

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
Option : Fiabilité et Evaluation de Performances des Réseaux

Mémoire de Master

Thème

*Modélisation et étude de performances d'une
approche coopérative de gestion des ressources d'un
réseau sans fil*

Présenté par :

M^{elle} Asma ROUBECHE

Devant le jury composé de :

Présidente	M ^{elle} S. Boulefkhar	M.A.A	Université de Béjaïa.
Rapporteur	D ^r L. Bouallouche	M.C.A	Université de Béjaïa.
Rapporteur	M ^r A. Larbi	M.A.A	Université de Béjaïa.
Examinatrice	M ^{elle} S. Hakmi	Doctorante	Université de Béjaïa.
Examinatrice	M ^{elle} S. Lahlah	Doctorante	Université de Béjaïa.

Juin 2012

Dédicaces

*À mes chères parents,
À mes sœurs et frères,
À mes grands parents.*

Remerciements

Initialement, ce mémoire n'aurait pas été réalisé sans la bénédiction du Bon Dieu qui m'a permis de m'instruire et qui a récompensé mes prières.

Au terme de ce travail, je veux bien exprimer ma profonde gratitude à mes encadreurs :

Madame Louiza BOUALLOUCHE, Maître de conférence à l'université A/MIRA de Béjaïa, pour les précieux conseils qu'elle m'a prodigué, ses idées enrichissantes ainsi que la gentillesse qu'elle m'a toujours éprouvé, et qui sans cesse su m'encourager.

Monsieur Ali LARBI, Maître assistant à l'université A/MIRA de Béjaïa, d'avoir accepté, d'assurer le suivi de ce mémoire et pour la qualité de son encadrement.

Je remercie très vivement Mademoiselle Samra BOULEFKHAR, Maître assistant à l'université A/MIRA de Béjaïa, pour l'honneur qu'il m'a fait en présidant le jury de mon mémoire de Master.

Je remercie très vivement Mademoiselle Sedda HAKMI et Mademoiselle Souad LAHLAH, Doctorantes à l'université A/MIRA de Béjaïa d'avoir accepté de participer à ce jury.

Je ne peux conclure ces remerciements sans exprimer ma reconnaissance à toute ma famille, à tous mes amis et à tous ceux qui m'ont aidé et encouragé durant ce projet.

Table des matières

<i>Dédicaces</i>	I
<i>Remerciements</i>	II
<i>Table des matières</i>	VI
<i>Liste des tableaux</i>	VII
<i>Table des figures</i>	IX
Introduction générale	1
1 Les réseaux <i>ad hoc</i>	4
1.1 Introduction	4
1.2 Définition	4
1.3 Domaines d'application	5
1.3.1. Domaines militaires	5
1.3.2. Domaines civiles	6
1.4 Caractéristiques des réseaux <i>ad hoc</i>	6
1.4.1 L'absence d'infrastructure	6
1.4.2 La topologie dynamiques	6
1.4.3 La bande passante limitée	7
1.4.4 Des contraintes d'énergie	7
1.4.5 La sécurité physique limitée	7
1.5 La modélisation	7
1.6 Le routage	7

1.6.1	Protocoles de routage proactifs	8
1.6.2	Protocoles de routage réactifs	9
1.6.3	Protocoles hybrides	9
1.7	Localisation des ressources dans les réseaux <i>ad hoc</i>	10
1.8	Conclusion	11
2	Les caches et les réseaux <i>ad hoc</i>	12
2.1	Introduction	12
2.2	Définition	12
2.3	Fonctionnement des caches	13
2.4	Localisation des caches	13
2.4.1	Le cache client	13
2.4.2	Le cache serveur	13
2.4.3	Le cache proxy	14
2.5	L'apport d'utilisation des cache dans les réseaux <i>ad hoc</i>	14
2.5.1	Réduction de la latence	14
2.5.2	Sauvegarde de la bande passante	14
2.5.3	Augmentation de l'accessibilité des données	14
2.6	Les systèmes de caches coopératifs	15
2.6.1	Architecture hiérarchique	15
2.6.2	Architecture distribuée	16
2.6.3	Les schémas de caches coopératifs MANET	17
2.6.3.1	CachePath	17
2.6.3.2	CacheData	18
2.6.3.3	HybridCache	19
2.7	Les politiques de remplacement de cache dans les réseaux <i>ad hoc</i>	19
2.7.1	Les métriques des politiques de remplacement	20
2.7.1.1	Taux de succès (Hit rate)	20
2.7.1.2	Taux de succès en octet (Byte hit rate)	20
2.7.1.3	Économie de bande passante	20
2.7.1.4	Réduction de latence	20
2.7.1.5	Performance de la CPU	21
2.7.2	Les politiques de remplacement traditionnel	21
2.7.3	Les stratégies de remplacement de cache dédiées aux environnements mobiles	22

2.7.3.1 La politique TDS (Time and Distance Sensitive)	22
2.7.3.2 La politique ZC (Zone coopérative)	22
2.7.3.3 FSDV (Frequency, Size and Distance based Value)	24
2.8 Conclusion	24
3 Les réseaux de Pétri	25
3.1 Introduction	25
3.2 Réseaux de Petri places-transitions	25
3.2.1 Définitions fondamentales	26
3.2.2 Comportement dynamique	27
3.3 Propriétés des réseaux de Petri	31
3.3.1 Les propriétés dynamiques	31
3.3.1.1 Réseau de Petri borné et Réseau sauf	31
3.3.1.2 Vivacité et blocage	31
3.3.1.3 Conflits et parallélisme	32
3.3.1.4 La réversibilité	33
3.3.1.5 Réinitialisabilité et état d'accueil	33
3.3.1.6 Graphe d'événement, la connexité et la forte connexité	33
3.3.2 Les propriétés structurelles	34
3.3.2.1 Représentation matricielle d'un RdP	34
3.3.2.2 Les P-invariants et les T-invariants	34
3.4 Extensions des réseaux de Petri	35
3.4.1 Les réseaux de Petri étendus	35
3.4.2 Les Réseaux de Petri T-temporels	36
3.4.3 Les Réseaux de Petri temporisés	37
3.4.4 Les Réseaux de Petri colorés	38
3.4.5 Les Réseaux de Petri interprétés	38
3.4.6 Les Réseaux de Petri stochastiques	39
3.4.6.1 Définition d'un RdPS	39
3.4.6.2 Durée de sensibilisation stochastique	40
3.4.7 Les Réseaux de Petri stochastiques généralisés	40
3.4.7.1 Définition d'un RdPSG	40
3.4.7.2 Analyse des RdPSG	42
3.4.7.3 Evaluation des indices de performances	43
3.5 Conclusion	44

4	Modélisation et Évaluation des performances	45
4.1	Introduction	45
4.2	Présentation du schéma de coopération	45
4.3	La modélisation via les réseaux de Petri	47
4.3.1	Définition du modèle R_1	47
4.3.2	Interprétation du modèle R_1	51
4.3.3	L'analyse du modèle R_1	52
4.3.3.1	L'analyse qualitative	52
4.3.3.2	L'analyse quantitative	52
4.3.4	Les indices des performances liés à R_1	56
4.4	Le modèle de SimpleCache	57
4.4.1	Définition du modèle R_2	58
4.4.2	Les indices de performances	60
4.5	Amélioration du schéma de coopération	61
4.5.1	Définition du modèle R_3	61
4.5.2	L'analyse qualitative du modèle R_3	63
4.6	Conclusion	63
5	Application	65
5.1	Introduction	65
5.2	Description de l'environnement MATLAB	65
5.3	Exemple d'application	67
5.4	Evaluation des performances du modèle R_1	68
5.5	Evaluation des performances du modèle R_2	69
5.6	Variation des paramètres, Interprétation des résultats et comparaison . . .	69
5.7	Le temps de réponse et la politique de remplacement FSDV	70
5.8	Conclusion	71
	Conclusion générale et perspectives	72
	Bibliographie	74
	Annexe	78

Liste des tableaux

5.1	Les probabilités stationnaires liées à R_1	68
5.2	Les probabilités stationnaires liées à R_2	69
5.3	La variation du taux μ_3	69

Table des figures

1.1	Exemple de réseau <i>Ad hoc</i>	5
1.2	La modélisation d'un réseau <i>ad hoc</i>	8
1.3	Le chemin utilisé dans le routage entre la source et la destination	8
2.1	La réduction de la latence par la technique du cache	15
2.2	Schéma de cache hiérarchique	16
2.3	Schéma de cache distribuée	17
2.4	Cache path dans les réseaux <i>Ad hoc</i>	18
3.1	Exemple d'un réseau de Petri	26
3.2	Franchissement d'une transition	28
3.3	Exemple d'un RdP avec le graphe des marquage associé	30
3.4	Franchissement de transition avec arc inhibiteur	36
3.5	Un exemple de réseau de Petri temporel	37
3.6	Transition immédiate et transition stochastique	41
3.7	Exemple de réseau de Petri stochastique généralisé	41
4.1	Diagramme de coopération des caches	46
4.2	Modélisation du schéma de coopération via les RdPSG et Interprétés	49
4.3	Le graphe des marquages	53
4.4	La chaîne de Markov	54
4.5	Le modèle R_2	58
4.6	Le graphe des marquages associé à R_2	59
4.7	La chaîne de Markov du modèle R_2	59
4.8	Le modèle R_3	62

5.1	Exemple d'un réseau <i>Ad hoc</i>	67
5.2	Comparaison des temps moyen de réponse	70

Introduction générale

Actuellement, les réseaux *Ad hoc* sont employés dans de nombreux contextes (réunions, services de secours, compagnes scientifiques sur le terrain,...). Les réseaux *Ad hoc* sont le plus souvent déployés sur des plate-formes mobiles aux ressources limitées, que ce soit la batterie, la mémoire ou la puissance de calcul. L'utilisation de mobiles devient de plus en plus gourmande en termes de ces ressources, car elles nécessitent plus de données, alors que tous les nœuds mobiles ne peuvent vraisemblablement pas stocker toutes les données disponibles dans leur propre mémoire. Il faut donc trouver une solution pour que chaque nœud puisse accéder à toutes les données en les plaçant de façon adéquate sur le réseau. Évidemment, ceci n'est pas toujours possible puisque la particularité des réseaux *Ad hoc* est la mobilité et il est très envisageable qu'un nœud se trouve séparé du détenteur de la donnée qu'il recherche par suite à un partitionnement du réseau.

L'utilisation des caches dans les réseaux mobiles *Ad hoc* s'avère très efficace dans l'amélioration de la performance de ces réseaux, et ceci en termes de latence, de charge du serveur, de connectivité, de bande passante et de consommation d'énergie. De même, le fait de faire coopérer plusieurs caches d'une même zone géographique peut améliorer de manière significative les performances de caches. Ces approches de coopération se basent sur le partage des données cachées dans les différents nœuds mobiles. Les performances du cache sont aussi étroitement liées à la politique de remplacement utilisée. En effet, lorsque le cache est plein, les algorithmes de remplacement de cache se chargent de trouver un sous-ensemble approprié d'éléments de données à remplacer à partir du cache. Le choix d'une politique de remplacement de cache coopératif pour les réseaux mobiles *ad hoc*, est très important afin de pouvoir améliorer d'une manière significative l'efficacité et les performances du schéma de coopération.

L'évaluation des performances d'une approche de cache coopératif dans un réseau

mobile *Ad hoc* est primordiale pour étudier son comportement dans le système. Il existe deux types d'approches d'évaluation de performance de systèmes : les méthodes analytiques et les techniques de simulation.

La simulation consiste à implémenter un modèle simplifié du système à l'aide d'un programme adéquat [41]. Elle rend possible l'analyse systématique des systèmes lorsque les solutions analytiques ne sont pas disponibles. Cependant, l'élaboration et l'exécution du programme de simulation prennent beaucoup de temps, et malgré tout, ce dernier fournit beaucoup moins d'informations. C'est pourquoi la simulation est généralement considérée comme une technique de dernier recours.

Les méthodes analytiques consistent à réduire le système en un modèle mathématique et à l'analyser numériquement [41]. Ces Méthodes mathématiques sont exactes mais la résolution numérique suscite souvent des limitations qui font adopter une résolution approchée.

Il existe de nombreux outils mathématiques permettant de telle évaluations, tels que les approches probabilistes (chaîne de Markov, files d'attente, les automates, les réseaux de Petri, etc), les approches déterministes (Network Calculus, analyse du pire cas, etc).

La complexité grandissante des systèmes informatiques nécessite le développement de méthodes et outils efficaces dans le but de contrôler leur comportement logique. En effet, il est souvent important de prouver le bon fonctionnement d'un système suivant certaines propriétés. Les résultats observés en analysant qualitativement un système sont complétées par des résultats obtenus en l'analysant quantitativement. Ainsi, des méthodes formelles de validation telles que les réseaux de Petri ont vu le jour par A.C. Petri en 1962.

Contrairement aux autres approches de modélisation citées ci-dessus, la modélisation par des processus stochastiques des RdP offre la possibilité de prendre en considération la synchronisation et de pouvoir générer automatiquement le graphe des états du système. Il ne s'avère donc plus nécessaire de construire naturellement ce graphe, comme dans l'approche markovienne, un tel graphe pouvant contenir plusieurs milliers d'états. De plus, les réseaux de Petri stochastiques nous permettent de faire une analyse quantitative par l'évaluation des indices de performances comme les durées moyennes de service et les temps moyens de réponse.

Différentes grandeurs peuvent être affectées à un réseau de Petri; on parle alors de réseaux de Petri interprétés. Une des formes les plus simples d'interprétation consiste à échanger des messages de type booléen entre les nœuds du réseau. Cet échange prend la forme "condition/action".

Pour toutes ces raisons, nous avons privilégié les réseaux de Petri stochastiques généra-

lisés et interprétés pour modéliser un schéma de coopération et évaluer ses performances.

Organisation du document

Ce mémoire est organisé en cinq chapitres.

Dans le premier chapitre, nous introduisons un état de l'art sur les réseaux mobiles *ad hoc* avec leurs différentes caractéristiques, nous abordons le problème d'acheminement des paquets (le routage), et nous terminons par des généralités sur la localisation des ressources dans ce type de réseau.

Le deuxième chapitre est dédié aux concepts liés au cache et son principe de fonctionnement, nous décrivons la notion de cache coopératif, ses métriques de performances, ainsi que l'apport d'utilisation de cache dans les réseaux mobiles *ad hoc*, puis nous présentons quelques politiques de remplacement de cache.

Le troisième chapitre est consacré à l'étude des réseaux de Petri, en commençant par la présentation de quelques définitions de leur structure, et leurs propriétés structurelles et dynamiques. Les différentes extensions des réseaux de Petri sont également présentées dans ce chapitre. Nous nous intéressons particulièrement aux réseaux de Petri Stochastiques généralisés et interprétés.

Le quatrième chapitre a pour objectif de présenter la partie modélisation via les réseaux de Petri stochastiques généralisés et interprétés. Pour cela, nous commençons par la présentation du schéma de cache coopératif, ensuite nous proposons trois modèles : le cas de coopération, le cas d'absence de coopération, et le cas d'amélioration l'efficacité d'un schéma de cache coopératif par l'intégration d'une politique de remplacement de cache. Finalement nous effectuons une analyse qualitative et une autre quantitative pour l'évaluation des indices de performances des modèles afin d'effectuer une comparaison entre les temps moyens de réponse.

Le cinquième chapitre présente une brève application numérique qui nous permet d'obtenir et d'interpréter les indices de performance.

Nous terminons ce mémoire par une conclusion générale suivie de perspectives qui découlent de ce travail.

Les réseaux *ad hoc*

1.1 Introduction

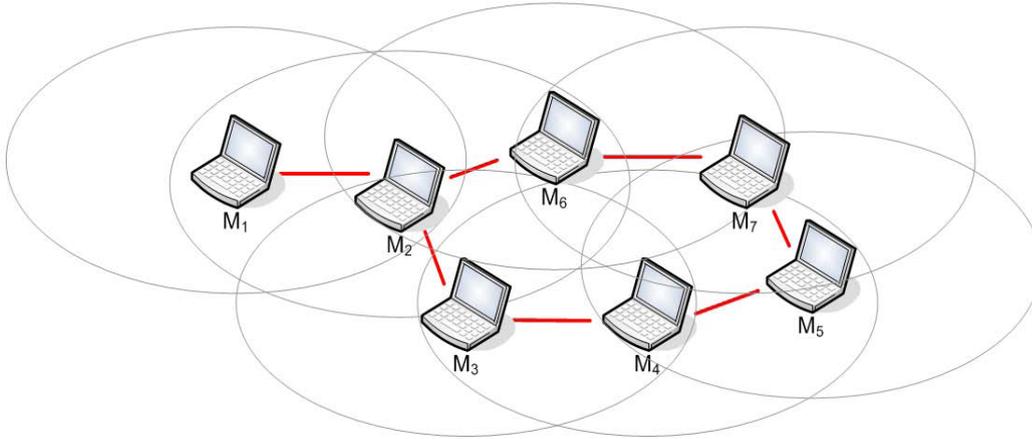
Les environnements mobiles offrent aujourd'hui une grande flexibilité d'emploi, en particulier, ils permettent la mise en réseau des sites dont le câblage serait trop onéreux à réaliser dans leur totalité, voire même impossible. Une alternative intéressante pour pallier à cette limitation est d'utiliser les réseaux sans fil. Dans ce type de réseaux, la portée de transmission des équipements sans fil détermine les terminaux qui peuvent communiquer directement les uns avec les autres, ce qui définit la topologie du réseau. Les nœuds mobiles sont capables de relayer les paquets grâce à un protocole de routage. Ceci permet d'avoir des communications entre des terminaux non adjacents, c'est ce qu'on appelle une communication multi-saut. Un tel réseau est appelé un réseau *ad hoc* ou réseau sans fil multi saut. Un réseau *ad hoc* est un concept qui a fait l'objet de recherches scientifiques depuis les années 1970 par L'IETF¹.

1.2 Définition

Un réseau *ad hoc* ou MANET (Mobile Ad hoc Network) est un ensemble de nœuds mobiles autonomes qui se déplacent et communiquent par une transmission sans fil qui ne suppose pas d'infrastructure préexistante [4]. Le réseau ad hoc se forme de manière spontanée et provisoire dès que plusieurs nœuds mobiles se trouvent à portée radio les uns des autres. Les nœuds communiquent, selon la distance qui les sépare, par deux modes de communication : soit les nœuds mobiles peuvent directement communiquer (en

1. Internet Engineering Task Force

transmission ad hoc) car ils sont dans à portée de transmission, soit ils doivent utiliser d'autres nœuds mobiles comme des relais pour acheminer les paquets à destination (la transmission est multi-sauts).

FIGURE 1.1 – Exemple de réseau *Ad hoc*

1.3 Domaines d'application

Le champ d'application d'un réseau *ad hoc* est vaste et la recherche dans ce domaine est abondante. Initialement, ces réseaux ont été construits dans une optique militaire, mais maintenant, ils sont utilisés dans des contextes divers allant de l'absence de câblage (lié à la vétusté des bâtiments, au manque de place ou à une catastrophe naturelle) aux environnements hostiles.

