

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

Université A. Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle



Mémoire De Magister

En Mathématiques Appliquées

Option

Modélisation Mathématique et Techniques de Décision

Thème

Evaluation des Performances des Systèmes
Prioritaires à l'aide des Réseaux de Petri
Stochastiques Généralisés (RdPSG)

Présenté par :

M^{elle} Sedda HAKMI

Devant le jury composé de :

Présidente	M ^{me}	L. Bouallouche	M. C. A	U. de Béjaïa
Rapporteur	M ^r	D. Aïssani	Professeur	U. de Béjaïa
Examineur	M ^r	S. Adjabi	M. C. A	U. de Béjaïa
Examineur	M ^r	B. Rabta	Maître de Rech	U. Amsterdam
Invitée	M ^{me}	O. Iamouchene	M. A. A	U. de Béjaïa

Béjaïa, 2011

Remerciements

En premier lieu, je remercie Dieu, le miséricordieux, sans lui rien de tout cela n'aurait pu être.

*J'*exprime mes profonds remerciements à mon promoteur, M^r D. Aïssani professeur à l'université A. Mira de Béjaïa, pour l'honneur qu'il m'a fait en assurant la direction de ce mémoire. Je le remercie pour sa patience, ses encouragements et pour l'efficacité de son aide inestimable tout au long de ce travail. Ses compétences ont été un atout indéniable à la réussite de ce travail.

Ce travail a été co-encadré par M^{me} O. Iamouchene à qui j'exprime ma gratitude, pour son soutien permanent, son aide précieuse et pour la disponibilité dont elle a toujours fait preuve à mon égard.

Je suis très reconnaissante envers M^{me} L. Bouallouche, pour l'honneur qu'elle me fait en présidant le Jury de ma soutenance, ainsi, que pour toute l'aide qu'elle m'a prodiguée.

*J'*exprime toute ma gratitude à Monsieur S. Adjabi et Monsieur B. Rabta qui ont accepté d'examiner avec attention ce travail. Je les remercie également pour toute l'aide qu'ils m'ont apportée.

Dans ces moments importants, je pense très fort à ma famille en commençant par mes chers parents pour leur soutien et leurs encouragements, ainsi qu'à mes frères et soeurs qui m'ont soutenue en toute circonstance. Merci pour la confiance qu'ils ont placée en moi.

Enfin, je remercie également tous ceux qui ont contribué de près ou de loin au parachèvement de ce travail, soit par leur savoir scientifique ou par leur amitié.

Dédicaces

Je dédie ce modeste travail :

- * *A mes chers parents pour leur soutien et leur amour ;*
- * *A mes frères et sœurs Linda, Abdou, Samira, A.Aziz et Mohammed
qui attendent avec impatience ma réussite ;*
- * *A mes neveux et nièces ;*
- * *A mon fiancé ;*
- * *A mes amis (es) ;*
- * *A tous les miens ;*

Sedda Hakmi.

Table des matières

Table des Matières	i
Table des Figures	vi
Liste des tableaux	vii
Introduction Générale	1
1 Etat de l'Art sur les réseaux de Petri	5
1.1 Introduction	5
1.2 Notions de base	6
1.2.1 Structure d'un Réseau de Petri	6
1.2.2 Réseau marqué	7
1.2.3 Graphe associé et notations matricielles	8
1.3 Interprétation d'un réseau de Petri	8
1.3.1 Evolution d'un réseau de Petri	9
1.3.2 Conflit et parallélisme	13
1.3.3 Propriétés d'un réseau de Petri	14
1.4 Extension des réseaux de Petri	16
1.4.1 Réseaux de Petri généralisés	16
1.4.2 Réseaux de Petri à arcs inhibiteurs	17
1.4.3 Réseaux de Petri colorés	18

1.4.4	Réseaux de Petri temporisés	19
1.4.5	Réseaux de Petri temporels	21
1.4.6	Réseaux de Petri stochastiques	22
1.5	Réseaux de Petri Stochastiques Généralisés	24
1.6	Conclusion	26
2	Systèmes de files d'attente	27
2.1	Introduction	27
2.2	Représentation d'un système d'attente	28
2.3	Formule de Little	30
2.4	Mesures de Performance	30
2.5	Processus stochastiques	31
2.5.1	Propriété : sans mémoire	31
2.5.2	Processus Markovien	31
2.5.3	Processus de Poisson	32
2.5.4	Processus de Naissance et de Mort	32
2.5.5	Chaînes de Markov	33
2.6	Modélisation des Systèmes de Files d'Attente	35
2.6.1	Modèles de files d'attente Markoviens	35
2.6.2	Modèle de files d'attente non Markovien	36
2.7	Quelques systèmes d'attente classiques	36
2.7.1	Système M/M/1	36
2.7.2	Système M/M/1/K	36
2.7.3	Système M/M/c	37
2.7.4	Système M/M/∞	38
2.8	Systèmes avec priorités	38
2.8.1	Priorité relative	39
2.8.2	Système $M_R/G_R/1$ avec priorité relative	40

2.8.3	Système $M_2/G_2/1$ avec priorité relative	43
2.8.4	Système $M_2/M_2/1$ avec priorité relative	44
2.9	Systèmes de files d'attente prioritaires markoviens avec sources finies	46
2.10	Systèmes de files d'attente prioritaires avec source finie unique	48
2.11	Conclusion	48
3	Analyse de systèmes prioritaires à l'aide des RdPSG	49
3.1	Introduction	49
3.2	Analyse des RdPSG	50
3.3	Notions préliminaires	50
3.4	La construction de la chaîne de Markov	51
3.5	Evaluation des indices de performances	51
3.6	Modélisation de systèmes avec priorité relative à une seule source finie par les RdPSG	53
3.6.1	Analyse des performances de R_1	55
3.6.2	Application sur le système $M_2/M_2/1//3$	57
3.7	Modélisation de systèmes avec priorité relative à deux sources finies par les RdPSG	64
3.7.1	Analyse des performances de R_2	65
3.7.2	Application sur le système $M_2/M_2/1//(2, 2)$	66
3.8	Modélisation de systèmes avec priorité relative à trois sources finies par les RdPSG	73
3.8.1	Analyse des performances de R_3	75
3.9	Conclusion	76
4	Applications	77
4.1	Introduction	77
4.2	Description de TimeNet	78
4.3	Applications numériques	79

4.3.1	Application 1 : Systèmes mono-serveur à deux sources finies	
	$M_2/M_2/1// (N_1, N_2)$	80
4.3.2	Application 2 : Systèmes mono-serveur à trois sources finies	
	$M_3/M_3/1// (N_1, N_2, N_3)$	81
4.3.3	Application 3 : Systèmes mono-serveur à une seule source finie . . .	82
4.4	Conclusion	85
Conclusion générale et perspectives		86
Bibliographie		88

Table des figures

1.1	Exemple d'un réseau de Petri	7
1.2	Exemple 1 de Franchissement dans un réseau de Petri	11
1.3	Exemple 2 de Franchissement dans un réseau de Petri	11
1.4	Exemple de graphe des marquages accessibles d'un RdP	12
1.5	Exemples de conflits dans un réseau de Petri	14
1.6	Exemple d'un réseau de Petri généralisé	17
1.7	Représentation d'un arc inhibiteur	18
1.8	Temporisation d'un réseau de Petri	20
1.9	Temporisation des transitions d'un réseau de Petri	21
1.10	Représentation des transitions immédiates et temporisées	25
2.1	Système de files d'attente à n serveurs	28
2.2	Les probabilités de transition de $M_2/M_2/1/3$ avec priorité relative	45
3.1	Le RdPSG modélisant le système $M_2/M_2/1//N$ avec priorité relative	53
3.2	Graphe d'accessibilité associé à R_1	58
3.3	Chaîne de Markov réduite associée à R_1 pour $N = 3$	59
3.4	La chaîne de <i>Markov</i> réduite associée à R_1	63
3.5	Le RdPSG modélisant le système $M_2/M_2/1//N_1, N_2$ avec priorité relative	64
3.6	Graphe d'accessibilité associé à R_2	67
3.7	Chaîne de Markov réduite associée à R_2 pour $N_1 = 2, N_2 = 2$	68

3.8	La chaîne de <i>Markov</i> réduite associée à R_2	72
3.9	Le RdPSG modélisant le système $M_3/M_3/1/(N_1, N_2, N_3)$ avec priorité relative	74
4.1	Comparaison de quelques logiciels	79
4.2	Organigramme principal de simulation	83

Liste des tableaux

4.1	Résultats obtenus par les RdPSG pour le cas $M_2/M_2/1// (N_1, N_2)$	80
4.2	Résultats obtenus par Chandra et Sargent pour le cas $M_2/M_2/1// (N_1, N_2)$	80
4.3	Résultats obtenus par les RdPSG pour le cas $M_3/M_3/1// (N_1, N_2, N_3)$. . .	81
4.4	Résultats obtenus par Chandra et Sargent pour le cas $M_3/M_3/1// (N_1, N_2, N_3)$	81
4.5	Comparaison des résultats pour le cas $M_2/M_2/1//N$	84
4.6	Comparaison des résultats pour le cas $M_2/M_2/1//N$	84

Introduction Générale

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. C'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 phénomènes d'attente surviennent naturellement dans la plupart des environnements rencontrés dans la vie courante. Les premiers travaux sur la théorie des files d'attente ont été initiés par l'ingénieur danois A. A. Erlang dans le cadre de la conception des réseaux téléphoniques. Par la suite, il s'est avéré que les problèmes surgissant dans les réseaux téléphoniques apparaissent aussi dans d'autres domaines : industriels, économiques, systèmes informatiques, télécommunications, . . .

Un système de files d'attente (S.F.A.) est un modèle mathématique qui peut décrire un phénomène d'attente. C'est une entité constituée d'une file d'attente et d'un ou plusieurs serveurs. Des clients arrivent à ce système, attendent éventuellement dans la file afin de recevoir un service et quittent par la suite le système. Il existe des situations concrètes où l'on peut avoir à faire à plusieurs classes de clients dans un S.F.A. [57], ces classes se distinguent souvent, soit par des arrivées, des services ou des disciplines de service propres à chaque classe. Pour des raisons particulières, spécifiques au système donné, on est amené à associer des priorités à ces classes. Ces priorités sont souvent dictées par l'urgence d'une tâche, son importance ou son influence sur l'état globale du système, etc. Ainsi, dans ces cas, on parle de disciplines de service prioritaires. Il est à noter que dans chaque classe prioritaire une discipline de service classique est adoptée aux clients de cette classe, souvent la discipline considérée est FIFO (First In First Out). L'importance de cette catégorie de systèmes de files d'attente fait que ces derniers temps plusieurs recherches se sont focalisées sur l'étude et l'analyse de ces systèmes.

Jaiswal dans [54] a fait une étude détaillée sur les différentes disciplines prioritaires ainsi que leurs propriétés. De cette étude, il s'avère que deux catégories principales de disciplines prioritaires sont à considérer : Priorité absolue et Priorité relative. Dans la première discipline, un client verra son service interrompu suite à l'arrivée d'un client plus prioritaire. Alors que, dans la deuxième discipline, un service ne peut être interrompu i.e. l'arrivée d'un client prioritaire n'influe pas sur les clients déjà en service. Mais cette dernière discipline se traduit uniquement par un dépassement des clients de moindre priorité dans la file d'attente. Donc, si un client prioritaire ne trouve pas de serveur libre, il va se positionner derrière les clients qui possèdent une priorité supérieure ou égale à la sienne.

La priorité relative a été introduite par Cobham [24], puis considérée par plusieurs chercheurs : Holloy [48], Dressin et Reich [30], etc. Le modèle multi-serveurs avec plusieurs classes de clients et priorité relative a été analysé par Davis dans [27]. Ce dernier a déterminé la distribution des temps d'attente de chaque classe de clients pour un modèle $M/M/C$. Altinkemer, Bose et Pal ont étudié un modèle $M/D/C$ avec priorité relative [15]. Wagner a traité un modèle multi-serveurs à priorité relative à capacité limitée [34]. Quant à la priorité absolue, elle a été introduite par White et Christie [97]. Pour des temps de service de loi générale, l'analyse de systèmes de cette discipline prioritaire a été faite par Van-Der-Mei et al. [72]. Miller a étudié le calcul des probabilités au régime stationnaire avec les deux types de priorité absolue et relative dans [74]. Pour cela, il a adopté l'approche numérique de la méthode de matrice géométrique établie par Neuts [83]. Chacune de ces deux classes principales de priorités, décrite précédemment, est classifiée à son tour en plusieurs disciplines prioritaires particulières qui seront exposées dans le deuxième chapitre de ce mémoire. Pour plus de détails sur ces systèmes prioritaires, le lecteur peut se référer à [16, 43, 52, 54, 62, 95, 97], etc.

Il a été constaté que les S.F.A. à serveur non fiable peuvent être considérés comme des systèmes d'attente avec priorité absolue ou relative. Dans le cas de priorité absolue, ceci s'explique par l'occurrence d'une panne qui interrompe le service. Ainsi, les périodes de pannes ou de réparations peuvent être vues comme les périodes de service des clients prioritaires (voir White et Chris [97], Keilson [58], Gaver [34]). De même, si la réparation d'un élément défaillant se fait à la fin de service, la situation peut s'interpréter en faisant appel à la discipline de priorité relative. Comme exemple illustratif, le lecteur peut consulter [1], où l'auteur a considéré le système $M_2/G_2/1$ avec priorité relative comme un modèle non fiable avec réparation de l'élément défaillant qui se fait à la fin de service.

Dans certains S.F.A. prioritaires, les clients proviennent d'au moins une population finie. On parle dans ce cas de S.F.A. à source(s) finie(s). La classe de S.F.A. avec source(s) finie(s) a été considérée par Thiruvengadam dans [53] et détaillée par Jaiswal dans [54]. Le problème de la caractérisation du flot des arrivées dans les S.F.A. prioritaires à source(s) finie(s) a donné naissance à deux classifications [54] :

- Les modèles à K classes de priorité avec K sources finies.
- Les modèles où deux classes de priorité ou plus proviennent d'une source finie unique.

Ces deux types de modèles sont intrinsèquement différents et nécessitent des analyses distinctes, contrairement à leurs homologues à sources infinies.

En effet, pour les modèles avec priorité et source infinie, des résultats analytiques existent même si ils sont compliqués. La majorité d'entre eux sont donnés via les transformées de Laplace-Stielje ou les fonctions génératrices. Quand aux modèles prioritaires avec source finie unique, les résultats analytiques sont inexistant. La difficulté d'analyse de ces systèmes est principalement due à la dépendance des flots des arrivées des différentes classes de priorité. Ainsi, leurs études sont basées sur des méthodes approximatives ou des méthodes algorithmiques. Enfin, pour les systèmes prioritaires à sources finies multiples, des résultats analytiques existent, cependant leur analyse présente d'immenses difficultés algébriques [54]. Ainsi, même pour ces derniers systèmes, le recours aux méthodes approximatives ou les méthodes algorithmiques est souvent la seule issue. Les structures de ces différents systèmes prioritaires, leurs modélisations ainsi que leurs analyses sont détaillées dans la monographie de Jaiswal [54].

Dans ce mémoire, nous proposons de faire usage d'une approche basée sur les réseaux de Petri (RdP), pour étudier des systèmes prioritaires particuliers à savoir : $M_2/M_2/1//N$, $M_2/M_2/1//(N_1, N_2)$ et $M_3/M_3/1//(N_1, N_2, N_3)$.

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" [85]. Ce travail a été développé par Anatol W. Holt, F. Commoner, M. Hack et leurs collègues dans le groupe de recherche de Massachusetts Institute of Technology (MIT) dans les années 70. En 1975, la première conférence sur les réseaux de Petri et les méthodes relationnelles a été organisée à MIT. En 1981 le premier livre sur les réseaux de Petri a été publié par J. Peterson.

Plusieurs extensions ont été portées à la définition initiale des RdP. En effet, l'introduction de certains paramètres, caractéristiques ou critères particuliers aux systèmes à étudier, a conduit à la définition de plusieurs formalismes tels que : les réseaux de Petri

stochastiques (RdPS), les réseaux de Petri stochastiques généralisés (RdPSG), les réseaux de Petri temporels, etc. Ces différents formalismes avec leurs propriétés et leurs méthodes d'analyses seront exposées par la suite. Les ouvrages [5, 6, 12, 28, 35, 60, 70, 76, 77, 78, 80, 81, 88] constituent de bons outils pour un état de l'art sur les RdPS.

Vu la particularité des systèmes prioritaires qu'on va étudier, on a choisi de faire usage du formalisme RdPSG pour leurs analyses. En effet, ce choix est dicté par le fait que les systèmes de files d'attente sont modélisés par des processus aléatoires, ainsi, que pour l'existence d'évènements immédiats à caractériser. D'ailleurs, ce formalisme a été exploité pour l'étude de systèmes de files d'attente. A titre d'exemple, on peut citer le travail de C. Oliver et S. Kishor [84] où un système $M/M/1//K$ avec vacance a été étudié via les RdPSG ainsi que les travaux de N. Gharbi [36, 37, 38, 39, 40, 41] qui ont porté sur l'analyse de systèmes de files d'attente avec rappels à un seul serveur, multi-serveurs, avec pannes, avec vacances.

Organisation du document

Ce mémoire est composé :

- d'une introduction où l'idée générale du sujet à traiter est exposée ;
- quatre chapitres où le vif du sujet est traité. Dans le chapitre 1, nous exposons un état de l'art sur les réseaux de Petri et différents formalismes. Un intérêt particulier est accordé au formalisme RDPSG (Réseaux de Petri Stochastique Généralisés). Le chapitre 2 est dédié à quelques systèmes de files d'attente qui nous seront nécessaires dans les chapitres suivants, et là , un intérêt particulier est donné aux systèmes prioritaires. Le chapitre trois est consacré à la présentation de notre approche de modélisation et d'analyse des performances des systèmes mono-serveur avec priorité et source(s) finie(s) à l'aide des RdPSG. Les chaînes de Markov associées aux modèles des réseaux de Petri construits, sont définies. Quand au quatrième chapitre, il est dédié à la modélisation et l'évaluation des indices de performances des systèmes prioritaires considérés en faisant usage des logiciels TimeNet et PIPE.
- et d'une conclusion générale qui donne un bilan de notre travail et une idée sur les perspectives à venir.

1

Etat de l'Art sur les réseaux de Petri

1.1 Introduction

Un réseau de Petri (RdP) est un outil graphique de modélisation et d'évaluation des performances de systèmes complexes. Sa puissance d'expression permet d'étudier des systèmes composés de sous systèmes fonctionnant en parallèle, communiquant et partageant des ressources. Les réseaux de Pétri ont largement été utilisés dans différents domaines [36, 75, 86, 100]. Un point crucial qui est inhérent dans les RdP est qu'ils peuvent être étendus selon les besoins. En plus de ses caractéristiques analytiques, ils offrent une possibilité d'une analyse qualitative. Les réseaux de Petri proposent une modélisation à deux visages :

- Une modélisation statique, qui décrit l'architecture du système, les dépendances existant entre les différentes parties, les actions possibles, les conditions requises pour qu'elles soient effectuables et leurs effets sur différents composants du système.
- Une modélisation dynamique, qui décrit les comportements possibles du système (via un principe de modélisation).

Dans ce chapitre, nous allons introduire les principales définitions, les concepts fondamentaux, les notions de base et les propriétés générales des réseaux de Petri.

1.2 Notions de base

Un réseau de Petri est un graphe biparti orienté qui a deux types de nœuds : un ensemble de places représentant les ressources et un ensemble de transitions symbolisant les évènements ou les opérations. L'état d'un système décrit par un *RdP* est représenté par le marquage de ce graphe [12, 35].

1.2.1 Structure d'un Réseau de Petri

Un réseau de Petri non marqué est un quadruplet :

$$R = (P, T, Pré, Post); \quad (1.1)$$

tel 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,
- $Pré : P \times T \longrightarrow \mathbb{N}$ est l'application d'incidence avant (*places précédentes*),
- $Post : P \times T \longrightarrow \mathbb{N}$ est l'application d'incidence arrière (*places suivantes*).

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

$$C = Post - Pré;$$

Les places sont représentées par un cercle et les transitions en général par un rectangle. 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 des places aux transitions (les préconditions) et des transitions aux places (les postconditions). Par défaut, un arc possède un poids avec la valeur entière 1, ainsi, il peut être un entier supérieur à 1, mais une telle valeur devra être indiquée sur l'arc correspondant. Si tous les arcs ont un poids égal à 1, le RdP est dit ordinaire.

1.2.2 Réseau marqué

Un réseau de Petri marqué est caractérisé par le couple :

$$N = (R, M_0); \quad (1.2)$$

où

- R est un réseau de Petri,
- M_0 est le marquage initial qui est une application définie par :

$$\begin{aligned} M_0 : P &\longrightarrow \mathbb{N}; \\ p &\longmapsto M_0(p); \end{aligned}$$

avec $M_0(p)$ est le nombre de marques (jetons) contenus dans la place p .

Exemple de réseau de Petri

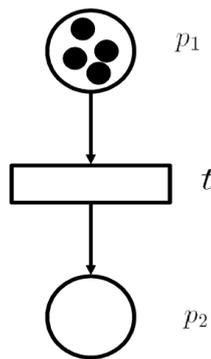


FIG. 1.1 – Exemple d'un réseau de Petri

Cette figure représente un réseau de Petri marqué. Les places p_1 et p_2 contiennent des nombres entiers (positifs ou nuls) de marques dites aussi jetons. Le nombre de jetons dans p_1 est $M_0(p_1) = 4$ et le nombre de jetons dans p_2 est noté par $M_0(p_2) = 0$. Le marquage M du réseau entier est défini par le vecteur de ces marquages i.e.

$$M = (M(p_1), M(p_2)) = (4, 0).$$

L'évolution du marquage par franchissement des transitions dans un réseau de Petri traduit l'évolution du système modélisé dans ces différents états après l'occurrence de certains évènements.

1.2.3 Graphe associé et notations matricielles

A un réseau de Petri on peut associer un graphe qui possède deux types de nœuds : 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 $Pré(p, t) \neq 0$.

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 $Pré$ et $Post$ sont associées aux arcs comme étiquettes (par défaut on prend la valeur 1).

Le marquage M peut être représenté par un vecteur ayant pour dimension le nombre de places. Les fonctions $Pré$, $Post$ et C seront présentées par des matrices dont le nombre de lignes est égal au nombre de places et le nombre de colonnes est égal au nombre de transitions.

Proposition 1.1.

- *Chaque place d'un réseau de Petri peut avoir deux états : elle est marquée ou non marquée .*
- *Un 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'évènement qui lui est associé peut avoir lieu.*
- *Le poids d'un arc est donné par les fonctions $Pré$ et $Post$.*

Pour plus de détails le lecteur peut se référer à [29, 35, 36, 88]

1.3 Interprétation d'un réseau de Petri

Interpréter un réseau de Petri, c'est d'abord donner un sens concret à un modèle mathématique en associant des places, des transitions et des jetons à des entités existantes. Ainsi, les places peuvent être interprétées comme des activités d'un système à évènements discrets, par exemple, les stocks d'un atelier de production, des procédures en cours d'exécution dans un système informatique, etc.

Aux transitions on peut faire correspondre des évènements. On peut également leurs associer des activités ou des procédures à condition qu'elles soient indivisibles ou encore ininterrompibles.

Les *jetons* peuvent être interprétés comme des objets physiques, des informations, des structures de données ou des ressources. Ces entités sont soumises à des événements (franchissement de transitions) qui les font changer d'état (passer d'une place à une autre).

1.3.1 Evolution d'un réseau de Petri

A partir d'un marquage initial, un réseau de Petri peut évoluer. L'évolution, c'est-à-dire la transition d'un marquage vers un marquage suivant, ne peut avoir lieu que si toutes les conditions relatives à cette transition sont satisfaites. Ceci définit la règle de *tir* ou de *franchissement* d'une transition.

Dans les réseaux de Petri, le comportement pourra évoluer lorsque toutes les places relatives à une transition contiendront un nombre suffisant de jetons. Plus précisément, lorsque le nombre de jetons dans chaque place d'entrée d'une transition sera supérieur ou égal au poids de l'arc joignant cette place à la transition. La transition sera alors franchissable (sensibilisée ou tirable). Lorsqu'elle sera franchie (ou tirée), son franchissement définira le marquage suivant du réseau. Le tir dépend donc des jetons, ainsi, la règle d'évolution des réseaux marqués leurs donne une dynamique tout en précisant comment les transitions permettent de modifier l'état du système [25, 42].

Les règles de base qui régissent la procédure de modification du marquage lors du franchissement d'une transition sont les suivantes :

Règle n°1 : une marque est enlevée à chacune des places de l'ensemble d'entrée de la transition.

Règle n°2 : une marque est déposée dans chacune des places de l'ensemble de sortie de la transition.

Règle n°3 : lorsque plusieurs transitions sont validées par un même marquage, une et seulement une transition peut être franchie.

Définition 1.1. (Sensibilisation et franchissement [25])

Une transition t est franchissable (sensibilisée) pour un marquage M si et seulement si :

$$\forall p \in P \quad M(p) \geq \text{Pré}(p, t);$$

On peut exprimer que t est franchissable par les notations :

$$M \geq \text{Pré}(\cdot, t);$$

$$M(t > \cdot.$$

Si t est franchissable pour le marquage M , le franchissement de t donne le nouveau marquage M' tel que :

$$\forall p \in P, M'(p) = M(p) - \text{Pré}(p, t) + \text{Post}(p, t). \quad (1.3)$$

Ceci est exprimé par la notation suivante :

$$M(t > M'.$$

L'évolution d'un réseau, c'est-à-dire le franchissement d'une transition t , est donc une action indivisible qui consiste à retirer un nombre de jetons dans chaque place en amont p égal au poids de l'arc reliant p à t et à ajouter dans chaque place en aval p' de t un nombre de jetons égal au poids de l'arc reliant t à p' . La notion de franchissabilité exprime une condition d'exécution d'une action.

Exemples de Franchissement dans un réseau de Petri**Exemple1**

Les règles de franchissement et de sensibilisation des transitions sont illustrées dans la figure 1.2 en utilisant la réaction chimique connue : $2H_2 + O_2 \longrightarrow 2H_2O$.

Deux jetons dans chaque place en entrée dans : 1.2(a) indique que deux unités de H_2 et deux unités O_2 sont disponibles, donc la transition t est franchissable. Après avoir franchi t , le marquage va changer et on obtient le réseau ayant le nouveau marquage comme 1.2(b). Ceci est le seul franchissement possible de t puisque les conditions de franchissement ne sont plus vérifiées, à savoir qu'il n'y a pas 2 unités de H_2 .

