510-518, 1984.
- [28] J. Haas. Peter. *Stochastic Petri Nets : Modelling, Stability,Simulation*. Springer Series in Operations Research, Springer-Verlag New York, 2002.
- [29] C. A. Petri. *Kommunikation mit automaten*. PhD thesis, University of Bonn, 13 : 5-13, 1962.
- [30] N. Philippe. *Basic elements of queueing theory : application to medeling of computer systems*. Tech report, The University of Massachusetts, 2004.
- [31] G ; Pujolle and S. Fdida. *Modèle de système et de réseaux*. Eyrolles, 1989.
- [32] M. Reizer. *Mean Value Analysis of Queueing Networks, a New Look at an Old Problem*. In M. Arato, A. Butrimenko et E. Gelenbe (Eds), Performance of Computer Systems : Prod. of the 4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems, North Holland. Amsterdam. p.636, 1980.
- [33] M. Reizer. *Mean Value Analysis and Convolution Method for Queue Dependent Servers in Closed Networks*. Performance Evaluation, 1 :718, 1981.
- [34] M. Reizer and S. S. Lavenberg. *Mean Value Analysis of Closed Multichain Queueing Networks*. J.A.C.M., 27 :313–323, 1980.
- [35] V. Robert. *Les Réseaux de Petri*. Cours sur les réseaux de petri, LAAS-CNRS Toulouse, 2002.
- [36] P. Rongier. *Développement et évaluation de méthodes multi-agents pour les applications aux véhicules robotisés*. Thèse de Doctorat en Génie Informatique, Université de MONTPELLIER II, 2007.
- [37] R. Rugg. *Processus stochastique*. Presses Polytechniques Romandes, 1989.
- [38] J. S. Simonoff. *Smoothing Methods in Statistics*. Springer Series in Statistics. Springer-Verlag, New York, 1996.
- [39] W. J. Stewart. *Probability Markov Chains Queues and Simulation*. Published by Princeton University Press, United Kingdom : Princeton University Press, 6 Oxford Street, 777 pages, 2009.
- [40] J. Wang. *Petri Nets for Dynamic Event-Driven System Modeling*. Department of Software Engineering Monmouth University, pp. 1 – 17, 2006.

Résumé

Dans ce mémoire, nous montrons comment utiliser les *RdP* pour modéliser et évaluer les performances d'un réseau de Jackson fermé $[M/M/1 \rightarrow M/M/1]$. Pour les réseaux de Jackson fermés il y'a des résultats analytiques exactes, mais ils sont en fonction du nombre de stations que contient le réseau à étudier et du nombre de clients qui peuvent circuler dans ces stations donc ces résultats se compliquent en fur et à mesure que le nombre des stations et/ou le nombre de clients dans ces stations augmente. Ainsi, après une modélisation appropriée en utilisant le formalisme des *RdPSG* (Réseaux de Petri Stochastiques Généralisés) qui s'adapte à la structure de ces réseaux de files d'attente, nous avons pu construire le graphe de marquage du modèle qui nous a permis de construire la chaîne de Markov qui lui est associée. A partir de cette chaîne de Markov la distribution stationnaire est calculée et elle nous a permis d'obtenir les indices de performances. Ces indices de performance nous les avons obtenus en parallèle via le simulateur *GRIF*.

Mots clés : Réseaux de files d'attente, Réseaux de Jackson fermé, Réseaux de Gordon Noel, Réseaux de Petri généralisés, Markov chain, Modélisation.

Abstract

In this work, we show how to use *PN* to model and evaluate the performance of a closed Jackson network $[M/M/1 \rightarrow M/M/1]$. For closed Jackson networks there are accurate analytical results, but they are a function of the number of stations in the network to be studied and the number of clients that can flow through those stations, so these results become more complicated as the number of stations and/or the number of clients in those stations increases. Thus, after an appropriate modeling using the *GSPN* formalism (Generalized Stochastic Petri Networks) that adapts to the structure of these queueing networks, we were able to build the marking graph of the model which allowed us to build the Markov chain associated with it. From this Markov chain the stationary distribution is computed and it allowed us to obtain the performance indexes. We obtained these performance indices in parallel via the *GRIF* simulator.

Keywords : Queue networks, Jackson Networks closed, Networks of Gordon Noel, Generalized Petri nets, Markov chain, Modeling.