1.3.1. Domaines militaires

Dans un champ de bataille, les réseaux *ad hoc* peuvent se composer de plusieurs commandants et un groupe de soldats [15]. Les soldats peuvent accéder à des centres d'information des officiers pour des informations géographiques détaillées, des informations sur l'ennemi, de nouvelles commandes, et ainsi de suite. Puisque les soldats voisins ont tendance à avoir des missions similaires et des intérêts communs, ainsi, plusieurs soldats peuvent avoir besoin d'accéder aux mêmes données au différents moments. Avoir un soldat voisin sert plus tard à accéder à ces données au lieu du centre d'information lointain afin d'économiser l'énergie de la batterie, la bande passante, et le temps de latence.

1.3.2. Domaines civiles

Les réseaux mobiles *ad hoc* ont des applications potentielles dans des environnements civils ce qui les rendent un foyer de recherche courant [4], [5].

- **Les services d'urgence** : Opération de recherche et de secours des personnes, tremblement de terre, feux, inondation.
- **Application industrielle** : Les nœuds seront en fait des robots mobiles qui pourront se déplacer librement dans l'usine.
- **Environnement hostiles** : tels que des cratères des volcans pour surveiller leur activité ou bien le long d'une taille géologique.
- **Campus universitaire** : L'utilisation des réseaux sans fil peut être très utile pour les étudiants qui pourront se connecter sur leurs comptes et travailler à partir de la bibliothèque ou de leur chambre.
- **Les réseaux routier** : Le besoin de l'utilisation des réseaux *ad hoc* se fait de plus en plus sentir notamment pour l'échange d'informations routières. En effet, le déploiement de se types de réseaux permet de véhiculer des messages concernant l'état de la route, du trafic, ou la signalisation d'obstacles.
- ...etc

1.4 Caractéristiques des réseaux *ad hoc*

Un réseau mobile *ad hoc* intègre les caractéristiques spécifiques de tout réseau mobile classique entre autre [4], [5] :

1.4.1 L'absence d'infrastructure

L'absence d'infrastructure est une propriété principale pour les réseaux *ad hoc*. Les hôtes mobiles sont responsables de l'établissement et se maintien de la connectivité du réseau de manière continue.

1.4.2 La topologie dynamiques

Causés notamment par des déconnexions volontaires ou involontaires des utilisateurs, ainsi que par des obstacles et des interférences dans les liaisons radio, qui perturbent les communications.

1.4.3 La bande passante limitée

Une des caractéristiques primordiales des réseaux *ad hoc* est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un hôte soit modeste.

1.4.4 Des contraintes d'énergie

L'optimisation de la conservation de l'énergie sera un critère essentiel des solutions proposées pour les nœuds du réseau *ad hoc* fonctionnant sur batterie ou sur des ressources énergétiques limitées.

1.4.5 La sécurité physique limitée

Les réseaux *ad hoc* sont constitués des terminaux mobiles qui sont généralement plus sensibles aux menaces physiques causées par l'absence d'une administration centralisée. Pour cela, le contrôle des données transférées doit être minimisé pour préserver à la fois les ressources et l'énergie.

1.5 La modélisation

Un réseau *ad hoc* peut être modélisé par un graphe $G_t = (V_t, E_t)$ [4], où : V_t représente l'ensemble des nœuds (i.e. les unités ou les hôtes mobiles) du réseau et E_t modélise l'ensemble des connections qui existent entre ces nœuds.

Si $e = (u, v) \in E_t$, cela veut dire que les nœuds u et v sont en mesure de communiquer directement à l'instant t . La figure 1.2 représente un réseau *ad hoc* de 10 unités mobiles sous forme d'un graphe.

1.6 Le routage

Généralement, le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le but d'une stratégie de routage est l'établissement de routes qui soient correctes et efficaces entre une paire quelconque d'hôtes ce qui assure l'échange des messages d'une manière continue [4].

Vu les limitations des réseaux *ad hoc*, la construction des routes doit être faite avec un

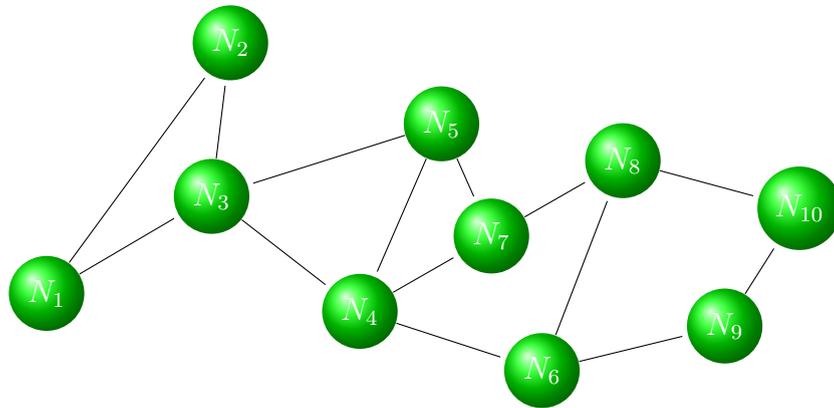


FIGURE 1.2 – La modélisation d'un réseau *ad hoc*.

minimum de contrôle et de consommation de la bande passante. Par exemple, le chemin indiqué dans la figure 1.3 est le chemin reliant une station source et une station destination dans un réseau *ad hoc*.

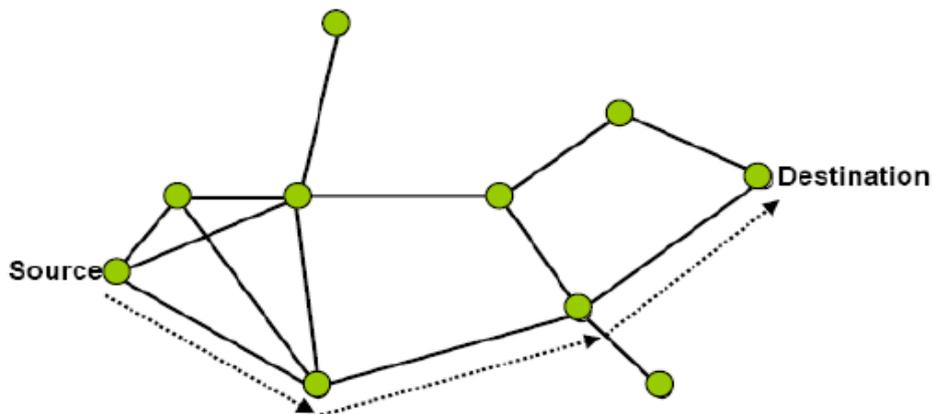


FIGURE 1.3 – Le chemin utilisé dans le routage entre la source et la destination

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage dans les réseaux *ad hoc* peuvent être classés en trois catégories : Les protocoles proactifs, les protocoles réactifs et les protocoles hybrides.

1.6.1 Protocoles de routage proactifs

Les protocoles proactifs sont directement inspirés des protocoles de routages déployés dans l'Internet et sont donc des adaptations des routages à état de lien et à vecteur de

distances.

Afin de prendre en compte les changements de topologie du réseau *ad hoc*, le principe de base des protocoles proactifs est de maintenir à jour les tables de routage, de sorte que lorsqu'une application désire envoyer un paquet à un autre mobile, une route soit immédiatement connue. Dans le contexte des réseaux *ad hoc*, les nœuds peuvent apparaître ou disparaître de manière aléatoire et la topologie même du réseau peut changer ; cela signifie qu'il va falloir un échange continu d'informations pour que chaque nœud ait une image à jour du réseau. Cette approche réagit plus rapidement à la demande puisque les routes sont connues. Les tables de routage mettent beaucoup de temps pour converger vers un état stable à chaque fois que les changements dans la topologie deviennent plus fréquents. Notons aussi que le trafic de contrôle peut être important partiellement inutile, en particulier si les changements de topologies sont moins fréquents. Les tables sont donc maintenues grâce à des paquets de contrôle, et il est possible d'y trouver directement et à tout moment un chemin vers les destinations connues en fonction de divers critères. On peut par exemple privilégier les routes comportant peu de sauts, celles qui offrent la meilleure bande passante, ou encore celles où le délai est le plus faible. Les protocoles les plus importants de cette classe sont **OLSR** (Optimized Link State Routing) [4], [5] et **DSDV** (Destination-Sequenced Distance Vector) [4], [5].

1.6.2 Protocoles de routage réactifs

Un protocole de routage réactif ne cherche une route dans le réseau que lorsqu'un mobile désire envoyer un message. Il n'utilise donc de la bande passante que lorsqu'il en a besoin puisqu'il ne maintient pas à jour d'informations sur le réseau quand ce n'est pas nécessaire. Par contre, il peut s'écouler un temps plus ou moins long avant que la route soit construite et que le mobile puisse envoyer ses données. Ainsi, les informations contenues dans les tables de routage, n'étant pas rafraîchies périodiquement, ne sont utilisables que pendant un temps limité (déterminé par la durée de vie d'une route), ou dans le pire des cas, une seule fois. Actuellement les protocoles de routage réactifs les plus connus sont **DSR** (Dynamic Source Routing) et **AODV** (Ad hoc On Demand distance Vector) [4].

1.6.3 Protocoles hybrides

Les protocoles hybrides combinent les deux approches. Ils utilisent un protocole proactif, pour apprendre le proche voisinage (voisinage à deux ou trois sauts) ; ainsi ils disposent des routes immédiates dans le voisinage. Au-delà de cette zone prédéfinie, le protocole

hybride fait appel aux techniques des protocoles réactifs pour chercher des routes. Avec ce découpage, le réseau est partagé en plusieurs zones, et la recherche de route en mode réactif peut être améliorée. A la réception d'une requête de recherche réactive, un nœud peut indiquer immédiatement si la destination est dans le voisinage ou non, et par conséquent savoir s'il faut aiguiller ladite requête vers les autres zones sans déranger le reste de sa zone. Ce type de protocole s'adapte bien aux grands réseaux, cependant, il accumule aussi les inconvénients des protocoles réactifs et proactifs : messages de contrôle périodique, le coût d'ouverture d'une nouvelle route.

1.7 Localisation des ressources dans les réseaux *ad hoc*

Dans les différents contextes d'application, un utilisateur a besoin d'accéder aux ressources numériques présentes dans le réseau *ad hoc*. Ces ressources numériques peuvent correspondre à des documents mis à disposition, par exemple lors d'une réunion de travail. Elles peuvent aussi désigner des ressources matérielles telles que des capteurs réalisant des prélèvements pour la domotique ou la surveillance des désastres naturels. Enfin, elles font référence à des services numériques effectuant des traitements et des calculs. C'est le cas par exemple d'un service numérique évaluant les pertes survenues sur une scène de désastre ou sur un champ de bataille et fournissant le résultat aux coordinateurs tout en proposant une stratégie [17].

Dans les réseaux filaires ou sans fil basés sur des infrastructures, les solutions proposées afin de localiser les ressources se basent majoritairement sur un répertoire centralisant les informations relatives aux ressources du réseau. Ces solutions sont inexploitable dans les réseaux ad hoc en raison du manque de connectivité induit notamment par les changements de topologie du réseau, et par l'épuisement des réserves énergétiques des batteries.

Dans un environnement sans fil *ad hoc*, la conception d'une stratégie de localisation de ressources doit prendre en compte les contraintes suivantes :

- L'absence d'infrastructures et les changements fréquents de topologie,
- Les faibles capacités du réseau induites par le médium de communication utilisant des liens sans fil de type radio,
- Les faibles capacités et l'hétérogénéité des terminaux,
- Le manque de connectivité des terminaux.

Afin de limiter l'impact de chacune de ces contraintes, une des solutions proposées est d'utiliser la technique de cache (cf. chapitre 2) dont le but est d'améliorer les performances.

Ces performances peuvent ainsi être quantifiée en termes de :

- Minimiser la latence moyenne perçue par les utilisateurs.
- Réduire la consommation d'énergie, ce qui peut améliorer la durée de vie des batteries des nœuds mobiles.
- Réduire la charge de travail sur les sources de données.
- Réduire la consommation de la bande passante du réseau.

1.8 Conclusion

Dans ce chapitre, nous avons présenté le concept des réseaux *Ad hoc* et le problème de routage dans cet environnement. Dans la pratique, les réseaux *Ad hoc* connaissent aujourd'hui plusieurs applications telles que les applications militaires et les applications de secours et de façon générale, toutes les applications caractérisées par une absence d'infrastructure préexistante.

Après avoir défini l'environnement mobile *ad hoc* et décrit ses principales applications et caractéristiques, nous avons abordé le problème de localisation des ressources dans ces types de réseaux. Cette étude nous a conduit à présenter une approche coopérative pour gérer la localisation de ressources dans ce type de réseau, et qui fera l'objet du prochain chapitre.

Les caches et les réseaux *ad hoc*

2.1 Introduction

La mise en cache des données dans les nœuds mobiles d'un réseau *ad hoc* est largement considérée comme une solution efficace pour améliorer les performances du réseau. La gestion coopérative, fondée sur l'idée du partage et la coordination des données en cache entre plusieurs utilisateurs, peut être particulièrement efficace à l'accès à l'information dans les réseaux mobiles *ad hoc*, où les clients mobiles se déplacent fréquemment et la topologie du réseau change dynamiquement.

2.2 Définition

Un cache correspond à une zone de stockage utilisée pour conserver temporairement les contenus en vue d'une réutilisation. Un cache vise à réduire le temps d'attente de l'utilisateur lors de la localisation et de l'accès à un contenu [17]. Cette notion de cache a été introduite en 1965 par **WILKES** sous le nom de mémoire esclave (Slave Memory), l'antémémoire ou mémoire cache, ou encore cache.

Lors d'un lancement d'une recherche d'un document, si les données demandées figurent dans le cache on dit un succès de cache (*cache hit*), sinon on dit un défaut de cache (*cache miss*).

2.3 Fonctionnement des caches

Le cache contient une copie des données originelles lorsqu'elles sont coûteuses (en termes de latence, énergie) à récupérer par rapport au temps d'accès au cache. Une fois les données stockées dans le cache, l'utilisation future de ces données peut être réalisée en accédant à la copie en cache plutôt qu'en récupérant les données, ce qui diminue le temps d'accès moyen.

Le principe de fonctionnement est simple, lorsqu'un client cherche un document, il va à accéder aux ressources de son cache, s'il le trouve pas, il va accéder aux ressources d'un serveur, il envoie une requête au proxy. Celui-ci cherche ce document dans son cache et le retransmet alors au client, on parle alors de succès de cache (*cache hit*). Si le document demandé par le client n'existe pas dans le cache, on parle alors de défaut de cache (*cache miss*), le proxy envoie à son tour la requête au serveur. A la réception du document par le proxy, ce dernier va le retransmettre au client en stockant au passage le document dans son cache s'il le juge intéressant [15].

2.4 Localisation des caches

La gestion du cache dans un réseau se fait à différents niveaux, afin d'obtenir le maximum de résultats. Il existe trois types de caches : les caches client, les caches serveur et les caches proxy. En général, les proxies peuvent être localisés près du serveur (dans ce cas ils agissent comme des serveurs miroirs), ou près des clients dans le but de réduire le temps de latence [2].

2.4.1 Le cache client

Il est disponible au niveau de la machine du client sur son navigateur web. En utilisant ce cache, le client accède à une page sans faire de requête à un serveur si les fichiers correspondants sont stockés dans sa mémoire.

2.4.2 Le cache serveur

Il offre les mêmes fonctionnalités au niveau du serveur, en disposant des documents dans sa mémoire vive le plus fréquemment accédé, de ce fait, il accélère l'accès à ses propres ressources.

2.4.3 Le cache proxy

Est un dispositif commun à plusieurs clients permettant à ceux-ci d'optimiser leur accès au document. La solution du proxy permet d'interposer une capacité de stockage entre le client et le serveur.

2.5 L'apport d'utilisation des cache dans les réseaux *ad hoc*

La mise en cache est une technique importante pour l'amélioration des performances lors d'une récupération des données dans les environnements mobiles *ad hoc*. Avec la mise en cache, le délai d'accès aux données est réduit car les demandes peuvent être servies à partir du cache local, éliminant ainsi la nécessité pour la transmission de données sur les liens sans fil [1].

Les clients mobile dans les réseaux *ad hoc* peuvent avoir des tâches similaires et partagent des intérêts communs, ce qui permet de réduire la bande passante, la latence, augmente l'accessibilité des données et minimise la consommation d'énergie.

2.5.1 Réduction de la latence

Comme illustré dans la figure 2.1 le gain en latence peut être considérable pour une requête servie depuis un nœud intermédiaire plutôt que depuis la source du document demandé.

2.5.2 Sauvegarde de la bande passante

Un autre avantage pour l'utilisation des caches dans les réseaux mobiles *ad hoc* et dans les réseaux en général, est la réduction de la consommation de la bande passante. Avec cette réduction, l'avantage évident est non seulement la réduction des coûts globaux des transferts à travers le réseau, mais elle réduit également la charge de lien vers la source, et donc elle diminue indirectement la latence [1].

2.5.3 Augmentation de l'accessibilité des données

Vu le caractère de topologie dynamique des réseaux mobiles *Ad hoc* qui entraîne de fréquentes déconnexions et d'éventuelles partitionnements de réseaux, une source de données

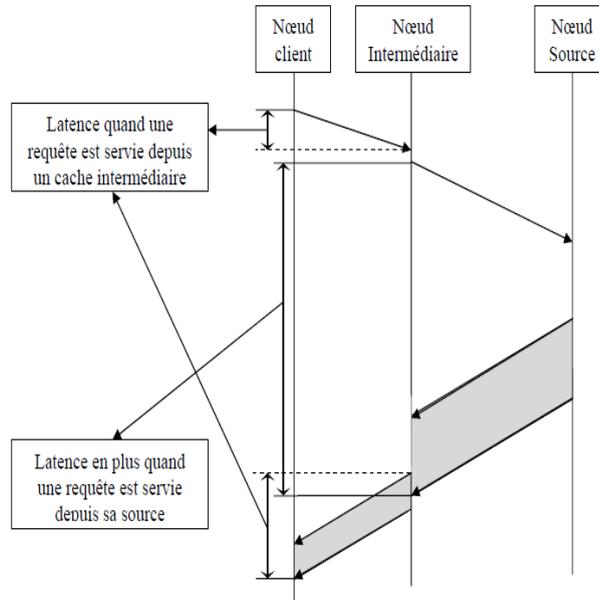


FIGURE 2.1 – La réduction de la latence par la technique du cache

peut devenir inaccessible, mais les requêtes qui lui sont destinées peuvent être satisfaites par les caches des autres nœuds accessibles [1].

2.6 Les systèmes de caches coopératifs

Un système de cache coopératif a été largement étudié pour économiser la bande passante et de réduire la latence et de congestion dans l'Internet. Dans un environnement de cache coopératif, un certain nombre de serveurs cache coopère entre eux pour servir un ensemble de clients afin de les aider à cacher plusieurs objets collectivement, sans surcharger le serveur d'origine.

L'architecture des caches coopératives peut être divisée en deux catégories, hiérarchique et distribuée.

2.6.1 Architecture hiérarchique

Les caches sont organisés selon une certaine hiérarchie. Un cache parent est utilisé pour résoudre les défauts d'objets ayant eu lieu chez un fils. La demande d'un objet est propagée récursivement de manière ascendante jusqu'à trouver l'objet ou atteindre les serveurs de données qui sont virtuellement la racine de la hiérarchie. Lorsque l'objet est

trouvé, il est transféré vers le bas de la hiérarchie jusqu'à atteindre le demandeur. L'objet est caché à chaque niveau [2].