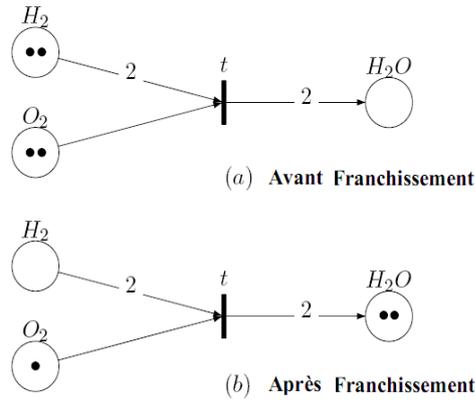


FIG. 1.2 – Exemple 1 de Franchissement dans un réseau de Petri

Exemple2

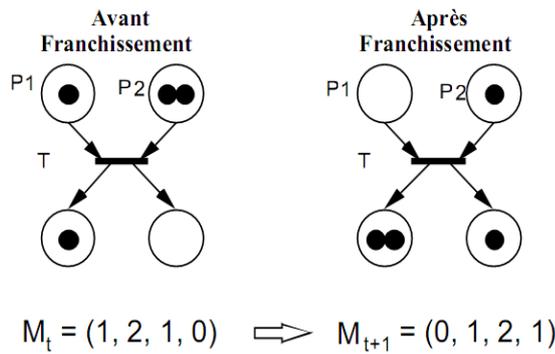


FIG. 1.3 – Exemple 2 de Franchissement dans un réseau de Petri

La figure 1.3 illustre le franchissement de la transition T . En effet, l'état initial était $M_t = (1, 2, 1, 0)$, alors qu'après le franchissement de T le marquage devient $M_{t+1} = (0, 1, 2, 1)$. Ceci est réalisé en ôtant un jeton de chaque place en amont. On constate que M_{t+1} n'est plus franchissable vu qu'il n'y a plus de jetons dans la place P_1 .

Définition 1.2. (Séquence de franchissement [32, 35])

Soit (R, M_0) un réseau de Petri marqué, $s = t_1, t_2, \dots, t_n \in T^*$ une séquence de transitions. La séquence s est franchissable depuis M si et seulement si il existe des marquages M_1, M_2, \dots, M_n tels que :

$$M_1(t_1 > M_2(t_2 > \dots M_{n-1}(t_n > M_n. \tag{1.4}$$

Il est à noter que T^* est un sous ensemble de T constitué des transitions qui forment la séquence de franchissement.

Dans ce cas le tir de s conduit au marquage M_n donc on note $M(s > M_n)$.

Définition 1.3. (Marquage accessible [28])

Soit (R, M_0) un réseau de Petri marqué. Un marquage M est accessible si et seulement si il existe une séquence de franchissement $s \in T^*$ telle que $M_0(s > M)$.

Définition 1.4. (Ensemble d'accessibilité [28])

Soit (R, M_0) un réseau de Petri. L'ensemble des marquages accessibles ou ensemble d'accessibilité d'un réseau est noté $A(R, M_0)$ où A est l'ensemble des marquages atteints par une séquence de franchissement :

$$A(R, M_0) = \{M \in \mathbb{N}^P / \exists s \in T^* \text{ tel que } M_0(s > M)\}. \tag{1.5}$$

Définition 1.5. (Graphe des marquages accessibles [28])

Soit (R, M_0) un réseau de Petri. 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 de $A(R, M_0)$ et dont les arcs, étiquetés par les noms des transitions, sont définis par la relation de tir entre les marquages. Un arc étiqueté par t joint M à M' si et seulement si $M(t > M')$.

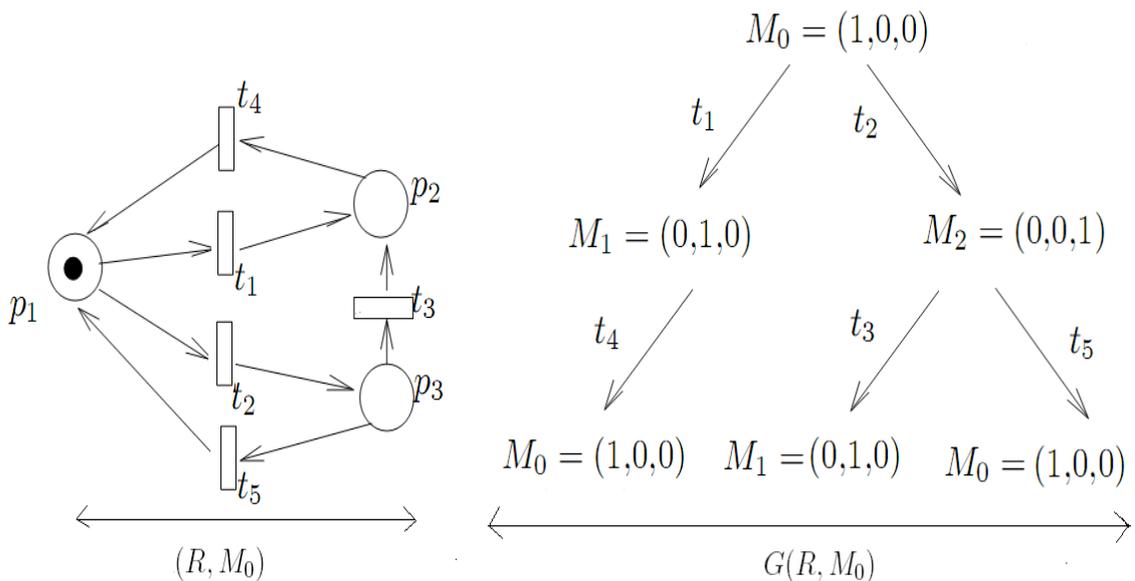


FIG. 1.4 – Exemple de graphe des marquages accessibles d'un RdP

1.3.2 Conflit et parallélisme

Nous venons de voir que l'évolution d'un réseau de Petri est réalisée par le franchissement de transitions sensibilisées. Pour un marquage donné, plusieurs transitions peuvent être sensibilisées mais la règle d'évolution impose qu'une seule transition puisse être franchie à la fois.

Si les transitions sensibilisées n'ont pas de place en amont commune, le tir de l'une d'entre elles ne remet pas en cause celui des autres. On parle dans ce cas d'un RdP sans conflit (toute place a au plus une transition de sortie). Si les transitions sensibilisées ont une place amont en commun, le tir de l'une d'entre elles peut remettre en question celui des autres. Ces transitions sont en conflit qui conduit à un indéterminisme dans l'évolution d'un réseau lié à la compétition entre ces transitions [29, 35].

Définition 1.6. *Deux transitions t_1 et t_2 sont en conflit structurel si et seulement si elles ont au moins une place commune en entrée, i.e. :*

$$\exists p \in P, \text{Pré}(p, t_1) \times \text{Pré}(p, t_2) \neq 0. \quad (1.6)$$

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

$$M(t_1) >; \quad (1.7)$$

et

$$M(t_2) >; \quad (1.8)$$

et

$$\exists p \in P : M(p) < \text{Pré}(p, t_1) + \text{pré}(p, t_2). \quad (1.9)$$

Définition 1.7. *Deux transitions t_1 et t_2 sont en parallèles structurellement si :*

$$(\text{Pré}(\cdot, t_1))^T \times \text{Pré}(\cdot, t_2) = 0. \quad (1.10)$$

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

$$M(t_1) >; \quad (1.11)$$

$$M(t_2) >. \quad (1.12)$$

Un conflit est l'existence d'une place qui a au moins deux transitions de sortie. La notion de conflit modélise un choix ou une décision.

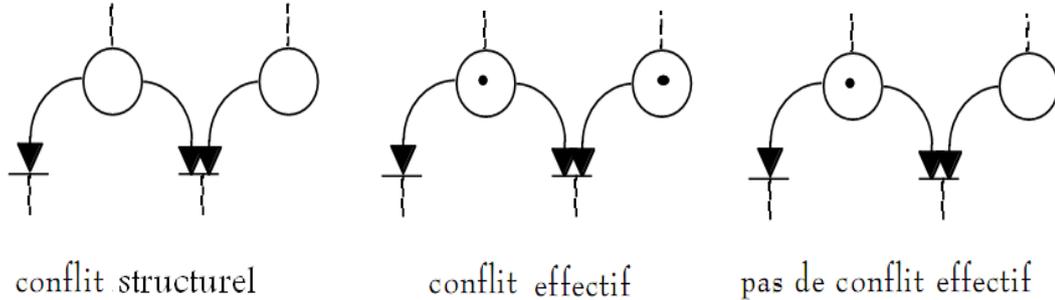


FIG. 1.5 – Exemples de conflits dans un réseau de Petri

1.3.3 Propriétés d'un réseau de Petri

Après avoir modélisé un système par RdP, la question naturelle qui se pose est :
Qu'est-ce qu'on peut faire avec ce modèle ?

Ce modèle de réseau de Petri nous donne des techniques pour analyser les propriétés qui sont les caractéristiques qui permettent d'évaluer la qualité d'un système donné. Parmi les propriétés des RdP on citera dans ce qui suit les plus importantes d'entre elles [28, 35] :

a) Réseau borné

La bornitude d'un RdP exprime le fait que le nombre d'états que peut prendre le système modélisé par ce RdP est fini, autrement dit, le nombre de marquages accessibles est fini. Dans le cas contraire, où le RdP est non borné, le nombre d'états est infini et ceci est dû au fait que certains paramètres de ce système sont non bornés. Par exemple le paramètre "Taille" d'un buffer peut être non limitée ce qui introduit la non bornitude du modèle RdP.

Définition 1.8. Soit un réseau de Petri $R = (P, T, Pré, Post)$; une place $p \in P$ est dite k -bornée pour un marquage initial M_0 si et seulement si :

$$\exists k \in \mathbb{N}, \forall M' \in A(R, M_0), M'(p) \leq k; \quad (1.13)$$

où $A(R, M_0)$ est l'ensemble des marquages accessibles.

Si $k = 1$, on dit que la place p est sauf.

Définition 1.9. Un réseau de Petri $R = (P, T, Pré, Post)$ est dit k -borné (respectivement sauf) pour un marquage initial, si et seulement si toutes ses places sont k -bornées (respectivement saufs).

Définition 1.10. Un réseau marqué (R, M_0) est borné, si le nombre de marquages accessibles est fini.

b) Réseau sans blocage

Un RdP est dit bloqué si à une certaine étape son franchissement s'arrête et aucune transition n'est franchissable. Formellement ceci est donné par la définition suivante :

Définition 1.11. Un marquage M' d'un réseau (R, M_0) est appelé marquage puits (mort) si aucune transition n'est franchissable à partir de M' . Un réseau (R, M_0) est dit sans blocage si tout marquage accessible depuis M_0 n'est pas un marquage puits. Un réseau R est dit sans blocage s'il existe un marquage initial M_0 tel que $\langle R, M_0 \rangle$ soit sans blocage.

c) Vivacité

La vivacité est une propriété importante pour traduire le bon fonctionnement d'un système. D'ailleurs, un réseau vivant modélise un système en fonctionnement permanent sans aucun blocage.

Définition 1.12. Un réseau de Petri (R, M_0) est vivant si :

$$\forall M \in A(R, M_0), \forall t \in T, \exists M' \in A(R, M) \text{ tel que } M'(t > . \quad (1.14)$$

Autrement dit,

$$\forall M \in A(R, M_0), \forall t \in T, \exists s \in T^* \text{ tel que } M(st > . \quad (1.15)$$

d) Etat d'accueil

Un état d'accueil caractérise le fait que le système peut se retrouver dans cet état à partir des marquages en aval. Si le marquage initial est un état d'accueil, cela signifie que l'on peut toujours réinitialiser le système. Ce qui est exprimé formellement dans la définition suivante :

Définition 1.13. Un marquage accessible M d'un réseau marqué (R, M_0) est un état d'accueil s'il est accessible à partir de n'importe quel autre marquage accessible :

$$\forall M' \in A(R, M_0), \exists s \in T^* / M'(s > M. \quad (1.16)$$

e) **Graphe fortement connexe**

- Une composante fortement connexe d'un graphe est un sous graphe tel qu'il existe un chemin (orienté) entre tout point A et tout point B de ce sous graphe.
- Une composante fortement connexe est dite terminale, si aucun sommet de celle-ci ne possède de successeurs dans une autre composante fortement connexe.
- Un graphe est dit fortement connexe s'il possède une seule composante fortement connexe.

L'existence d'un état d'accueil dans un RdP est liée à la connexité de ce RdP, la proposition suivante nous donne cette liaison.

Proposition 1.2.

Soit (R, M_0) un réseau de Petri borné. Le réseau (R, M_0) a un état d'accueil si et seulement si le graphe d'accessibilité comprend une seule composante fortement connexe terminale.

Remarque 1.1. *Si le marquage initial d'un RdP est un état d'accueil alors le système modélisé par ce RdP est réinitialisable.*

1.4 Extension des réseaux de Petri

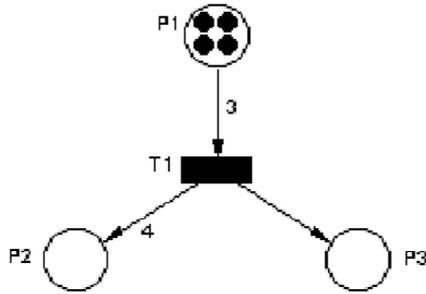
Divers extensions ont été portées aux RdP classiques, ce qui a donné naissance à plusieurs formalismes de RdP. Ces formalismes introduisent entre autres l'aspect temporel, aspect stochastique, etc. Ce qui a enrichi les structures des RdP.

1.4.1 Réseaux de Petri 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 est augmenté par le poids p . Cette extension des RdP permet la réduction de la taille du RdP ordinaire. Toutefois, la transformation est possible du RdP généralisé vers le RdP ordinaire.

Exemple

Avant franchissement :



Après franchissement :

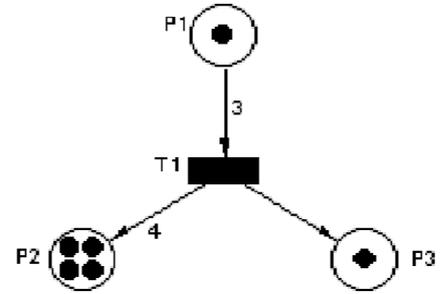


FIG. 1.6 – Exemple d'un réseau de Petri généralisé

1.4.2 Réseaux de Petri à arcs inhibiteurs

On a vu précédemment qu'une transition est franchissable si la place contient des jetons. Pour conditionner le tir d'une transition par l'absence de jetons dans une ou plusieurs places, il faut donc enrichir le modèle. On ajoute pour cela un nouveau type d'arcs, allant d'une place vers une transition, appelé arc inhibiteur. Cet arc indique l'une des conditions de franchissabilité de la transition est que la place ne contienne aucun jeton. La définition formelle des réseaux de Petri à arcs inhibiteur est la suivante [35] :

Définition 1.14. *Un réseau de Petri à arcs inhibiteur est défini par un 5-uplet*

$R = (P, T, Pré, Post, Inh)$ où :

- P est un ensemble fini de places et T un ensemble fini de transitions ;
- $Pré$ et $Post : P \times T \rightarrow \mathbb{N}$ sont les fonctions d'incidence avant et d'incidence arrière respectivement ;
- $Inh : P \times T \rightarrow \mathbb{N} \setminus \{0\}$ est la fonction d'inhibition.

La règle de franchissement d'une transition dans un RdP à arc inhibiteur est expliquée dans la définition suivante :

Définition 1.15. *Soit M un marquage d'un réseau de Petri à arcs inhibiteurs et t une transition :*

la transition t est franchissable à partir de M si et seulement si :

$$\forall p \in P, M(p) \geq Pré(p, t) \text{ et } M(p) < Inh(p, t). \quad (1.17)$$

Le franchissement de t à partir de M conduit au marquage M' défini par :

$$\forall p \in P, M'(p) = M(p) - \text{Pré}(p, t) + \text{Post}(p, t). \quad (1.18)$$

On note que dans les réseaux de Petri à arcs inhibiteurs, seulement la condition de franchissabilité est modifiée. Sa représentation graphique ne diffère de celle des réseaux standards que par le fait qu'un arc inhibiteur est représenté par une flèche avec un petit cercle à l'extrémité au lieu d'une flèche tout court.

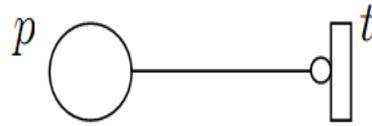


FIG. 1.7 – Représentation d'un arc inhibiteur

1.4.3 Réseaux de Petri colorés

Les réseaux de Petri colorés ont été définis pour enrichir l'information portée par les jetons. Ils facilitent la modélisation de systèmes de grande taille, ainsi, ils présentent un grand intérêt pour modéliser certains systèmes complexes. En effet :

- Dans le but de différencier les jetons, on leur associe des couleurs. Ainsi, on associe à chaque place l'ensemble de couleurs des jetons qui peuvent y séjourner. A chaque transition on associe un ensemble de couleurs correspondant aux manières avec lesquelles la transition sera franchie.

- Le franchissement des jetons d'un RdP coloré peut être effectué de plusieurs manières en fonction des couleurs associées aux transitions. La relation entre les couleurs de franchissement et le marquage coloré est définie par des fonctions associées aux arcs.

- La couleur dans un RdP coloré peut être un n-uplet et peut donc porter une information complexe telle que par exemple la nature et la position d'un objet dans un stock, ou bien une trame de donnée avec une valeur et une adresse de destination [61]. A partir des notions introduites précédemment, nous obtenons la définition suivante d'un RdP coloré [55, 56] :

Définition 1.16. *Un réseau de Petri coloré est un 6-uplet défini par :*

$$R_c = (P, T, C_{oul}, C_{sec}, W, M_o);$$

où

- P est un ensemble fini de places ;
- T est un ensemble fini de transitions ;
- C_{oul} est un ensemble fini de couleurs ;
- C_{sec} est la fonction qui à chaque place et à chaque transition associe un sous ensemble de C_{oul} :

$$C_{sec} : P \cup T \longrightarrow \mathcal{P}(C_{oul});$$

- W est la fonction d'incidence C=Post- Pré :

$$W(p, t) : C_{sec}(t) \times C_{sec}(p) \longrightarrow \mathbb{N};$$

- M_o est le marquage initial, pour chaque place et pour chaque couleur possible dans cette place. Il associe un nombre de jetons :

$$M_o(p) : C_{sec}(p) \longrightarrow \mathbb{N};$$

Remarque 1.2. *Les réseaux de Petri colorés ne sont pas plus puissants que les réseaux de Petri classiques. Ils permettent simplement une abréviation de ces derniers. L'opération de passage d'un réseau de Petri simple au réseau de Petri coloré est appelée opération de pliage et l'opération inverse est appelée dépliage.*

1.4.4 Réseaux de Petri temporisés

Les RdP temporisés sont une extension des RdP ordinaires. Cette extension est caractérisée par l'ajout de temporisations i.e. l'introduction de la variable temps. Il existe deux sortes de RdP temporisés : RdP T -temporisés [86] et des RdP P -temporisés [92, 93]. Dans le premier cas une durée de temps est associée aux transitions. La transition est donc validée après écoulement de la durée qui lui est allouée. Dans le second cas, la temporisation est associée aux places. Ces temporisations peuvent traduire des durées de déroulement des actions et des opérations associées aux transitions ou aux places [35, 59].

A) Réseau de Petri P -temporisé

Définition 1.17. Un réseau de Petri P -temporisé est défini par le couple (R, d) avec :

- R est un réseau de petri $(P, T, Pré, Post, M_0)$,
- $d : T \rightarrow Q^+$ est la fonction de temporisation. La règle de tir doit tenir compte du temps. Soit t_0 l'instant où une marque est déposée dans une place p . La marque est indisponible (on dit qu'elle est gelée) pendant une durée égale à $d(p)$, i.e. dans l'intervalle $[t_0, t_0 + d(p)]$. Ensuite, la marque peut être utilisée pour le tir d'une transition.

Les réseaux temporisés sont souvent utilisés pour la représentation formelle des synchronisations dans les systèmes. Dans la figure suivante est donné un exemple de RdP P -temporisé.

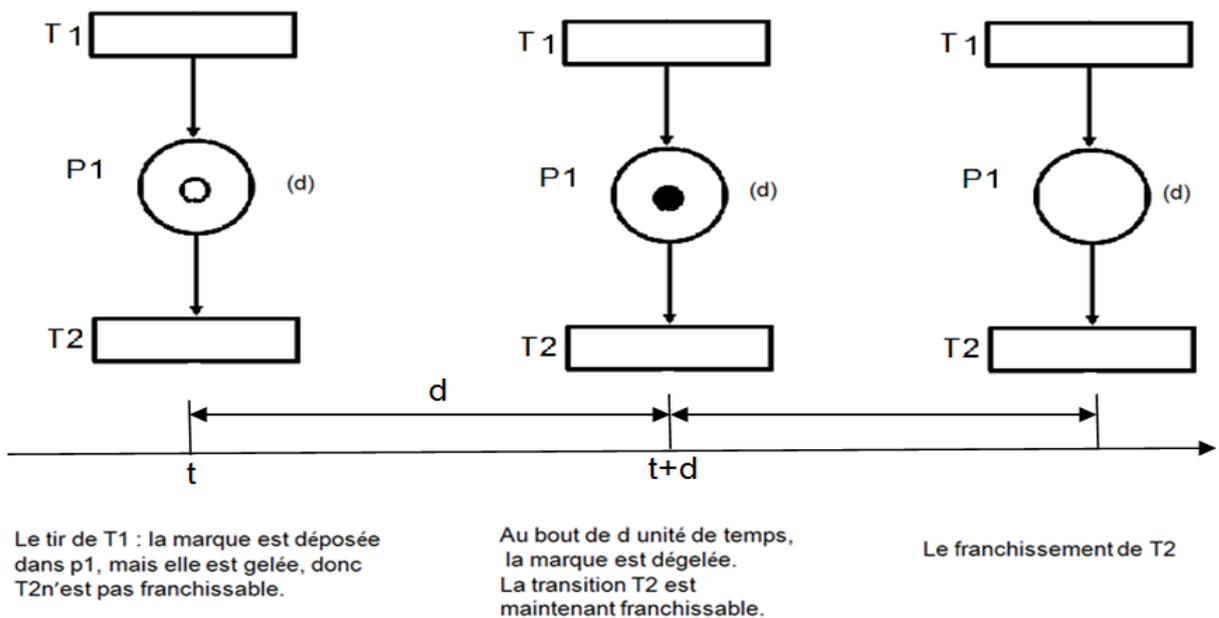


FIG. 1.8 – Temporisation d'un réseau de Petri

Dans ce modèle, un paramètre temporel est associé à chaque place. La sémantique de ce paramètre correspond au temps de séjour minimum d'une marque dans une place (temps d'indisponibilité), et d représente la durée d'indisponibilité de la marque pour la validation des transitions. Après le franchissement de la transition T_1 , le jeton reste gelé dans la place P_1 pendant d unités de temps. Il ne peut pas être consommé avant l'écoulement du temps qui lui est associé. Dès que ce temps est achevé, la transition T_2 sera sensibilisée.

B) Réseaux de Petri T-temporisés

La temporisation des transitions consiste à associer une durée de temps à chaque transition. Le tir d'une transition est décomposé en deux durée de temps : l'enclenchement du tir et la fin du tir, séparés par la durée spécifiée par la fonction de temporisation. Les premiers travaux sur cette classe de réseaux ont été réalisés par Ramchandani [87].

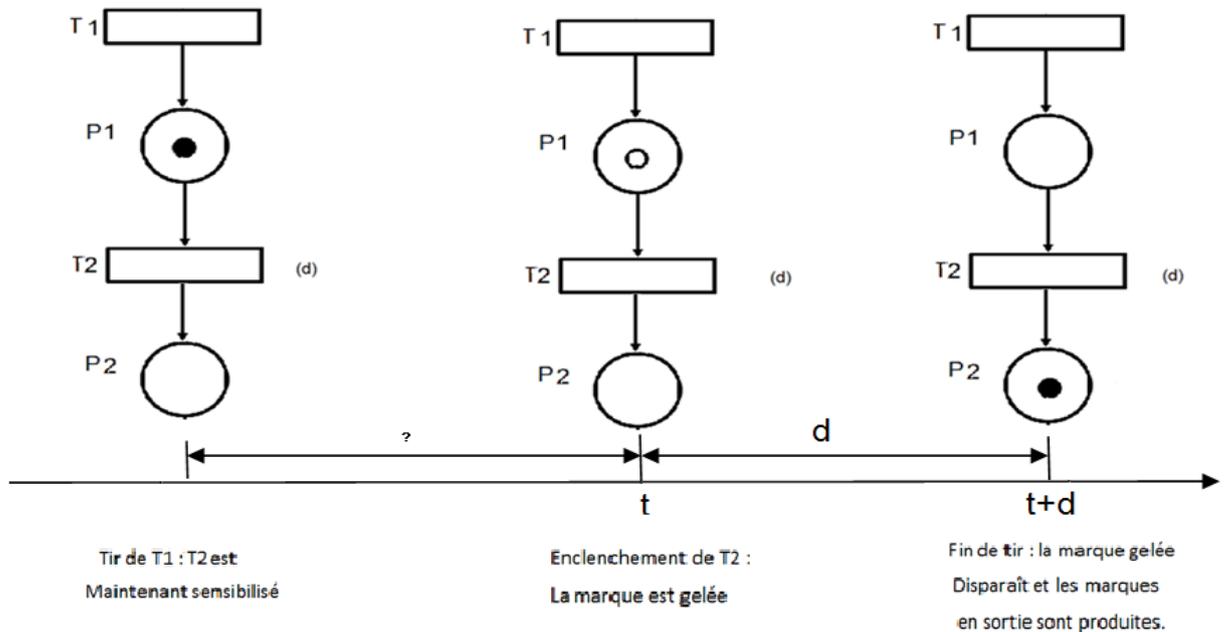


FIG. 1.9 – Temporisation des transitions d'un réseau de Petri

Dans cette figure le franchissement de la transition T_2 est ininterrompible. Après sensibilisation de la transition T_2 la marque est gelée. Il faut attendre d unité de temps pour l'apparition du jeton dans la place P_2 . Pendant cet intervalle de temps, ce jeton ne peut plus être utilisé pour valider d'autres transitions.

1.4.5 Réseaux de Petri temporels

Ces modèles de RdP ont été introduits par [10, 73] pour la modélisation et l'analyse des systèmes de communication. Une transition dans ce RdP peut être franchie seulement lorsqu'une durée de temps comprise dans l'intervalle du temps associé à la transition ou bien à la place s'est écoulée depuis l'instant de sa validation. La transition t_i doit rester sensibilisée durant au moins d_{imin} unités de temps et au plus d_{imax} unités de temps avant

d'être franchie. Par contre les jetons peuvent à tout moment être consommés par une autre transition.

Définition 1.18. *Un réseau de Petri temporel est une paire $N_t = (R, IS)$ avec :*

- R est un réseau de petri $(P, T, Pré, Post, M_0)$,
- IS est la fonction durée de franchissement :

$$IS : T \longrightarrow (\mathbb{Q}^+ \cup \infty) \times (\mathbb{Q}^+ \cup \infty)$$

$$t_i \longrightarrow IS_i(t_i) = [a_i, b_i].$$

La fonction IS associe à chaque transition t_i du réseau un intervalle de temps à bornes rationnelles $IS_i = [a_i, b_i]$, où a_i est la date de tir au plus tôt et b_i est la date de tir au plus tard de t_i [90].

1.4.6 Réseaux de Petri stochastiques

Les réseaux de Petri stochastiques ont été introduits par Natkin, Molloy [76, 77, 81]. Ces auteurs ont été les premiers chercheurs, qui séparément, ont introduit cette nouvelle extension (RdPS). Depuis, de très nombreux modèles ont été définis, des synthèses complètes sont proposées dans [22, 29, 33, 69, 71, 82] afin de répondre à certains problèmes d'évaluation quantitative des systèmes informatiques industriels.

Dans les réseaux de Petri stochastiques, les délais associés aux transitions sont aléatoires contrairement aux durées déterministes et constantes associées aux RdP temporisés. Ces temps sont modélisés par des variables aléatoires dont la loi est souvent considérée comme étant exponentielle. Ceci, permet d'approcher le graphe des marquages à un processus markovien.

Dans les réseaux de Petri temporels on prend toutes les valeurs d'un intervalle $([\theta_{min}, \theta_{max}])$, par contre, dans les RdPS la durée de sensibilisation est une variable aléatoire θ , avec une distribution de probabilité, dans le cas de distribution exponentielle :

$$P_\theta(x) = P[\theta \leq x] = 1 - e^{-\lambda x}.$$

La fonction $P_\theta(x)$ décrit la probabilité pour que le franchissement ait lieu avant x , c'est à dire pour que la durée de sensibilisation soit inférieure ou égale à x .

On en déduit que la valeur moyenne de la durée de sensibilisation est :

$$\bar{\theta} = \int_0^\infty (1 - P_\theta(x)) dx = \int_0^\infty e^{-\lambda x} dx = \frac{1}{\lambda};$$

λ est appelé *taux de franchissement de la transition*.

La propriété d'absence de mémoire des distributions exponentielles des délais de franchissement a permis à Molloy [77] de montrer que les réseaux de Petri stochastiques sont isomorphes aux processus de Markov à temps continu et à espace d'états discret. En particulier, un réseau de Petri stochastique k -borné est isomorphe à un processus de Markov à espace d'états fini. Les techniques des processus markoviens sont alors exploitées pour calculer les mesures de performances.

Les avantages des réseaux de Petri stochastiques sont nombreux : ils sont très flexibles, leur puissance de modélisation est équivalente à celle des processus de Markov, ils ont une représentation graphique naturelle.

La vérification des propriétés structurelles des RdP permet la validation des modèles associés. Même si la construction des modèles de RdPS pour des systèmes réels sont primitives i.e. qu'elle nécessite beaucoup de travail et la représentation graphique est, par conséquent, complexe. Cependant, une fois les modèles construits, les possibilités d'analyse qu'ils offrent couvrent le coût de ces complexités [17].

Définition 1.19. *Un réseau de Petri stochastique est le couple (R, Λ) avec :*

- R est le réseau de Petri $(P, T, Pré, Post, M_0)$,
- Λ est une fonction qui à chaque transition t associe un taux de franchissement $\lambda_t = \Lambda(t)$.

Avec les RdPS, on pourra par exemple, calculer le temps de bon fonctionnement entre deux défaillances, le temps de réparation ou dans certains cas la durée opérationnelle d'une machine, les taux de production, l'évolution des stocks, etc. La modélisation des systèmes nécessite parfois de prendre en compte des transitions non stochastiques (immédiates). Dans ce but une extension des réseaux de Petri stochastiques a été proposée elle porte le nom de " Réseaux de Petri Stochastiques Généralisés RdPSG" [3].

L'approche que nous proposons pour l'évaluation des performances des files d'attente avec priorité est basée sur ces réseaux de Petri stochastiques. Notre choix s'est porté sur les RdPSG pour l'évaluation des performances des systèmes avec priorité et source(s) finie(s). Ce choix est dicté pour les avantages des RdPSG à savoir [9] :

- leur puissance de modélisation et en particulier leur aptitude à prendre en compte les différents types de dépendances qui caractérisent le comportement d'un système : la concurrence, le parallélisme et la synchronisation,
- leur puissance d'analyse des propriétés du système et de validation du modèle établi,
- leur aptitude à évaluer des mesures du comportement du système en particulier des indices de performances.

Suite à ces avantages les RdPSG ont été explicités pour l'analyse des performances des systèmes de files d'attente compliqués. Citons à titre d'exemple les travaux de N. Gharbi [36, 37, 38, 39, 40, 41]. Dans sa thèse de doctorat [36], elle s'est intéressée à l'évaluation de performances et de la fiabilité de systèmes d'attente avec rappels exponentiels. Dans ses autres travaux, elle a proposé des modélisations via les RdPSG pour des systèmes à un seul serveur, multi-serveurs, avec pannes, avec vacances. Ces modélisations lui ont permis d'obtenir une analyse quantitative des performances de ces systèmes.

Les RdPSG sont présentés en détail dans les références suivantes le lecteur peut se référer aux ouvrages suivants [2, 3, 5, 11, 26, 66, 67, 68, 69, 71, 76, 77, 81]. Dans la section suivante, on va se limiter à présenter les principales propriétés de ce formalisme RdPSG et à introduire quelques définitions qui nous seront utiles par la suite.

1.5 Réseaux de Petri Stochastiques Généralisés

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 associées 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.

Remarque 1.3. Dans la représentation graphique d'un RdPSG, les transitions immédiates sont représentées par des traits et les transitions temporisées par des rectangles.



FIG. 1.10 – Représentation des transitions immédiates et temporisées

La définition formelle d'un RdPSG est donnée par la définition suivante :

Définition 1.20. Un réseau de Petri stochastique généralisé est un huit-uplet $(P, T, Pré, post, Inh, pri, W, M_0)$ [7] où :

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

- $M_0 : P \longrightarrow \mathbb{N}$ est le marquage initial du réseau.

Remarque 1.4. On peut utiliser des poids pour le calcul des probabilités de franchissement des transitions immédiates et éventuellement pour la résolution probabiliste des conflits entre plusieurs transitions immédiates sensibilisées [68].

Remarque 1.5. De ce formalisme RdPSG une question légitime se pose : L'introduction des transitions immédiates remet-elle en cause le caractère markovien homogène du processus de marquage ?

Ajmore et al ont montré que s'il n'existe pas de suite infini d'états évanescents successivement atteignables alors le processus de marquage après élimination des états évanescents reste un processus de Markov homogène [2].

1.6 Conclusion

Le formalisme des RdP, adapté à la prise en compte des problèmes de concurrence, de synchronisme et de parallélisme, constitue un excellent outil de spécification fonctionnelle d'un problème et de mise en évidence des contraintes. Les propriétés mathématiques qui découlent de l'analyse des RdP permettent une étude comportementale et structurelle essentielle à la validation d'une spécification.

Dans ce chapitre, nous avons introduit le modèle de base des RdP. Nous nous sommes intéressés principalement aux aspects dynamiques et à l'étude comportementale de ces derniers. Par ailleurs, nous avons exposé les différentes extensions apportées à ces RdP de base. L'un de ces formalismes qui est RdPSG sera exploité dans les chapitres suivants pour l'analyse des systèmes prioritaires.

2

Systemes de files d'attente

2.1 Introduction

La théorie des files d'attente définit les lois mathématiques qui régissent un système de files d'attente en vue de dégager des paramètres qui le caractérisent, tels que le temps moyen d'attente, le nombre moyen de clients dans la file, le nombre moyen de clients dans le système, etc.

Les exemples de phénomènes d'attente dans les sociétés dites modernes sont nombreux. Cela va des phénomènes les plus visibles, par exemple dans le domaine des transports (terrestre, aérien, ...) ou des services (banque, poste, ...) aux phénomènes d'attente plus discrets que l'on retrouve dans certains systèmes tels que les réseaux téléphoniques, les systèmes informatiques, ou les réseaux de télécommunication. Le formalisme des files d'attente et les résultats associés ont été introduits, puis ont été développés comme une discipline très pratique dont l'objectif était de construire des modèles permettant de prédire le comportement des systèmes fournissant un service à des demandes aléatoires.

Dans ce chapitre, nous allons donner quelques notions générales de la théorie des files d'attente. Un intérêt particulier sera accordé aux systèmes prioritaires, qui sont l'objet

de notre travail.

2.2 Représentation d'un système d'attente

Tous les exemples de phénomènes d'attente ont des caractéristiques communes que l'on peut résumer ainsi : Des entités ou des ressources parviennent suivant un processus, souvent aléatoire, acquérir un service auprès d'une autre entité dite serveur.

Une file commencera à se manifester dès que le taux des arrivées excède le taux de service. Cette file d'attente peut ne pas se manifester de façon personnifiée. On peut parler d'une file de machines dans un atelier qui attendent la réparation, ou d'une file de bateaux qui attendent le déchargement devant un quai, etc.

Remarquons que le processus ainsi décrit est un processus stochastique, car on ne peut connaître à l'avance ni le temps d'arrivée d'un client, ni sa durée de service qu'il demandera.

Les paramètres suivants permettent de décrire un système de files d'attente dont le

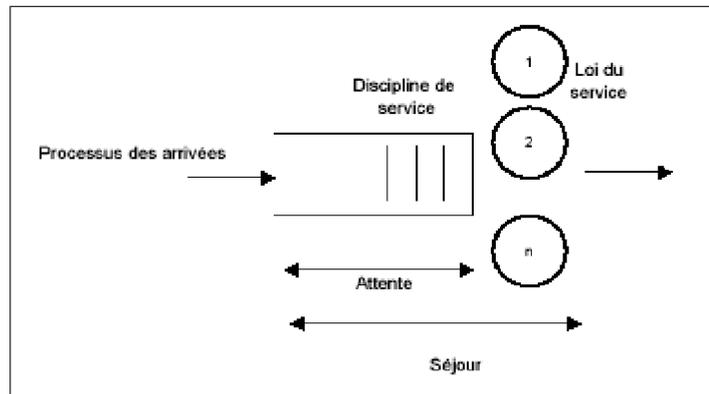


FIG. 2.1 – Système de files d'attente à n serveurs

schéma général est illustré dans la figure 2.1 :

- **Le processus des arrivées des clients :** L'arrivée des clients à un système de files d'attente peut être décrit par un ensemble de variables aléatoires qui forme un processus dit processus des arrivées. Souvent, on suppose que les temps entre deux arrivées successives sont indépendants et de même loi. Dans beaucoup de situations pratiques, ce flux des arrivées est poissonnien. Bien sûr les arrivées des clients peuvent être individuelles ou groupés [46].

- **Source** : C'est la population d'où provient les clients. Elle peut être finie ou infinie suivant le système étudié.
- **Le processus de service** : De la même manière que dans le cas des arrivées, les services se décrivent par un processus aléatoire.
- **Capacité du système** : Elle représente le nombre maximal de clients que le système peut accueillir dans sa file d'attente et dans ses serveurs.
- **Nombre de serveurs** : Un système de files d'attente peut être constitué d'un seul serveur ou de plusieurs serveurs en parallèle et dans ce dernier cas il est dit multi-serveurs.
- **Discipline de service** : Elle spécifie la manière avec laquelle le serveur sélectionne le prochain client à servir. Les principales disciplines de service utilisées sont :
 - FIFO (First In First Out)** : Le premier arrivé sera le premier servi.
 - LIFO (Last In First Out)** : Le dernier arrivé sera le premier servi.
 - FIRO (First In Random Out)** : Les clients sont servis de manière aléatoire.
 - ⋮

La discipline de service usuelle est FIFO.

Ainsi, pour décrire un système de files d'attente, il suffit de donner ses paramètres par la notation dite de Kendall suivante :

$$A/B/m/K/N/D$$

où :

- A** : le processus des arrivées ;
- B** : le processus des services ;
- m** : le nombre de serveurs ;
- K** : la capacité du système ;
- N** : la taille de la source ;
- D** : la discipline de service.

Pour décrire les lois des processus **A** et **B**, des symboles tels ceux donnés ci dessous, sont utilisés :

- M** : loi markovienne (lois sans mémoire);
- E_k** : loi d'Erlang;
- H_k** : loi hyper exponentielle à k étapes;
- D** : loi déterministe;
- G** : loi générale;
- GI** : loi générale indépendante;
- ⋮

2.3 Formule de Little

Soit λ le taux des arrivées, W le temps moyen d'attente et L le nombre moyen de clients présents dans le système. Alors, si λ , W et L existent, ils sont reliés l'un à l'autre par l'équation :

$$L = \lambda W. \quad (2.1)$$

Cette formule est l'un des résultats les plus généraux et utiles dans la théorie des files d'attente [65].

2.4 Mesures de Performance

Précédemment, on a donné les principaux paramètres qui permettent de décrire un système de files d'attente. Dans ce qui suit, on donnera les principales caractéristiques que l'on cherche à dégager d'une analyse de ce système. En effet, parmi ces caractéristiques, on trouve :

- L_s : Nombre moyen de clients dans le système.
- L_f : Nombre moyen de clients dans la file.
- W_f : Durée moyenne d'attente d'un client dans la file.
- W_s : Durée moyenne de séjour d'un client dans le système.

Les valeurs de L_s et L_f peuvent être obtenues, en fonction de celles de W_s et W_f , respectivement, en utilisant la formule de Little :

$$\begin{aligned}L_s &= \lambda W_s; \\L_f &= \lambda W_f;\end{aligned}$$

où λ est le taux des arrivées.

2.5 Processus stochastiques

Pour étudier un phénomène d'attente on le rapporte généralement à un système décrit par un processus stochastique. Un processus stochastique est un système dynamique dont l'état évolue au cours du temps de manière aléatoire.

Définition 2.1. : *On appelle processus stochastique, une famille de variables aléatoires $\{X_t, t \in T\}$ où t est un paramètre parcourant l'ensemble T .*

En pratique, T est soit l'ensemble \mathbb{R} des réels, soit l'ensemble \mathbb{R}^+ des réels positifs, soit l'ensemble \mathbb{N} des entiers naturels. Si l'ensemble T est dénombrable, le processus est dit discret et les X_t forment une suite stochastique. Sinon, le processus est dit continu (permanent).

On appelle espace des états l'ensemble E où les variables X_t prennent leurs valeurs. E peut être discret, dans ce cas on parle de chaîne, ou continu et on parle de processus [21].

2.5.1 Propriété : sans mémoire

X est une variable aléatoire sans mémoire si :

$$P(X < t + t_0 / t > t_0) = P(X < t).$$

La loi exponentielle de paramètre λ est la seule loi continue sans mémoire.

2.5.2 Processus Markovien

Un processus $\{X_t\}_{t \geq 0}$, à espace d'état E , est un processus de Markov si l'équation :

$$P(X_t \leq x / X_{t_1} = y_1, X_{t_2} = y_2, \dots, X_{t_n} = y_n) = P(X_t \leq x / X_{t_n} = y_n) \quad t > 0,$$

est vérifiée pour tout $t_1 < t_2 < \dots < t_n < t; n \in \mathbb{N}$. Ce qui signifie, que connaissant le présent du processus, son futur dépend seulement de l'instant présent, i.e. il est indépendant de son passé. Cette propriété est connue sous le nom de Propriété de Markov ou propriété sans mémoire. E est l'ensemble des valeurs possibles du processus $X(t)$.

2.5.3 Processus de Poisson

Parmi les processus stochastiques à temps continu, le processus de Poisson occupe une place privilégiée. Il est utilisé, avant tout, pour décrire la réalisation dans le temps d'événements aléatoires de types donnés tels que les arrivées des clients vers un système de files d'attente. Le processus de poisson est un cas particulier du processus de naissance et de mort, satisfaisant les hypothèses suivantes :

1. Le processus est strictement stationnaire ;
2. Le processus est homogène dans le temps ;
3. Il arrive au plus un événement à la fois.

La probabilité d'observer k occurrences consécutives d'un événement dans un intervalle de longueur t est :

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

2.5.4 Processus de Naissance et de Mort

Soit un processus stochastique $\{X_t\}_{t \geq 0}$ à espace d'état discret $n \in \mathbb{N}$ et homogène dans le temps, c'est-à-dire que :

$$P(X_{t+s} = j / X_s = i) = p_{ij},$$

ne dépend pas de s . Alors $\{X_t, t \geq 0\}$ est un processus de naissance et de mort s'il satisfait les postulats suivants [89] :

1. $P_{i,i+1}(\Delta t) = \lambda_i \Delta t + \theta(\Delta t)$, ($i \geq 0$) ;
2. $P_{i,i-1}(\Delta t) = \mu_i \Delta t + \theta(\Delta t)$, ($i \geq 1$) ;
3. $P_{i,i}(\Delta t) = 1 - [\lambda_i + \mu_i + \theta(\Delta t)]$, ($i \geq 0$).

Un processus de naissance et de mort est un processus markovien pour lequel à partir d'un état i , deux transitions seulement sont possibles : le passage à l'état $(i+1)$ "naissance" avec un taux λ_i ou à l'état $(i-1)$ "mort" avec un taux μ_i . Le processus peut aussi rester dans le même état i .

2.5.5 Chaînes de Markov

A. A. Markov fut membre de l'école mathématique de Saint-Pétersbourg fondée par Tchebychev [94]. Leurs travaux sur la théorie des probabilités font preuve d'une rigueur mathématique peu commune dans le domaine à leur époque (entre 1880 et 1922). Les travaux de recherche de A. A. Markov l'on amené indirectement à concevoir les chaînes de Markov finies.

Ces chaînes ont fait leur première apparition dans un article publié en 1906. Kolmogorov a généralisé ce concept en introduisant, en 1931, les chaînes de Markov à temps continu et en 1936 les chaînes de Markov avec un nombre infini d'états.

En 1950, Feller fut le premier à unifier l'ensemble de ces travaux et les a rendus très populaires pour modéliser toutes sortes de phénomènes naturels.

Définition 2.2. Une suite $\{X_t, t=0, 1, 2, \dots\}$ est dite possédant une propriété Markovienne ou simplement c'est une chaîne de Markov, si pour chaque instant t , on a [21] :

$$P[X_{t+1} = j / X_0 = a, X_1 = b, \dots, X_t = i] = P[X_{t+1} = j / X_t = i], \quad t > 0.$$

Cette propriété signifie qu'étant donné l'ensemble des états passés et présents du système, la probabilité d'un état future quelconque de ce système est indépendante de son état passé et dépend seulement de son état actuel. Une chaîne de Markov homogène est définie par sa matrice de transition $P = [p_{ij}]_{ij}$ où :

$$p_{ij} = P[X_{t+1} = j / X_t = i], \quad \forall t \in T.$$

Les p_{ij} constituent une matrice carrée d'ordre n , avec n le nombre d'états. Les éléments p_{ij} vérifient la propriété de normalisation :

$$\sum_{j=1}^n p_{ij} = 1, \quad \forall i = \overline{1, n}.$$

Analyse des CMTD

Pour analyser les CMTD, il faut introduire quelques définitions et propriétés dont les détails peuvent être consultés dans [21] :

Définition 2.3. (Ensemble fermé d'états)

Un ensemble C d'états est fermé si :

$$\forall i \in C \text{ et } \forall j \in C, \quad p_{ij} = 0.$$

Proposition 2.1.

$$\forall n > 0, \forall i \in C, \text{ on a : } \sum_{j \in C} p_{ij}^n = 1.$$

I.e. au bout de n transitions, on est toujours dans C .

Définition 2.4. (Chaîne irréductible)

Une chaîne est irréductible si elle ne contient aucun sous-ensemble fermé autre que celui de tous ses états.

Définition 2.5. (Périodicité)

Un état E_i est dit périodique de période $d(i)$ si :

$$d(i) = \text{pgcd}\{n/p_{ii}^n > 0\} > 1.$$

Si $d(i) = 1$, E_i est dit apériodique.

Proposition 2.2. Une chaîne de Markov finie, apériodique et irréductible est ergodique.

Définition 2.6. Stationnarité

Une chaîne de Markov admet une distribution stationnaire π si elle est ergodique. Le vecteur π est solution du système d'équations linéaires suivant :

$$\begin{cases} \pi = \pi.P; \\ \sum_{i \in E} \pi_i = 1. \end{cases}$$

Définition 2.7. Chaîne de Markov à temps continu (CMTC)

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

- l'espace d'états S est dénombrable ;
- le temps d'observation est de nature continue ;
- le processus vérifie la propriété de Markov :

$$P[X(t_n) = j / X(t_{n-1}) = i_{n-1}, \dots, X(t_0) = i_0] = P[X(t_n) = j / X(t_{n-1}) = i_{n-1}], \\ \forall t_0 < t_1 < \dots < t_n, n \in \mathbb{N}$$

Une chaîne de Markov homogène à temps continu est définie par ses probabilités de transition :

$$p_{ij} = P [X(s+t) = j / X(s) = i] \quad \forall s \geq 0.$$

C'est à dire que les probabilités $P[X(t_n) = j / X(t_n - 1) = i]$ ne dépendent pas des instants d'observations t_n et t_{n-1} , mais uniquement de la durée $t_n - t_{n-1}$ qui sépare les deux observations.

Proposition 2.3. *Dans une chaîne de Markov ergodique, le vecteur π des probabilités stationnaires existe et est l'unique solution du système matriciel suivant :*

$$\begin{cases} \pi \cdot Q = 0; \\ \sum_{i \in E} \pi_i = 1; \end{cases}$$

où E est l'espace d'états de cette chaîne et Q est son générateur infinitésimal.

2.6 Modélisation des Systèmes de Files d'Attente

L'étude mathématique d'un système d'attente se fait le plus souvent par l'introduction d'un processus stochastique qui le décrit de façon appropriée. Le processus souvent considéré est le processus donnant le nombre de clients dans le système à l'instant t , " $\{X(t)\}_{t \geq 0}$ ". En fonction de ce processus descriptif, on cherche à calculer :

◇ Le régime transitoire du processus stochastique, défini par les probabilités d'état :

$$p_n = P[X(t) = n].$$

Les fonctions $p_n(t)$ dépendent de l'état initial ou de la distribution initiale du processus.

◇ Le régime stationnaire du processus stochastique, défini par :

$$p_n = \lim_{t \rightarrow \infty} p_n(t) = P[X = n], \forall n \in \mathbb{N}.$$

2.6.1 Modèles de files d'attente Markoviens

Les Modèles Markoviens de files d'attente sont des systèmes où les deux quantités stochastiques principales "les temps des inter-arrivée" et "la durée de service" sont des variables aléatoires indépendantes, exponentiellement distribuées. La propriété *sans mémoire* de la loi exponentielle facilite l'analyse de ces modèles.

2.6.2 Modèle de files d'attente non Markovien

En s'écartant de l'hypothèse d'exponentialité de l'une ou des deux quantités stochastiques "les temps des inter-arrivées" et "les durées de service" ou bien en introduisant des paramètres supplémentaires spécifiques au modèle tels que : la priorité, les rappels, les blocages, ... , on n'aura plus de processus Markovien. Ce qui rend l'analyse du modèle très délicate, voire impossible.

2.7 Quelques systèmes d'attente classiques

2.7.1 Système M/M/1

Ce système est composé d'une file de capacité infinie et d'un unique serveur. La discipline de service de la file est FIFO et le processus des arrivées est poissonnien de taux λ , de même la durée de service d'un client est une variable aléatoire exponentielle de taux μ .

L'état de ce système est complètement décrit par le processus $\{n(t)\}_{t \geq 0}$ qui donne le nombre de clients dans le système à l'instant t . $\{n(t)\}_{t \geq 0}$ est un processus stochastique à espace d'état discret et à temps continu. Puisque la file est à capacité illimitée, l'espace d'état E est infini : $E = \mathbb{N}$. Lorsqu'à un instant donné t il y a $n(t) > 0$ clients dans le système deux évènements peuvent se produire : l'arrivée d'un nouveau client ou le départ d'un client ayant terminé son service.

A l'état stationnaire, on obtient les probabilités suivantes :

$$p_n = p_0 \left(\frac{\lambda}{\mu}\right)^n = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^n, \quad n = 0, 1, 2, \dots$$

Avec $p_0 = 1 - \frac{\lambda}{\mu}$.

2.7.2 Système M/M/1/K

Considérons un système à serveur simple à la file M/M/1 excepté que la capacité de la file d'attente est finie. On a donc toujours les hypothèses suivantes : le processus d'arrivées des clients dans la file est un processus de Poisson de taux λ et le temps de service d'un client est une variable aléatoire exponentielle de taux μ . Soit K la capacité de la file d'attente : c'est le nombre maximal de clients qui peuvent être présents dans le système, soit en attente, soit en service. Quand un client arrive alors qu'il y a déjà K

clients présents dans le système, il est perdu, dans ce cas le système est dit système avec refus [4] ou bien le système se bloque.

Soit la description d'état suivante : $\{n(t)\}_{t \geq 0}$ le nombre de clients dans le système à l'instant t . L'espace d'état E est fini : $E = \{0, 1, \dots, K\}$.

Les hypothèses données correspondent au formalisme suivant :

$$\lambda_k = \begin{cases} \lambda, & \text{si } k < K; \\ 0, & \text{si } k \geq K; \end{cases}$$

$$\mu_k = \mu, \quad \text{si } k=1, \dots, K$$

Ainsi, la distribution stationnaire du système est donnée par :

$$p_k = \begin{cases} 0, & \text{si } k > K; \\ \rho^k p_0, & \text{si } k \leq K; \end{cases} \quad \rho = \frac{\lambda}{\mu};$$

$$p_0 = \left[1 + \sum_{k=1}^K \rho^k \right]^{-1} = \frac{1 - \rho}{1 - \rho^{K+1}}.$$

2.7.3 Système M/M/c

On considère un système identique à la file M/M/1 excepté qu'il comporte c serveurs. On conserve les hypothèses : processus d'arrivée des clients poissonien de taux λ et les temps de service sont exponentiels de taux $\mu_i, i = \overline{1, c}$.