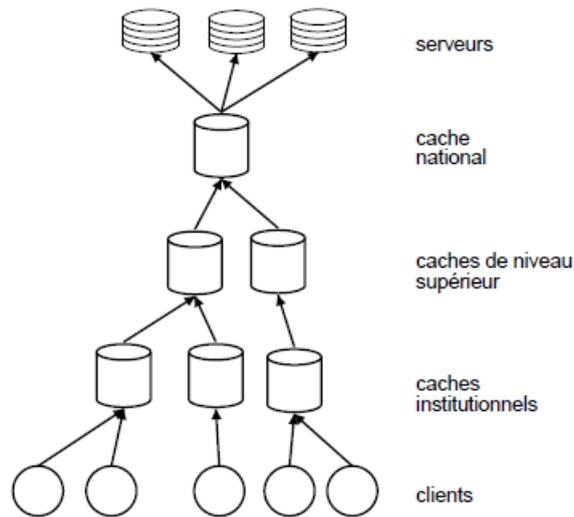


FIGURE 2.2 – Schéma de cache hiérarchique

2.6.2 Architecture distribuée

Ici les caches sont considérés de même niveau. La résolution d'un défaut d'objet peut être demandée à des caches dits frères. Un tel cache frère répond selon son contenu mais ne propage pas la demande. La première réponse positive est prise en compte, les autres étant ensuite ignorées [2].

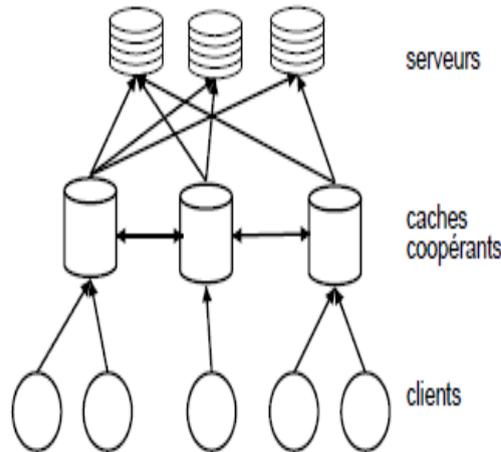


FIGURE 2.3 – Schéma de cache distribuée

2.6.3 Les schémas de caches coopératifs MANET

En mode *Ad hoc*, les clients mobiles peuvent communiquer les uns avec les autres et peut accéder aux éléments de données de cache par les autres clients mobiles connectés. Trois schémas de cache coopérative ont proposé dans [15] pour les réseaux ad hoc : CachePath, CacheData et HybridCache.

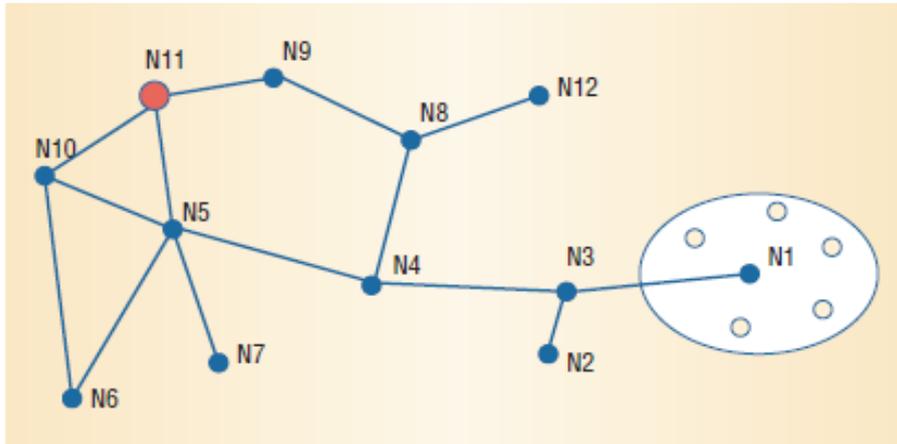
Supposons que dans le réseau ad hoc de la figure 2.4, N_{11} est une source de données contenant une base de données de n articles $d_1 d_2 \dots d_n$. Les nœuds en bleus sont des nœuds routeur, le nœud N_1 est un cluster head de plusieurs nœuds mobiles voisins, ou membres du cluster.

2.6.3.1 CachePath

Les nœuds mobiles cache le chemin de données et l'utiliser pour rediriger les requêtes futures au nœud voisin qui a les données au lieu d'aller au serveur d'origine.

La figure 2.4 illustre le concept CachePath [15].

Supposons que le nœud N_1 demande une donnée d_i à partir de N_{11} . Lorsque N_3 passe une donnée d_i à N_1 , N_3 sait que N_1 a une copie des données. Plus tard, si N_2 demande les mêmes données d_i , N_3 sait que la source de données N_{11} est une suite de trois sauts de distance alors que N_1 est un seul saut de distance. Ainsi, N_3 demande la donnée à N_1 au lieu de demandé à N_4 .

FIGURE 2.4 – Cache path dans les réseaux *Ad hoc*

Beaucoup d’algorithmes de routage fournissent les informations de comptage de saut entre la source et de destination. La cache d’un chemin d’un document réduit la bande passante et économise l’énergie parce que les nœuds peuvent obtenir les documents à un minimum de sauts.

En CachePath, un nœud n’a pas besoin d’enregistrer le chemin des informations de toutes les données passant. Au contraire, il ne sauvegarde que du chemin de données quand ils sont plus proche de cache que de nœud source. Par exemple, quand N_{11} transmet d_i au nœud de destination N_1 empruntant le chemin $\{N_5 - N_4 - N_3 - N_1\}$ et N_5 ne cache pas le chemin d’informations d_i parce qu’ils sont plus proches aux données source que le cache N_1 .

2.6.3.2 CacheData

Les nœuds intermédiaire cache les données pour servir les requêtes futures au lieu d’aller chercher les données du centre d’origine [15].

En CacheData, le nœud routeur met en cache les données au lieu du chemin quand il constate que les données sont souvent consultées. Par exemple, dans la figure2.4, si les deux N_6 et N_7 demande un document d_i à N_5 , ce dernier pourrait penser que d_i est populaire et il le cache localement. N_5 peut alors répondre à des demandes futures de N_4 directement. Comme l’Approche CacheData a besoin d’un espace supplémentaire pour sauver des données, il doit être l’utilisé avec prudence.

Supposons que N_3 passe plusieurs demandes de document d_i à N_{11} , les nœuds du chemin $\{N_3-N_4-N_5\}$ veulent mettre d_i en cache comme un objets fréquemment utilisé.

Cependant, ils vont perdre une grande quantité d'espace mémoire. Pour éviter cela, CacheData applique une autre règle : Un nœud ne cache pas les données si toutes les demandes de ces données proviennent de même nœud.

2.6.3.3 HybridCache

CachePath et CacheData peuvent d'une manière significative améliorer les performances du réseau [15]. L'analyse de CachePath a montré qu'il fonctionne mieux lorsque le cache est petit en terme d'espace mémoire ou la fréquence d'actualisation des données est faible, tandis que CacheData se comporte mieux dans les autres situations. Pour améliorer encore les performances, l'approche HybridCache se produit. Un système hybride est un système qui exploite les avantages de CacheData et CachePath tout en évitant leurs faiblesses. Plus précisément, quand les nœuds mobiles transmettent un document, ils cachent la donnée ou son chemin sur la base de certains critères. Ces critères comprennent la taille de document s_i et la durée de vie (time to live TTL) TTL_i .

Pour cacher un document d_i , on utilise les heuristiques suivantes afin de décider de mettre en cache les données ou le chemin :

- Si s_i est petite, alors CacheData est plus optimale parce que l'élément de données n'a pas besoin qu'une petite taille de mémoire cache, sinon, CachePath est préférable car il économise de l'espace cache.
- Si TTL_i est petit, CacheData est préférable, sinon, pour un TTL_i grand, CachePath est acceptable.

2.7 Les politiques de remplacement de cache dans les réseaux *ad hoc*

L'espace de cache est limité, c'est une première contrainte incontournable qui implique donc que l'on ne peut pas tout cacher. Ceci s'exprime par :

$\sum s_i \leq S$ avec s_i la taille des documents dans le cache, et S la taille du cache.

Il faut donc mettre en place une gestion du cache qui s'assure d'abord que l'on ne dépasse pas cette limite et surtout que l'espace de cache soit utilisé de façon optimale. La gestion du cache doit donc, le cas échéant, supprimer des objets de l'espace cache pour y stocker d'autres objets. Cette opération se fait selon une politique de remplacement qui détermine, en fonction de critères prédéfinis et d'observations statistiques, les objets dont la présence

dans l'espace cache contribue le moins aux gains de performances.

Plusieurs stratégies de remplacement de cache ont été proposées dans le cadre de la mise en cache coopérative dans les réseaux mobile *ad hoc*. La différence entre ces stratégies se situe dans la manière d'expulser les documents [1], [9].

2.7.1 Les métriques des politiques de remplacement

Plusieurs critères ont été utilisés pour définir quand un objet doit être expulsé du cache. Ces critères sont [1] :

2.7.1.1 Taux de succès (Hit rate)

Il représente le pourcentage des documents demandés qui sont servis à partir du cache par rapport au nombre total de documents demandés, les petits documents sont généralement favorisés par les algorithmes de remplacement pour rester dans le cache afin d'augmenter le taux de succès.

2.7.1.2 Taux de succès en octet (Byte hit rate)

Il représente le pourcentage d'octets de données servies à partir du cache par rapport au nombre totale d'octets servis. Cette métrique favorise pour le remplacement les petits documents tandis qu'un algorithme de remplacement basé sur le taux de succès favorise les grands.

2.7.1.3 Économie de bande passante

Cette métrique vise à quantifier le degré de diminution du nombre d'octets transférés à travers le réseau. C'est une métrique très importante pour beaucoup d'ISPs (Internet Service Provider) pour réduire leurs coûts, qui est directement liée à la bande passante.

2.7.1.4 Réduction de latence

La latence se réfère au délai de transmission d'une information d'un point à l'autre. Cette métrique mesure le degré de réduction de cette latence pour une politique de remplacement de cache donnée.

2.7.1.5 Performance de la CPU

Une politique de remplacement de cache excessivement compliquée pourrait faire de la CPU un goulot d'étranglement. Il est également important de noter que les politiques de remplacement qui doivent examiner une grande quantité de données pourraient surcharger la mémoire et ainsi donc réduire sensiblement les performances de la CPU.

2.7.2 Les politiques de remplacement traditionnel

Parmi les politiques de remplacement les plus populaire , on peut citer [40] :

LRU (Last Recently Used) : Pour tout objet caché, on stocke et on met à jour à chaque nouvel accès la date d'accès. La politique d'éviction consiste alors à éliminer les objets auxquels on n'a pas accédé depuis le plus longtemps. On trouve aussi une variante : LRU-Threshold, qui consiste à ne pas cacher les objets dépassant une certaine taille ;

LFU (least frequently used) : Pour tout objet caché, on stocke et on met à jour un compteur du nombre d'accès à cet objet. La politique d'éviction consiste alors à éliminer les objets auxquels on accède le moins fréquemment ;

SIZE : Remplacer le document de grande taille ;

Log (Size) + LRU : Log-size propose de partitionner l'espace des objets du cache selon une fonction $\log(\text{size})$ et d'évincer des ressources dans chaque partition en appliquant LRU. Le principe étant de favoriser la sortie des documents de plus grande taille .

HYPHER-G : Un affinement de LFU avec prise en compte de la taille et de la dernière date d'accès ;

LRV (Lowest Relative Value) : Inclut la taille pour calculer la valeur qui estime si un document est gardé en cache (remplacer celui qui a la plus petite valeur). Le calcul est basé sur des analyses de la trace des données.

GreedyDual-Size : combine la localité, la taille et le temps de latence pour atteindre une meilleure performance.

2.7.3 Les stratégies de remplacement de cache dédiées aux environnements mobiles

2.7.3.1 La politique TDS (Time and Distance Sensitive)

Le mécanisme de remplacement de cache TDS (Time and Distance Sensitive) [1] proposé dans le cadre de la mémoire cache globale. Elle prend en considération pour le remplacement le facteur de distance et un facteur temporel.

Le premier est la distance δ mesurée par le nombre de sauts par rapport aux points d'accès ou terminaux mobiles, qui ont la donnée demandée. Si la donnée élémentaire avec un δ plus élevé est choisie en tant que victime, alors la latence d'accès serait haute. Par conséquent, la donnée avec une petite valeur δ est choisie en tant que victime. En raison de la mobilité des terminaux, la topologie de réseau peut changer fréquemment, et les valeurs δ peuvent devenir obsolètes. Le deuxième facteur τ , capture le temps écoulé depuis la dernière mise à jour de δ . La valeur de τ est obtenue par la formule suivante :

$$\frac{1}{t_{cur} - t_{update}}$$

où t_{cur} et t_{update} sont respectivement le temps courant et le temps de la dernière mise à jour de δ pour un document dans le cache. TDS calcule pour chaque document i une valeur $v(i) = \delta_i \times \tau_i$.

L'inconvénient de cette politique de remplacement est qu'elle ne prend pas en considération la récence et la fréquence des accès.

2.7.3.2 La politique ZC (Zone coopérative)

Un schéma de mise en cache coopérative pour les réseaux mobiles *ad hoc* est présenté dans [1],[9] et [10]. Dans leurs plans, ils définissent une zone d'un terminal mobile comme un ensemble de mobiles voisins dans un certain nombre de sauts. Si une demande est générée, elle est d'abord diffusée sur les terminaux mobiles dans la zone pour récupérer l'élément de données demandé. Si les données ne sont pas mises en cache dans la zone, une communication peer-to-peer est utilisée pour demander la requête à la station de base. En limitant la diffusion à un nombre de sauts fixe, le coût de l'énergie et la bande passante de l'émission peut être réduite.

Le mécanisme de remplacement de cache ZC (Zone Cooperative) prend en considération pour le remplacement quatre facteurs, la popularité, la distance, la cohérence et la taille

du document. Ces facteurs sont définis comme suit :

La popularité : Elle est la probabilité d'accès d'un document se trouvant dans le cache. Un document avec une petite probabilité d'accès devrait être choisi pour le remplacement, donc le serveur enregistre pour chaque document d_i une probabilité d'accès A_i initialiser à zéro. Pour chaque accès à ce document, cette probabilité est mise à jour suivant cette formule :

$$A_i^{new} = \alpha \frac{1}{t_c - t_l} + (1 - \alpha)A_i^{old}$$

Où

- t_c est le temps courant ;
- t_l est le temps de dernier accès ;
- α est un facteur constant pour peser l'importance de la recense d'accès.

La distance δ : Elle est mesurée en nombre de sauts entre le nœud demandeur du document et le nœud répondant (la source ou à partir d'un cache). Les documents avec une petite distance sont favorisés pour le remplacement. Ce paramètre est particulièrement important pour la conservation de la bande passante et la réduction de la latence.

La cohérence : Un document d_i est valide pour une durée de vie limitée, qui est connue à partir de champ TTL_i . les documents avec des petits TTL sont favorisés pour le remplacement.

La taille du document S_I : Les documents avec une plus grande taille sont choisis pour le remplacement. Dans ce cas le cache peut contenir plus de documents et ainsi peut satisfaire plus de demandes d'accès.

ZC calcule une valeur $v(i)$ pour chaque document i de la manière suivante :

$$w_1 A_i + w_2 \delta_i + w_3 TTL_i + \frac{w_4}{S_i}$$

où w_1 , w_2 , w_3 et w_4 sont des poids associés à chaque facteur tels que :

$$\sum_{j=1}^4 w_j = 1$$

La politique ZC remplace le document ayant la plus petite valeur. L'inconvénient de cette politique est la difficulté réside dans la détermination optimale des poids w_i pour le remplacement.

2.7.3.3 FSDV (Frequency, Size and Distance based Value)

Cette politique de remplacement se base sur le calcul d'une valeur pour chaque document présent dans le cache [1]. Cette fonction inclue quatre paramètres, la fréquence d'accès, la distance en nombre de sauts séparant le nœud qui répond à une demande d'un document de celui qui le demande, le temps écoulé depuis la dernière mise à jour de cette même distance et la taille de document.

Donc pour chaque document d'identifiant id on calcule la fonction $f(id)$ puis on choisit pour le remplacement le document ayant la plus petite valeur. Cette fonction peut se calculer par l'équation suivante :

$$f(id) = \frac{frequence(id) \times distance(id)}{taille(id) \times age_distance(id)}$$

Mais l'utilisation de la fréquence d'accès pose le problème de la pollution de cache. Ce problème est posé lorsqu'un document était très populaire dans une période de temps, alors il peut demeurer toujours dans le cache même s'il reste non référencé pour une longue autre période de temps.

Pour remédier à ce problème, on utilise un autre paramètre, Age_Max qui représente le seuil maximum pour la moyenne de l'âge des documents présents dans le cache.

Quand la moyenne dépasse ce seuil, toutes les fréquences des documents seront divisées par deux.

Dans cette stratégie, le document ayant la plus petite valeur est sélectionné pour le remplacement.

2.8 Conclusion

Après avoir définie la notion de cache coopératif, ainsi que leurs principe de fonctionnement, on a décrit quelques politiques de remplacement de cache coopérative avec des schémas de coopération existant dans la littérature pour savoir comment choisir à remplacer un objet dans un cache et comment prendre en considération les différentes contraintes liées aux réseaux *Ad hoc*.

Les futurs chapitres, ont pour but la modélisation d'un nouveau schéma de coopération dédié aux environnements, en faisant appel à des méthodes graphiques mathématiques est particulièrement les réseaux de Petri Stochastiques Généralisés et Interprétés.

Les réseaux de Pétri

3.1 Introduction

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

3.2 Réseaux de Petri places-transitions

Un Réseau de Pétri est un graphe bi-parti avec deux types de nœuds

- Des places symbolisées par des cercles et représentant des conditions : une ressource d'un système (ex : une machine, un stock,...), l'état d'une ressource du système (machine libre, stock vide,...).
- Des transitions symbolisées par des traits représentant l'ensemble d'événements dont l'occurrence provoque un changement de l'état du système..

- Un ensemble fini d’arcs orientés qui assurent la liaison entre les places et les transitions.

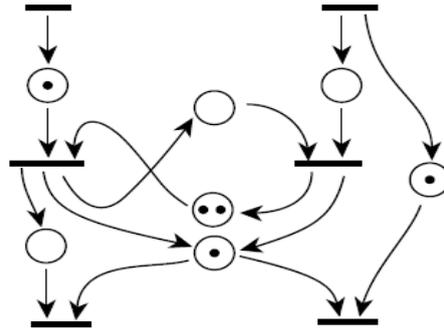


FIGURE 3.1 – Exemple d’un réseau de Petri

3.2.1 Définitions fondamentales

Définition 3.1 :

Un réseau de Petri (RdP) est un 5 – uplet $R = \langle P, T, Pre, Post, W \rangle$ où [20] :

- P est un ensemble fini et non vide de places, $P = (p_1, p_2, \dots, p_m)$ tel que $m = card(P)$ le nombre de places,
- T est un ensemble fini et non vide de transitions, $T = (t_1, t_2, \dots, t_n)$ tel que $n = card(T)$ le nombre de transitions,
- $P \cap T = \emptyset$,
- $Pre : P \times T \longrightarrow \mathbb{N}$ est l’application d’incidence avant, correspondant aux poids des arcs directs reliant les places aux transitions,
- $Post : T \times P \longrightarrow \mathbb{N}$ est l’application d’incidence arrière, correspondant aux poids des arcs directs reliant les transitions aux places.
- W est une fonction qui associe un poids aux arcs sous la forme d’un entier strictement positif tel que $W : (P \times T) \cup (T \times P) \longrightarrow \mathbb{N}$
- t^* (respectivement *t) : l’ensemble des places de sortie (respectivement d’entrée) de la transition t .
- p^* (respectivement *p) : l’ensemble des transitions de sortie (respectivement d’entrée) de la place p .