La description d'état la plus simple à considérer est $\{n(t)\}_{t \geq 0}$ le nombre de clients dans le système à l'instant t . Si à un instant donné, $n(t)$ est inférieur ou égal à c , cela signifie que tous les clients sont en service. Si $n(t)$ est supérieur à c , alors c clients sont en service et les $(n(t) - c)$ restant sont dans la file d'attente. L'espace d'états E est infini : $E = \{0, 1, \dots\}$.

Lorsque le processus est dans un état $n < c$, tous les clients sont en service et sont donc susceptibles de quitter la file au bout d'un "temps exponentiel" de taux μ_k avec :

$$\mu_k \equiv \mu \min(k, c);$$

La distribution stationnaire du système M/M/c est alors donnée par :

$$p_k = \begin{cases} \frac{(\lambda/\mu)^k}{k!} p_0, & \text{si } k \leq c; \\ \frac{(\lambda/\mu)^k}{c! c^{k-c}} p_0 & \text{si } k \geq c; \end{cases}$$

où :

$$p_0 = \left[\sum_{k=0}^c \frac{(\lambda/\mu)^k}{k!} + \frac{(\lambda/\mu)^{c+1}}{c!(c - \lambda/\mu)} \right]^{-1}.$$

2.7.4 Système M/M/ ∞

On considère un système composé d'un nombre illimité de serveurs indépendants les uns des autres. Dès qu'un client arrive, il rentre instantanément en service. Cette file particulière ne comporte donc pas de file d'attente. Le processus d'arrivée des clients est poissonien de taux λ et les temps des services sont exponentiels de taux $\mu_i, i = \overline{1, \infty}$. Ce système est complètement décrit par la chaîne de Markov à temps continu.

Pour ce système, on a :

$$\begin{aligned} \lambda_k &= \lambda, \quad (k \geq 0); \\ \mu_k &= k\mu, \quad (k \geq 1); \end{aligned}$$

La distribution stationnaire de ce système d'attente est donnée par :

$$p_n = \left(\frac{\lambda}{\mu}\right)^n \frac{e^{-\lambda/\mu}}{n!}.$$

2.8 Systèmes avec priorités

Il existe des situations concrètes où l'on peut faire une distinction entre différentes classes des clients, où on est amené à imposer des priorités d'utilisation de service. Dans ce cas, on associe à chaque classe un indice de priorité $i, (1 \leq i \leq k)$ caractérisant son importance. En effet, la discipline, accordée à la manière avec laquelle le serveur sélectionne le client à servir et comment le servir, est dite discipline prioritaire de service. On a deux catégories principales de disciplines prioritaires :

- Exogène : elle ne doit dépendre que de la nature de la classe de priorité à laquelle le client appartient.
- Endogène : la décision sera basée globalement ou partiellement sur d'autres considérations relatives à l'état du système, par exemple la nature de la classe du client dernièrement servi ou du temps d'attente des clients présents dans le système.

Peu de résultats sont connus sur la discipline prioritaire endogène. Avec la discipline exogène, la décision à prendre pour choisir le prochain client à servir dépend de la classe

prioritaire associée à ce client, i.e. un client de la $i^{\text{ème}}$ classe est toujours sélectionné pour être servi avant un autre client de priorité inférieure. Cependant, si un client de classe j est en service lors de l'arrivée d'un client de classe i , plus prioritaire, alors différentes alternatives peuvent se présenter suivant les disciplines suivantes :

- 1- **Priorité absolue** : Le service du client de la $j^{\text{ème}}$ classe est immédiatement interrompu.
- 2- **Priorité relative** : Le service du client de la $j^{\text{ème}}$ classe continue sans être interrompu.
- 3- **Priorité libre** : Le serveur peut choisir librement d'utiliser la priorité relative ou la priorité absolue suivant le temps écoulé de la durée de service du client de classe j .

La priorité absolue peut aussi se diviser en deux catégories :

- **Priorité absolue conservatrice** : Le client interrompu reprend son service à partir du moment où il est interrompu, i.e. il conserve le bénéfice de la proportion du service effectuée avant qu'il soit interrompu.
- **Priorité absolue non conservatrice** : Le client interrompu doit refaire tout son service comme si c'est sa première arrivée au service, i.e. il ne conserve rien de la partie de service qu'il a effectuée juste avant son interruption.

Une autre classification importante des systèmes avec priorité relative à source(s) finie(s) sera exposée par la suite.

2.8.1 Priorité relative

La priorité relative a été introduite par Cobham [24], puis elle a été étudiée par plusieurs chercheurs, Jaiswal [54], Holloy [48], Dressin et Reich [30], L. Bouallouche et D. Aïssani [16], K. Thiruvengadam [96]. Dans ce qui suit, nous présentons quelques systèmes avec priorité relative.

2.8.2 Système $M_R/G_R/1$ avec priorité relative

Ce système représente une file d'attente avec priorité relative à R classes de clients. La classe 1 a la haute priorité et R a la faible priorité. Les clients arrivent indépendamment les uns des autres suivant un flux poissonnien de taux λ_i où $i = \overline{1, R}$. Le taux d'arrivée total est donné par ($\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_R$). Le service se fait selon une loi générale, de fonction de répartition $B_k(\cdot)$, et de moyenne $\frac{1}{\mu_k}$ pour les clients de la classe k .

Soit $\rho_k = \frac{\lambda_k}{\mu_k}$ est la condition d'ergodicité géométrique, elle est donnée par $\rho_1 + \rho_2 + \dots + \rho_k < 1$. Le temps d'attente W_k du dernier client arrivé C_k de classe k , avant qu'il entre en service pour la première fois, est donné comme suit [98] :

$$W_k = W_0 + W_{1,k} + W_{2,k}, \quad k = 1, \dots, R; \quad (2.2)$$

où :

- W_0 est le temps résiduel du client en cours de service ;
- $W_{1,k}$ est le temps de service de tous les clients qui sont déjà présents, à l'arrivée du client C_k et qui ont une priorité égale ou supérieure à celle de C_k ;
- $W_{2,k}$ est le temps de service de tous les clients qui sont arrivés pendant le temps d'attente du client C_k et qui ont une priorité supérieure à celle de C_k .

- **Le temps moyen résiduel du client en cours de service :**

Le temps moyen résiduel du client en cours de service est donné par :

$$E[W_0] = \sum_{i=1}^R \frac{\lambda_i E[B_i^2]}{2}. \quad (2.3)$$

On remarque que, le temps résiduel moyen de service d'un client dépend des moments d'ordre deux des distributions des temps de service.

- **Le temps moyen de service des clients de classes prioritaires C_i , présents dans le système, avant l'arrivée du client C_k avec $k \geq i$**

Supposons maintenant qu'un client C_k de classe k arrive, et soit $N_{i,k}$ le nombre de clients de classe i qui sont déjà présents dans la file et qui sont servis avant C_k . On a donc :

$$E[W_{1,k}] = \sum_{i=1}^k E[N_{i,k}] \cdot E[B_i], \quad (2.4)$$

Grâce à la formule de Little on obtient :

$$E[N_{i,k}] = \lambda_i E[W_i], \quad (2.5)$$

où W_i est le temps d'attente dans la file pour les clients de classe i , donc :

$$\begin{aligned} E[W_{1,k}] &= \sum_{i=1}^k E[N_{i,k}] \cdot E[B_i], \\ &= \sum_{i=1}^k \lambda_i E[W_i] \cdot E[B_i], \\ &= \sum_{i=1}^k \rho_i E[W_i], \quad k = 1, \dots, R. \end{aligned} \quad (2.6)$$

- Le temps moyen de service de tous les clients de classe C_i qui sont arrivés pendant le temps d'attente du client C_k , avec $k \geq i$.

Soit $M_{i,k}$ le nombre de clients de classe i , avec $k \geq i$ arrivant durant le temps d'attente W_k du client C_k . En utilisant la formule de Little, on exprime $E[M_{i,k}]$ par :

$$E[M_{i,k}] = \lambda_i E[W_k]. \quad (2.7)$$

Donc on aura :

$$E[W_{2,k}] = \sum_{i=1}^{k-1} E[M_{i,k}] E[B_i], \quad (2.8)$$

$$= \sum_{i=1}^{k-1} \lambda_i E[W_k] E[B_i], \quad (2.9)$$

$$= E[W_k] \sum_{i=1}^{k-1} \rho_i, \quad k = 1, \dots, R. \quad (2.10)$$

- **Le temps moyen d'attente d'un client de classe k**

En incluant les trois résultats trouvés dans (2.3), (2.6) et (2.10) dans l'expression de base de $E[W_k]$ on obtiendra :

$$\begin{aligned} E[W_k] &= E[W_0] + E[W_{1,k}] + E[W_{2,k}] \\ &= E[W_0] + \sum_{i=1}^{k-1} \rho_i E[W_i] + E[W_k] \sum_{i=1}^k \rho_i. \end{aligned} \quad (2.11)$$

De cette dernière équation, on trouve :

$$E[W_k] = \frac{1}{1 - \sum_{i=1}^k \rho_i} \left(E[W_0] + \sum_{i=1}^{k-1} \rho_i E[W_i] \right). \quad (2.12)$$

Pour $k = 1, \dots, R$, on obtient un système d'équations linéaires, qu'on peut résoudre d'une manière récursive. On note :

$$\sigma_k = \sum_{i=1}^k \rho_i. \quad (2.13)$$

Ainsi pour $k = 1$, on aura :

$$E[W_1] = \frac{E[W_0]}{1 - \sigma_1}. \quad (2.14)$$

Pour $k = 2$ on obtient :

$$\begin{aligned} E[W_2] &= \frac{1}{1 - \sigma_2} \left(E[W_0] + \rho_1 \frac{E[W_0]}{1 - \sigma_1} \right), \\ &= \frac{E[W_0]}{1 - \sigma_2} \left(\frac{1 - \rho_1 + \rho_1}{1 - \sigma_1} \right), \\ &= \frac{E[W_0]}{(1 - \sigma_1)(1 - \sigma_2)}. \end{aligned} \quad (2.15)$$

Il est facile à présent de prouver d'une manière récursive que pour $k < R$:

$$\begin{aligned} E[W_k] &= \frac{E[W_0]}{(1 - \sigma_k)(1 - \sigma_{k-1})}, \\ &= \frac{\sum_{i=1}^R \lambda_i E[B_i^2]}{2(1 - \sum_{i=1}^k \rho_i)(1 - \sum_{i=1}^{k-1} \rho_i)}, \quad k = \overline{1, R}. \end{aligned} \quad (2.16)$$

- **Le temps moyen de séjour d'un client de classe k dans le système :**

$$E[S_k] = E[W_k] + E[B_k] = \frac{\sum_{i=1}^R \lambda_i E[B_i^2]}{2(1 - \sum_{i=1}^k \rho_i)(1 - \sum_{i=1}^{k-1} \rho_i)} + E[B_k], \quad k = \overline{1, R}. \quad (2.17)$$

Enfin, à l'aide de la formule de Little, on peut obtenir :

a) Nombre moyen de clients de classe k dans la file :

$$Q_k = \lambda_k E[W_k] = \lambda_k \left(\frac{\sum_{i=1}^R \lambda_i E[B_i^2]}{2(1 - \sum_{i=1}^k \rho_i)(1 - \sum_{i=1}^{k-1} \rho_i)} \right), \quad \overline{1, R}. \quad (2.18)$$

b) Nombre moyen de clients de classe k dans le système

$$L_k = \lambda_k E[S_k] = \lambda_k \left(\frac{\sum_{i=1}^R \lambda_i E[B_i^2]}{2(1 - \sum_{i=1}^k \rho_i)(1 - \sum_{i=1}^{k-1} \rho_i)} \right) + \rho_k, \quad k = \overline{1, R}. \quad (2.19)$$

2.8.3 Système $M_2/G_2/1$ avec priorité relative

Ce système est un cas particulier des systèmes $M_R/G_R/1$ décrit précédemment. En effet, dans ce système, deux classes de clients de priorité relative y circulent. Ces classes arrivent selon un processus de poisson de paramètres λ_1 et λ_2 . La distribution de service est générale de fonction de répartition $\beta_i, i = \overline{1, 2}$ et de moyenne $\frac{1}{\mu_1}$ et $\frac{1}{\mu_2}$. L'adaptation des formules (2,11)et (2,18) à ce système à deux classes de priorités, nous permet d'avoir :

Le temps moyen d'attente d'un client prioritaire dans la file :

$$E[W_1] = \frac{(\lambda_1 E[B_1^2] + \lambda_2 E[B_2^2])}{2(1 - \rho_1)}. \quad (2.20)$$

Le temps moyen d'attente d'un client non prioritaire dans la file :

$$E[W_2] = \frac{(\lambda_1 E[B_1^2] + \lambda_2 E[B_2^2])}{2(1 - \rho_1 - \rho_2)(1 - \rho_1)}. \quad (2.21)$$

Le temps moyen de séjour d'un client prioritaire dans le système :

$$E[S_1] = \frac{(\lambda_1 E[B_1^2] + \lambda_2 E[B_2^2])}{2(1 - \rho_1)} + E[B_1]. \quad (2.22)$$

Le temps moyen de séjour d'un client non prioritaire dans le système :

$$E[S_2] = \frac{(\lambda_1 E[B_1^2] + \lambda_2 E[B_2^2])}{2(1 - \rho_1 - \rho_2)(1 - \rho_1)} + E[B_2]. \quad (2.23)$$

2.8.4 Système $M_2/M_2/1$ avec priorité relative

En considérant les durées de service des clients prioritaires et non prioritaires dans le système $M_2/G_2/1$, décrit dans la section précédente, de distribution exponentielle, alors les deux types de clients arrivent suivant un processus poissonnien de taux λ_i , où $i = 1, 2$ qui correspond à chaque classe. Le service se fait suivant la loi exponentielle de taux μ_i pour $i = \overline{1, 2}$. les états de la chaîne de Markov associés à ce système dépendent de n_1, n_2, r où :

n_1 = Le nombre de clients de classe 1 dans le système à l'instant t .

n_2 = Le nombre de clients de classe 2 dans le système à l'instant t .

$$r = \begin{cases} 1 & \text{si le client en service est de classe 1,} \\ 2 & \text{si le client en service est de classe 2.} \end{cases}$$

La condition d'ergodicité géométrique est donnée par $\rho = \lambda/\mu < 1$, avec $\lambda = \lambda_1 + \lambda_2$ et $\mu = \mu_1 + \mu_2$. Si nous limitons la file à trois $M_2/M_2/1/3$, ses probabilités de transition $P_{n,m,r}$ sont donnée par le système d'équation suivant :

$$\begin{aligned} P_{0,0}(\lambda_1 + \lambda_2) &= P_{0,1,2}\mu_2 + P_{1,0,1}\mu_1, \\ P_{0,1,2}(\lambda_2 + \mu_2 + \lambda_1) &= P_{0,0}\lambda_2 + P_{0,2,2}\mu_2 + P_{1,1,1}\mu_1, \\ P_{1,0,1}(\lambda_1 + \lambda_2 + \mu_1) &= P_{0,0}\lambda_1 + P_{2,0,1}\mu_1 + P_{1,1,2}\mu_2, \\ P_{1,1,2}(\lambda_1 + \lambda_2 + \mu_2) &= P_{1,2,2}\mu_2 + P_{2,1,1}\mu_1 + P_{0,1,2}\lambda_1, \\ P_{0,2,2}(\lambda_1 + \lambda_2 + \mu_2) &= P_{0,1,2}\lambda_2 + P_{1,2,1}\mu_1 + P_{0,3,2}\mu_2, \\ P_{2,0,1}(\lambda_1 + \lambda_2 + \mu_1) &= P_{1,0,1}\lambda_1 + P_{2,1,2}\mu_2 + P_{3,0,1}\mu_1. \end{aligned}$$

Ces probabilités de transition sont illustrées dans la figure suivante :

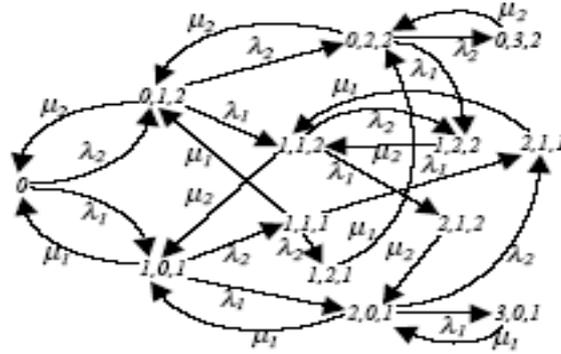


FIG. 2.2 – Les probabilités de transition de $M_2/M_2/1/3$ avec priorité relative

Nous remarquons que la description de la chaîne de Markov associée à $M_2/M_2/1/3$ est compliquée malgré qu'on a que trois clients seulement. Si nous généralisons pour le système $M_2/M_2/1/\infty$, $P_{n,m,r}$ sera donnée par le système d'équation suivant :

$$\begin{aligned}
 0 &= -(\lambda_1 + \lambda_2)P_{0,0} + \mu_1 P_{1,0,1} + \mu_2 P_{0,1,2}, \\
 0 &= -(\lambda_1 + \lambda_2 + \mu_1)P_{1,0,1} + \lambda_1 P_{0,0} + \mu_1 P_{2,0,1} + \mu_2 P_{1,1,2}, \\
 0 &= -(\lambda_1 + \lambda_2 + \mu_1)P_{0,1,2} + \lambda_2 P_{0,0} + \mu_1 P_{1,1,1} + \mu_2 P_{0,2,2}), \\
 0 &= -(\lambda_1 + \lambda_2 + \mu_1)P_{m,0,1} + \lambda_1 P_{m-1,0,1} + \mu_1 P_{m+1,0,1} + \mu_2 P_{m,1,2}, \\
 0 &= -(\lambda_1 + \lambda_2 + \mu_2)P_{0,n,2} + \lambda_2 P_{0,n-1,2} + \mu_1 P_{1,n,1} + \mu_2 P_{0,n+1,2} \\
 0 &= -(\lambda_1 + \lambda_2 + \mu_2)P_{1,n,1} + \lambda_2 P_{1,n-1,1} + \mu_1 P_{2,n,1} + \mu_2 P_{1,n+1,2}, \\
 0 &= -(\lambda_1 + \lambda_2 + \mu_1)P_{m,n,1} + \lambda_1 P_{m-1,n,1} + \lambda_2 P_{m,n-1,1} + \mu_1 P_{m+1,n,1} + \mu_2 P_{m,n+1,2}; \quad m > 1, n > 0, \\
 0 &= -(\lambda + \mu)P_{m,n,2} + \lambda_1 P_{m,n-1,2}; \quad m > 0, n > 1.
 \end{aligned} \tag{2.24}$$

On constate que dans ce cas, le nombre d'états est très grand. Ceci compliquera l'obtention de la solution de ces équations de balance. Pour obtenir les distributions stationnaires $P_n(t)$ avec :

$$P_n = \sum_{m=0}^{n-1} (P_{n-m,m,1} + P_{n,n-m,2}) = (1 - \rho)\rho^n \quad (n > 0);$$

nous allons faire appel aux fonctions génératrices voir [45, 46, 98].

Nous allons introduire dans ce qui suit, une autre classification importante pour les systèmes d'attente prioritaires et qui dépend de la capacité des sources d'où émanent les clients.

2.9 Systèmes de files d'attente prioritaires markoviens avec sources finies

On avait souligné qu'une classification des systèmes prioritaires par rapport aux source(s) finie(s) est importante à considérer. Ainsi, si dans un S.F.A. avec priorités toutes les sources sont infinies, on parlera de S.F.A. avec priorité et source infinie. Cependant, si une ou plusieurs sources sont finies, on parlera de S.F.A. avec priorité et sources finies. Ce dernier modèle peut à son tour être classifié en deux catégories :

- (a) Un S.F.A. avec priorité et multiples sources finies, dans lequel chaque classe de clients émane d'une source finie différente.
- (b) Un S.F.A. avec priorité et source finie unique, dans lequel deux ou plusieurs classes de clients émanent de la même source finie.

Dans les chapitres à venir, il sera question d'analyse de systèmes appartenant à ces deux dernières catégories, à savoir les modèles $M_2/M_2/1//N_1, \dots, N_k$ respectivement $M_2/M_2/1//N$. Dans ces systèmes, $K \geq 2$ classes de clients arrivent de N sources finies (respectivement une seule source finie) vers un serveur. Chaque client de la source est supposé demander le service après avoir séjourné dans la source un temps distribué exponentiellement avec une moyenne $1/\lambda_i$. Un client de la classe j est supposé prioritaire au client de la classe i si $i < j$. Le serveur commence à servir les clients de haute priorité et les temps de service sont distribués selon une loi exponentielle de taux μ_i . A l'intérieur de chaque classe, les clients sont servis selon la discipline (FIFO) [79].

L'analyse du système $M_2/M_2/1//N$ est délicate d'où la non existence de résultats analytiques.

Par contre, pour l'analyse des systèmes $M_k/M_k/1//N_1, \dots, N_k$ des approches ont été conçues. L'une de ces approches a été utilisée par Thiruvengadam [96], Jaiswal et Thiruvengadam [53], Jaiswal [54]. Ils ont incorporé une variable supplémentaire qui est le temps écoulé ou le temps qui reste pour terminer le service d'un client afin de décrire le système par une chaîne de Markov. Cependant, malgré que la discipline est markovienne, la résolution des équations qui régissent la longueur conjointe des files est très compliquée, même avec uniquement deux classes. La généralisation à plus de deux classes, présente d'immenses difficultés algébriques [54]. Une seconde approche concerne l'étude de ce modèle en utilisant la méthode de la chaîne induite aux instants de départ des clients de la station de service

En effet, ces instants constituent un ensemble de points de régénération, ainsi, le processus donnant le nombre de clients présents dans le système à ces instants constitue un processus de Markov.

Processus d'arrivées

Les clients émanants d'une source infinie arrivent au serveur à des instants :

$$\{t_0 < t_1 < \dots, < t_k < \dots\}$$

Les variables aléatoires $\tau_k = t_k - t_{k-1}, k \geq 1$ sont les inter-arrivées. On suppose que les τ_k forment une suite de variables aléatoires indépendantes, identiquement distribuées avec une distribution :

$$A(x) = P[\tau_k \leq x] = 1 - \exp(-\lambda x),$$

où λ est le taux des inter-arrivées. Il est facile de montrer, dans ce cas que la distribution de $n(t)$ "le nombre de clients qui arrivent dans le système à l'instant t " suit une distribution de Poisson donnée par :

$$P[n(t) = n] = \frac{\lambda t}{n!} \exp(-\lambda t).$$

Cependant, quand la source est finie, les arrivées sont définies comme suit :

On considère un client particulier appartenant à la source. Ce client après une attente τ'_1 dans la source, demande service pour la première fois. Si il n'est pas servi, il retourne à la source et attend une durée de temps τ'_2 avant de faire une seconde demande de service. En général, le client attend une durée de temps τ'_k dans la source avant de faire sa $k^{\text{ème}}$ demande de service. Soit $A(x) = P[\tau'_k \leq x]$. Au lieu de spécifier le processus d'entrée à travers la distribution des inter-arrivées, ce qui englobe la distribution des τ'_k et la capacité de la source, on spécifie le processus d'arrivées à travers la distribution des τ'_k seulement. Prenons à titre d'exemple un système où la source est de capacité N et soit $A(x) = 1 - \exp(-\lambda x)$ la distribution des inter-arrivées, alors la distribution de $n(t)$ sera donnée par :

$$P[n(t) = n] = \binom{N}{n} [1 - \exp(-\lambda t)]^n \exp[-(N - n)\lambda t]. \quad (2.25)$$

2.10 Systèmes de files d'attente prioritaires avec source finie unique

Dans le cas des systèmes avec priorité et sources finies multiples, chaque source génère seulement une seule sorte de clients. Cependant, si on considère l'exemple de N machines, où chacune d'elles peut s'arrêter de fonctionner suite à k types de pannes. Une machine qui s'arrête de fonctionner a une probabilité λ_i/λ d'avoir la $i^{\text{ème}}$ panne et sa réparation suit une loi de probabilité de densité $S_i(x)$. Ainsi, dans un système pareil, à n'importe quel instant il y aura n_i machines à cause de la $i^{\text{ème}}$ panne, donc le nombre total de machines tombées en panne sera $\sum n_i = n$. Il est clair que, à l'instar des modèles prioritaires à sources finies multiples, les arrivées des clients de la $i^{\text{ème}}$ classe affectent le taux des arrivées de toutes les classes. A cause de cette dépendance des taux des arrivées, les modèles prioritaires à source finie unique sont difficiles à analyser. La première tentative d'analyse de ce genre de modèles est réalisée par Benson et Cox [14] pour un service exponentiellement distribué et une discipline prioritaire relative.

2.11 Conclusion

Dans ce chapitre, nous avons donné en premier lieu, un bref aperçu sur quelques notions de bases sur la théorie des files d'attente, ainsi que quelques modèles particuliers de systèmes de files d'attente. En second lieu, un intérêt particulier est donné aux systèmes prioritaires. En effet, nous avons exposé une classification détaillée de ces systèmes. Une idée générale sur les approches adoptées pour l'analyse des systèmes prioritaires montre que l'obtention de résultats analytiques est compliquée, voire impossible [54]. Le recours à des méthodes algorithmiques approximatives s'impose. Ainsi, dans le prochain chapitre, nous proposons une modélisation de ces systèmes avec priorité relative et source(s) finie(s) via les réseaux de Petri stochastiques généralisés.

3

Analyse de systèmes prioritaires à l'aide des RdPSG

3.1 Introduction

Durant ces dernières décennies, un effort considérable a été consacré par beaucoup de praticiens et de chercheurs théoriciens à l'évaluation des performances des systèmes prioritaires. L'importance accordée à ces systèmes est due à leurs utilisations dans la vie courante. En effet, les systèmes prioritaires peuvent se rencontrer dans les systèmes informatiques, les systèmes de télécommunication, les transports, etc.

Malgré leurs utilisations intensives, peu de résultats analytiques sont connus sur ces systèmes. Ceci est dû principalement à la complexité de leurs structures comme on l'a souligné dans le chapitre précédent. Ainsi, leur étude passe par des méthodes analytiques approximatives, des méthodes algorithmiques ou par la simulation.

Dans ce chapitre, nous ferons usage d'une autre approche qui est l'approche des RdPSG pour l'analyse de ces systèmes prioritaires. En effet, il sera question d'une étude d'un système à source finie unique $M_2/M_2/1//N$, puis d'une étude de systèmes à sources finies multiples $M_2/M_2/1//(N_1, N_2)$ et $M_3/M_3/1//(N_1, N_2, N_3)$

3.2 Analyse des RdPSG

Après la construction du modèle de *RdPSG* associé à un système, on passe à son analyse qui consiste à définir d'une part ses propriétés qualitatives et d'autre part, à calculer ses paramètres de performances quantitatifs.

On ne peut pas faire l'analyse quantitative sans l'analyse qualitative. En effet, l'évaluation des performances à l'état stationnaire nécessite l'ergodicité du modèle, ce qui revient à vérifier certaines propriétés qualitatives.

L'analyse qualitative : consiste à vérifier les propriétés que nous avons présentées dans le chapitre 1 tels que : la vivacité, la bornitude, les états d'accueils. . .etc. Alors que l'analyse quantitative : consiste à calculer les probabilités stationnaires et les indices de performances.

3.3 Notions préliminaires

Les notions suivantes relatives aux RdPSG nous seront utiles dans la suite de notre analyse [28, 31, 99].

Proposition 3.1. *Le graphe de marquage associé au RdPSG est isomorphe à une chaîne de Markov à temps continu et à espace d'états discret .*

Le traitement de la chaîne de Markov générée à partir d'un graphe des marquages d'un RdPSG permettra d'effectuer une analyse quantitative du comportement dynamique du système.

Théorème 3.1. *Un RdPSG borné et tel que son graphe des marquages accessibles est fortement connexe est ergodique.*

Théorème 3.2. *Un RdPSG borné est ergodique s'il admet le marquage initial comme état d'accueil .*

3.4 La construction de la chaîne de Markov

A partir d'un *RdPSG*, on peut générer l'arbre d'admissibilité qui représente les séquences de franchissements des transitions et qui nous permet de visionner tous les marquages possibles à partir du marquage initial. Cet arbre est taillé quand un précédent marquage est obtenu. L'étiquette sur chaque arête orientée représente la transition tirée qui a produit le marquage suivant. La chaîne de Markov à temps continu peut être construite à partir de ce graphe de marquage [84].

Le graphe de Markov des états tangibles est obtenue après l'élimination des marquages évanescents. Ainsi, les états de la chaîne de Markov associés au modèle étudié sont les marquages tangibles de l'arbre. Les marquages évanescents sont fusionnés avec leurs successeurs (marquages tangibles).

Le principe d'élimination des marquages évanescents est le suivant : Soit M_b un marquage évanescent directement accessible à partir du marquage tangible M_a , et soit S l'ensemble des marquages tangibles accessibles à partir de M_b en passant par une séquence de marquages évanescents seulement. Dans ce cas, le marquage évanescent M_b et tous ceux atteignable à partir de M_b par le franchissement de transitions immédiates peuvent être éliminés en introduisant simplement un arc reliant directement M_a à M_c $\forall M_c \in S(M_c \neq M_a)$, et ce en calculant convenablement la probabilité de transition de M_a à M_c [3, 36].

3.5 Evaluation des indices de performances

Une fois le modèle obtenu, nous pouvons vérifier ses propriétés qualitatives, puis déduire son ergodicité pour faire l'analyse quantitative. Si le modèle est ergodic, alors la distribution de probabilité des marquages à l'état stationnaire existe et elle est unique. Nous notons par $\pi_i = (\pi_0, \pi_1, \dots, \pi_n)$ cette distribution des probabilités où π_i est la probabilité que le processus est à l'état M_i .

Ayant π , plusieurs indices de performances peuvent être calculés. Parmi ces indices les plus importants on a [36, 84] :

• **Le taux moyen de franchissement d'une transition** ($\bar{\lambda}(t_i)$) :

On appelle le taux moyen (débit moyen) de franchissement d'une transition t_i , le nombre moyen de tirs de t_i en une unité de temps.

$$\bar{\lambda}(t_i) = \sum_{M_j \in E(t_i)} \lambda_i(M_j) \pi_j;$$

où :

- $E(t_i)$ est l'ensemble des marquages où la transition t_i est franchissable.
- $\lambda_i(M_j)$ est le taux de franchissement associé à la transition t_i dans le marquage M_j .

• **Le nombre moyen de jetons dans une place** ($n(p)$) :

Le nombre moyen de jetons dans une place p est donné par :

$$n(p) = \sum_{i: M_i \in E} M_i(p) \pi_i;$$

où :

- $M_i(p)$ est le nombre de jetons dans la place p pour le marquage M_i .
- E est l'ensemble des marquages accessibles.

• **Le temps moyen de séjour d'un jeton dans un sous-réseau** :

Le temps moyen de séjour (délai moyen) d'un jeton dans un sous réseau S d'un RdPSG à l'état stationnaire, peut être calculé en utilisant la formule de Little [65] :

$$E[T] = \frac{E[N]}{E[\beta]};$$

où :

- $E[N]$ est le nombre moyen de jetons dans S .
- $E[\beta]$ est le taux d'arrivée effectif des jetons dans S .

3.6 Modélisation de systèmes avec priorité relative à une seule source finie par les RdPSG

On considère une file d'attente avec priorité relative et source finie à un seul serveur $M_2/M_2/1//N$ dans laquelle arrivent deux classes de clients : clients non prioritaires et clients prioritaires.

Les deux types de clients arrivent indépendamment les uns des autres suivant un processus quasi-aléatoire de taux λ_1 et λ_2 respectivement.

Le service des clients non prioritaires et prioritaires se fait suivant la loi exponentielle de paramètre μ_1 et μ_2 respectivement.

La figure 3.1 représente une modélisation de ces systèmes par les RdPSG où la capacité de la population est représentée par le paramètre entier positif N qui apparaît comme marquage initial de la place P_A .

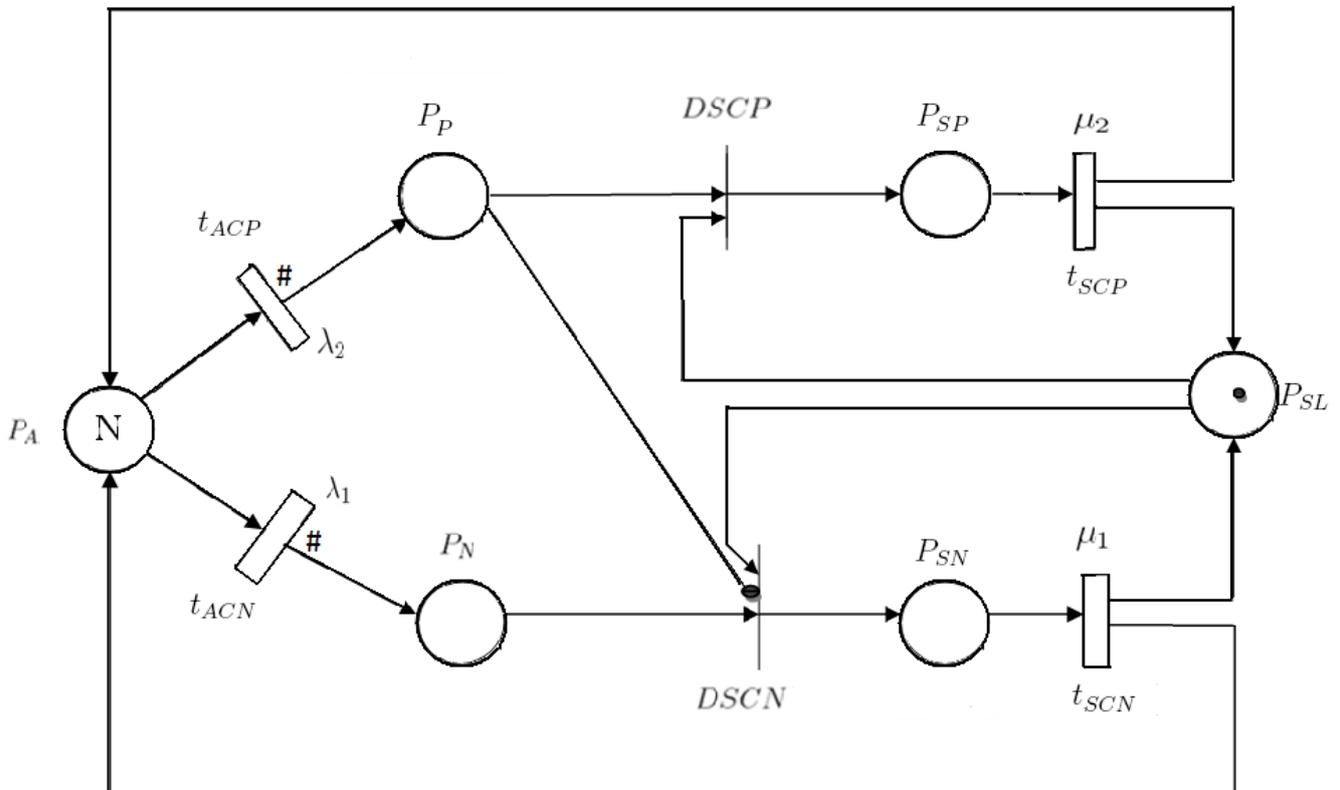


FIG. 3.1 – Le RdPSG modélisant le système $M_2/M_2/1//N$ avec priorité relative

Dans ce modèle on a :

- La place P_A contient les clients libres (source), représentés par N jetons. C'est-à-dire, aucun des N clients n'est arrivé dans le système ;
- La place P_N contient les clients non prioritaires ;
- La place P_p contient les clients prioritaires ;
- La place P_{SP} représente l'état 'Le serveur est occupé par le client prioritaire' ;
- La place P_{SN} représente l'état 'Le serveur est occupé par le client non prioritaire' ;
- La place P_{SL} représente l'état 'Le serveur est libre', représenté par un seul jeton.
- Le marquage initial du réseau est alors donné par :

$$M_o = (M(P_A), M(P_p), M(P_{SP}), M(P_N), M(P_{SN}), M(P_{SL})) = (N, 0, 0, 0, 0, 1).$$

Ceci, signifie que tous les clients sont initialement libres et que le serveur est disponible.

Remarque 3.1. *Les transitions immédiates sont représentées par des traits et les transitions temporisées par des rectangles. Les jetons restent comptabilisés dans les places en entrée d'une transition, tant que son franchissement n'est pas terminé.*

– Par la suite, il sera référé au modèle de RdPSG représenté dans la figure 3.1 par R_1 .

Lors du franchissement de la transition t_{ACN} , un jeton est enlevé de la place P_A et est déposé dans P_N . Le franchissement de t_{ACN} indique l'arrivée d'un client non prioritaire généré de la source. Les clients se comportent indépendamment les uns des autres et le taux de tir est dépendant du nombre de jetons dans la place P_A . Si on a N jetons dans la place P_A alors le taux de franchissement est égal à $N\lambda_1$. La condition de dépendance du taux de franchissement est représentée par le symbole \sharp placé à coté de la transition t_{ACN} .

Si le client est prioritaire, c'est la transition t_{ACP} qui sera franchie ainsi, la place P_p reçoit un jeton. La transition temporisée t_{ACP} est également à marquage dépendant alors le taux de franchissement est égal à $N\lambda_2$.

La transition immédiate $DSCN$ sera tirée instantanément si la place P_{SL} contient un jeton (qui correspond au serveur libre) et que la place P_p est vide (pas de clients prioritaires) et cela est exprimé par l'arc inhibiteur allant de la place P_p vers la transition $DSCN$ (conditionner la place P_p). Dans ce cas où $DSCN$ est tirée, un jeton est enlevé de la place P_{SL} et de P_N est déposé dans la place P_{SN} . Ce jeton représente le serveur qui est occupé par le client non prioritaire.

Par contre, si P_P contient un jeton, c'est la transition $DSCP$ qui sera tirée (car elle est prioritaire) à condition bien sûr que la place P_{SL} contient un jeton représentant le serveur libre. Dans ce cas, la place P_{SP} reçoit un jeton représentant le client prioritaire en service. D'autre part, quand la transition temporisée t_{SCP} est franchie (respectivement la transition temporisée t_{SCN}) qui détermine la fin de la période de service du client prioritaire (respectivement du client non prioritaire), un jeton est déposé dans P_A qui représente la condition que le client prioritaire (respectivement non prioritaire) devient libre et un autre jeton est déposé dans la place P_{SL} ce qui représentera la condition que le serveur est prêt à servir un autre client.

3.6.1 Analyse des performances de R_1

Le modèle R_1 proposé dans la figure 3.1 est borné, vivant, sans blocage et le marquage initial est un état d'accueil. Par conséquent, ce modèle admet un état stationnaire. Soit $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_n)$ sa distribution des probabilités des marquages à l'état stationnaire où π_i est la probabilité que le processus soit à l'état M_i .

Nous donnons dans ce qui suit les formules explicites de certains indices de performances du réseau R_1

- **Le taux moyen effectif des arrivées des clients non prioritaires :**

Ce taux est le nombre moyen de tir en une unité de temps :

$$\eta_N = \sum_{j \in SM_j} \lambda_1(M_j) \pi_j.$$

où :

- (SM_j) est l'ensemble des marquages où la transition t_{ACN} est franchissable.
- $\lambda_1(M_j)$ est le taux de franchissement associé à la transition t_{ACN} dans M_j .

- **Le taux moyen effectif des arrivées des clients prioritaires :**

Ce taux est donné par :

$$\eta_P = \sum_{j \in (SM_j)} \lambda_2(M_j) \pi_j;$$

où :

- (SM_j) est l'ensemble des marquages où la transition t_{ACP} est franchissable.
- $\lambda_2(M_j)$ est le taux de franchissement associé à la transition t_{ACP} dans M_j .

- **Le nombre moyen de clients non prioritaires dans la file :**

Ce nombre moyen est donné par la formule suivante :

$$\eta_{0N} = \sum_j M_j(P_N)\pi_j;$$

où :

$M_j(P_N)$ est le nombre de jetons dans la place P_N dans le marquage M_j . La somme dans cette formule se fait sur tous les marquages accessibles.

- **Le nombre moyen de clients prioritaires dans la file :**

De la même manière que précédemment, ce nombre moyen est donné par :

$$\eta_{0P} = \sum_j M_j(P_P)\pi_j;$$

où :

$M_j(P_P)$ est le nombre de jetons dans la place P_P dans le marquage M_j .

- **Le nombre moyen de clients non prioritaires dans le système :**

Il est donné par :

$$\eta_{SN} = \sum_j [M_j(P_N) + M_j(P_{SN})]\pi_j.$$

- **Le nombre moyen de clients prioritaires dans le système :**

Il est donné par :

$$\eta_{SP} = \sum_j [M_j(P_P) + M_j(P_{SP})]\pi_j.$$

- **Le temps moyen d'attente des clients non prioritaires :**

Le temps d'attente d'un client correspond à la durée de temps entre l'arrivée du client et son début de service. Il est obtenu à l'aide de la formule de Little :

$$W_N = \frac{\eta_{0N}}{\eta_N}.$$

- **Le temps moyen d'attente des clients prioritaires :**

Le temps moyen d'attente pour les clients prioritaires est donné par :

$$W_P = \frac{\eta_{0P}}{\eta_P}.$$

- Le temps moyen de réponse des clients non prioritaires :

$$\tau_N = \frac{\eta_{SN}}{\eta_N}.$$

- Le temps moyen de réponse des clients prioritaires :

$$\tau_P = \frac{\eta_{SP}}{\eta_P}.$$

3.6.2 Application sur le système $M_2/M_2/1//3$

On considère une file d'attente $M_2/M_2/1//3$ avec priorité relative et source finie de taille 3. Les deux types de clients (non prioritaires et prioritaires) arrivent indépendamment les uns des autres suivant un processus quasi-aléatoire de taux λ_1 et λ_2 respectivement. Les taux des services des clients non prioritaires (resp. prioritaires) sont μ_1 (resp. μ_2). Le marquage initial est :

$$M_0 = (M(P_A), M(P_P), M(P_{SP}), M(P_N), M(P_{SN}), M(P_{SL})) = (3, 0, 0, 0, 0, 1).$$

Le graphe des marquages de ce RdPSG est donné dans la figure 3.2. Ce graphe est un arbre qui permet de calculer tous les marquages possibles à partir du marquage initial. Cet arbre représente les séquences de franchissement des transitions à partir du marquage initial. Il contient 5 marquages évanescents qui sont : $M_1, M_2, M_{15}, M_{16}, M_{17}$ et les autres marquages sont des marquages tangibles.

Rappelons que dans les marquages évanescents au moins une transition immédiate est sensibilisée et dans les marquages tangibles seulement les transitions temporisées sont sensibilisées.

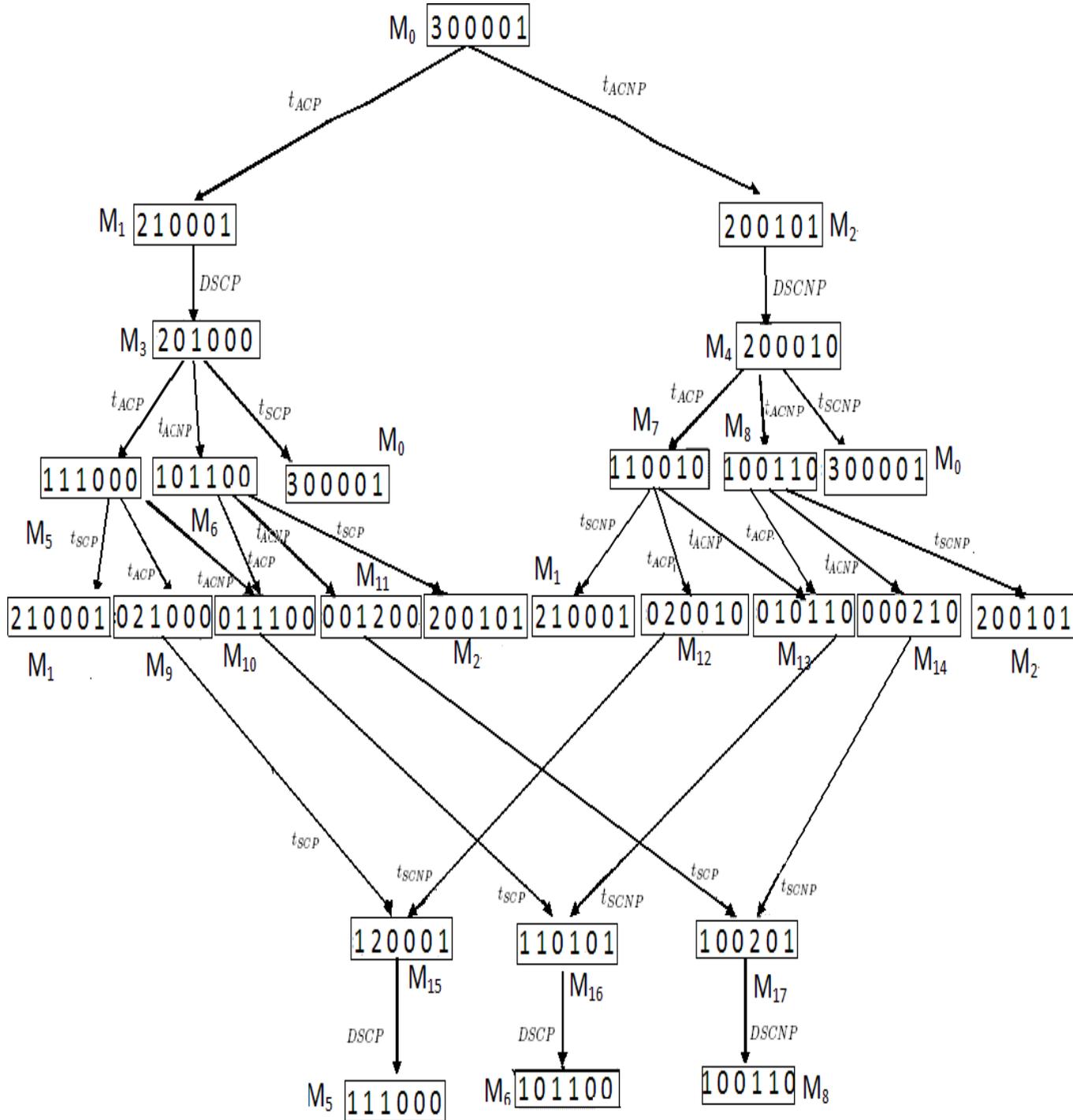


FIG. 3.2 – Graphe d'accessibilité associé à R_1

Maintenant, on peut construire la chaîne de Markov continue à partir de ce graphe des marquages. Les états de la *CMTC* sont les marquages tangibles, les marquages évanescents sont fusionnés avec leurs successeurs. Exemple : M_1 est fusionné avec M_3 ; M_2 avec M_4 ; M_{15} avec M_5 ; M_{16} avec M_6 et M_{17} avec M_8 . Les taux de transitions de la *CMTC* associée au graphe du système $M_2/M_2/1//3$ sont les taux de tir des transitions temporisées de R_1 avec $N = 3$. La figure 3.3 illustre cette chaîne de Markov.

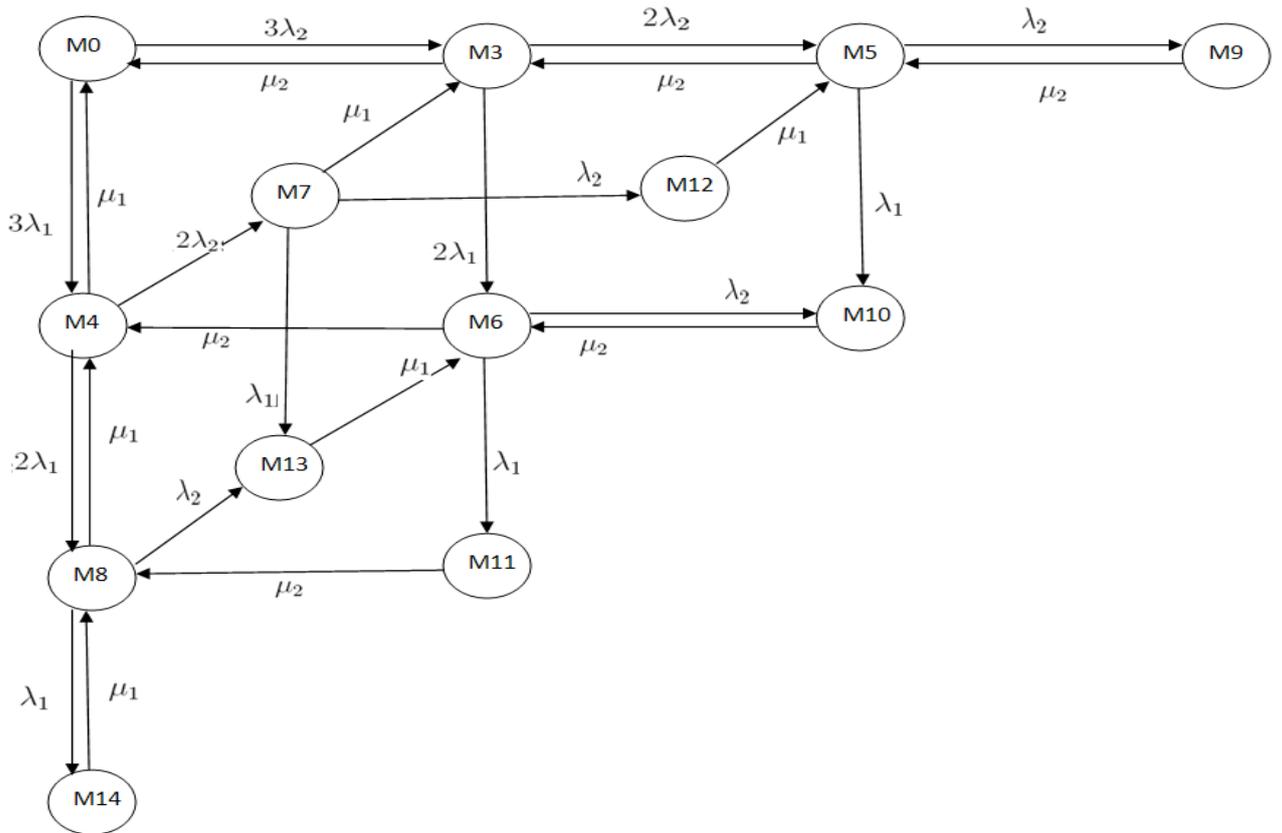


FIG. 3.3 – Chaîne de Markov réduite associée à R_1 pour $N = 3$

Le générateur infinitésimal Q de cette chaîne de Markov est donné par la matrice suivante :

Cette matrice carrée est de dimension $(s \times s)$, où s est le nombre fini de marquages tangibles du *RdPSG* qui regroupe l'ensemble des taux de transitions d'un marquage vers un autre.

Le vecteur des probabilités stationnaires $\pi = (\pi_0, \pi_3, \pi_4, \pi_5, \pi_6, \pi_7, \pi_8, \pi_9, \pi_{10}, \pi_{11}, \pi_{12}, \pi_{13}, \pi_{14})$ est la solution du système :

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

où :

π_i représente la probabilité stationnaire que le processus soit à l'état M_i .