Lorsque Pre et $Post$ prennent leurs valeurs dans l'ensemble $\{0, 1\}$, le réseau est alors qualifié de réseau **ordinaire**.

Les places dans un RdP peuvent contenir des marques (jetons) qui permettent de modéliser la dynamique du système considéré. Le RdP dans ce cas est dit marqué et la distribution des jetons dans les places est appelée **marquage du RdP**.

Définition 3.2 (Réseau marqué) :

Un **RdP marqué** est un couple $\langle R, M_0 \rangle$ dans lequel

- R est un RdP.
- M_0 est le marquage initial de R définissant ainsi l'état initial du réseau, c'est à dire, avant tout franchissement .

Dans un RdP marqué, chaque place contient un nombre entier (positif ou nul) de marques ou jetons. Le nombre de marques contenu dans une place p_i est noté $M(p_i)$.

Définition 3.3 (Transition sources) :

Une transition t d'un RdP est une transition source si et seulement si *t est un ensemble vide.

Définition 3.4 (Transition puits) :

Une transition t d'un RdP est une transition puits si et seulement si t^* est un ensemble vide.

Définition 3.5 (Boucles élémentaire) :

Une boucle élémentaire est définie par la présence d'une place p et d'une transition t telles que $p \in {}^*t$ et $p \in t^*$.

3.2.2 Comportement dynamique

Le comportement dynamique d'un réseau de Petri représente son évolution au cours du temps et de l'occurrence des événements (on dit sa sémantique opérationnelle). les définitions suivantes spécifie les différents aspect du comportement dynamique d'un réseau de Petri [23].

Définition 3.6 (Règle de fonctionnement) :

Une transition t est dite franchissable ou tirable ou validée pour un marquage M si et seulement si :

$$\forall p \in {}^*t, \quad M(p) \geq \text{pre}(p, t) \quad (3.1)$$

et on le note par $M(t >$, avec *t représente l'ensemble de places en amont de t .

Le franchissement d'une transition validée modifie le marquage des places en amont et en aval de cette transition. Le franchissement d'une transition a pour conséquences le retrait d'un nombre de marques égale aux poids des arcs pour les places en amont de cette transition et l'ajout d'un nombre de marques égal aux poids des arcs pour les places en aval.

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

Le tirage de la transition t provoque le passage d'un marquage M à un marquage M' tel que :

$$\forall p \in P, \quad M'(p) = M(p) + \text{Post}(p, t) - \text{Pre}(p, t) \quad (3.2)$$

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

La figure 3.2 illustre le franchissement d'une transition.

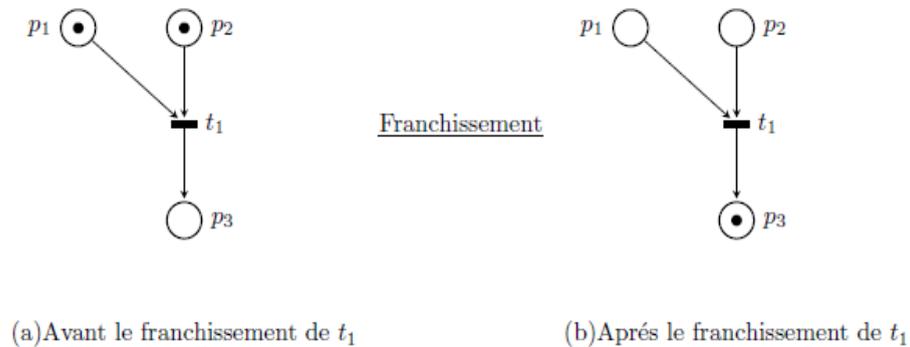


FIGURE 3.2 – Franchissement d'une transition

Remarque

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

Définition 3.7 (Séquence de franchissement) [21] :

Soit R un RdP, $\sigma = t_1 t_2 \dots t_n \in T$ une séquence de transition et M un marquage. La séquence σ est franchissable depuis M si et seulement s'il existe des marquages $M_1, M_2 \dots M_n$ tels que : $M(t_1 > M_1(t_2 > M_2(\dots M_{n-1}(t_n > M_n$.

Si la séquence de franchissement σ est réalisable à partir de M_i , le marquage obtenu M_k est donné par l'équation fondamentale :

$$M_k = M_i + C \cdot \bar{\sigma} \tag{3.3}$$

où C est la matrice d'incidence du RdP et $k = 1..n$. Le tir de σ conduit alors au marquage M_n . On le note $M(\sigma > M_n$.

Remarque

Notons que, l'équation fondamentale ne garantit pas que σ soit franchissable mais nous permet simplement de trouver le nouveau marquage lorsqu'on connaît le marquage M_I et la séquence franchissable σ .

Définition 3.8 (Marquage accessible) :

Soit (R, M_0) un RdP marqué.

Un marquage M est accessible si et seulement s'il existe une séquence de franchissement σ telle que $M_0(\sigma > M$.

Définition 3.9 (Ensemble d'accessibilité) [22] :

Soit (R, M_0) un RdP marqué.

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

$$Acc(R, M_0) = \{M \in \mathbb{N}^p / \exists \sigma \in t^* \text{ tel que } M_0(\sigma > M\} \tag{3.4}$$

Définition 3.10 (Graphe des marquages accessible) [22] :

Soit (R, M_0) un RdP marqué.

Le graphe des marquages accessibles (ou graphe d'accessibilité) de ce réseau, noté $G(R, M_0)$ est défini comme le graphe dont les nœuds (ou sommets) sont les marquages accessibles de $Acc(R, M_0)$ 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 $M \xrightarrow{t} M'$.

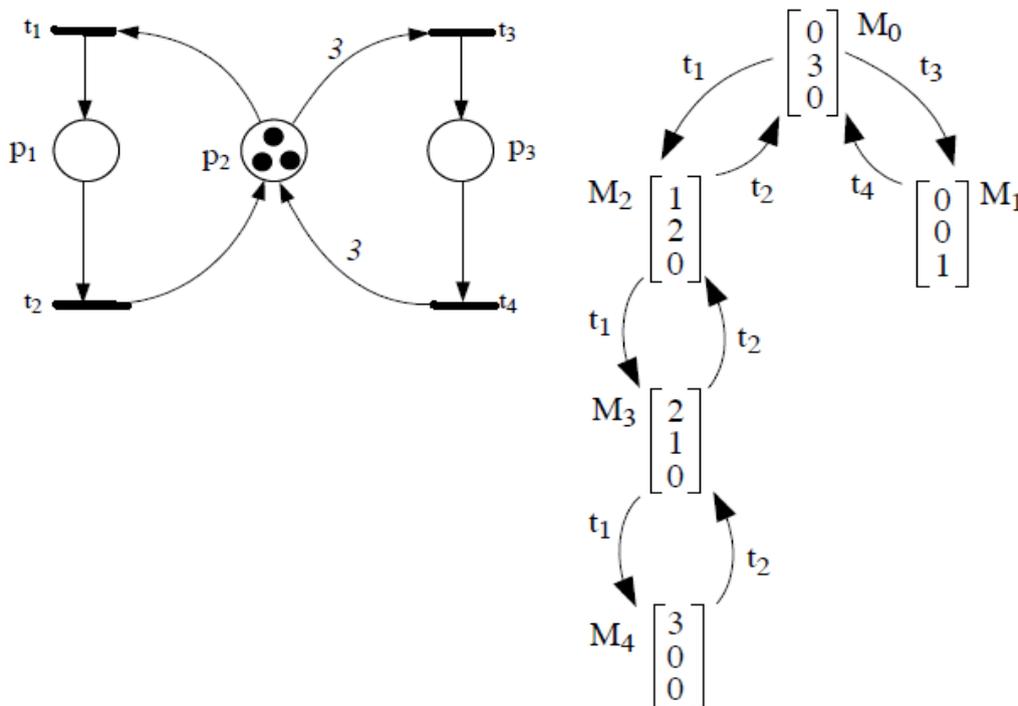


FIGURE 3.3 – Exemple d'un RdP avec le graphe des marquage associé

Définition 3.11 (Graphe de couverture) :

Arborescence de couverture et graphe de couverture : On peut avoir une situation où l'ensemble d'accessibilité n'est pas fini, le graphe des marquages est ainsi infini et donc il n'est pas possible de le construire dans sa globalité. Dans le cas où le graphe des marquages est infini on remplace le graphe des marquages par le graphe de couverture, qui est une approximation. Ce graphe de couverture est fini, mais fournit moins d'information que le graphe des marquages.

3.3 Propriétés des réseaux de Petri

Les réseaux de Petri permettent de modéliser des systèmes complexes [20]. Ceux-ci se doivent de répondre à un certain nombre d'exigences (absence de blocage, préservation des fonctionnalités, existence d'un fonctionnement en régime permanent, etc...). Nous définissons maintenant ces propriétés élémentaires.

3.3.1 Les propriétés dynamiques

3.3.1.1 Réseau de Petri borné et Réseau sauf

Définition 3.12 (bornétude) :

Une place p_i est **bornée** pour un marquage initial M_0 si pour tout marquage accessible à partir de M_0 , le nombre de marques dans p_i reste borné. Elle est dite **k-bornée** si le nombre de marques dans p_i est toujours inférieur ou égal à k . Un RdP marqué est **k-borné** si toutes ses places sont k -bornées.

Définition 3.13 (RdP sauf) :

Un RdP marqué est **sauf** ou **binaire** pour un marquage initial M_0 s'il est **1-borne**.

3.3.1.2 Vivacité et blocage

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

Définition 3.14 (Vivacité) :

Une transition t_j est **vivante** pour un marquage initial M_0 si pour tout marquage accessible M_k , il existe une séquence de franchissements à partir de M_k contenant t_j . Si une transition t_j est vivante alors, à tout instant, on sait que t_j peut être franchie dans le futur.

Dans le cas d'un RdP modélisant un système fonctionnant en permanence, si une transition n'est pas vivante et si une fonction du système est associée au franchissement de cette transition, cela veut dire qu'à partir d'un certain instant, cette fonction ne sera plus disponible dans le futur, ce qui peut traduire une erreur ou une panne.

Définition 3.15 :

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

Définition 3.16 (Blocage) :

Un **blocage** (ou état puits) est un marquage pour lequel aucune transition est validée. Un RdP marqué est dit sans blocage pour un marquage initial M_0 si aucun marquage accessible n'est un blocage.

3.3.1.3 Conflits et parallélisme

Définition 3.17 (Conflits structurel et effectif) [22] :

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

$$\exists p \in P, Pre(p, t_1) \times Pre(p, t_2) \neq 0. \quad (3.5)$$

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

$$M(t_1) > \text{ et } M(t_2) > \quad (3.6)$$

et

$$\exists p \in P, M(p) < Pre(p, t_1) + Pre(p, t_2). \quad (3.7)$$

Définition 3.18 (Parallélisme) :

Deux transition t_1 et t_2 sont en **parallèles structurellement** si

$$(Pre(., t_1))^T \times (Pre(., t_2)) = 0. \quad (3.8)$$

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

$$M(t_1 > \text{ et } M(t_2 > . \quad (3.9)$$

3.3.1.4 La réversibilité

Définition 3.19 :

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

3.3.1.5 Réinitialiabilité et état d'accueil

Définition 3.20 (État d'accueil) : [21]

Un RdP a un **état d'accueil** M_a pour un marquage initial M_0 si pour tout marquage M_i accessible depuis M_0 , il existe une séquence de franchissement σ tel que M_a est accessible depuis M_i en franchissant σ .

Définition 3.21 (Réinitialiabilité) :

Le RdP est dit **réinitialisable** pour un marquage initial M_0 si M_0 est un état d'accueil.

3.3.1.6 Graphe d'événement, la connexité et la forte connexité

Définition 3.22 (La connexité) : [21]

Un **Graphe d'événement** est un RdP dans lequel chaque place a exactement une transition d'entrée et une transition de sortie. Il est caractérisé par l'absence de conflits structurels.

Définition 3.23 (La connexité) :

Un RdP est dit connexe si et seulement si il existe un chemin (non forcément orienté) depuis n'importe quel nœud vers un autre nœud. Un nœud peut être une place ou une transition.

Définition 3.24 (la forte connexité) :

Un RdP est dit fortement connexe si et seulement si il existe un chemin orienté entre chaque paire de nœuds de types différent.

3.3.2 Les propriétés structurelles

3.3.2.1 Représentation matricielle d'un RdP

La matrice correspondant à Pre et celle correspondant à Post est construit de la manière suivante. Les lignes identifient les places et les colonnes, les transitions. Une intersection (i,j) , dans Pre, correspond au coût associé à l'arc menant de la place p_i à la transition t_j , tandis qu'une intersection (i,j) , dans Post, correspond au coût associé à l'arc menant de la transition t_j à la place p_i . La matrice d'incidence d'un RdP est la matrice entière $C \in Mat(P, T)$ définie par

$$\forall (p, t) \in P \times T, C(p, t) = Post(p, t) - Pre(p, t) \quad (3.10)$$

3.3.2.2 Les P-invariants et les T-invariants

La matrice d'incidence C définie précédemment nous permet de définir les notions importantes d'invariants

Définition 3.25 (P-invariant) [21] :

Un **P-invariant** est un vecteur X de dimension m , dont ses éléments sont des entiers, vérifiant :

$$X^T.C = 0. \quad (3.11)$$

Soient X un P-invariant et M_0 un marquage initial d'un RdP quelconque. Pour tout marquage M_i atteignable depuis M_0 la relation suivante est vérifiée :

$$X^T.M_0 = X^T.M_i \quad (3.12)$$

Cette relation déduite à partir de l'équation fondamentale (3.4), est une relation de conservation du nombre de marques. Elle exprime qu'une certaine combinaison linéaire entre les jetons figurant dans les places d'un P-invariant pour tous les marquages accessibles depuis M_0 est constante.

Définition 3.26 (T-invariant) [21] :

Un **T-invariant** est un vecteur Y d'entiers positifs et de dimension n vérifiant :

$$C.Y = 0 \quad (3.13)$$

Soit σ une séquence de franchissements et $\bar{\sigma}$ son vecteur caractéristique. Si $\bar{\sigma} = Y$ est un T-invariant et si M_0 est le marquage initial, le marquage M_i obtenu en franchissant la séquence σ est tel que :

$$M_i = M_0 \tag{3.14}$$

Autrement dit que σ est une séquence cyclique.

Deux définitions découlent des définitions précédentes. La définition d'une composante conservative et celle d'une composante répétitive. Un ensemble de places est une composante conservative si et seulement si ses places forment un P-invariant. Un ensemble de transitions forme une composante répétitive si et seulement si elles forment un T-invariant.

3.4 Extensions des réseaux de Petri

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

3.4.1 Les réseaux de Petri étendus

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

Arcs inhibiteurs

La plus simple extension d'un RdP, permettant le test à zéro, est l'arc inhibiteur. Il sera représenté sur le RdP par un arc terminé par un petit cercle. Cet arc partira nécessairement d'une place p et aboutira à une transition t . la transition t ne sera alors franchissable que lorsque le marquage de la place amont sera strictement inférieur au poids de l'arc inhibiteur.

Définition 3.27 (RdP à arc inhibiteur) [24] :

Un RdP à arc inhibiteur est défini par un couple $\langle R, IN \rangle$ tel que :

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

Pour un marquage M donnée, la condition du tirage d'une transition t , devient alors

$$\forall p \in P, \quad M(p) \geq Pre(p, t) \wedge (M(p) < IN(p, t) \vee IN(p, t) = 0) \quad (3.15)$$

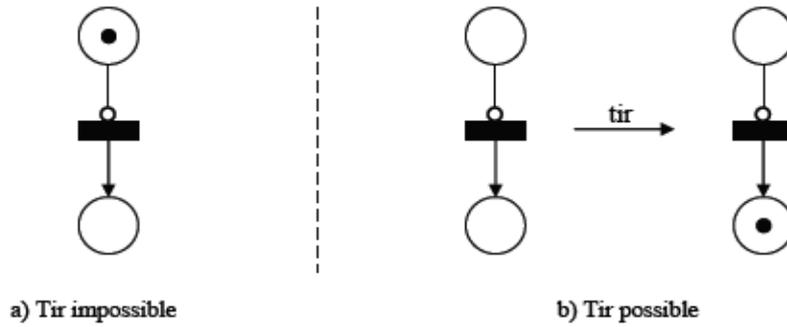


FIGURE 3.4 – Franchissement de transition avec arc inhibiteur

Priorité

Lors d'un conflit effectif, il convient de préciser un choix pour la transition à tirer [24]. Ce choix peut être aléatoire ou défini par une certaine notion sur ces transitions. Une de ces possibilités consisterait à associer à chaque transition du RdP, un nombre entier croissant avec la priorité, établissant ainsi un ordre total. Il suffit souvent d'introduire une relation binaire $>$, c'est-à-dire vérifiant

$$\forall n \geq 1, \quad \neg(\exists(t_1, t_2 \dots t_{n-1}, t_n) \in T^n, t_1 > t_2 > \dots t_{n-1} > t_n) \quad (3.16)$$

où \neg représente la négation logique.

3.4.2 Les Réseaux de Petri T-temporels

Les réseaux de Petri T-temporels (RdPT) [25] étendent les réseaux de Petri avec des intervalles de temps $[a(x), b(x)]$ associés à chaque transition t du réseau. Pour être tirée,

une transition t doit non seulement être sensibilisée mais aussi l'avoir été continûment pendant une durée comprise entre $a(x)$ et $b(x)$.

Donc une fois la date $b(x)$ atteinte, la transition t devra obligatoirement être franchie, à moins d'avoir été désensibilisé puis nouvellement sensibilisée par le tir d'une autre transition.

Définition 3.28 (RdP T-temporel) :

Un RdP temporel est un couple $\langle R, I_s \rangle$ où R est un RdP marqué et $I_s : T \rightarrow I^+$ est une fonction appelée Intervalle statique, avec I^+ l'ensemble des intervalles réels non vides à bornes rationnelles non négatives.

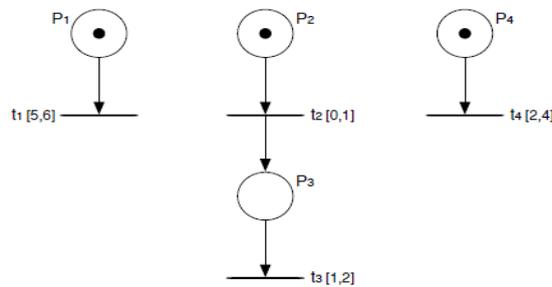


FIGURE 3.5 – Un exemple de réseau de Petri temporel

3.4.3 Les Réseaux de Petri temporisés

Un réseau de Petri T-temporisé est un couple $\langle R, Tempo \rangle$ avec [26] :

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

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

3.4.4 Les Réseaux de Petri colorés

Dans les Réseaux de Petri Colorés (RdPC), en associant à chaque jeton une valeur appelée la couleur du jeton et permet de distinguer les jetons entre eux [23]. Le franchissement d'une transition est alors conditionné par la présence dans les places en entrée du nombre de jetons nécessaires, qui satisfont les couleurs qui étiquettent les arcs. Cela est décrit dans le RdP coloré en associant une fonction de precondition $f = Pre(P, T)$ à chaque arc entrant reliant une place p à une transition t qui représente la valeur renvoyée par cette fonction est la couleur du jeton autorisée à passer par cet arc, et une fonction de postcondition $f' = Post(T, P)$ à chaque arc sortant reliant une transition t à la place P qui représente la valeur renvoyée par cette fonction est la couleur du jeton pour laquelle la validation de la transition t est envisagée.