Nous pouvons maintenant calculer les indices de performances :

- Le taux moyen effectif des arrivées des clients non prioritaires (η_N) :

$$\eta_N = \sum_{j \in (SM_j)} \lambda_1(M_j) \pi_j = 3\lambda_1\pi_0 + 2\lambda_1(\pi_3 + \pi_4) + \lambda_1(\pi_5 + \pi_6 + \pi_7 + \pi_8).$$

- Le taux moyen effectif des arrivées des clients prioritaires (η_P) :

$$\eta_P = \sum_{j \in (SM_j)} \lambda_2(M_j) \pi_j = 3\lambda_2\pi_0 + 2\lambda_2(\pi_3 + \pi_4) + \lambda_2(\pi_5 + \pi_6 + \pi_7 + \pi_8).$$

- Le nombre moyen de clients non prioritaires dans la file (η_{0N}) :

$$\eta_{0N} = \sum_j M_j(P_N) \pi_j = \pi_6 + \pi_8 + \pi_{10} + \pi_{13} + 2(\pi_{11} + \pi_{14}).$$

- Le nombre moyen de clients prioritaires dans la file (η_{0P}) :

$$\eta_{0P} = \sum_j M_j(P_P) \pi_j = \pi_5 + \pi_7 + \pi_{10} + \pi_{13} + 2(\pi_9 + \pi_{12}).$$

- Le nombre moyen de clients non prioritaires dans le système (η_{SN}) :

$$\begin{aligned} \eta_{SN} &= \sum_j [M_j(P_N) + M_j(P_{SN})] \pi_j. \\ &= \pi_4 + \pi_6 + \pi_7 + \pi_{10} + \pi_{12} + 2(\pi_8 + \pi_{13} + \pi_{11}) + 3\pi_{14}. \end{aligned}$$

- Le nombre moyen de clients prioritaires dans le système (η_{SP}) :

$$\begin{aligned} \eta_{SP} &= \sum_j [M_j(P_P) + M_j(P_{SP})] \pi_j. \\ &= \pi_3 + \pi_6 + \pi_7 + \pi_{11} + \pi_{13} + 2(\pi_5 + \pi_{10} + \pi_{12}) + 3\pi_9. \end{aligned}$$

- Le temps moyen d'attente des clients non prioritaires (W_N) :

$$W_N = \frac{\eta_{0N}}{\eta_N}.$$

- Le temps moyen d'attente des clients prioritaires (W_P) :

$$W_P = \frac{\eta_{0P}}{\eta_P}.$$

- Le temps moyen de réponse des clients non prioritaires (τ_N) :

$$\tau_N = \frac{\eta_{SN}}{\eta_N}.$$

- Le temps moyen de réponse des clients prioritaires (τ_P) :

$$\tau_P = \frac{\eta_{SP}}{\eta_P}.$$

Nous avons analysé le système $M_2/M_2/1//N$ avec plusieurs valeurs de N . Notre choix s'est porté sur le cas du système $M_2/M_2/1//3$ pour le présenter dans cette section. Néanmoins, le graphe de la chaîne de Markov associé à R_1 avec N quelconque a été obtenu d'une manière récursive (voir 3.4).

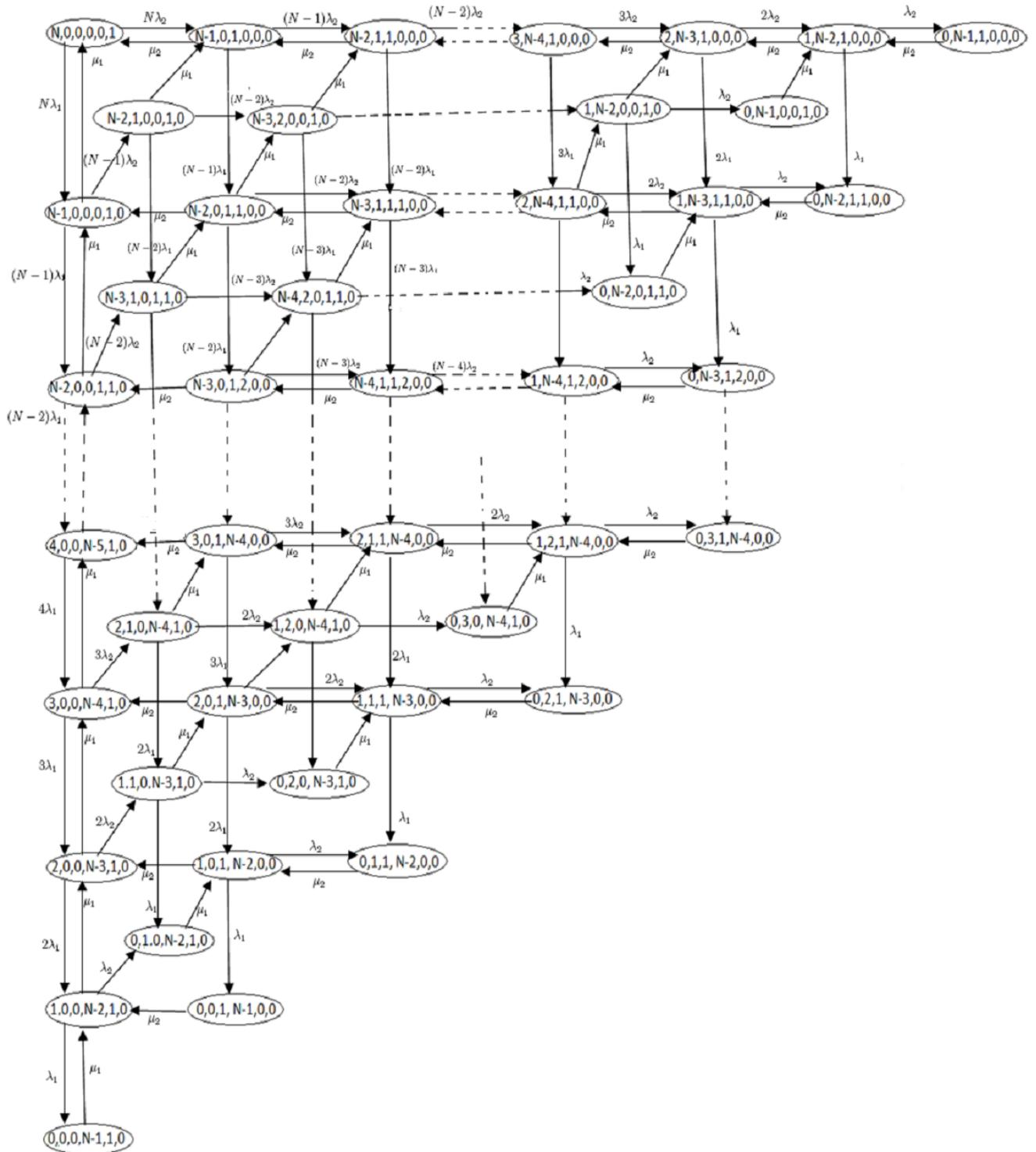


FIG. 3.4 – La chaîne de Markov réduite associée à R_1

3.7 Modélisation de systèmes avec priorité relative à deux sources finies par les RdPSG

On considère un système d'attente avec priorité relative à deux sources finies $M_2/M_2/1// (N_1, N_2)$ dans lequel arrivent deux classes de clients : Clients non prioritaires venant de la source N_1 et clients prioritaires venant de la source N_2 .

Les deux types de clients arrivent indépendamment les uns des autres suivant un processus quasi-aléatoire avec un taux de λ_1 et λ_2 respectivement. Le service des clients prioritaires et non prioritaires se fait suivant la loi exponentielle de paramètre μ_2 et μ_1 respectivement. La figure 3.5 représente une modélisation de ces systèmes par les RdPSG où la capacité de la population est représentée par les paramètres entiers positifs N_1 et N_2 qui apparaissent comme marquage initial des places P_B et P_A respectivement.

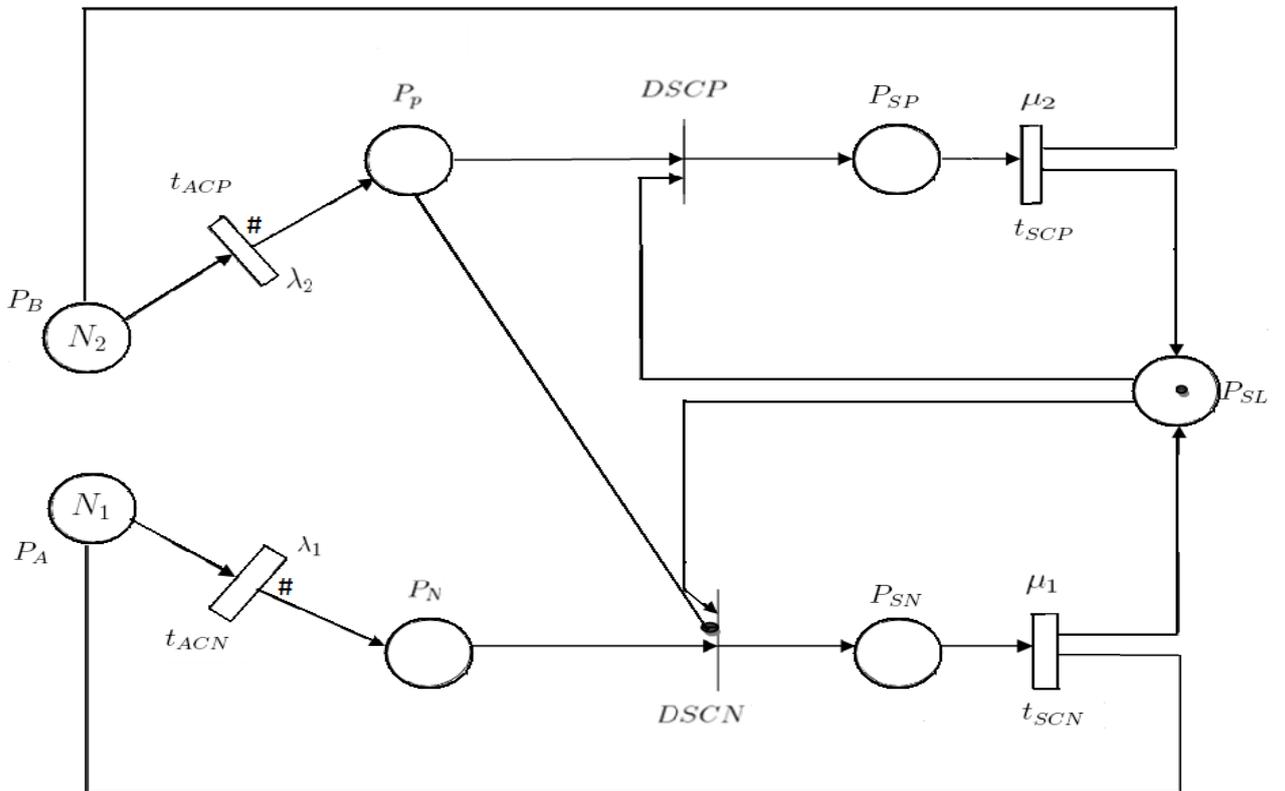


FIG. 3.5 – Le RdPSG modélisant le système $M_2/M_2/1// (N_1, N_2)$ avec priorité relative

Dans ce modèle :

- La place P_A contient les clients non prioritaires libres (source), représentés par N_1 jetons.
- La place P_B contient les clients prioritaires libres (source), représentés par N_2 jetons.
- Le marquage initial du réseau est :

$$M_o = (M(P_B), M(P_p), M(P_{SP}), M(P_A), M(P_N), M(P_{SN}), M(P_{SL})) = (N_2, 0, 0, N_1, 0, 0, 1)$$

Ce qui signifie que tous les clients sont initialement libres, et que le serveur est disponible.

Il sera référé à ce RdPSG par l'abréviation R_2 .

Le franchissement de la transition t_{ACN} représente l'arrivée d'un client non prioritaire généré d'une source. Ainsi, la place P_N reçoit un jeton. Les clients se comportent indépendamment les uns des autres et le taux de tir est dépendant du nombre de jetons dans la place P_A . Si on a N_1 jetons dans la place P_A , alors le taux de franchissement est égal à $N_1\lambda_1$.

Si le client est prioritaire, c'est la transition t_{ACP} qui sera franchie avec un taux $N_2\lambda_2$ et la place P_p reçoit un jeton.

A la fin de service du client prioritaire (respectivement non prioritaire), un jeton sera déposé dans la place P_B (respectivement P_A). Ce qui représente le fait que le client devient libre, ainsi, un autre jeton est déposé dans la place P_{SL} puisque le serveur devient libre.

Même principe de description et de calculs que ceux de la section 3.5 sont adoptés pour ces systèmes avec sources finies multiples. En effet, il suffit juste de subdiviser la source et associer les taux de franchissement à ces sous sources.

3.7.1 Analyse des performances de R_2

Le modèle R_2 est borné et le marquage initial est un état d'accueil, par conséquent, ce modèle admet un état stationnaires $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_n)$.

Une fois ces probabilités d'état stationnaire sont obtenus, plusieurs paramètres de performances peuvent être calculés en appliquant les formules explicites déjà données dans la section (3.6.1) et qui sont valables aussi pour R_2 .

3.7.2 Application sur le système $M_2/M_2/1//(2, 2)$

On considère une file d'attente $M_2/M_2/1//(2, 2)$ avec priorité relative et à deux sources finie de taille $N_1 = 2$ pour les clients non prioritaires et $N_2 = 2$ pour les clients prioritaires. Les deux types de clients arrivent indépendamment les uns des autres suivant un processus markovien de taux λ_1 et λ_2 et leurs taux de service est μ_1 et μ_2 , respectivement. Le marquage initial est alors donné par :

$$M_o = (M(P_B), M(P_p), M(P_{SP}), M(P_A), M(P_N), M(P_{SN}), M(P_{SL})) = (2, 0, 0, 2, 0, 0, 1)$$

Le graphe des marquages de ce RdPSG est illustré dans la figure 3.6, il nous permet de calculer tous les marquages possibles à partir du marquage initial. Ce graphe est un arbre qui représente les séquences de franchissements des transitions à partir du marquage initial. Il contient 7 marquages évanescents qui sont : $M_1, M_2, M_{13}, M_{15}, M_{17}, M_{18}, M_{19}$ et le reste des marquages sont tangibles.

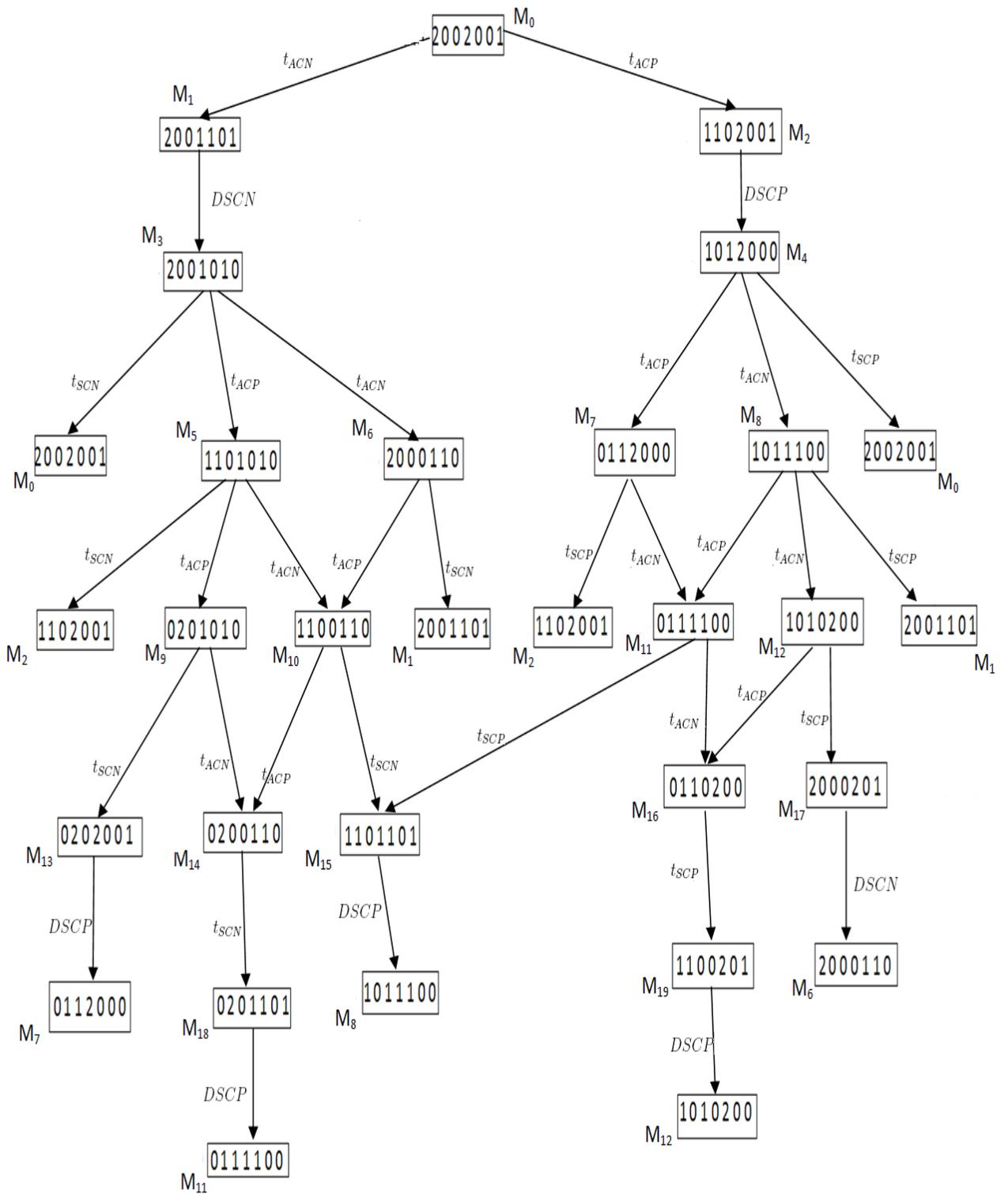


FIG. 3.6 – Graphe d'accessibilité associé à R_2

Une chaîne de Markov à temps continu est construite à partir de ce graphe des marquages. Les états de cette *CMTC* sont les marquages tangibles. Les marquages évanescents sont fusionnés avec leurs successeurs : M_1 est fusionné avec M_3 ; M_2 avec M_4 ; M_{13} avec M_7 ; M_{18} avec M_{11} ; M_{15} avec M_8 ; M_{19} avec M_{12} et M_{17} avec M_6 . Les taux de transitions de cette *CMTC* sont les taux de tir des transitions du R_2 . La figure 3.7 illustre la chaîne de Markov associée à R_2 pour $N_1 = 2$ et $N_2 = 2$.

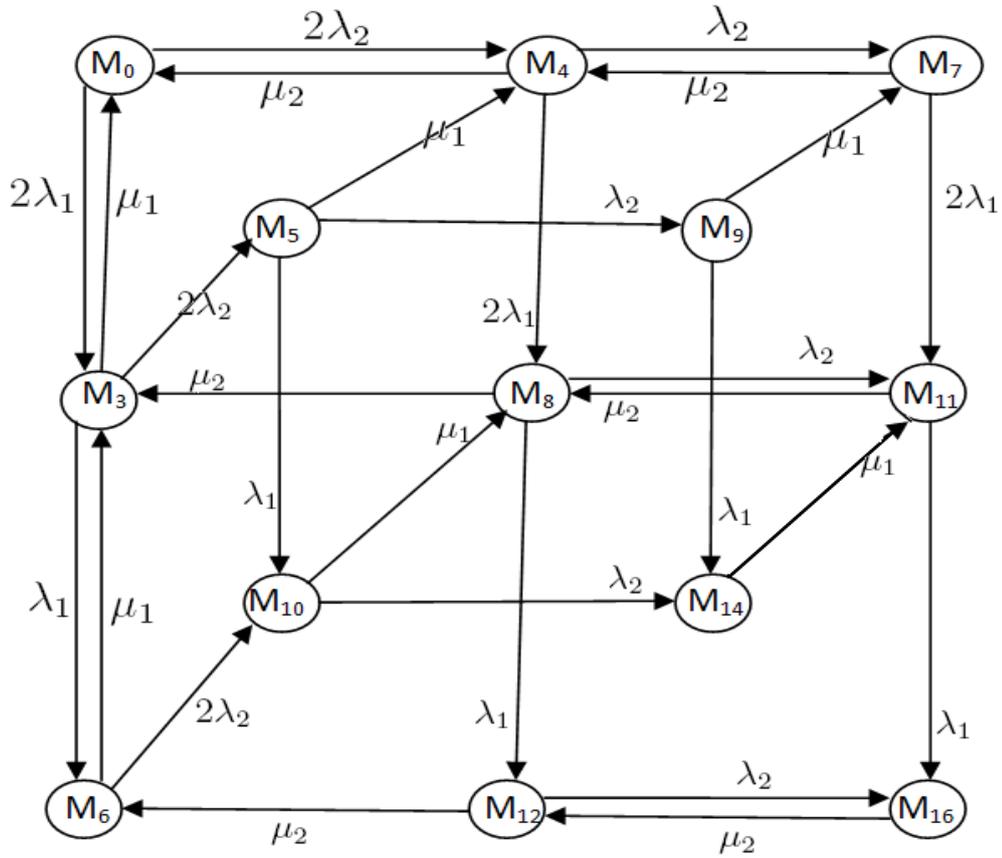


FIG. 3.7 – Chaîne de Markov réduite associée à R_2 pour $N_1 = 2$, $N_2 = 2$

Le générateur infinitésimal Q de la chaîne de Markov ainsi définie est donné par la matrice suivante :

Le vecteur des probabilités stationnaires $\pi = (\pi_0, \pi_3, \pi_4, \pi_5, \pi_6, \pi_7, \pi_8, \pi_9, \pi_{10}, \pi_{11}, \pi_{12}, \pi_{14}, \pi_{16})$ est la solution du système :

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

où :

π_i représente la probabilité stationnaire que le processus soit à l'état M_i .

De ce générateur Q nous pouvons calculer la distribution stationnaire π , ainsi que les indices de performances de R_2 pour $N_1 = 2$ et $N_2 = 2$:

– Le taux moyen effectif des arrivées des clients non prioritaires (η_N) :

$$\eta_N = \sum_{j \in (SM_j)} \lambda_1(M_j) \cdot \pi_j = 2\lambda_1(\pi_0 + \pi_4 + \pi_7) + \lambda_1(\pi_3 + \pi_5 + \pi_8 + \pi_9 + \pi_{11}).$$

– Le taux moyen effectif des arrivées des clients prioritaires (η_P) :

$$\eta_P = \sum_{j \in (SM_j)} \lambda_2(M_j) \cdot \pi_j = 2\lambda_2(\pi_0 + \pi_3 + \pi_6) + \lambda_2(\pi_4 + \pi_5 + \pi_8 + \pi_{10} + \pi_{12}).$$

– Le nombre moyen de clients non prioritaires dans la file (η_{0N}) :

$$\eta_{0N} = \sum_j M_j(P_N) \pi_j = 2(\pi_{12} + \pi_{16}) + \pi_6 + \pi_8 + \pi_{10} + \pi_{11} + \pi_{14}.$$

– Le nombre moyen de clients prioritaires dans la file (η_{0P}) :

$$\eta_{0P} = \sum_j M_j(P_P) \pi_j = 2(\pi_9 + \pi_{14}) + \pi_5 + \pi_7 + \pi_{10} + \pi_{11} + \pi_{16}.$$

– Le nombre moyen de clients non prioritaires dans le système (η_{SN}) :

$$\eta_{SN} = \sum_j [M_j(P_N) + M_j(P_{SN})] \pi_j.$$

$$\eta_{SN} = 2(\pi_6 + \pi_{10} + \pi_{14} + \pi_{12} + \pi_{16}) + \pi_3 + \pi_5 + \pi_9 + \pi_8 + \pi_{11}.$$

– Le nombre moyen de clients prioritaires dans le système (η_{SP}) :

$$\eta_{SP} = \sum_j [M_j(P_P) + M_j(P_{SP})] \pi_j.$$

$$\eta_{SP} = 2(\pi_7 + \pi_{11} + \pi_{16} + \pi_9 + \pi_{14}) + \pi_4 + \pi_5 + \pi_8 + \pi_{10} + \pi_{12}.$$

- Le temps moyen d'attente des clients non prioritaires (W_N) :

$$W_N = \frac{\eta_{0N}}{\eta_N}.$$

- Le temps moyen d'attente des clients prioritaires (W_P) :

$$W_P = \frac{\eta_{0P}}{\eta_P}.$$

- Le temps moyen de réponse des clients non prioritaires (τ_N) :

$$\tau_N = \frac{\eta_{SN}}{\eta_N}.$$

- Le temps moyen de réponse des clients prioritaires (τ_P) :

$$\tau_P = \frac{\eta_{SP}}{\eta_P}.$$

Nous avons analysé le système $M_2/M_2/1//(N_1, N_2)$ avec plusieurs valeurs de N_1 et N_2 . Notre choix s'est porté sur le cas du système $M_2/M_2/1//(2, 2)$ pour le présenter dans cette section. Néanmoins, le graphe de la chaîne de Markov associé à R_2 avec N_1 et N_2 quelconques a été obtenu d'une manière récursive (voir 3.8).

Il est clair que l'analyse des systèmes prioritaires avec sources finies multiples par les RdPSG se complique au fur et à mesure que le nombre de sources augmente. Ceci, peut se constater dans la section suivante où un système $M_3/M_3/1//(N_1, N_2, N_3)$ sera analysé.

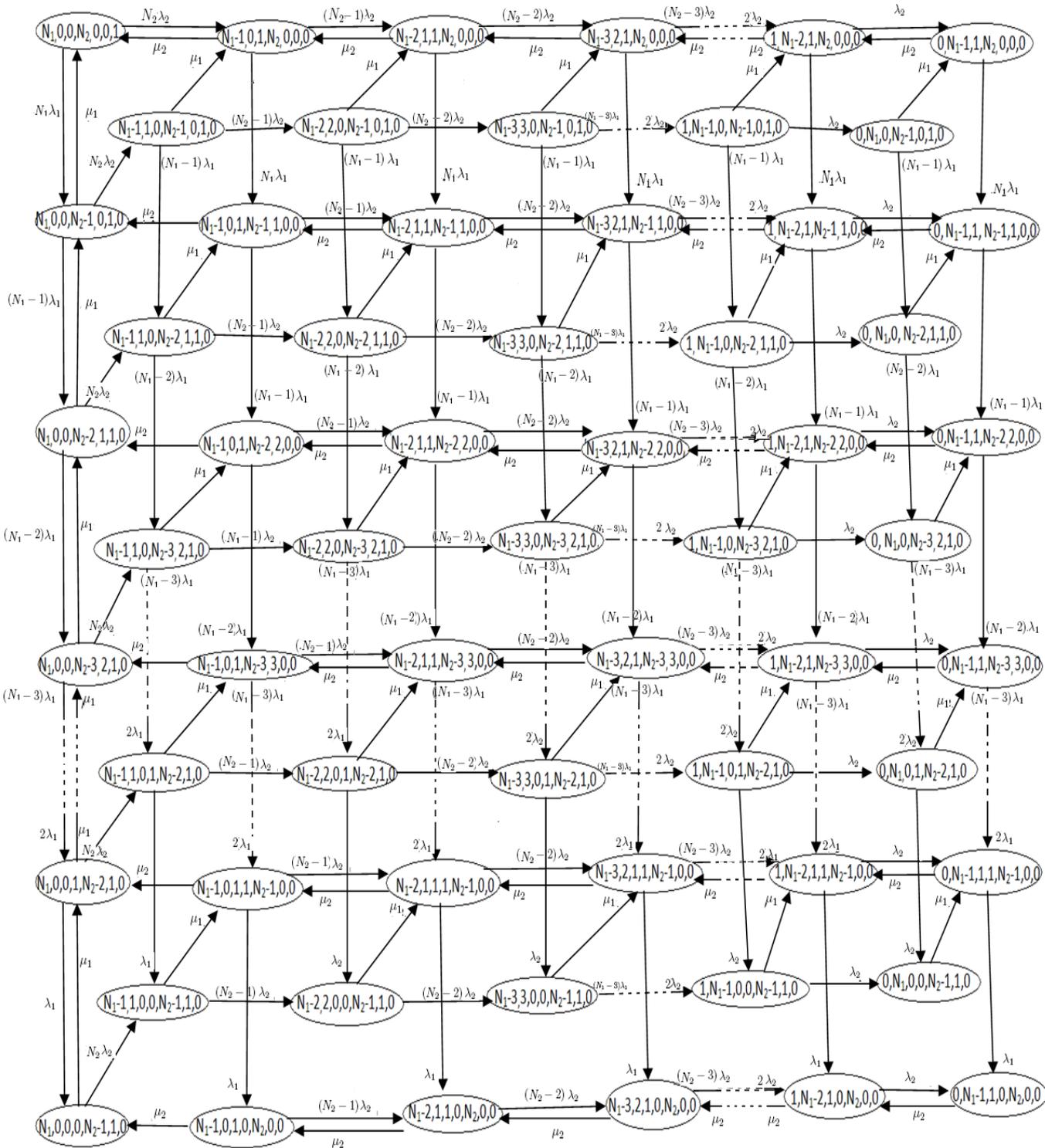


FIG. 3.8 – La chaîne de Markov réduite associée à R_2

3.8 Modélisation de systèmes avec priorité relative à trois sources finies par les RdPSG

Dans cette section, nous allons analyser un système de files d'attente à 3 classes de priorités $M_3/M_3/1/(N_1, N_2, N_3)$. Les clients de classe C_i sont prioritaires à ceux de la classe C_j si $i > j$. Ces trois types de clients arrivent indépendamment les uns des autres suivant un processus markovien avec un taux λ_i , $i = \overline{1, 3}$.

Le service des clients de classe C_i , $i = \overline{1, 3}$ se fait suivant la loi exponentielle de paramètre μ_i .

La figure 3.9 décrit le modèle *RdPSG* correspondant à ce système. La capacité des trois populations est représentée par les paramètres entiers positifs N_1 , N_2 et N_3 qui apparaissent comme marquage initial des places P_A , P_B et P_C . Par la suite, il sera référé au modèle de RdPSG représenté dans la figure 3.9 par R_3 .

- La place P_C est la source qui contient les clients prioritaires de classe 3, représentés par N_3 jetons ;
- La place $P_{\bar{p}}$ contient les clients prioritaires de classe 3 ;
- La place $P_{\overline{SP}}$ représente l'état 'Le serveur est occupé par le client prioritaire de classe 3.
- Le marquage initial du réseau est :

$$\begin{aligned} M_o &= (M(P_C), M(P_{\bar{p}}), M(P_{\overline{SP}}), M(P_B), M(P_p), M(P_{SP}), M(P_A), M(P_N), M(P_{SN}), M(P_{SL})) \\ &= (N_3, 0, 0, N_2, 0, 0, N_1, 0, 0, 1). \end{aligned}$$

Ce qui signifie que tous les clients sont initialement libres, et que le serveur est disponible. L'arrivée d'un client prioritaire de classe 3 est représentée par le franchissement de la transition $t_{\overline{ACP}}$, d'où la place $P_{\bar{p}}$ est incrémentée de un. Les clients se comportent indépendamment les uns des autres et le taux de tir est dépendant du nombre de jetons dans la place P_C . Si on a N_3 jetons dans la place P_C alors le taux de franchissement est égal à $N_3\lambda_3$.

Si $P_{\bar{p}}$ contient un jeton et que le serveur est oisif (la place P_{SL} contient un jeton) alors la transition immédiate \overline{DSCP} sera franchie, la place $P_{\overline{SP}}$ reçoit un jeton représentant le serveur qui est occupé par le client de la classe 3.

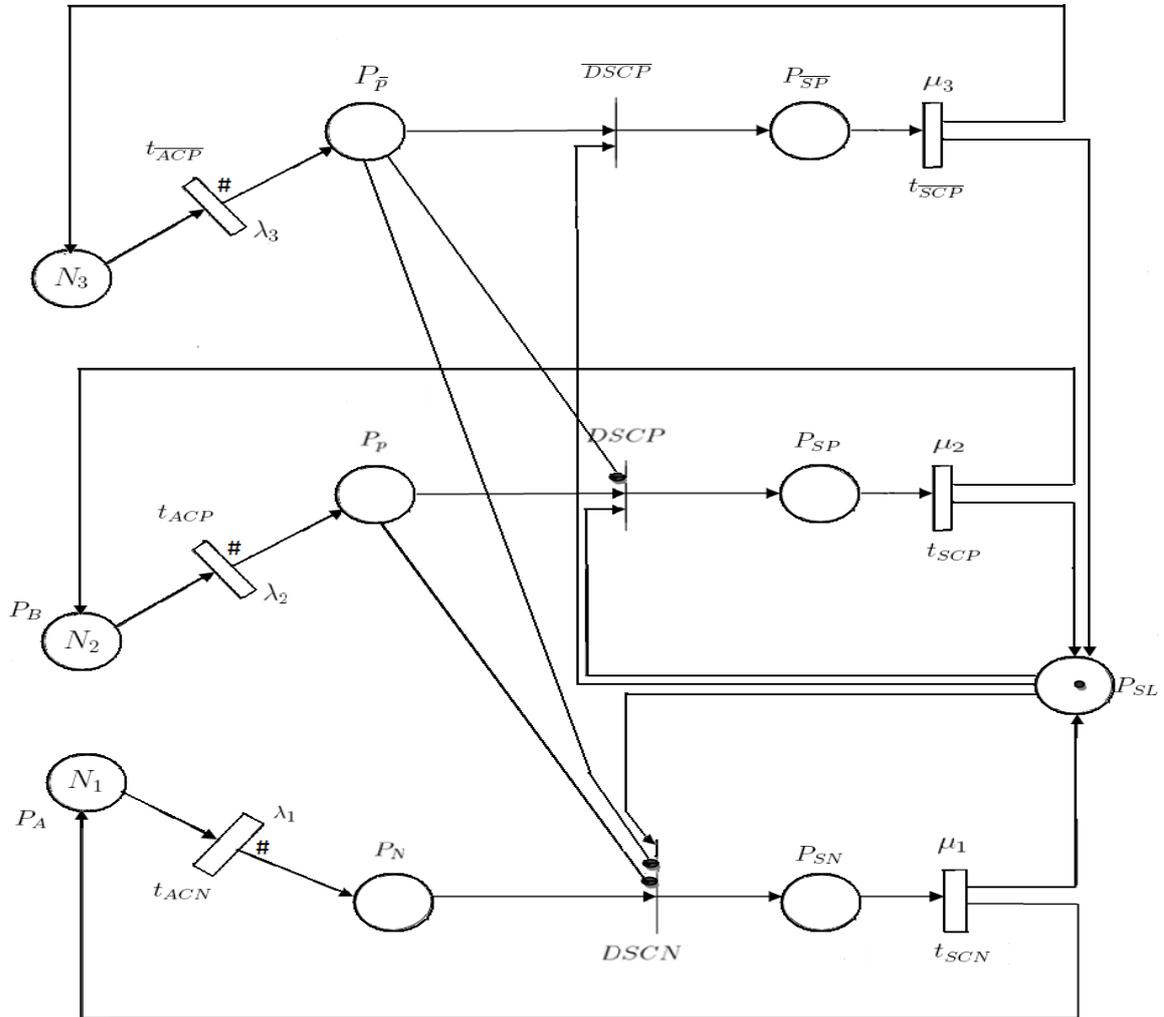


FIG. 3.9 – Le RdPSG modélisant le système $M_3/M_3/1/(N_1, N_2, N_3)$ avec priorité relative

La transition $DSCP$ sera franchie, si la place $P_{\bar{P}}$ est vide (i.e. il n’y a pas de clients de classe 3), cela est exprimé par l’arc inhibiteur allant de la place $P_{\bar{P}}$ vers la transition immédiate $DSCP$.

Pour franchir la transition $DSCN$, il faut que les deux places $P_{\bar{P}}$ et P_P soient vides (les arcs inhibiteurs de $P_{\bar{P}}$ vers la transition immédiate $DSCP$ et de $P_{\bar{P}}$ vers $DSCN$), que la place P_N contienne au moins un jeton et que le serveur soit libre (i.e. la place P_{SL} est non vide).

La fin de service du client prioritaire de classe 3 est représentée par le franchissement de la transition $t_{S\bar{C}P}$. Ainsi, deux jetons seront produits dont l’un est déposé dans P_C , qui

représente la fin du service de ce client, le second jeton représente le serveur qui devient libre et qui rejoint la place P_{SL} .

3.8.1 Analyse des performances de R_3

L'évaluation des performances à l'état stationnaire nécessite l'ergodicité du modèle, or, le modèle R_3 est borné, vivant et son marquage initial est un état d'accueil, par conséquent, ce modèle admet un état stationnaire π .

Après le calcul des probabilités d'état stationnaire, plusieurs paramètres de performances peuvent être calculés de la même manière que ceux adoptés dans les applications précédentes. Nous donnons dans ce qui suit juste les indices de performances ayant rapport avec les clients de la classe 3.

- **Le taux moyen effectif des arrivées des clients prioritaires de classe 3** ($\eta_{\bar{p}}$) :

$$\eta_{\bar{p}} = \sum_{j \in (SM_j)} \lambda_3(M_j) \pi_j;$$

où :

- (SM_j) est l'ensemble des marquages où la transition $t_{\overline{ACP}}$ est franchissable.
- $\lambda_3(M_j)$ est le taux de franchissement associé à la transition $t_{\overline{ACP}}$ dans M_j .
- **Le nombre moyen de clients prioritaires de classe 3 dans la file** ($\eta_{\overline{0P}}$)

$$\eta_{\overline{0P}} = \sum_j M_j(P_{\bar{p}}) \pi_j;$$

où : $M_j(P_{\bar{p}})$ est le nombre de jetons dans la place $P_{\bar{p}}$ dans le marquage M_j et la somme concerne tous les marquages accessibles.

- **Le nombre moyen de clients prioritaires de classe 3 dans le système** ($\eta_{\overline{SP}}$) :

$$\eta_{\overline{SP}} = \sum_j [M_j(P_{\bar{p}}) + M_j(P_{\overline{SP}})] \pi_j.$$

- **Le temps moyen d'attente des clients prioritaires de classe 3** ($W_{\bar{p}}$) :

$$W_{\bar{p}} = \frac{\eta_{\overline{0P}}}{\eta_{\bar{p}}}.$$

- **Le temps moyen de réponse des clients prioritaires de classe 3** ($\tau_{\bar{p}}$) :

$$\tau_{\bar{p}} = \frac{\eta_{\overline{SP}}}{\eta_{\bar{p}}}.$$

3.9 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la modélisation et à l'évaluation de performances de systèmes de files d'attente avec priorité relative à source(s) finie(s) via les réseaux de Petri stochastiques généralisés.

Dans un premier temps, nous avons présenté quelques théorèmes et notions fondamentales sur l'analyse des RdPSG, qui nous permettent de définir les propriétés qualitatives et quantitatives d'un système à modéliser. Nous avons, par ailleurs, montré comment construire la chaîne de Markov correspondante au *RdPSG* construit, et donné les formules permettant l'obtention des paramètres de performance.

Dans un second temps, nous avons considéré trois exemples d'applications qui se sont portés sur les systèmes suivants : $M_2/M_2/1//N$, $M_2/M_2/1//(N_1, N_2)$ et $M_3/M_3/1//(N_1, N_2, N_3)$. Dans ces exemples, l'étude a été très détaillée, en effet, on a commencé par la description graphique, puis la construction du graphe des marquages, l'extraction de la CMTC associée à ce graphe, l'étude de l'existence de l'état stationnaire et enfin, le calcul des indices de performances.

Dans le chapitre suivant, il sera question de généralisation des applications précédentes aux cas où la population de la source est de grande taille, en faisant appels à des logiciels spéciaux dédiés aux RdPSG afin de valider nos résultats.

4

Applications

4.1 Introduction

L'objectif de ce chapitre est de valider numériquement l'étude théorique faite via les RdPSG sur les systèmes $M_2/M_2/1//N$, $M_2/M_2/1//(N_1, N_2)$ et $M_3/M_3/1//(N_1, N_2, N_3)$, dans le chapitre précédent, en faisant usage de deux logiciels (PIPE [13] et TimeNet [101]) dédiés aux RdPSG. Nous allons évaluer les indices de performances de ces systèmes en faisant varier différents paramètres. Nous avons dû faire appel à deux logiciels pour la simple raison, que chacun d'eux nous offre des possibilités que l'autre n'a pas. En effet, à l'instar de TimeNet, PIPE ne prend pas en considération la dépendance existante entre le taux de franchissement et le nombre de clients dans la source. Cependant, PIPE nous permet de visualiser les marquages évanescents et la CMTC associée au graphe du modèle étudié (pour des espaces d'états limités), ce que TimeNet n'offre pas.

4.2 Description de TimeNet

Le logiciel TimeNet (TIME Net Evaluation Tool) [101] est une boîte à outils graphique et interactive qui permet la modélisation et l'analyse qualitative et quantitative des systèmes en utilisant les réseaux de Petri stochastiques généralisés et réseaux de Petri stochastiques généralisés colorés (RdPSGC).

TimeNet a été développé par "Real-Time Systems" et le groupe de " Robotique " de Technische Universität à Berlin, Allemagne. La première version de cet outil a été développée en 1991. Le projet a été motivé par la nécessité d'un logiciel puissant pour une évaluation efficace des systèmes de RdPSG.

TimeNet et son prédécesseur DSPNexpress ont été influencés par les expériences des autres outils de réseaux de Petri bien connue, GreatSPN et SPNP. TimeNet fournit une interface graphique conviviale, il est particulièrement adapté à l'analyse des réseaux de Petri déterministes, stochastiques et stochastiques généralisés. L'analyse est aussi disponible, en exclusivité, pour les réseaux de Petri stochastiques non-markoviens.

La version TimeNet 4.0 est une version stable disponible depuis 2007 dans laquelle de nouveaux algorithmes ont été ajoutés, elle comprend une interface graphique qui intègre les différents formalismes des réseaux de Petri et des algorithmes d'analyse générique en Java [102]. Elle prend en charge une nouvelle classe de réseaux de Petri stochastiques colorés (SCPN) qui permet la conception de systèmes complexes d'une manière compacte. Une simulation standard à événements discrets a été mise en œuvre pour l'évaluation des performances des modèles RdPSC.

Plusieurs outils ont été développés pour l'analyse des RdPSG citons : GreatSPN [20] utilisé par N. Gharbi, Pipe [13], sharp [47], SPNP [23], UltraSAN [91], HiQPN [8], DSPNexpress [63, 64].

Comparaison entre quelques logiciels :

Le tableau 4.1 représente une comparaison de quelques caractéristiques qui ne sont pas partagées entre ces logiciels [13].

Features	GreatSPN 2.0	Sharpe	TimeNet 4.0	PIPE 2.5
Operating system	Unix	Any	Linux/Windows	Any
Arc weighs	Yes	Yes	Yes	Yes
Capacity Limitation	Yes	No	No	Yes
Server Semantics	Yes	No	Yes	No
Token Game				
Step by Step	Yes	No	Yes	Yes
Backwards	No	No	No	Yes
Continuous	Yes	No	Yes	Yes
Analysis				
Transient Analysis	Yes	No	Yes	No
Transient Simulation	Yes	No	Yes	No

FIG. 4.1 – Comparaison de quelques logiciels

4.3 Applications numériques

M. J. Chandra and R. G. Sargent [18, 19] se sont intéressés à l'analyse d'un système de files d'attente mono-serveur avec priorité relative à plusieurs sources finies. En utilisant la technique de la chaîne de Markov induite, ils ont obtenu des mesures de performances de ce système. Les formules analytiques qu'ils ont obtenues font intervenir le paramètre ρ_i qui est la proportion de temps où le serveur est occupé par les clients de la classe i . Ce paramètre est difficile à calculer analytiquement, ainsi, Chandra et Sargent ont établi un algorithme permettant le calcul numérique de ce paramètre.

Nous allons exploiter les résultats de ces deux chercheurs pour valider nos résultats qui seront exposés dans les sections suivantes.

Dans ce qui va suivre, nous allons présenter l'étude faite pour l'évaluation des performances des systèmes modélisés dans le chapitre précédent. En effet, en variant les différents paramètres $\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3$ et pour différents nombres de clients dans les sources on obtient les caractéristiques :

t_{Q_i} : Le temps moyen d'attente d'un client de la classe i .

t_{N_i} : Le temps moyen de réponse d'un client de la classe i .

\bar{Q}_i : Le nombre moyen de clients de la classe i dans la file.

\bar{N}_i : Le nombre moyen de clients de la classe i dans le système.

$\rho(i)$: La proportion du temps d'occupation du serveur par la classe i .

4.3.1 Application 1 : Systèmes mono-serveur à deux sources finies $M_2/M_2/1// (N_1, N_2)$

				RdPSG				
i	N_i	λ_i	μ_i	$\rho(i)$	t_{Q_i}	t_{N_i}	\bar{Q}_i	\bar{N}_i
1	25	0.06	1	0.07634	309.7897	310,7897	23.6512	23.7275
2	25	0.1	2	0.92365	3.0332	3.5332	5.6032	6.5269
1	10	1	10	0.12252	7.0617	7.1617	8.6522	8.7747
2	10	0.8	6	0.87689	0.4839	0.6506	2.5464	3.4233

TAB. 4.1 – Résultats obtenus par les RdPSG pour le cas $M_2/M_2/1// (N_1, N_2)$

Les résultats obtenus par les RdPSG sont validés par les résultats analytiques obtenus par Chandra et Sargent, et sont résumés dans le tableau 4.2. Nous constatons que les résultats obtenus par les RdPSG sont très proches de ceux obtenus dans [18, 19].

				Analytiques				
i	N_i	λ_i	μ_i	$\rho(i)$	t_{Q_i}	t_{N_i}	\bar{Q}_i	\bar{N}_i
1	25	0.06	1	0.0763	309.7898	310.7898	23.6512	23.7275
2	25	0.1	2	0.9236	3.0332	3.5332	5.6032	6.5269
1	10	1	10	0.1225	7.0621	7.1621	8.6523	8.7748
2	10	0.8	6	0.8768	0.4840	0.6506	2.5464	3.4233

TAB. 4.2 – Résultats obtenus par Chandra et Sargent pour le cas $M_2/M_2/1// (N_1, N_2)$

4.3.2 Application 2 : Systèmes mono-serveur à trois sources finies $M_3/M_3/1// (N_1, N_2, N_3)$

				RdPSG				
i	N_i	λ_i	μ_i	$\rho(i)$	t_{Q_i}	t_{N_i}	\bar{Q}_i	\bar{N}_i
1	5	0.2	1	0.3773	7.2491	8.2491	2.7357	3.1130
2	2	0.4	5	0.0717	2.8779	3.0779	1.0319	1.1036
3	3	0.125	0.5	0.5166	1.6135	3.6135	0.4168	0.9334
1	2	0.4	5	0.0195	17.7502	17.9506	1.7359	1.7555
2	3	0.125	0.5	0.3347	7.9236	9.9245	1.3262	1.6609
3	5	0.2	1	0.6142	2.1394	3.1394	1.3142	1.9285
1	20	0.05	5	0.1829	1.6644	1.8644	1.5225	1.7054
2	4	0.3	2.5	0.3767	0.5138	0.9138	0.4839	0.8607
3	4	0.2	3	0.2365	0.3028	0.6362	0.2149	0.4515

TAB. 4.3 – Résultats obtenus par les RdPSG pour le cas $M_3/M_3/1// (N_1, N_2, N_3)$

				Analytiques				
i	N_i	λ_i	μ_i	$\rho(i)$	t_{Q_i}	t_{N_i}	\bar{Q}_i	\bar{N}_i
1	5	0.2	1	0.377	7.249	8.249	2.735	3.113
2	2	0.4	5	0.072	2.878	3.078	1.031	1.103
3	3	0.125	0.5	0.517	1.614	3.614	0.416	0.933
1	2	0.4	5	0.020	17.750	17.950	1.735	1.755
2	3	0.125	0.5	0.335	7.924	9.924	0.326	0.661
3	5	0.2	1	0.614	2.139	3.139	1.314	1.928
1	20	0.05	5	0.183	1.665	1.864	1.522	1.705
2	4	0.3	2.5	0.376	0.513	0.913	0.483	0.860
3	4	0.2	3	0.236	0.303	0.636	0.215	0.451

TAB. 4.4 – Résultats obtenus par Chandra et Sargent pour le cas $M_3/M_3/1// (N_1, N_2, N_3)$

Même chose pour ce cas, les résultats obtenus sont validés par Chandra et Sargent, représentés dans le tableau 4.4. Nous remarquons que les résultats obtenus par les RdPSG, représentés dans le tableau 4.3, sont très proches de ceux obtenus par Chandra et Sargent.

4.3.3 Application 3 : Systèmes mono-serveur à une seule source finie

Modélisation par la simulation à évènements discrets

Dans cette partie, nous proposons une modélisation du système $M_2/M_2/1//N$ par simulation à évènements discrets. Les étapes de la simulation sont données dans l'organigramme de la figure 4.2 où la légende des notations utilisées est la suivante :

N : taille de la population.

λ_n : paramètre de la loi exponentielle des inter-arrivées des clients non prioritaires.

λ_p : paramètre de la loi exponentielle des inter-arrivées des clients prioritaires.

μ_n : paramètre de la loi exponentielle des durées de service des clients non prioritaires.

μ_p : paramètre de la loi exponentielle des durées de service des clients prioritaires.

ρ : charge théorique du système.

N_n : nombre de clients non prioritaires dans le système.

N_p : nombre de clients prioritaires dans le système.

$tmax$: durée de la simulation.

$ACNP$: Arrivée d'un Client Non Prioritaire.

$DSCNP$: Début de Service d'un Client Non Prioritaire.

$FSCNP$: Fin de Service d'un Client Non Prioritaire.

ACP : Arrivée d'un Client Prioritaire.

$DSCP$: Début de Service d'un Client Prioritaire.

$FSCP$: Fin de Service d'un Client Prioritaire.

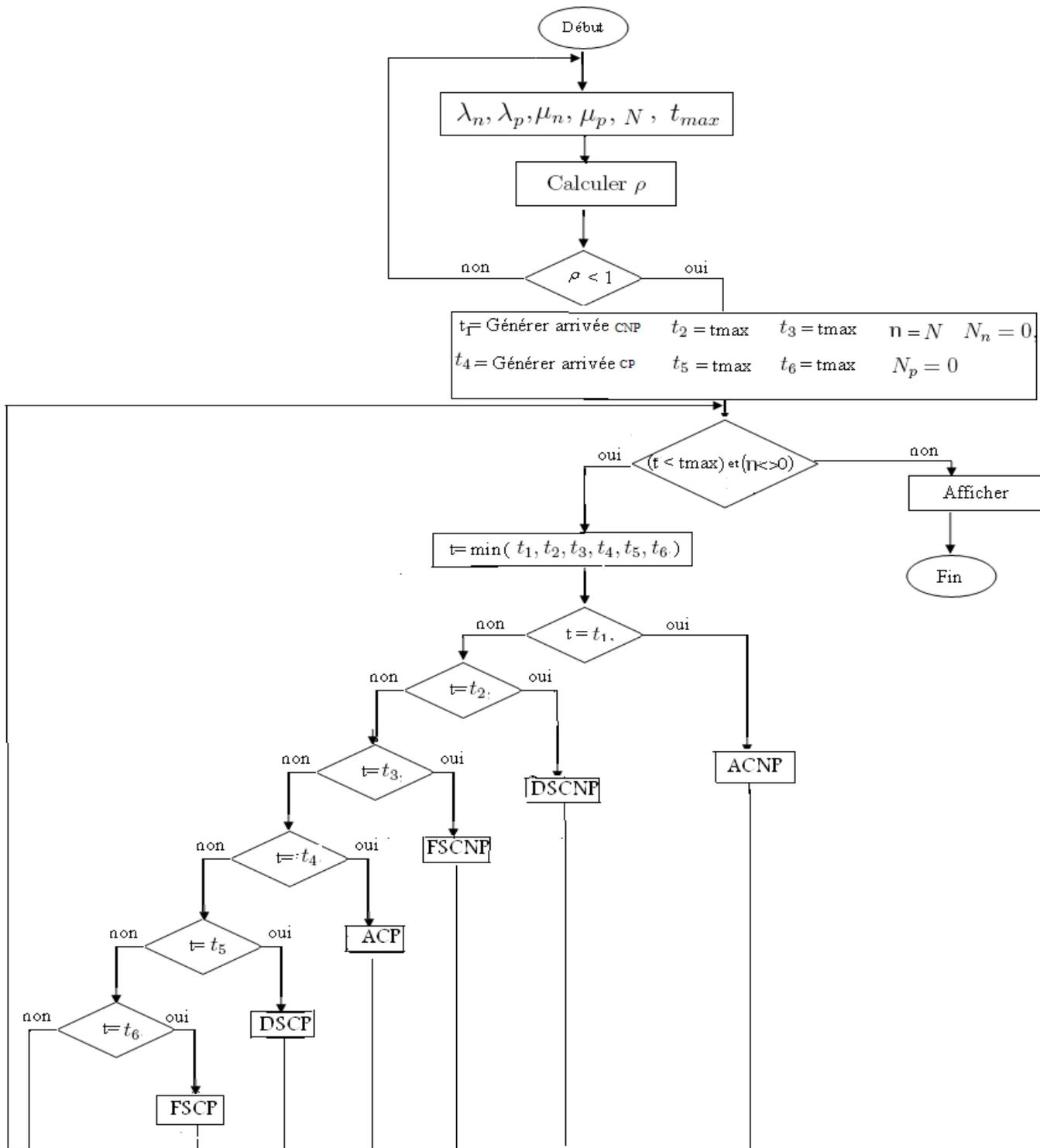


FIG. 4.2 – Organigramme principal de simulation

Comparaison numérique

Les résultats affichés dans le tableau 4.5 représentent les résultats obtenus par la simulation et ceux obtenus par les RdPSG pour les paramètres pour $N=20$, $\lambda_n=1$, $\lambda_p=0.8$, et $\mu_n=10$, $\mu_p=6$, on obtient les résultats suivants :

RdPSG					Simulation			
i	t_{Q_i}	t_{N_i}	\bar{Q}_i	\bar{N}_i	t_{Q_i}	t_{N_i}	\bar{Q}_i	\bar{N}_i
1	3.2455	3.3453	13.9088	14.3373	3.2508	3.3490	13.9035	14.3235
2	0.2349	0.4015	0.8054	1.3768	0.2424	0.4090	0.8296	1.3996

TAB. 4.5 – Comparaison des résultats pour le cas $M_2/M_2/1//N$

Les résultats affichés dans le tableau 4.6 représentent les résultats obtenus par la simulation et ceux obtenus par les RdPSG pour les paramètres $N=50$, $\lambda_n=0.06$, $\lambda_p=0.1$, et $\mu_n=1$, $\mu_p=2$.