Remarque

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 RdP simple au RdP coloré est appelée opération de *pliage* et l'opération inverse est appelée *dépliage*.

3.4.5 Les Réseaux de Petri interprétés

Les réseaux de Petri interprétés aussi appelés RdP Prédicat/Transition présentent la première classe de réseaux de petri à haut niveau introduit par Genrich et Lautenbach en 1981 [42]. Dans un RdP Interprété, le contrôle et les données sont séparés : le contrôle est modélisé par un RdP Place-transition, tant dis qu'un environnement externe fait d'un ensemble fini de variable est associé à l'état du réseau pour traiter les données. Un prédicat sur cet environnement et une action qui puisse modifier cet environnement, sont associés à chaque transition [27].

Dans la description d'une transition, il est possible de spécifier une liste d'expressions booléennes qui doivent être nécessairement vérifiées pour que la transition soit valide. Ces expressions sont séparées par des virgules. Il s'agit des conditions de validation.

Définition 3.29 (Réseau de Petri Interprété) :

Un RdP Interprété est un quadruplet $\langle R, X, Pred, Act \rangle$ où

- R est un RdP marqué ;

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

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

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

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

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

3.4.6 Les Réseaux de Petri stochastiques

Les réseaux de Petri stochastiques (RdPS) sont surtout utilisés pour modéliser les phénomènes de file d'attente dans le domaine de l'informatique et dans les réseaux de communications [26]. Ce sont des modèles formels qui permettent à la fois des analyses qualitatives, par la vérification de propriétés cité précédemment, et quantitatives par l'évaluation des performances.

3.4.6.1 Définition d'un RdPS

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

chaque arc par un taux de franchissement. Le comportement aléatoire du RdPS est alors identique à celui d'une chaîne de Markov.

Définition 3.30 (Réseau de Petri stochastique) [29], [33] :

Un Réseau de Petri stochastique (RdPS) est un couple $\langle R, Taux \rangle$ tel que :

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

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

3.4.6.2 Durée de sensibilisation stochastique

Au lieu de prendre, de façon égale, toutes les valeurs d'un intervalle $([a(x), b(x)])$ comme dans le réseau de Petri temporels, on peut supposer que la durée de sensibilisation est une variable stochastique θ avec une distribution de probabilité exponentielle [24].

$$Pr[\theta \leq x] = 1 - e^{-\lambda x} \quad (3.17)$$

La probabilité $Pr_\theta(x)$ est la probabilité pour que le franchissement ait lieu avant un temps x c'est-à-dire pour que la durée de sensibilisation soit inférieure à x .

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

$$\theta = \int_0^\infty (1 - Pr_\theta(x)).dx = \int_0^\infty e^{-\lambda x}.dx = \frac{1}{\lambda} \quad (3.18)$$

où λ est appelé *taux* de transtion.

3.4.7 Les Réseaux de Petri stochastiques généralisés

3.4.7.1 Définition d'un RdPSG

Dans les réseaux de Petri stochastiques, toutes les transitions sont associées à une temporisation selon une distribution de loi exponentielle. Les réseaux de Petri stochastiques généralisés ont été introduits par Ajmone [30], [31] afin de limiter certaines restrictions.

Cette classe de réseau autorise deux types de transitions :

- Les transitions temporisées basées sur des distributions exponentielles.
- Les transitions déterministes à temporisation nulle (transition immédiate). Ces transitions sont franchies immédiatement dès lors qu'elles sont sensibilisées.



FIGURE 3.6 – Transition immédiate et transition stochastique

Les transitions immédiates permettent notamment de modéliser les synchronisations, ou bien encore d'approximer des temporisations très faibles par rapport aux transitions stochastiques. Graphiquement, les transitions immédiates sont généralement représentées par un trait noir, et les transitions temporisées par un rectangle blanc.

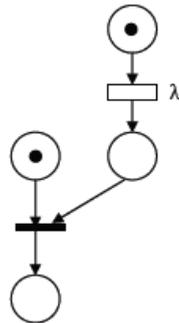


FIGURE 3.7 – Exemple de réseau de Petri stochastique généralisé

L'existence de transitions franchies immédiatement a pour conséquence de faire coexister deux types d'états :

1. **les états tangibles** : les transitions sensibilisées sont des transitions temporisées.

2. **les états évanescents (ou virtuels ou instantanés)** : il existe au moins une transition sensibilisée qui est immédiate. Celle-ci est alors franchie immédiatement. Le temps de séjour dans cet état est donc nul.

Dans un marquage où plusieurs transitions sont sensibilisées, les transitions immédiates sont obligatoirement tirées avant les transitions temporisées. On définit ainsi deux types de marquages : les marquages *tangible* dans lesquels seules des transitions temporisées sont sensibilisées, et les marquages *évanescents* dans lequel au moins une transition immédiate est sensibilisée.

La définition formelle d'un RdPSG est donnée par la définition ci-après.

Définition 3.31 (Réseau de Petri Stochastique Généralisé) :

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

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

3.4.7.2 Analyse des RdPSG

L'analyse des RdPSG est basée sur deux aspects essentiels [22]

Analyse qualitative : consiste à vérifier les propriétés que nous avons présenté dans la section 3.3 tel que la vivacité, la bornitude, les états d'accueil, ...etc.

Analyse quantitative : Cette partie d'analyse consiste à calculer les probabilités stationnaires et les indices de performance. Elle est basée sur la chaîne de Markov associée au RdPSG. Cette chaîne peut être construite à partir du graphe des marquages accessibles de la manière suivante :

la première étape consiste à éliminer tous les états évanescents. À partir de ce nouveau réseau, on cherche le graphe des marquages relatif aux états tangibles. Le graphe de Markov possède alors la même structure que le graphe des marquages. Il ne reste plus qu'à étiqueter les arcs par les taux de franchissement dépendants

des taux de transition du réseau de Petri et du marquage. On obtient le générateur infinitésimal Q de cette CTMC qui est alors une matrice carrée de dimension $(r \times r)$ (r le nombre fini de marquages tangibles du RdPSG) qui regroupe l'ensemble des taux de transition d'un marquage vers un autre.

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

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

Proposition 3.1 *Le graphe de marquage d'un RdPSG est isomorphe à une chaîne de Markov à temps continue.*

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

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

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

3.4.7.3 Evaluation des indices de performances

Une fois le modèle est obtenu, on vérifie ses propriété qualitative pour déduire son ergodicité pour faire de l'analyse quantitative. Si le modèle est ergodique, alors la distribution de probabilité des marquages à l'état stationnaire existe et elle est unique [22].

Plusieurs indices de performances peuvent être calculé. Parmi ces indices les plus important on cite :

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

$$\bar{\lambda}(t_j) = \sum_{M_j \in E(t_i)} \lambda_i(M_j) \pi_j \tag{3.19}$$

où :

$E(t_i)$ est l'ensemble des marquages où la transition t_i est franchissable.

$\lambda(M_j)$ est le taux de franchissement de t_i en M_j .

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

$$n(p) = \sum_{i:M_i \in E} M_i(p)\pi_i \quad (3.20)$$

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'une marque dans un sous-réseau** : Le délai moyen qu'un jeton passe dans une partie S (sous-réseau) d'un RdPSG à l'état stationnaire, peut être calculé en utilisant la formule de Little :

$$E[T] = \frac{E[N]}{E[\beta]} \quad (3.21)$$

où :

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

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

3.5 Conclusion

Les réseaux de Petri ont prouvé leurs adaptation à l'analyse des systèmes de communication informatique. L'analyse d'un réseau de Petri peut révéler des caractéristiques importantes du système concernant sa structure et son comportement dynamique. Les résultats de cette analyse sont utilisés pour évaluer le système et en permettre la modification ou l'amélioration.

Dans ce chapitre, et après l'introduction informelle, nous avons introduit les principales définitions associées au RdP places-transition, nous nous sommes également intéressé aux différents propriétés dynamiques et structurelles exprimables par RdP. Par suite, nous avons décrit les différentes extensions qui sont des améliorations du modèle de RdP classique. Parmi ces extensions, nous allons s'intéresser dans ce travail à la combinaison entre les réseaux de Petri Stochastiques Généralisés et les réseaux de Petri Interprétés pour effectuer la modélisation et l'évaluation des performances d'un schéma de coopération géré par une politique de remplacement de cache, et qui fera l'objet but du prochain chapitre.

Modélisation et Évaluation des performances

4.1 Introduction

Les outils de modélisation qui ont été retenus dans le cadre de cette étude sont les réseaux de Petri Stochastiques Généralisés Interprétés. Les RdPSG et interprétés permettent la modélisation, la simulation et l'évaluation des performances des systèmes complexes. Ces outils sont largement utilisés pour analyser plusieurs systèmes, et particulièrement les systèmes informatiques. Dans ce chapitre, nous allons modéliser le schéma de cache coopératif en utilisant ces types de réseaux en évaluant les performances liées à ce dernier.

4.2 Présentation du schéma de coopération

Dans un réseau *Ad hoc*, chaque nœud est équipé d'une petite mémoire appelée "cache". Cette mémoire cache est utilisée pour sauvegarder les documents déjà demandés, et qui sont les plus susceptibles d'être réutilisés dans le futur proche.

Le principe de la coopération des caches, est le fait de permettre à un nœud, dans une situation d'un cache miss local après une requête pour un document, de pouvoir accéder aux documents cachés dans les autres nœuds.

Dans ce schéma de coopération, un certain nœuds proportionnel au nombre total de nœud dans le réseau, sont élues pour jouer le rôle de fournisseurs de services du cache. Cette élection est faite selon un algorithme exécuté de manière périodique pour désigner à chaque fois un autre groupe de fournisseurs de services, afin d'assurer une bonne répartition de charge.

Lorsqu'un nœud est élu comme fournisseur de services cache, il diffuse un message d'infor-

mation pour les voisins à un ou deux sauts afin de les solliciter à envoyer le résumé de leur cache. Si un nœud reçoit plusieurs messages le sollicitant à envoyer son résumé de cache, il ne répond qu'à un seul, en choisissant le plus proche. Ce nœud est tenu de valider son résumé périodiquement. Si le fournisseur de services cache s'aperçoit qu'un résumé n'est toujours pas validé au bout de la troisième période, alors, ce résumé sera supprimé. Lorsqu'un nœud compte qu'il est hors de la portée de communication de son fournisseur de services cache, il diffuse alors une demande d'adhésion à un autre fournisseur de services cache.

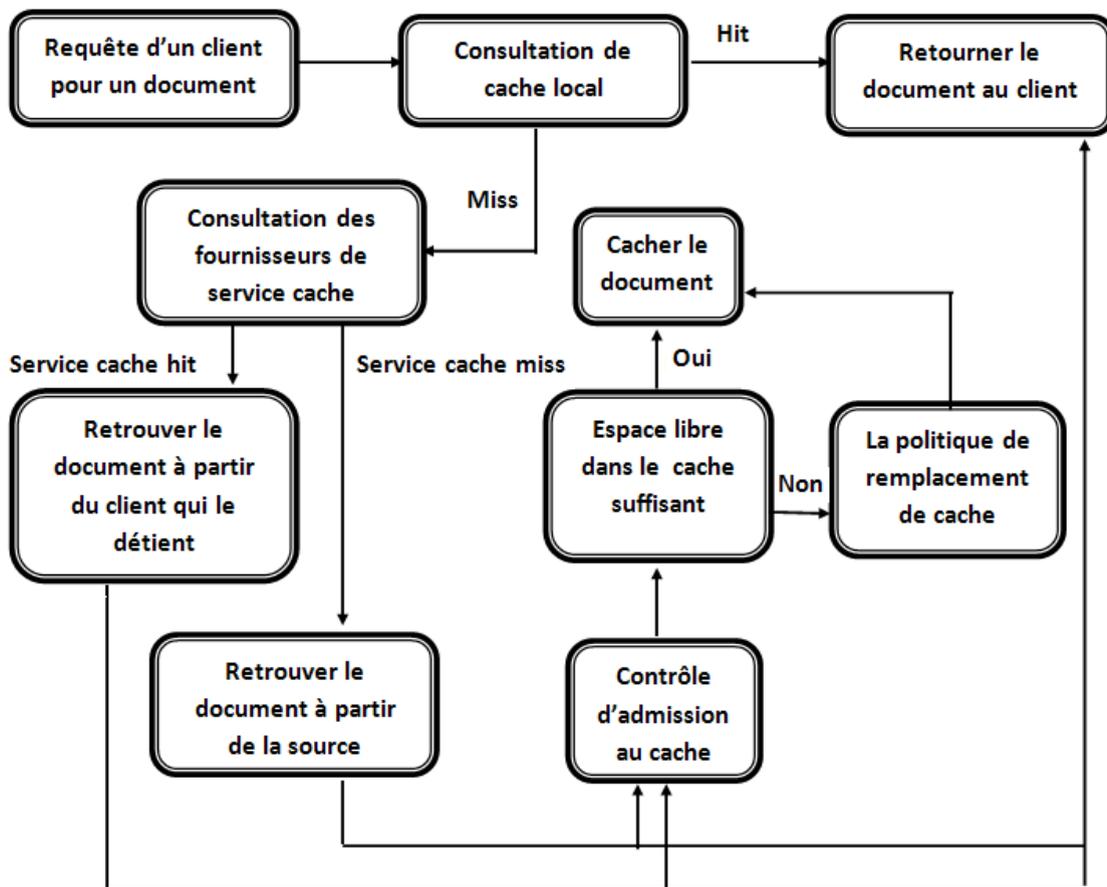


FIGURE 4.1 – Diagramme de coopération des caches

Lorsqu'une requête pour un document est lancée pour un client, ce dernier consulte d'abord son cache local, si le document s'y retrouve, alors on dit que c'est un succès de cache local (Local cache hit), sinon c'est un défaut de cache (local cache miss).

Dans le cas d'un cache miss local, le client envoie une requête directement au premier fournisseur de services cache qui n'est pas forcément le détenteur de son résumé de cache à la réception de cette requête, le fournisseur de services cache recherche dans sa mémoire une association entre l'identité du document recherché et l'identité d'un nœud du réseau. Si une association est trouvée, alors le fournisseur de services caches renvoie au client l'identité du nœud qui possède le document recherché. Par la suite, le client contacte directement ce nœud.

Si aucune association n'est trouvée, le client contacte le dernier fournisseur de services cache et procède de la même manière. Au bout du troisième fournisseur de services cache, si le document n'est toujours pas trouvé, alors le client contacte directement le serveur d'origine.

Quand un nœud reçoit une réponse pour un document à partir d'un fournisseur de services cache ou bien de l'origine, et si l'espace libre dans le cache est suffisant pour recevoir ce document, alors il est inséré directement dans le cache, sinon la politique de remplacement de cache est invoquée pour libérer de l'espace pour ce nouveau document. La figure 4.1 montre le comportement du cache recevant une requête pour un document.

4.3 La modélisation via les réseaux de Petri

Parmi les différentes classes des réseaux de Petri, nous avons privilégié les réseaux de Petri stochastiques généralisés (RdPSG) principalement pour modéliser le schéma de coopération parce qu'ils permettent de fournir des résultats en régime permanent, qui donneront directement les probabilités stationnaires du système. De plus, ces outils prennent en considération l'aspect aléatoire des paramètres du système (ici, l'arrivée des requêtes suit une loi de Poisson et les durées de services (de recherche d'un document) sont exponentiellement distribuées). L'interprétation de ce réseau est liée à l'existante du document recherché au niveau du réseau.

4.3.1 Définition du modèle R_1

Soit RdP le réseau de Petri associé au schéma de coopération défini par :

1. R_1 est un réseaux de Petri Stochastique Généralisé.
2. $X = \{x\}$ tel que $x = [\exists, Id]$ signifie que le document d'identifiant Id existe dans une place précise.

3. Pred est l'application qui associe à chaque transition un prédicat, qui est une expression booléenne, construite sur la variable x de X . Par défaut, la valeur du prédicat associée à une transition est vraie.

Les hypothèses du modèle R_1

Les hypothèses du modèle sont définis comme suit :

- Le nombre de fournisseurs de service cache est limité à trois.
- Les arrivées des requêtes sont indépendantes les unes des autres suivant un processus de Poisson de paramètre λ .
- Le service de recherche s'effectue au niveau du cache local, des trois fournisseurs de service cache, et de serveur d'origine et les durées de services suivent une loi exponentielle de paramètres :
 - ✓ μ_1 le taux de recherche dans le cache local.
 - ✓ μ_2 le taux de recherche dans l'un des fournisseurs de services cache et $\frac{1}{2}\mu_2$ est le taux de réponse dans les cas de succès .
 - ✓ μ_3 le taux d'aller chercher le document à l'origine et $\frac{1}{2}\mu_3$ est le taux de réponse correspondant.

Le modèle de coopération

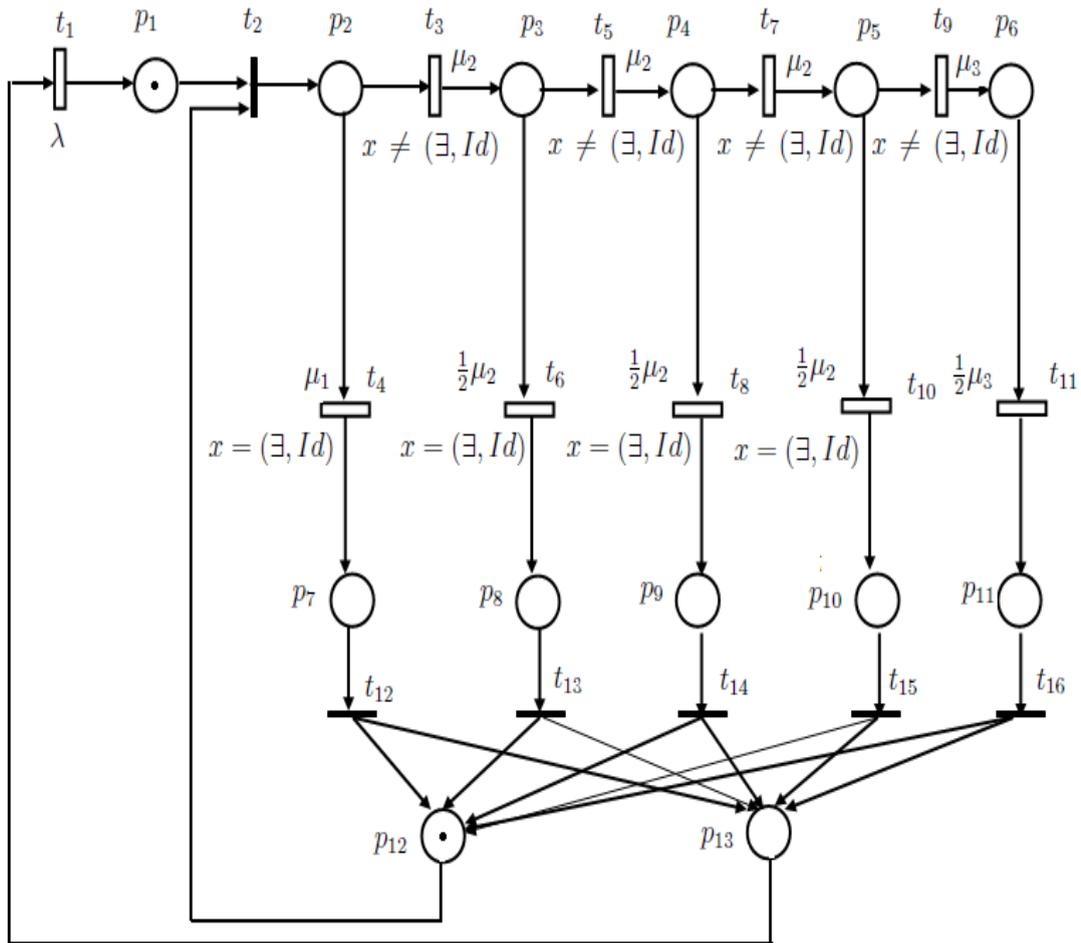


FIGURE 4.2 – Modélisation du schéma de coopération via les RdPSG et Interprétés

Les places

Les places	Description
p_1	La file d'attente des requêtes.
p_2	La place qui représente le cache local.
p_3	Le succès à partir du cache local.
p_4	Le premier fournisseur du service cache (FSC1).
p_5	Le succès à partir du FSC1.
p_6	Le deuxième fournisseur du service cache (FSC2).
p_7	Le succès à partir du FSC2.
p_8	Le troisième fournisseur du service cache (FSC3).
p_9	Le succès à partir du FSC3.
p_{10}	La place du serveur d'origine.
p_{11}	Le document renvoyé à partir de l'origine.
p_{12}	Le service de recherche d'un document.
p_{13}	La fin de recherche.

Les transitions

Les transitions	Description	Type
t_1	L'arrivée des requêtes	stochastique
t_2	Début de recherche d'un document dans le cache local	immédiat
t_3	Début de recherche dans FSC1	stochastique
t_4	Fin de recherche local avec un succès	stochastique
t_5	Debut de recherche d'un document dans FSC2	stochastique
t_6	Fin de recherche au niveau de FSC1 avec un succès	stochastique
t_7	Début de recherche d'un document dans FSC3	stochastique
t_8	Fin de recherche au niveau de FSC2 avec un succès	stochastique
t_9	Debut de recherche à partir de l'origine	stochastique
t_{10}	Fin de recherche au niveau de FSC3 avec un succès	stochastique
t_{11}	Fin de recherche à partir de l'origine	stochastique
t_{12}	/	immédiat
t_{13}	/	immédiat
t_{14}	/	immédiat
t_{15}	/	immédiat
t_{16}	/	immédiat

4.3.2 Interprétation du modèle R_1

Soit M_0 le marquage initial du modèle tel que

$$M_0 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0)$$

- La présence d’un jeton dans la place p_1 signifie qu’une requête est arrivée à la file d’attente pour être servie. Les requêtes arrivent selon un processus quasi-aléatoire de Poisson de paramètre λ .
 - La présence d’un jeton dans la place p_{12} signifie que le service de recherche d’un document n’est pas encore entamé.
 - La première transition à franchir est la transition t_2 , ce que signifie le début de recherche d’un document d’identifiant Id .
 - À l’arrivée du jeton à la place p_2 , ça veut dire que la requête doit tout d’abord consulter le cache local, et la prochaine transition à franchir est soit t_3 ou bien t_4 . En fait, le franchissement de t_3 ou t_4 est lié à la condition d’existence du document dans le cache, c’est pour cette raison que l’interprétation des transitions est effectuée en associant à chaque transition une valeur booléenne x qui doit prendre la valeur vraie.
 - Si le document existe dans le cache local, l’instruction $x = (\exists, Id)$ est exécutée, une marque est ensuite déposée dans la place p_7 avec un taux de réponse μ_1 .
 - Si le document n’existe pas dans le cache (i.e $x \neq (\exists, Id)$ qui est exécutée), la transition t_3 est franchie avec un taux μ_2 qui dépose un jeton dans la place p_3 pour commencer la recherche dans le premier fournisseur de service cache FSC1.
 - Si le résumé de document Id existe dans FSC1, dans ce cas, la prochaine transition à franchir est t_6 avec un taux $\frac{1}{2}\mu_2$, à ce moment, un jeton est déposé dans la place p_8 qui signifie qu’un succès est effectué à partir du FSC1.
 - Si le résumé du document Id n’existe pas dans FSC1, l’instruction $x \neq (\exists, Id)$ est exécutée, et la transition t_5 est franchie avec un taux μ_2 pour commencer la recherche à partir du deuxième fournisseur de services cache.
- Avec le même raisonnement effectué pour FSC1, on interprète FSC2, FSC3.
- Dans le cas où le résumé du document Id n’existe pas dans FSC3, l’instruction $x \neq (\exists, Id)$ est exécutée et la transition t_9 est franchie avec un taux μ_3 et un jeton est déposé dans la place p_6 . La prochaine transition à franchir est t_{11} avec un taux $\frac{1}{2}\mu_3$ qui dépose un jeton dans la place p_{11} .
 - Si un jeton est présent dans la place p_{11} , cela signifie que la requête est servie à

partir du serveur l'origine.

- Si les transitions t_4 , t_6 , t_8 , ou bien t_{10} sont franchissables, cela signifie que la requête a été servie à partir du cache local, FSC1, FSC2, FSC3 respectivement, et un succès est effectué.
- Les transitions t_{12} , t_{13} , t_{14} , t_{15} , t_{16} présentent la fin de recherche et un jeton est déposé dans p_{13} pour signifier que le client est servi.
- L'existence d'un arc de p_{13} à t_1 signifie qu'une nouvelle requête est lancée avec un taux λ .

4.3.3 L'analyse du modèle R_1

4.3.3.1 L'analyse qualitative

Le modèle de RdP R_1 admet comme propriétés qualitatives :

- La bornitude : toutes les places du réseau sont 1__ bornées donc on déduit que le RdP est 1__ borné.
- La vivacité : toutes les transitions sont vivantes donc le RdP est vivant.
- La propriété de graphe d'événement : Le RdP n'est pas un graphe d'événement.
- Les états de blocage : le RdP ne possède aucun état de blocage.
- La connectivité : le RdP est fortement connexe.
- La réinitialiabilité : le RdP est réinitialisable et admet M_0 comme état d'accueil.

4.3.3.2 L'analyse quantitative

Cette analyse quantitative consiste à étudier le graphe des marquages et les indices de performances associés au modèle R_1 . Puisque ce dernier est borné et admet M_0 comme état d'accueil donc il est ergodique, et l'évaluation des performances à l'état stationnaire est possible. Le graphe des marquages associé au modèle du réseau proposé est donné par la figure 4.3.

La construction de la chaîne de Markov

Pour le calcul des différents paramètres de performances du RdPSG correspondant, il faut tout d'abord construire la chaîne de Markov associée au graphe des marquages pour calculer les probabilités d'état en régime permanent, i.e. les probabilités de se situer dans un marquage bien précis.

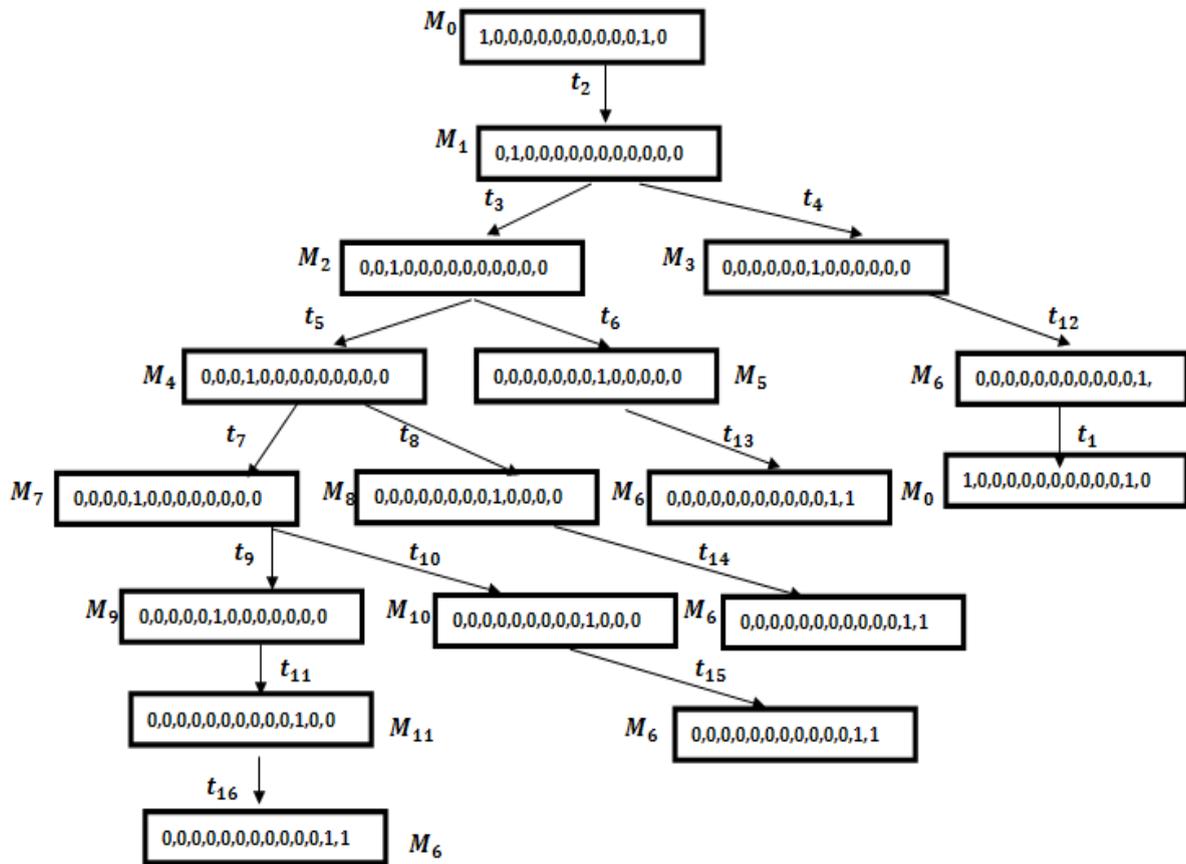


FIGURE 4.3 – Le graphe des marquages

On construit cette chaîne de Markov à partir du graphe des marquages accessibles où les états de la chaîne de Markov représentent les marquages tangibles de ce dernier. Les marquages évanescents sont fusionnés avec leurs successeurs (marquages tangibles). Les taux de transition de la chaîne de Markov à temps continu sont les taux de tir des transitions du RdPSG.

La figure 4.4 montre le graphe des marquage réduit à partir des états tangibles, autrement dit, c'est la chaîne da Markov associé.

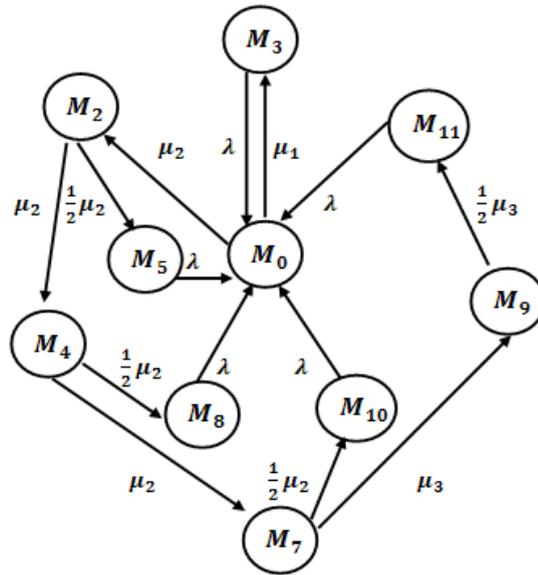


FIGURE 4.4 – La chaîne de Markov

Comme la chaîne de Markov de la figure 4.4 est ergodique, alors la distribution de probabilité des marquages à l'état stationnaire existe et elle est unique. On note $\pi = (\pi_1, \pi_2, \dots, \pi_n,)$ cette distribution de probabilité, où π_i , est la probabilité que le processus est à l'état M_i et $i = 1..n$.

Le générateur infinitésimal Q associé à la chaîne de Markov est donné par :

$$Q = \begin{pmatrix} -\mu_1 - \mu_2 & \mu_2 & \mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{3}{2}\mu_2 & 0 & \mu_2 & \frac{1}{2}\mu_2 & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & -\lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{3}{2}\mu_2 & 0 & \mu_2 & \frac{1}{2}\mu_2 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & -\lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{2}\mu_2 - \mu_3 & 0 & \mu_3 & \frac{1}{2}\mu_2 & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2}\mu_3 & 0 & \frac{1}{2}\mu_3 \\ \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda \end{pmatrix} \quad (4.1)$$

Pour obtenir les probabilités stationnaires, on a résolu le système d'équations suivant :

$$\begin{cases} \pi Q = 0 \\ \sum_{i=1}^{10} \pi_i = 1 \end{cases}$$

et on a obtenu les probabilités stationnaires suivantes :

$$\pi_1 = \frac{9\lambda(\mu_2 + 2\mu_3)}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_2 = \frac{6\lambda(\mu_2 + 2\mu_3)}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_3 = \frac{9\mu_1(\mu_2 + 2\mu_3)}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_4 = \frac{4\lambda(\mu_2 + 2\mu_3)}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_5 = \frac{3(\mu_2^2 + 2\mu_2\mu_3)}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_6 = \frac{8\lambda\mu_2}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_7 = \frac{2(\mu_2^2 + 2\mu_2\mu_3)}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_8 = \frac{16\lambda\mu_2}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_9 = \frac{4\mu_2^2}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

$$\pi_{10} = \frac{8\mu_2\mu_3}{43\lambda\mu_2 + 38\lambda\mu_3 + 9\mu_1\mu_2 + 18\mu_1\mu_3 + 18\mu_2\mu_3 + 9\mu_2^2}$$

4.3.4 Les indices des performances liés à R_1

Dans ce travail, on s'intéresse à évaluer les performances liées au schémas de coopération étudié ainsi qu'à la politique de remplacement FSDV. Pour cela, les indices de performances essentiels à évaluer sont les taux de franchissement et le délai moyen de réponse pour une requête. **★ Fréquence moyenne de franchissement d'une transition :**

$$\bar{\lambda}(t_1) = \lambda(\pi_3 + \pi_5 + \pi_7 + \pi_9 + \pi_{10})$$

$$\bar{\lambda}(t_3) = \mu_2\pi_1$$

$$\bar{\lambda}(t_4) = \mu_1\pi_1$$

$$\bar{\lambda}(t_5) = \mu_2\pi_2$$

$$\bar{\lambda}(t_6) = \frac{1}{2}\mu_2\pi_2$$

$$\bar{\lambda}(t_7) = \mu_2\pi_4$$

$$\bar{\lambda}(t_8) = \frac{1}{2}\mu_2\pi_4$$

$$\bar{\lambda}(t_9) = \mu_3\pi_6$$

$$\bar{\lambda}(t_{10}) = \frac{1}{2}\mu_2\pi_6$$

$$\bar{\lambda}(t_{11}) = \frac{1}{2}\mu_3\pi_8$$

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

$$\begin{aligned} n_s &= \sum_j [M_j(p_1) + M_j(p_2) + M_j(p_3) + M_j(p_4) + M_j(p_5) + M_j(p_6)]\pi_j \\ &= \pi_1 + \pi_2 + \pi_4 + \pi_6 + \pi_8 \end{aligned} \quad (4.2)$$

★ **Le taux moyen effectif des arrivées :**

$$\eta_{t_1} = \lambda(\pi_3 + \pi_5 + \pi_7 + \pi_9 + \pi_{10}) \quad (4.3)$$

★ **Le temps moyen de réponse :**

$$\tau = \frac{n_s}{\eta_{t_1}} = \frac{\pi_2 + \pi_4 + \pi_6 + \pi_8}{\lambda(\pi_3 + \pi_5 + \pi_7 + \pi_9 + \pi_{10})} \quad (4.4)$$

4.4 Le modèle de SimpleCache

La façon traditionnelle de résoudre une demande de données est de vérifier tout d'abord la mémoire cache local et d'envoyer la demande au serveur d'origine après un défaut de cache local. Ce régime est considéré comme SimpleCache dans [19]. Ce système fonctionne bien tant que la connexion au serveur est fiable et n'est pas trop coûteuse, sinon, il en résulte des demandes échouées ou des délais d'attente très élevé.

Le modèle de RdPSG et Interprété R_2 défini ci-dessous représente le régime de SimpleCache.

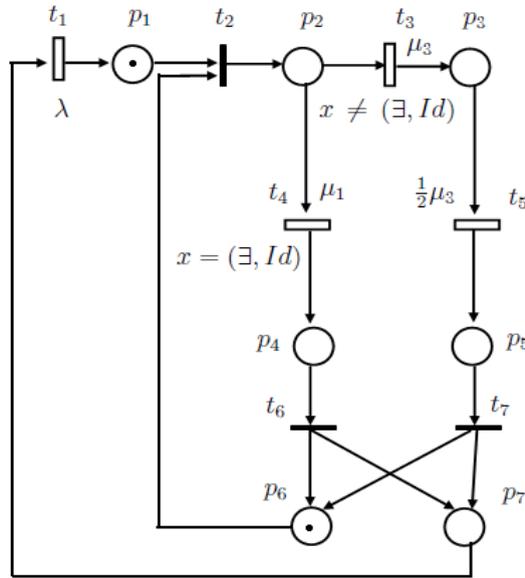


FIGURE 4.5 – Le modèle R_2

4.4.1 Définition du modèle R_2

Soit R_2 le modèle de RdPSG et interprété pour le régime SimpleCache. Les mêmes définitions données pour le modèle R_1 sont valables pour ce modèle. La différence réside dans le fait qu'après un défaut de cache, la requête consulte directement le serveur d'origine qui est représenté par la place p_3 avec un taux de recherche μ_3 . Le renvoi du document au client à partir de ce dernier s'effectue avec un taux $\frac{1}{2}\mu_3$.

Les propriétés qualitatives sont aussi conservées comme le modèle R_1 , et l'analyse quantitative est effectuée par la construction du graphe des marquages et la chaîne de Markov, de la même manière que le modèle R_1 et on obtient les deux figures associées à ces deux derniers.