RdPSG					Simulation			
i	t_{Q_i}	t_{N_i}	\bar{Q}_i	\bar{N}_i	t_{Q_i}	t_{N_i}	\bar{Q}_i	\bar{N}_i
1	71.0176	72.0176	38.7330	39.2784	71.0949	72.0958	38.7402	39.2856
2	1.2937	1.7937	1.1760	1.6305	1.2972	1.7976	1.1781	1.6326

TAB. 4.6 – Comparaison des résultats pour le cas $M_2/M_2/1//N$

D'après ces tableaux, il s'avère que les résultats obtenus par la simulation sont très proches de ceux obtenus par les RdPSG. Par conséquent, les résultats obtenus par les RdPSG sont très significatifs.

4.4 Conclusion

Dans ce chapitre, nous avons évalué les indices de performances des systèmes $M_2/M_2/1//N$, $M_2/M_2/1//(N_1, N_2)$, $M_3/M_3/1//(N_1, N_2, N_3)$ par les RdPSG pour différents nombres de clients dans la (les) source (s). Les résultats du premier système sont validés par simulation seulement, vu l'inexistence des résultats analytiques dans ce cas de source finie unique. Alors que pour les autres systèmes avec sources finies multiples la validation est faite avec des résultats analytiques.

L'avantage de l'approche de RdPSG proposée et de pouvoir analyser des systèmes complexes. Cette complexité peut être due à l'introduction de plusieurs phénomènes, telle que la priorité que nous avons considérée.

A partir des résultats de cette étude nous prouvons l'efficacité de l'approche proposée. De nombreux problèmes dans les systèmes avec priorité et sources finies peuvent être simplifiés en utilisant cette approche, en particulier, cette approche nous évite d'avoir à faire au problème de la détermination du débit ρ_i .

Conclusion générale et perspectives

L'analyse d'un système prioritaire présente des difficultés puisque la description de l'état du système doit tenir compte de l'information de chaque classe. Si les sources sont finies, cela introduit de grandes difficultés analytiques et complique énormément l'analyse des performances du système. C'est pourquoi, nous avons proposé dans ce mémoire une approche de modélisation, d'analyse et d'évaluation de performances des systèmes prioritaires mono-serveur à source(s) finie(s) basée sur les RdPSG.

L'aspect stochastique est ici très important car il fait partie intégrante de l'évolution de ces systèmes, et la prise en compte de ces phénomènes aléatoires, tels que les aléas de fonctionnement (pannes, arrêts, ...) accentue l'efficacité dans l'étude de ces systèmes. Pour les représentations, les RdPSG constituent alors les modèles les mieux adaptés. La facilité d'emploi de leur représentation graphique justifie déjà ce choix. Leur prolongement vers les processus de Markov, outils analytiques très puissants, accroît considérablement l'intérêt pour l'analyse et la synthèse de ces systèmes.

L'intérêt majeur des RdPSG est de pouvoir combiner l'analyse qualitative et l'analyse quantitative. Les RdPSG permettent l'obtention de résultats des indices de performances aisément. Ils permettent également d'inclure facilement les différentes structures de synchronisation.

Dans ce mémoire, nous nous sommes intéressés à l'évaluation des performances des systèmes d'attente $M_2/M_2/1//N$, $M_2/M_2/1//(N_1, N_2)$ et $M_3/M_3/1//(N_1, N_2, N_3)$ avec priorité relative et source(s) finie(s) via les RdPSG.

Dans un premier temps, nous avons étudié l'outil de modélisation RdP, leurs propriétés, leurs fonctionnements ainsi que quelques unes de ses extensions.

Dans un second temps, nous avons modélisé les systèmes $M_2/M_2/1//N$, $M_2/M_2/1//(N_1, N_2)$ et $M_3/M_3/1//(N_1, N_2, N_3)$, par les RdPSG. Après la construction des modèles RdPSG associés aux systèmes, nous nous sommes intéressés à leurs analyses

qui consiste à définir d'une part leurs propriétés qualitatives et d'autre part à calculer leurs paramètres de performances quantitatifs.

Pour effectuer l'analyse qualitative, nous avons dû étudier les propriétés des modèles de RdPSG associés à ces systèmes. La bornitude, la vivacité et l'admission du marquage initial comme état d'accueil dans ces RdPSG ont permis d'affirmer que l'état stationnaire de chacun de ces systèmes existe. Par conséquent, on peut établir l'analyse quantitative.

Pour l'analyse quantitative, nous avons généré les graphes des marquages accessibles (GMA) qui représentent les séquences de franchissements des transitions après l'occurrence de certains événements et qui permettent de visionner tous les marquages possibles à partir du marquage initial. A partir de ces GMA, nous avons construit les chaînes de Markov à temps continu, déduit les conditions d'ergodicité et défini ainsi la distribution stationnaire associée à chaque modèle. Ceci nous a permis de calculer les différents indices de performances.

Pour un nombre de clients important, nous avons pu évaluer les principaux indices de performances à l'aide du logiciel dédié aux RdPSG nommé TimeNet. Les résultats du système $M_2/M_2/1//N$ sont validés par simulation tandis que pour les deux autres systèmes $M_2/M_2/1//(\mathcal{N}_1, \mathcal{N}_2)$ et $M_3/M_3/1//(\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3)$, la validation est faite avec des résultats analytiques. Les résultats numériques obtenus par ce logiciel ont prouvé l'efficacité de l'approche proposée. En effet, nous avons pu constater que les RdPSG sont très puissants en terme d'expressivité et qu'ils constituent une approche fiable pour le calcul des paramètres de performances.

Comme perspectives de recherche, nous suggérons de considérer les points suivants :

- Lorsque les systèmes étudiés contiennent plusieurs classes de clients, leurs représentations en RdPSG deviennent complexes donc difficilement exploitables [44, 51, 50, 49]. Les RdPSG colorés constituent un modèle mathématique de haut-niveau, pouvant réduire la taille du modèle et dans lesquels chaque jeton correspondant à un client ou à un serveur doit porter une information qui dénote son type. Ainsi, il serait judicieux de les exploiter par l'analyse des systèmes étudiés dans ce mémoire quand le nombre de classes prioritaires est assez grand.
- Il serait intéressant d'élargir l'étude au cas non-borné $M_R/M_R/m//\infty$ à population infinie. Dans ce cas, il y'a lieu de faire appel au graphe de couverture.
- Il sera pertinent de considérer l'analyse des réseaux de files d'attente.

Bibliographie

- [1] D. Aïssani. Application of the operator methods to obtain inequalities of stability in an $M_2/G_2/1$ system with a relative priority. *Annales Maghrébines de l'Ingénieur*, 2 :2 : 790–795, 1991.
- [2] M. Ajmone, G. Balbo, and G. Conte. Performance models of multiprocessor systems. *The MIT Press, Cambridge, MA*, 1986.
- [3] M. Ajmone, G. Conte, and G. Balbo. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transaction on computer systems*, 2 :93–122, 1984.
- [4] J. Artalejo and G.Falin. Standard and retrial queueing systems : A comparative analysis. *Revista Matematica Complutense*, (1) :101–129, 2002.
- [5] G. Balbo. Introduction to generalized stochastic Petri nets. *Springer Verlag, Berlin*, 4486 :83–131, 2007.
- [6] G. Balbo, S.C. Bruell, and S. Ghanta. Combining queueing networks and generalized stochastic Petri nets for the solution of complex models of system behavior. *IEEE Transactions On Computers*, 37(10), 1988.
- [7] G. Balbo, G. Chiola, G. Franceschinis, and G. Molina Roet. On the efficient construction of the tangible reachability graph of generalized stochastic Petri nets. *Proc. International Workshop on Petri Nets and Performance Models (PNPM 87), Madison, Wisconsin*, 136-145, Août 1987.
- [8] F. Bause, P. Bucholz, and P. Kemper. QPN-tool for the specification and analysis of hierarchically combined queueing Petri nets. *Quantitative evaluation of computing and communication systems, Springer Verlag*, 977 :224–238, 1995.
- [9] C. Beounes and J.C. Laprie. Dependability evaluation of complex computer systems : Stochastic Petri net modeling. *Proc. 15th IEEE Int. Symp. Fault-Tolerant Computing, Ann Arbor, Michigan*, 364-369, Juin 1985.

-
- [10] P. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE Transaction on Software Engineering*, 3(17) :259–273, March 1991.
- [11] A. Blakemore. The cost of elimination of vanishing markings from generalized stochastic Petri nets. *Proc. International Workshop on Petri Nets and Performance Models (PNPM 89), Kyoto, Japon*, 85-92, 1989.
- [12] G. Bolch, S. Greiner, H. Demeue, and K. S. Trivedi. *Queueing networks and Markov chains*. Published by John Wiley and Sons, Hoboken, New Jersey, 2 edition, 2006.
- [13] P. Bonet, C. Llado, R. Puijaner, and W. Knottenbelt. PIPE 2.5 : A Petri net tool for performance modelling. *In Proc. 23rd Latin American Conference on Informatics CLEI*, october 2007.
- [14] F. Bonson and D. R. Cox. The productivity of machines requiring attention at random intervals. *Royal Statistical Society*, 13(1) :65–82, 1951.
- [15] K. Bose, K. Altinkemer, and R. Pal. Average waiting time of customers in an M/D/k queue with non preemptive priorities. *Computers Operations Research*, 14 :14, 133–136, 1998.
- [16] L. Bouallouche and D. Aïssani. Quantitatives estimates in an $M_2/G_2/1$ priority queue with non-preemptive priority : Method of strong stability. *International Journal Stochastics Models*, 2008.
- [17] H. Cancela. Evaluation de la sûreté de fonctionnement : Modèles combinatoires et markoviens. *PhD thesis, Université de Rennes*, Décembre 1996.
- [18] M. J. Chandra. A study of multiple finite-source queueing models. *The Journal of the Operational Research Society*, 37(3) :275–283, March 1986.
- [19] M. J. Chandra and R. G. Sargent. A numerical method to obtain the equilibrium results for the multiple finite sources priority queueing model. *Informs Management Science*, 29(11) :1298–1308, November 1983.
- [20] G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaud. Great SPN 1.7 : Graphical Editor and Analyzer for Timed and Stochastic Petri Nets. *performance evaluation*, 24 :47–68, 1995.
- [21] Ph. Chrétienne and R. Faure. *Processus stochastiques, leurs graphes, leurs usages*. Gauthier-villars, Paris, 1957.
- [22] G. Ciardo. Generalized and deterministic stochastic Petri nets. *Tutorials Notes of the fifth PNPM, Toulouse*, October 1993.

-
- [23] G. Ciardo, J. Muppala, and K. S. Trivedi. SPNP : Stochastic Petri Nets Package. *3rd Int, Workshop on Petri Nets and Performance models, Kyoto, Japan*, (3) :142–151, 1989.
- [24] A. Cobham. Priority assignment in waiting line problems. *Operations Research*, 1 :2 :70–76, 1954.
- [25] R. David and H. Alla. *Du Grafset aux réseaux de Petri*. Hermès Science Publications, 2 edition, 2001.
- [26] R. David and H. Alla. *Discrete, Continuous, and hybrid Petri Nets*. Springer-Verlag Berlin Heidelberg, 2005.
- [27] R. H. Davis. Waiting time distribution of multiservers priority queueing system. *Operations Research*, 1(14) :133–136, 1966.
- [28] M. Diaz. *Les réseaux de Petri, modèles fondamentaux*. Hêrmès Science Publications, Paris, 1 edition, 2001.
- [29] M. Diaz. *Petri nets, fundamental models, verification and applications*. ISTE Ltd and John Wiley and Sons, Inc, 2009.
- [30] S. A. Dressin and E. Reich. Priority assignment on a waiting line. *Quart Appel Math*, 1 :15 :208–211, 1957.
- [31] C. Dutheillet. *Symétrie dans les réseaux colorés - Définition, analyse et application à l'évaluation des performances*. Thèse de doctorat, Universuté Pierre et Marie Curie Pari VI, Mars 1992.
- [32] J. F. Ereau. *Réseaux de Petri pour l'étude de la disponibilité opérationnelle des systèmes spatiaux*. Thèse de Doctorat de l'université Paul Sabatier de Toulouse, Informatique industrielle, 1997.
- [33] G. Florin, C. Fraize, and S. Natkin. Stochastic Petri nets : Properties applications and tools. *Rapport de Recherche, CEDRIC-CNAM*, 1990.
- [34] P. Gaver. A waiting line with interrupted service, including priority. *Journal of the Royal Statistical Society*, 24 :73–90, 1962.
- [35] A. C. Geniet. *Les réseaux de Petri : Un outil de modélisation*. Springer-Verlag Berlin, 2 edition, 2006.
- [36] N. Gharbi. *Evaluation des performances et de la fiabilité des systèmes Multi-classes avec rappels à l'aide des réseaux de Petri stochastiques colorés*. Thèse de Doctorat, Informatique, U.S.T.H.B, 2007.

-
- [37] N. Gharbi. On the applicability of stochastic Petri nets for analysis of multiserver retrial systems with different vacation policies. *Springer-Verlag Berlin Heidelberg*, pages 289–302, 2008.
- [38] N. Gharbi and M. Ioualalen. Performance analysis of retrial queueing systems using generalized stochastic Petri nets. *In theory and practice of timed systems (TPTS'02), Electronic Notes in Theoretical Computer Science, Grenoble, France*, 65(6), 2002.
- [39] N. Gharbi and M. Ioualalen. Performance evaluation of multi-server queues with station and server vacation. *Applied In International Industrial Simulation Conference ISC'04 Malaga, Spain*, pages 397–401, 2004.
- [40] N. Gharbi and M. Ioualalen. GSPN analysis of retrial systems with servers breakdowns and repairs. *Applied Mathematics and Computation*, 174(2) :1151–1168, 2006.
- [41] N. Gharbi and M. Ioualalen. Numerical investigation of finite-source multiserver systems with different vacation policies. *Journal of Computational and Applied Mathematics*, 234(3) :625–635, 2010.
- [42] G. Goos, J. Hartmanisand, and J. V. Leeuwen. *Petri Nets, Fundamental Models, Verification and Applications*. Springer-Verlag Berlin Heidelberg NewYork, 2001.
- [43] D. Gross and C. M. Harris. Fundamentals of queueing theory. *Wiley, New York*, 1998.
- [44] S. Haddad. *Une Catégorie Régulière de Réseaux de Petri de Haut Niveau : Définition, Propriétés et Réductions*. Thèse de Doctorat, Université Paris VI, 1987.
- [45] N. Hamadouche. Approximation dans les systèmes prioritaires. *Mémoire de Magister, Université de Bejaia*, 2004.
- [46] N. Hamadouche. *Approximation dans les systèmes de files d'attente prioritaires*. Thèse de Doctorat, Université de Béjaia, 2010.
- [47] C. Hirel, R. Sahner, X. Zang, and K. Trivedi. Reliability and performability modeling using sharpe. *On Computer Performance Evaluation Modelling Techniques and Tools, LNCS, Springer Verlag*, pages 345–349, 2000.
- [48] J . L. Holloy. Waiting line subject to priorities. *Operations Research*, 1 :2 :341–343, 1954.
- [49] M. Ioualalen and A. Aissani. Symétries dans les réseaux de Petri stochastiques, construction du graphe symbolique. *RAIRO –VT*, (5-99), 1999.

-
- [50] M. Ioualalen and M.C. Boukala. Les réductions dans les réseaux de Petri stochastiques. *4ième colloque sur la recherche en Informatique, CARI'98, Dakar*, pages 725–735, 1998.
- [51] M. Ioualalen, M.C. Boukala, and A. Aïssani. Les méthodes d'analyse hiérarchiques des réseaux de Petri stochastiques. *3ième Conférence Internationale sur l'Automatisme des Processus Hybrides, Reims*, 1998.
- [52] N. K. Jaiswal. Finite-source queuing models. *Case Institute of Technology, Cleveland, Ohio*, (45), 1966.
- [53] N. K. Jaiswal and K. Thiruvengadam. Finite source priority queues. *SIAM J. Appl. Math*, 15 :1278–1293, 1967.
- [54] N.K. Jaiswal. *Priority Queues*. Academic press new york and london, University of southern California, 1968.
- [55] K. Jensen. Coloured Petri nets : basic concepts, analysis methods and practical use. *Basic Concepts, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Berlin*, 2, 1992.
- [56] K. Jensen. Coloured Petri nets :Basic concepts, analysis methods and practical use. *Springer Verlag, Berlin*, 1, 1997.
- [57] K. Kant. Introduction to computer system performance evaluation. *Mc Graw-Hill, Inc*, 1992.
- [58] J. Keilson. Queues subject to service interruption. *The Annals of Mathematical Statistics*, 33 :1314–1322, 1962.
- [59] W. Khansa. *RdP P-temporels, contribution à l'étude des systèmes à évènements discrets*. Thèse de Doctorat, Ecole Supérieure d'Ingénieurs d'Annecy, 1997.
- [60] V. Kordic. *Petri Net Theory and Applications*. I-TECH Education and Publishing, Austria, 2008.
- [61] F. Kordon. *Prototypage de systèmes parallèles à partir de réseaux de Petri colorés*. Thèse de doctorat, Université Pière et Marie Curie, 1992.
- [62] O. Lekadir and D. Aïssani. Approximation de Deux Files d'Attente en Série avec Blocage et Priorité Relative. In *Actes du Colloque International Statistique des Processus et Applications, Constantine, Algérie*, pages 67–70, Octobre 2008.
- [63] Ch. Lindemann. Stochastic modeling using DSPNexpress. *Performance Evaluation, Oldenbourg*, 1994.

-
- [64] Ch. Lindemann. DSPNexpress : A software package for the efficient solution of Deterministic and Stochastic Petri Nets. *Performance Evaluation*, 22 :3–21, 1995.
- [65] J. D. C. Little. A proof of the formula : $L = \lambda w$. *Oper. Res.* 383-387, 1961.
- [66] A. Marsan, G. Balbo, G. Chiola, and G. Conte. Generalized stochastic Petri nets revisited : Random switches and priorities. *Proc. International Workshop on Petri Nets and Performance Models (PNPM 87), Madison, Wisconsin*, 44-53, 1987.
- [67] A. Marsan, G. Balbo, G. Chiola, and G. Conte. On Petri nets with deterministic and exponential transition firing time. *7th European Workshop on Application and Theory of Petri Nets, Oxford*, Juin 1986.
- [68] A. Marsan, G. Balbo, G. Chiola, G. Conte, S. Donatelli, and G. Franceschinis. An introduction to generalized stochastic Petri nets. *Microelectronics and Reliability, Pergamon Press*, 31(4) :699–725.
- [69] M. A. Marsan, G. Balbo, A. Bobio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on software Engineering*, 15 :832816, July 1989.
- [70] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modeling with Generalised Stochastic Petri Nets*. Università degli studi di Torino, Dipartimento di Informatica, 1994.
- [71] M. A. Marsan and G. Chiola. On Petri nets with deterministic and exponential transition firing time. *On 7th European Workshop on application and Theory of Petri Nets, Oxford*, June 1986.
- [72] R. D. Mei-Van-Der, J. L. Mei-Van-Berg, R. Vranken, and B. M. Gijsen. Sojourn time approximations for a multi server processor sharing system with priorities. *Performance evaluation*, 54(3) :249–261, 2003.
- [73] P. Merlin and D. J. Faber. Recoverability of communication protocols. *IEEE Transaction on Communication, COM*, 9(24), 1976.
- [74] D. R. Miller. Computation of steady state probabilities for M/M/1 priority queues. *Operations Research*, 29 :945–958, 1981.
- [75] A. Mkhida. *Contribution à l'évaluation de la sûreté de fonctionnement des Systèmes Instrumentés de Sécurité intégrant de l'Intelligence*. Thèse de Doctorat, Institut National Polytechnique de Lorraine, Nancy Université, INPL, 2008.
- [76] M. K. Molloy. On the integration of delay and throughput measures in processing models. *Ph.D.Thesis, University of California, Los Angeles, USA*, 1981.

-
- [77] M. K. Molloy. Performance analysis using stochastique Petri nets. *IEEE Transactions on Computers*, 913-917, 1982.
- [78] J. Moody and P. Antsaklis. Supervisory control of discrete event systems using Petri nets. *Kluwer, Boston (USA)*, 186, 1998.
- [79] J. M. Moore. Final report of the queueing standardization conference. *AIIE Trans*, 4 :72-74, 1972.
- [80] T. Murata. *Petri nets properties, analysis, and applications*. Proceedings of the IEEE, 77(4) :541-580 edition, April 1989.
- [81] S. Natkin. *Les Réseaux de Petri stochastiques*. Thèse de Docteur Ingénieur CNAM, Paris, 1980.
- [82] S. Natkin. On the integration of delay and throughput measures in distributed processing models. *PhD thesis, University of California, Los Angeles*, 1981.
- [83] M. F. Neuts. Matrix geometric solution in stochastic models- an algorithmic approach. *Press, T. J. H. U, Baltimore, Maryland*, 1981.
- [84] C. Oliver and S. Kishor. Stochastic Petri net analysis of finite population. *J.C. Baltzer A.G. Scientific Publishing Company, USA*, 1990.
- [85] C.A. Petri. Kommunikation mit automaten. *Phd dissertatio, Institut für Instrumentelle Mathematik, University of Bonn, West Germany*, 65-377, 1962.
- [86] C. Ramchandani. Analysis of asynchronous concurrent systems by timed Petri nets. *Technical Report, Laboratory of Computer science, MIT, Cambridge*, 120, 1974.
- [87] C. Ramchandani. Analysis of asynchronous concurrent systems by timed Petri nets. *Ph.d thesis, MIT, Project MAC TR*, 120, 1974.
- [88] W. Reisig. *Elements of distributed algorithms*. ISBN Springer- Verlag Berlin Heidelberg New York, 3 edition, 1998.
- [89] A. Ruegg. *Processus stochastiques avec applications aux phénomènes d'attente et de fiabilité*. Presses polytechniques Romandes, 1989.
- [90] N. Sadou. *Aide à la conception des systèmes embarqués sûrs de fonctionnement*. Thèse de Doctorat, laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, Toulouse, 2007.
- [91] W. H. Sanders, W. S. Obal, M. A. Qureshi, and F. K. Widjanarko. The ultras an modeling environment. *Performance Evaluation*, 24 :89-115, 1995.
- [92] J. Sifakis. Etude du comportement permanent des réseaux de Petri temporisés. *Journées AFCET sur les Réseaux de Petri, Paris*, 11, 1977.

-
- [93] J. Sifakis. Use of Petri nets for performance evaluation in measuring modelling and evaluating computer systems. *North Holland, Publ. Co.*, 75-93, 1977.
- [94] B. Solaiman. Processus stochastiques pour l'ingénieur. *Presses Polytechniques et Universitaires Romandes*, 2006.
- [95] H. Takagi. Queueing analysis. *North-Holland*, 1, 1991.
- [96] K. Thiruvengadam. *Studies in waiting line problems*. Doctoral thesis, University of Delhi, India, 1965.
- [97] H. White and L. S. Christie. Queueing with preemptive or with breakdown. *Operations research*, 6 :79-95, 1958.
- [98] A. Willig. *Lecture notes on performance evaluation techniques*. Dr-Ing. Andreas Willig Hasso-Plattner-Institut Universität Potsdam, 2004.
- [99] A. Zenie. *Les réseaux de Petri stochastiques colorés : Application à l'analyse des systèmes répartis en temps réels*. Thèse de Doctorat. Université P et M. Curie, Paris VI, Octobre 1987.
- [100] C. Ziegler. *Sûreté de fonctionnement d'architectures informatiques embarquées sur automobile*. Thèse de Doctorat, LAAS, 1996.
- [101] A. Zimmermann and M. Knoke. TimeNET 4.0 A software tool for the performability evaluation with stochastic and colored Petri nets. *Technische Universität Berlin Real-Time Systems and Robotics Group Faculty of EE and CS Technical Report*, (13) :1436-9915, August 2007.
- [102] A. Zimmermann, M. Knoke, A. Huck, and G. Hommel. Towards version 4.0 of TimeNET. *GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems*, (13) :477-480, March 2006.

Résumé

Dans ce mémoire, nous montrons comment utiliser les réseaux de Petri pour modéliser et évaluer les performances de systèmes d'attente prioritaires avec une seule source finie (pour lesquels il n'y a pas de résultats analytiques) et avec plusieurs sources finies (pour lesquels les résultats analytiques sont complexes)

En particulier, nous obtenons les indices de performances des systèmes d'attente avec priorité relative et source(s) finie(s) $M_2/M_2/1//N$, $M_2/M_2/1//(N_1, N_2)$, $M_3/M_3/1//(N_1, N_2, N_3)$ en utilisant le formalisme des RdPSG (Réseaux de Petri Stochastiques Généralisés) qui s'adapte à la structure de ces systèmes.

En raison de l'inexistence de résultats analytiques pour le premier système, une simple étude comparative avec des résultats de la simulation a été réalisée. Par contre, pour les deux derniers systèmes, la validation est faite avec des résultats analytiques.

Mots clés : Systèmes de files d'attente, Priorité relative, Source finie, Modélisation, Evaluation de performances, Réseaux de Petri stochastiques généralisés.

Abstract

In this work, we show how to use Generalized Stochastic Petri Nets to model and evaluate the performance of priority queuing systems with a single finite source (for which there is no analytical results) and with several finite sources (for which analytical results are complex).

In particular, we obtain the performance indices of systems with non-preemptive priority and finite source (s) $M_2/M_2/1//N$, $M_2/M_2/1//(N_1, N_2)$, $M_3/M_3/1//(N_1, N_2, N_3)$ using the GSPN (Generalized Stochastic Petri Nets) formalism that fits the structure of these systems.

Due to the nonexistence of analytical results for the first system, a simple comparative study with simulated results was performed. However, for the latter two systems, the validation was done with analytical results.

Keywords : Queueing systems, Non-preemptive priority, Finite source, Modeling, Performance evaluation, Generalized stochastic Petri nets.