Le générateur infinitésimal Q associé à la chaîne de Markov est donné par :

$$Q = \begin{pmatrix} -\mu_1 - \mu_3 & \mu_3 & \mu_1 & 0 \\ 0 & -\frac{1}{2}\mu_3 & 0 & \frac{1}{2}\mu_3 \\ \lambda & 0 & -\lambda & 0 \\ \lambda & 0 & 0 & -\lambda \end{pmatrix} \quad (4.5)$$

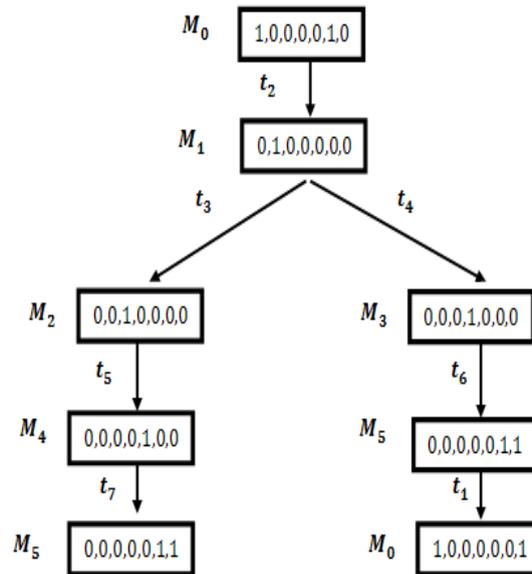


FIGURE 4.6 – Le graphe des marquages associé à R_2

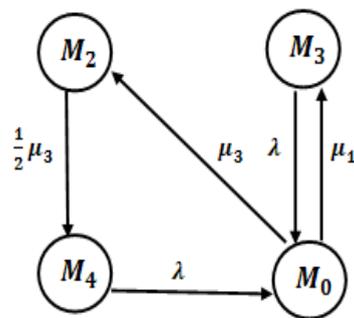


FIGURE 4.7 – La chaîne de Markov du modèle R_2

Pour obtenir les probabilités stationnaires, on a résolu le système d'équations suivant :

$$\begin{cases} \pi Q = 0 \\ \sum_{i=1}^4 \pi_i = 1 \end{cases}$$

et on a obtenue les probabilités stationnaires suivantes :

$$\pi_1 = \frac{\lambda}{3\lambda + \mu_1 + \mu_3}$$

$$\pi_2 = \frac{2\lambda}{3\lambda + \mu_1 + \mu_3}$$

$$\pi_3 = \frac{\mu_1}{3\lambda + \mu_1 + \mu_3}$$

$$\pi_4 = \frac{\mu_3}{3\lambda + \mu_1 + \mu_3}$$

π_1 , π_2 , π_3 , et π_4 sont les probabilité d'être dans les états M_0 , M_2 , M_3 , et M_4 respectivement.

4.4.2 Les indices de performances

★ **Fréquence moyenne de franchissement d'une transition :**

$$\bar{\lambda}(t_1) = \lambda(\pi_3 + \pi_4)$$

$$\bar{\lambda}(t_3) = \mu_3\pi_1$$

$$\bar{\lambda}(t_4) = \mu_1\pi_1$$

$$\bar{\lambda}(t_5) = \frac{1}{2}\mu_3\pi_2$$

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

$$n_s = \sum_j [M_j(p_1) + M_j(p_2) + M_j(p_3)] \pi_j = \pi_1 + \pi_2 \quad (4.6)$$

★ **Le taux moyen effectif des arrivées :**

$$\eta_{t_1} = \lambda(\pi_3 + \pi_4) \quad (4.7)$$

★ **Le temps moyen de réponse :**

$$\tau = \frac{n_s}{\eta_{t_1}} = \frac{\pi_1 + \pi_2}{\lambda(\pi_3 + \pi_4)} \quad (4.8)$$

4.5 Amélioration du schéma de coopération

Les politiques de remplacement de cache jouent un rôle d'amélioration du schéma de coopération par l'augmentation du taux de succès au niveau du cache local. En intégrant une politique de remplacement de cache dans le modèle qui est la politique FSDV (cf. chapitre 2), on obtient le nouveau modèle de réseau de Petri R_3 défini ci-dessous.

4.5.1 Définition du modèle R_3

Dans ce nouveau modèle, on a modifié le modèle R_1 par l'ajout et suppression des places et des transitions de la manière suivante :

- La place p_{13} maintenant représente le nombre de requêtes arrivées au système et on a ajouté la place p_{14} qui représente le nombre de requêtes servi à partir du cache local.
- La place p_{15} représente la politique de remplacement de cache FSDV.
- La place p_{16} signifie que si la requête a été servies à partir du FSC1, FSC2, FSC3, ou à partir de l'origine, un nouveau document est arrivé pour être caché dans le cache local.
- La place p_{17} représente le cache local et le nombre de jetons N représente le nombre de documents existe dans le cache.
- La transition t_{17} signifie que si l'espace disponible dans le cache n'est pas suffisant pour cacher le nouveau document, cette dernière doit être franchie pour effectuer un remplacement de afin de libérer de l'espace dans le cache.

- Contrairement à la transition t_{17} , la transition t_{18} signifie que si l'espace est suffisant dans le cache local pour y placer le nouveau document, alors c'est cette dernière qui doit être franchie.

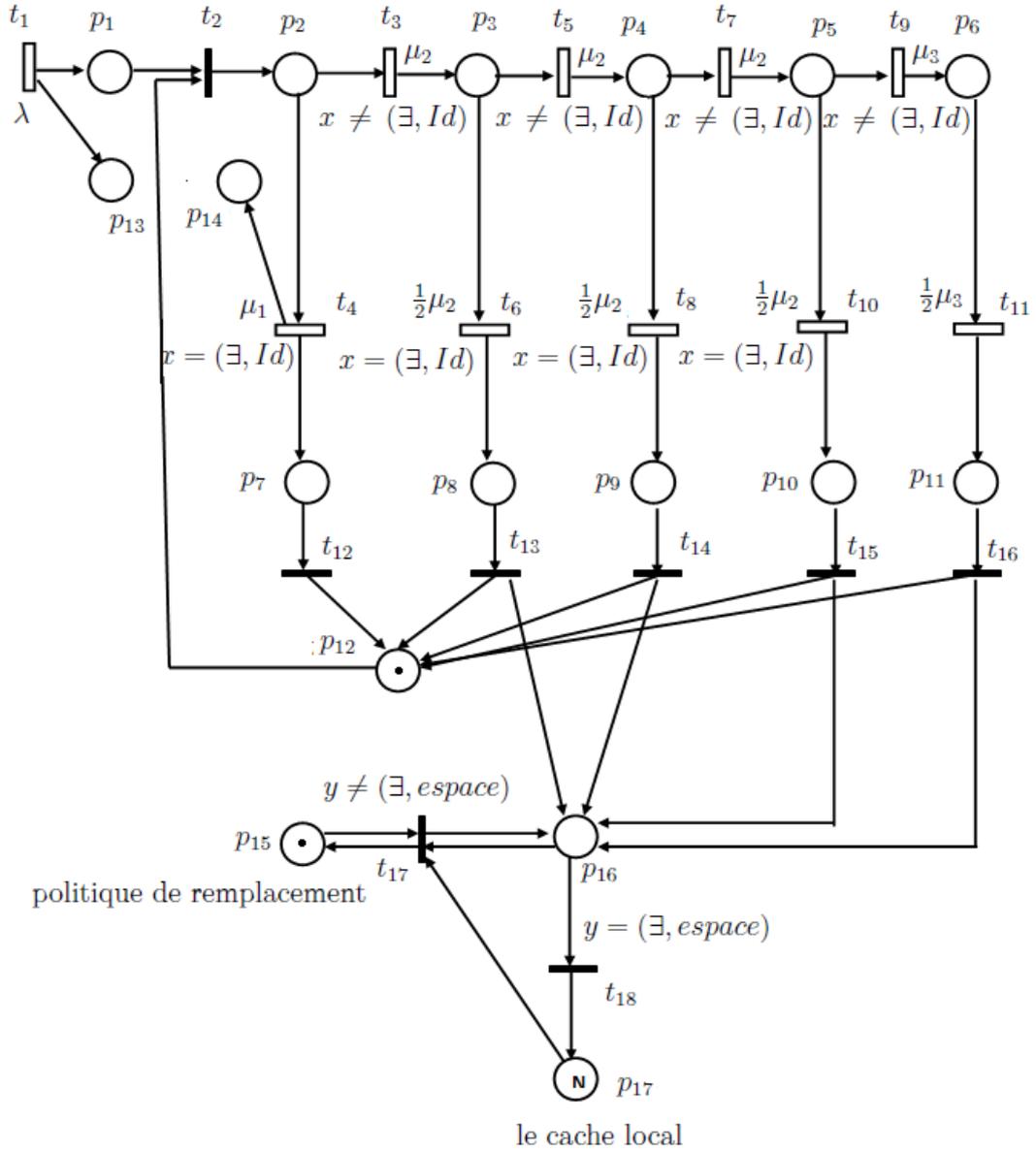


FIGURE 4.8 – Le modèle R_3

Remarque

Nous avons interprété les transitions t_{17} et t_{18} par des prédicats tels que la variable $y = (\exists, espace)$ est vraie si l'espace disponible dans le cache est suffisant, et dans ce cas là, c'est t_{17} qui doit être sensibilisée, sinon, si $y \neq (\exists, espace)$ qui est vraie, c'est la transition t_{18} qui doit être sensibilisée .

4.5.2 L'analyse qualitative du modèle R_3

Le modèle de RdP R_3 proposé admet comme propriétés qualitatives :

- La bornitude : Toutes les places du réseau ne sont pas bornées donc R_3 n'est pas borné.
- La vivacité : Toutes les transitions sont vivantes donc le RdP est vivant.
- La propriété de graphe d'événements : Le RdP n'est pas un graphe d'événements.
- Les états de blocage : Le RdP ne possède aucun état de blocage.
- La connectivité : Le RdP est connexe.
- La réinitialisabilité : Le RdP n'est pas réinitialisable.

Remarque 1

Pour le modèle R_3 , on va se limiter juste à une analyse qualitative puisque le modèle R_3 est un modèle non borné et non réinitialisable, donc une analyse quantitative pour ce modèle est peut être impossible.

Remarque 2

On peut calculer le taux de succès à partir du modèle R_3 par la division de nombre de requête servie à partir du cache local (i.e le nombre de jetons dans la place p_{14}) sur le nombre total de requêtes (le nombre de jetons dans la place p_{13}). En d'autre terme,

$$tauxdesuccès = \frac{M(p_{14})}{M(p_{13})}$$

4.6 Conclusion

Dans ce chapitre, nous avons présenté la modélisation d'un schéma de coopération par le modèle de réseau de Petri stochastique interprété R_1 . Après avoir expliqué le modèle R_1 , une analyse qualitative, décrivant les différentes propriétés du réseau est effectuée. De

même, une analyse quantitative, à travers la notion de chaîne de Markov est effectuée dans le but d'évaluer les performances du modèle. Un deuxième modèle de RdPSG et Interprétée R_2 est présenté pour le régime SimpleCache afin de comparer les délais moyens de réponse dans les cas d'absence et la présence d'un schéma de coopération. Un autre modèle R_3 à été également présenté comme amélioration du modèle R_1 avec l'intégration de la politique de remplacement de cache et d'étudier les propriétés qualitatives associés sont données.

Le prochain chapitre sera consacré à l'application numérique pour les performances des RdPs R_1 et R_2 afin de comparer les délais moyens de réponse dans le cas de SimpleCache et dans le cas de la coopération.

Application

5.1 Introduction

Ce chapitre est consacré à la présentation de quelques résultats exacts obtenus par les RdPSG et Interprétés. Nous présentons d'abord un exemple explicatif pour le schéma de coopération, ensuite, en utilisant l'environnement MATLAB, nous calculons les indices de performances définis dans le chapitre 4 pour les deux modèles R_1 (cas de coopération) et R_2 (cas de SimpleCache). Enfin, nous effectuons une comparaison des performances des deux modèles R_1 et R_2 en terme de délai moyen de réponse.

5.2 Description de l'environnement MATLAB

MATLAB (MATrix LABoratory) comprend de nombreuses fonctions graphiques, un système puissant d'opérateurs s'appliquant à des matrices, des algorithmes numériques (zeros d'une fonction, intégration, interpolation), ainsi qu'un langage de programmation extrêmement simple à utiliser.

Il fut conçu initialement (au milieu des années 1980) pour manipuler aisément des matrices à l'aide de fonctions pré-programmées (addition, multiplication, inversion, décompositions, déterminants,...), en s'affranchissant des contraintes des langages de programmation classiques :

- Plus de déclarations de variables.
- Plus de phase d'édition-compilation-exécution.

Cette orientation de calcul matriciel a depuis évolué vers un outil pouvant être vu comme une super-calculatrice graphique et regroupant dans la version de base la quasi-majorité

des problèmes numériques.

Plusieurs extensions plus «pointues» ont été conçues sous la forme de «TOOLBOXes», qui sont des paquets (payants) de fonctions supplémentaires dédiées à un domaine particulier :

- **COMNTROL** pour l'automatique.
- **SIGNAL** pour le traitement du signal.
- **OPTIMIZATION** pour l'optimisation.
- **NEURAL NETWORK** pour les réseaux de neurones.
- **PETRI NET** pour les réseaux de petri.

PNToolbox présente une facilité d'exploiter l'interface utilisateur graphique (GUI). PN-Toolbox est un outil pour la simulation, l'analyse et la conception de systèmes à événements discrets, basée sur les réseaux de Petri . Cet outil est intégré dans l'environnement MATLAB. La philosophie d'intégration de PNToolbox dans MATLAB présente l'avantage considérable de la création de puissants instruments algébriques, statistiques et graphiques qui exploitent les fonctions disponibles dans MATLAB. Dans la version actuelle de la boîte à outils PN, cinq types de modèles PN classiques sont acceptés, à savoir :

- RdP classiques.
- RdP t-temporisés et p-temporisés.
- RdPS et RdPSG.

Après avoir dessiné le réseau du modèle RdP, l'utilisateur peut :

- Visualiser la matrice d'incidence qui est construite automatiquement à partir de la topologie du réseau .
- Explorer les propriétés comportementales (telles que la vivacité, bornétude, etc) en consultant le graphe des marquages, qui est construit automatiquement à partir de la topologie du réseau et du marquage initial.
- Calculer P-invariants et T-invariants.
- effectuer une expérience de simulation.
- Afficher les résultats actuels de la simulation à l'aide de la portée et le Journal des installations.
- Évaluer la performance globale des indices (comme la moyenne de marquage des lieux, délai moyen de transitions de tir, etc).

Dans cette application on a utilisé Matlab pour les deux cas suivants :

Cas 1 Avoir le graphe d'accessibilité, via la boîte d'outils qui nous permet d'avoir des résultats exacts.

Cas 2 Utiliser son environnement pour calculer les performances à évaluer.

5.3 Exemple d'application

Dans cette partie, on va présenter un exemple d'un réseau *Ad hoc*, où les nœuds coopèrent entre eux pour partager les ressources. Ce réseau est représenté par la figure 5.1 tel que le nombre de nœuds est égal à 20 et chaque nœud est identifié par des indices de $N1$ jusqu'à $N20$.

Les nœuds en noir sont les nœuds élus comme fournisseur de services cache et le triangle rouge représente le serveur d'origine.

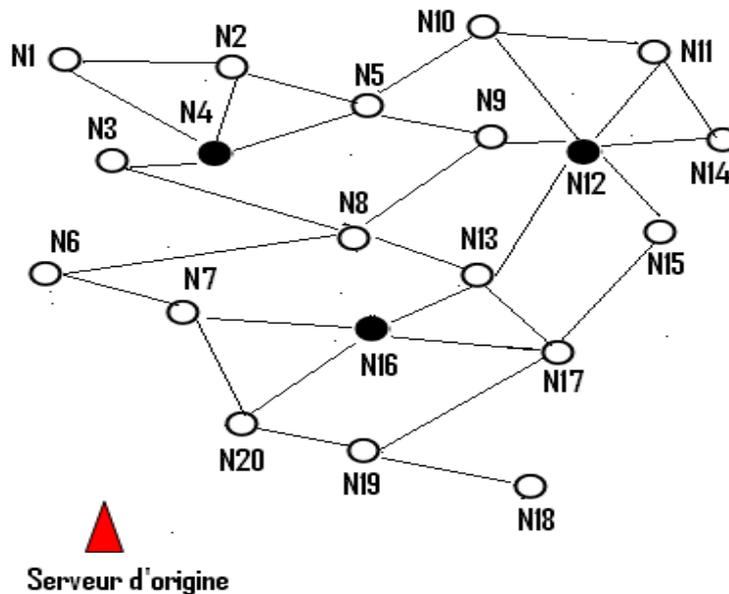


FIGURE 5.1 – Exemple d'un réseau *Ad hoc*

Choisissons le nœud $N8$ comme demandeur d'un document. Dans un premier temps, il consulte son cache local, si le document existe, le document sera renvoyé avec un taux μ_1 , sinon il va aller au premier fournisseur de service cache le plus proche (le nœud $N12$ par exemple) avec un taux de recherche μ_2 . Dans le cas d'un succès, le renvoi du document s'effectue avec la moitié du taux de recherche μ_2 puisque le temps de renvoi d'un document est lié à la récupération du document à partir du nœud qui le détient à travers un protocole

de routage, ce qui rend le service de renvoi lent par rapport au service de recherche. De la même manière, le taux de recherche à partir de l'origine est μ_3 , et le taux de renvoi est $\frac{1}{2}\mu_3$.

5.4 Evaluation des performances du modèle R_1

Les paramètres du modèle R_1

Pour des valeurs données de taux d'arrivée et taux des services, on calcule la distribution stationnaire et les paramètres de performances.

- $\lambda = 0.2$
- $\mu_1 = 20$
- $\mu_2 = 15$
- $\mu_3 = 1$

La distribution stationnaire

π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}
0.0064	0.0043	0.6479	0.0028	0.1079	0.0047	0.0719	0.0095	0.1199	0.0239

TABLE 5.1 – Les probabilités stationnaires liées à R_1

Quelques indices de performances lié à R_1

★ Nombre moyen de client dans le système :

$$n_s = \pi_1 + \pi_2 + \pi_4 + \pi_6 + \pi_8 = 0.0263$$

★ Le taux moyen effectif des arrivées :

$$\eta_{t_1} = \lambda(\pi_3 + \pi_5 + \pi_7 + \pi_9 + \pi_{10}) = 0.1947$$

★ Le temps moyen de réponse :

$$\tau_{R1} = \frac{n_s}{\eta_{t_1}} = \frac{\pi_2 + \pi_4 + \pi_6 + \pi_8}{\lambda(\pi_3 + \pi_5 + \pi_7 + \pi_9 + \pi_{10})} = 0.1355$$

5.5 Evaluation des performances du modèle R_2

Dans cette application, on prend le même taux d'arrivée λ et le même taux de recherche à partir de l'origine que dans le modèle R_1 , on obtient les probabilités stationnaires présentés dans le tableau 5.2.

π_1	π_2	π_3	π_4
0.0092	0.0185	0.9259	0.0462

TABLE 5.2 – Les probabilités stationnaires liées à R_2

Quelques indices de performances liées à R_2

★ Nombre moyen de client dans le système :

$$n_s = \pi_1 + \pi_2 = 0.0277$$

★ Le taux moyen effectif des arrivées :

$$\eta_{t_1} = \lambda(\pi_3 + \pi_4) = 0.1944$$

★ Le temps moyen de réponse :

$$\tau_{R2} = \frac{n_s}{\eta_{t_1}} = \frac{\pi_1 + \pi_2}{\lambda(\pi_3 + \pi_4)} = 0.1428$$

5.6 Variation des paramètres, Interprétation des résultats et comparaison

En faisant varier le taux μ_3 pour les deux modèles précédents, on obtient les résultats du tableau 5.3 et la figure 5.2

μ_3	0.1	1	2	5	10	15
τ_{R1}	0.1355	0.1275	0.1204	0.1066	0.0929	0.0875
τ_{R2}	0.1492	0.1428	0.1363	0.1200	0.1000	0.0875

TABLE 5.3 – La variation du taux μ_3

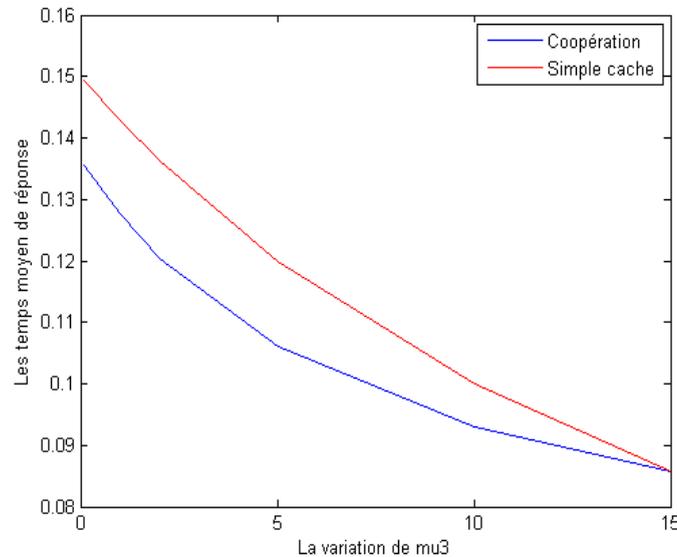


FIGURE 5.2 – Comparaison des temps moyen de réponse

La comparaison des résultats obtenus dans le cas SimpleCache et le cas du schéma de coopération, a montré que le délai moyen de réponse à partir du schéma de coopération est plus petit par rapport au délai moyen de réponse dans le cas du simple cache. Dans le cas où le taux μ_3 et μ_2 prennent les mêmes valeurs, c'est-à-dire le serveur d'origine a le même niveau en terme de distance avec les fournisseurs de services cache, dans ce cas, le temps moyen de réponse pour les deux cas sont les mêmes. Cela signifie que la distance parcourue dans le schéma de coopération aurait évité au client la latence de chargement de ces documents dans le cas où le serveur d'origine est plus loin. Dans le cas d'égalité des temps moyens de réponse, la réponse pour une requête à partir d'un fournisseur de services cache est meilleure dans ce cas pour éviter de surcharger le serveur d'origine.

5.7 Le temps de réponse et la politique de remplacement FSDV

La politique de remplacement FSDV (cf. chapitre 2) définie par la valeur

$$v(id) = \frac{\text{frequence}(id) \times \text{distance}(id)}{\text{taille}(id) \times \text{age_distace}(id)}$$

dépend du paramètre de la distance où cette dernière directement liée à la latence perçue par l'utilisateur. En effet, cacher les documents dont la source est accessible en longue distance (en terme de sauts) depuis le client aurait évité à ce dernier la latence de chargement de ces documents.

5.8 Conclusion

Dans ce chapitre, et après une brève présentation de l'environnement Matlab et l'outil PNtoolBox, nous avons donné un exemple explicatif pour le schéma de coopération. Par la suite, nous avons effectué une petite application numérique qui nous a permis de faire une comparaison des délais moyens de réponse du cas de présence de la coopération et dans le cas SimpleCache.

Conclusion générale et perspectives

Dans ce mémoire, nous avons réalisé une analyse mathématique d'un schéma de cache coopérative. L'approche analytique utilisée est basée sur l'utilisation des Réseaux de Petri Stochastiques Généralisés et Interprétés. Le but de cette analyse est d'étudier l'efficacité de cette approche coopérative en terme du délai moyen de réponse. Pour cela, notre étude a été scindée en deux volets.

- ★ Le premier volet consiste à présenter une synthèse sur la notion de coopération dans les réseaux mobiles *Ad hoc*. Le concept de la coopération a été proposé pour l'amélioration de la qualité de service (QoS) des réseaux mobiles *Ad hoc*. La coopération permet une gestion des caches globalement et de cette manière tous les caches d'une même zone seront traités comme un seul cache de grande taille. De là, il devient nécessaire de proposer des solutions afin de garantir une localisation efficace des ressources dans un tel type de réseau, c'est-à-dire la prise en compte des contraintes et caractéristiques des réseaux *ad hoc*, à savoir principalement le manque de connectivité, la bande passante réduite, et les capacités limitées des terminaux. Afin de garantir une bonne gestion d'un cache, l'introduction des stratégies de remplacement de cache est nécessaire. La meilleure stratégie de remplacement pour les réseaux mobiles *ad hoc* est celle qui prend en considération les caractéristiques spécifiques de ces réseaux.
- ★ Le deuxième volet consiste à effectuer une analyse mathématique par la modélisation via les réseaux de Petri Stochastiques Généralisés et Interprétés. Ces derniers forment une classe de modélisation de haut niveau qui nous permet d'effectuer une analyse qualitative et quantitative du schéma de coopération. Dans ce travail, nous avons proposé trois modèles via les RdPSG et Interprétés, qui sont les suivants :
 - Le premier modèle R_1 prend en considération un schéma de coopération. Ce modèle

nous a permis d'effectuer une analyse qualitative et une autre quantitative par l'évaluation de quelques indices de performances comme les fréquences moyennes des transitions et le temps moyen de réponse.

- Le deuxième modèle R_2 ne prend pas en considération le schéma de coopération (cas de SimpleCache), où la même analyse que pour R_1 est effectuée pour R_2 .
- Un troisième modèle R_3 est également présenté avec l'intégration de la politique de remplacement du cache afin d'améliorer l'efficacité du schéma de coopération, et pour ce modèle nous nous sommes contentés d'une analyse qualitative.

Une application numérique pour l'évaluation des performances des deux modèles R_1 et R_2 est effectuée afin de comparer les temps moyens de réponse. D'après les résultats obtenus, nous avons constaté que le temps moyen de réponse avec un schéma de cache coopératif est plus petit que le temps moyen de réponse dans le cas d'un SimpleCache. Ces résultats permettent d'affirmer que le schéma de cache coopératif réduit la latence de chargement d'un document par un utilisateur.

Afin de compléter ces premiers résultats, d'autres travaux doivent être menés à l'avenir, à savoir :

- La prise en considération du cas où le nombre de fournisseurs de services cache est très grand.
- La prise en considération de la politique de remplacement de cache dans l'étude analytique afin de réduire le temps moyen de réponse et le coût moyen d'énergie.
- D'autres lois stochastiques plus pertinentes pour les aléas rencontrés dans les systèmes de coopération en commun pourront ensuite être étudiées.
- L'optimisation multi-objectifs de la politique de remplacement de cache *FSDV*.
- La simulation à événements discrets pour des modèles des réseaux de Petri proposés dans le cadre de cette étude.

Bibliographie

- [1] A. Larbi. *Mécanisme de remplacement de cache dans les réseaux Ad Hoc*. Mémoire magistère en Informatique de l'université A/Mira de Bejaia, 2008.
- [2] M. Benmalek. *Sur les politiques cooperative de remplacement de cache dans les réseaux mobiles Ad hoc*. Mémoire magistère en Informatique de l'université A/Mira de Bejaia, 2009.
- [3] N. Benhamida. *Sur les politiques de gestion du cache d'un serveur web*. Mémoire magistère en Informatique de l'université A/Mira de Bejaia, 2007.
- [4] M. Frikha . *Réseaux Ad hoc : routage, qualité de service*. Edition Lavoisier, 2010.
- [5] H. Labiod. *Réseaux mobiles Ad hoc et réseaux de capteurs sans fil*, Edition Lavoisier.
- [6] C. Xiuzhen, H. Xiao and D. Ding-hu. *Ad hoc wireless networking*. Edition Kluwer Academic Publishers.
- [7] S. Busola. O and M. Natarjan. *A review of the energy efficient and secure multicast routing protocols for mobile ad hoc networks*. International journal on applications of graph theory in wireless ad hoc networks and sensor networks (GRAPH-HOC) Vol.2, No.2, June 2010.
- [8] R. Meraihi and A. Naimi. *Routage avec différenciation de terminaux dans les réseaux mobiles ad hoc* . RIST Vol.16 No.01. 2006.
- [9] S. Lim, W. Lee, G. Cao and R. Chita. Das. *A novel caching scheme for improving Internet-based mobile ad hoc networks performance* . in ICCCN 03 :IEEE International Conference on Computer Communications and Networks, Dallas, TX, U.S.A. pp 38-43. 2003.

- [10] E. Chan, W. Li and D. Chenb. *Energy saving strategies for cooperative cache replacement in mobile ad hoc networks* . City University of Hong Kong, Tat Chee Ave, Kowloon, Hong Kong. Pervasive and Mobile Computing pp 77-92. 2009.
- [11] N. Masahiko, M. Masayuki and M. Hideo. *Analysis and modeling of World Wide Web traffic for capacity dimensioning of Internet access lines*. Graduate School of Engineering Science, Osaka University, Toyonaka. Performance Evaluation 34. pp 249-271. 1998.
- [12] C. Narottam, R.C. Joshi, M. Manoj. *Cooperative Caching Strategy in Mobile Ad Hoc Networks Based on Clusters* . Wireless Personal Communications pp 41-63, 2007.
- [13] K. Georgios, M. Elias Christos, K.D Dimitris. *Using data envelopment analysis to evaluate the efficiency of web caching object replacement strategies* . Journal of Network and Computer Applications. pp 803-817. 2012.
- [14] L. JIA. *A Mobile Ad-hoc Network Data Cache Invalidation Method* . Department of Computer Appliance, Jiangsu Food science College, Huaian, Jiangsu, China. Procedia Engineering. pp 150 - 154. 2011.
- [15] C. Guohong, Y. Liangzhong and R.D. Chita. *Cooperative Cache-Based Data Access in Ad Hoc Networks*. Pennsylvania State University, IEEE Computer Society. Février. 2004.
- [16] D. Yu, K.S Sandeep G and V. Georgios. *Improving on-demand data access efficiency in MANETs with cooperative caching*. IMPACT Laboratory, School of Computing and Informatics, Arizona State University, Ad Hoc Networks pp 579-598. 2009.
- [17] F. Sailhan. *Localisation de ressources dans les réseaux ad hoc D*. Thèse doctorat, Université PARIS VI, 2010.
- [18] S. Guebli. *Intégration de la fonction de proxy cache aux routeurs actifs*. Rapport de DEA. Ecole Normale Supérieure de Lyon, 2003.
- [19] L. Yin and G. Cao. *Supporting Cooperative Caching in Ad Hoc Networks*, IEEE Trans. Mobile Computing, vol. 5, No. 1, pp 77-89, Jan. 2006.
- [20] S. Collart-Dutilleul. *Les Réseaux de Petri P-temporels. Modélisation et validation d'exigences temporelles.*, Mémoire d'habilitation à diriger des recherches en automatique et informatique, Université des Sciences et Technologies de Lille, 2008.

- [21] R. Bergoute Bekrar. *Identification des systèmes à événements discrets par réseaux de Petri*, These Doctorat, Université de Reims Champagne Ardenne, 2009.
- [22] S. Hakmi. *Evaluation des performances des systèmes prioritaires à l'aide des Réseaux de Petri Stochastiques Généralisées*, Mémoire magistère en Recherche Opérationnelle, Université A/Mira de Béjaia, 2011.
- [23] A.C. Geniet. *Les réseaux de Petri. Un outil de modélisation*, Spriger-Verlag Berlin, Edition 2, 2006.
- [24] P. Lezaud. *Etudes des réseaux de Petri Stochastiques*, DEA Mathématique appliquées, Université de Paul SABATIER, 1995.
- [25] M. Magnin. *Réseaux de Petri à chronomètres. Temps dense et temps discret*, Thèse de Doctorat, Automatique et Informatique Appliquée, Université de NANTES, 2007.
- [26] P. Rongier. *Développement et évaluation de méthodes multi-agents pour les applications aux véhicules robotisés*, Thèse de Doctorat en Génie Informatique, Université de MONTPELLIER II, 2007.
- [27] P. Azema, G. Juanole, E. Sanchis and M. Mombemard. *Specification and Veriication of Disrribwed Systems Using PROLOG Interpreted Petri Nets*, in Proc. Coof. '84 IEEE Software Engineering, pp.510-518, 1984.
- [28] M. Diaz. *Petri Nets fundamental models, verification and application*, Wiley edition, 2009.
- [29] V. Robert. *Les Réseaux de Petri*, Cours sur les réseaux de petri, LAAS-CNRS Toulouse, 2002.
- [30] Y. Atamna. *Réseaux de Petri Temporisés Stochastiques Classiques et Bien Formés. Définition, Analyse et Application aux Systèmes Distribuée Temps Réel*, Thèse Doctorat en Informatique Industrielle, Université de Paul SABATIER, 1994.
- [31] J. Marsan A. M. Balbo. G and G. Conte. *A class of Generalized Stochastic Petri Nets for the Performances of multiprocessor Systems*, In ACM TOCS, vol.2, May 1984.
- [32] L. Gallon. *Le modèle de réseaux de Petri Temporisés Stochastiques. Extensions et Applications* , Thèse Doctorat en Informatique Industrielle, Université de Paul SABATIER , 1997.

- [33] J. Haas. Peter. *Stochastic Petri Nets : Modelling, Stability,Simulation*, Springer Series in Operations Research, Springer-Verlag New York, 2002.
- [34] K. Prashant and C. Naveen. *A Review of Cooperative Cache Management Techniques in MANETs*, Department of Computer Science and Engineering National Institute of Technology, Hamirpur (H.P.) India, 2010.
- [35] G. Radhamani and S. Umamaheswari. *Comparison of Cooperative Caching Strategies in Mobile Ad-Hoc Network (MANET)*, School of Information Technology and Science, College of Science, Coimbatore, India, 2010.
- [36] M. Akon, M. Towhidul Islam, X. Shen and A. Singh. *A bandwidth and effective hit optimal cache scheme for wireless data access networks with client injected updates*, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1, 2012.
- [37] D. Aïssani, L. Bouallouche-Medjkoune and F. Semchedine. *Heavy-tailed distribution and QoS in distributed Web server systems*, Doctoral School in Computer Science and LAMOS, Laboratory of Modelisation and Optimization of Systems University of Béjaïa, 2008.
- [38] S. Youcef. *Méthodes et Outils d'évaluation de performances des services Web* Thèse Doctorat, Université de Paris Dauphine, 2009.
- [39] P. Barford and M. Crovella. *Generating Representative Web Workloads for Network and Server Performance Evaluation*. Measurement and Modeling of Computer Systems, pp. 151-160, 1998.
- [40] P. Rochat. *Proxys Internet Avancés*. Thèse Doctorat, École Polytechnique Fédérale de Lausanne, 2003.
- [41] L. Bouallouche-Medjkoune. *Modélisation et simulation des systèmes informatiques et Réseaux de télécommunications*. Cours École Doctorale Informatique ReSyD, Département Informatique, Université A/MIRA de Béjaïa 2009.
- [42] J. Wang. *Petri Nets for Dynamic Event-Driven System Modeling*. Department of Software Engineering Monmouth University, pp. 1-17, 2006.

Annexe

Caractérisation du trafic

Dans la littérature, plusieurs travaux ont été consacrés à la caractérisation du trafic Web. Parmi ces caractéristiques, on s'intéresse à :

- La distribution de la popularité des documents Web qui suit une loi Zipf-like ;
- La distribution de la taille des documents suit une loi heavy-tailed (Pareto) ;
- La distribution de temps entre requêtes suit une loi Pareto.

La popularité des documents

Une caractéristique commune des trafics Web est la distribution des références aux fichiers. Si l'on trie l'ensemble des fichiers disponibles sur l'internet en fonction de leur popularité, la distribution des requêtes suit une loi de Zipf :

$$F(rang) = \frac{C}{rang}$$

Le *rang* représente le classement du fichier lorsqu'on trie l'ensemble des fichiers suivant leur popularité (le fichier ayant $rang = 1$ étant le plus populaire). C est une constante de renormalisation dépendante du nombre total de fichiers n , elle est calculée en tenant compte du fait que la probabilité totale doit être égale à 1 (et l'on a $C = 1 / \sum_i^n \frac{1}{i}$). Cependant, plusieurs études sur l'applicabilité de la loi Zipf aux documents Web, ont conclues que la distribution de la popularité des documents Web suit généralement une distribution Zipf-like, dont la probabilité relative d'une requête pour le *ième* plus populaire document est proportionnelle à $\frac{1}{i^\beta}$ avec β compris entre 0 et 1. Selon cette loi, la popularité P_i d'un

document de rang i est donnée par la formule suivante :

$$P_i = ki^{-\beta}$$

avec $0 < \beta < 1$.

Où k est une constante de proportionnalité.

Temps entre requêtes

Dans [39], Barford et Crovella montrent que le temps entre deux requêtes web dans le système suit une loi de Pareto : la probabilité qu'il y ait t secondes entre deux requêtes est

$$D(t) = \gamma k^\gamma t^{-(\gamma+1)}$$

avec $k = 1$ et $\gamma = 1.5$.

Taille des fichiers

L'étude de la trace [39], montre que la répartition des tailles des fichiers transférés dans l'internet suit une distribution dite *heavy-tailed*. La probabilité que le fichier soit de la taille x est égale à :

$$P[X > x] \approx x^{-\alpha}$$

$x \rightarrow \infty, 0 < \alpha < 2$

Une forme simple de la distribution heavy-tailed est la distribution de Pareto. sa fonction de densité de probabilité est donnée par :

$$P(t) = \alpha k^\alpha x^{-(\alpha+1)}$$

Où : $\alpha, k > 0$ et $x \geq k$ et sa fonction de distribution est donnée par :

$$F(x) = P[X \leq x] = 1 - (k/\alpha)^\alpha$$

Le paramètre k représente la plus petite valeur possible de la variable aléatoire x , donc k est la plus petite taille que peut prendre un document, alors si k augmente les petites valeurs de taille de documents sont ignorées.

Résumé

La mise en cache des données fréquemment consultées dans un réseau *Ad hoc* est une technique qui permet d'améliorer de manière significative la disponibilité et l'accessibilité des données. La performance d'une telle technique est étroitement liée aux mécanismes de gestion qui lui sont associés, en particulier la politique de remplacement de cache utilisée. Les performances des caches peuvent davantage être améliorées si la gestion des caches est associée à une approche coopérative. Ce travail porte principalement, sur la modélisation et l'évaluation des performances d'un schéma de cache coopératif à travers un outil graphique mathématique "Les Réseaux de Petri Stochastiques Généralisés et Interprétés" en se basant sur l'analyse exacte pour calculer les temps moyens de réponse. Une comparaison a été effectuée dans le cas de la présence de la coopération et de l'absence de la coopération. Cette comparaison nous a confirmé que le schéma de cache coopératif diminue la latence de chargement d'un document dans un réseau mobile *Ad hoc*.

Mots clefs : Réseaux *Ad hoc*, Schémas de caches coopératifs, Politiques de remplacement de cache, Réseau de Petri Stochastiques généralisés et Interprétés, Evaluation de performances.

Abstract

Caching frequently accessed data in an *Ad hoc* network is a technique to significantly improve the availability and accessibility data. The performance of such a technique is closely related to mechanisms management associated with it especially the cache replacement policy used. The cache performance can be further improved if the cache management is associated with a cooperative approach. This work focuses on modeling and performance evaluation of a cooperative caching scheme through a graphical and mathematical tool "The Generalized Stochastic and Interpreted Petri Nets" based on the exact analysis to calculate the average response. A comparison was carried out in the case of the presence of cooperation and the absence of cooperation. This comparison confirmed that the cooperative caching scheme reduces the latency of loading a document in a mobile *Ad hoc* network.

Keywords : *Ad hoc* Networks, Cooperative Caching Schemes, Cache replacement policies, Generalized Stochastic and Interpreted Petri Nets, Performance evaluation.

