

**REREPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE**

UNIVERSITE SAAD DAHLEB BLIDA
Faculté des Sciences
Département d'Informatique



Mémoire de fin d'étude

En vue d'obtenir le diplôme de Master en informatique

Spécialité : Système Informatique et Réseaux

Thème :

Evaluation des Performances d'un Mécanisme d'Economie
d'Energie d'un Réseau de Capteurs Sans Fil

Devant le jury composé de :

M^{me} Boutoumi Bachira Promotrice.
M^r Hammouda Mohamed Président.
M^{me} Bey Fella Examinatrice.

Présenté par :

M^{elle} Yahiaoui Yasmine.
M^{elle} Zaidi Houda.

Année universitaire 2018/2019

Remerciement

En préambule à ce mémoire nous remerciant ALLAH qui nous aide et nous donne la patience et le courage durant ces longues années d'étude et qui nous a donné la force d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier notre promotrice Mme : (Boutoumi Bachira) pour l'orientation, la confiance, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port et pour son précieux conseil et son aide durant toute la période du travail.

Nos sincères remerciements vont également à tous nos enseignants, sans exception, qui ont veillé à notre formation.

Nous remercions infiniment nos chers parents qui nous ont soutenu et encouragé tout au long de nos études, tous les membres de nos familles pour leur contribution et leur soutien.

Enfin, nous remercions tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.



Résumé

Durant ces dernières années, nous avons été marqués par un développement très rapide des techniques et technologies dans les domaines de l'électronique, la mécanique et les technologies de communication sans fil. Ces innovations ont permis de créer de petits objets communicants équipés de capteurs à un coût raisonnable. De nos jours, l'essor des réseaux de capteurs sans fil (RCSF) représente une bonne thématique de recherche. Ces réseaux de capteurs sont composés d'un grand nombre de nœuds ou capteurs communiquant entre eux et déployés sur une zone donnée afin de mesurer une donnée telle que la température, l'humidité, la luminosité, etc. Ceci dans la plupart des cas, dans le but de surveiller un évènement.

La conservation d'énergie dans ces réseaux est très importante. Les capteurs que nous considérons sont alimentés par des batteries non rechargeables. Ainsi, la conservation de cette ressource très limitée au niveau des nœuds capteurs influe directement sur la durée de vie du RCSF.

Afin de conserver l'énergie dans ce vaste domaine qui est l'objectif de notre travail, nous allons proposer des modèles basés sur les files d'attente avec vacances sous la politique de N-Policy et les files d'attente avec rappel pour le principe de retransmission des paquets, en utilisant les chaînes de Markov à temps continu (CMTC). Après avoir fait l'analyse des chaînes modélisées, nous allons implémenter des algorithmes permettant de calculer le générateur infinitésimal de la chaîne de Markov et les différents indices de performances ou l'énergie consommée et l'une des indices en utilisant deux méthodes différentes une exacte et l'autre approximative, afin de faire une comparaison entre les deux méthodes en calculant quelques métriques de comparaison.

Mots clés :

Réseaux de capteurs sans fil, files d'attente avec vacance, files d'attente avec rappel, N_Policy, chaînes de Markov, Conservation d'énergie, indices de performances.

Abstract

In recent years, we have been marked by a very fast development of techniques and technologies in the fields of electronics, mechanics and wireless communication technologies. These innovations have made it possible to create small communicating objects equipped with sensors at a reasonable cost. Today, the growth of wireless sensor networks (WSN) is a good research theme. These sensor networks are composed of a large number of nodes or sensors communicating with each other and deployed over a given area in order to measure data such as temperature, humidity, brightness, etc. This in most cases, for the purpose of monitoring an event.

The conservation of energy in these networks is very important. The sensors that we consider are powered by non-rechargeable batteries. Therefore, the conservation of this very limited resource at the sensor nodes directly affects the life of the WSN.

In order to conserve energy in this vast area which is the goal of our work, we will propose models based on the queues with vacations under the N-Policy and retrial queues for the principle of retransmission of the packets using continuous time Markov chains (CMTC). After the analysis of the chains modeled we will implement algorithms allowing to calculate the infinitesimal generator of the Markov chain and different performance measures or the energy consumption is one of the measures using two different methods an exact and an approximation one in order to make a comparison between the two methods by calculating some comparison metrics.

Keywords:

Wireless sensor networks, Queues with vacancy, Retrial queues, N-Policy, Markov chains, Conservation of energy, performance indices.

Table de matière

Remerciement.....	I
Résumé.....	II
Abstract.....	III
Table de matière.....	IV
Liste des figures.....	VII
Liste des tableaux.....	IX
Liste des abréviations :.....	X
Introduction générale.....	1
1. Généralités sur les réseaux de capteurs sans fils.....	4
1.1. Introduction.....	4
1.2. Un nœud de capteur.....	4
1.3. Les composants d'un nœud capteur.....	5
1.3.1. Une unité d'acquisition.....	5
1.3.2. Une unité de traitement.....	5
1.3.3. Une unité de communication.....	5
1.3.4. Une batterie.....	6
1.4. Les réseaux de capteurs sans fils.....	7
1.4.1. La durée de vie d'un réseau de capteurs.....	7
1.4.2. Architectures adoptées pour les RCSF.....	7
1.5. Consommation d'énergie du capteur.....	9
1.5.1. Sources de surconsommation d'énergie.....	10
1.6. Techniques de minimisation d'énergie.....	12
1.6.1. L'énergie de capture.....	12
1.6.2. L'énergie de calcul.....	12
1.6.3. L'énergie de communication.....	13
1.6.4. Technique de duty cycling.....	13
1.7. Contraintes et caractéristiques.....	13
1.8. Les domaines d'application des réseaux de capteurs.....	14
1.9. Facteurs influençant la conception des RCSF.....	15
1.9.1. Contraintes Conceptuelles.....	15
1.9.2. Les contraintes matérielles.....	16
1.10. Modèle de communication des RCSF.....	17
1.10.1. Couches de communication.....	17

1.10.2.	Plans de gestion	18
1.11.	Conclusion	19
2.	Formalismes mathématiques.....	20
2.1.	Introduction.....	20
2.2.	Les files d'attente	20
2.2.1.	Définition.....	20
2.2.2.	Caractéristiques des files d'attente.....	21
2.2.3.	Notation de Kendall	22
2.3.	Types des systèmes d'attente.....	22
2.3.1.	Les systèmes de files d'attente avec priorité	22
2.3.2.	Les systèmes de file d'attente avec rappels.....	23
2.3.3.	Système de file d'attente avec vacances.....	24
2.4.	Chaines de Markov	26
2.4.1.	La loi exponentielle.....	27
2.4.2.	Variable aléatoire.....	27
2.4.3.	Processus stochastique.....	28
2.4.4.	Chaîne de Markov à temps discret	28
2.4.5.	Chaines de Markov à temps continu.....	30
2.5.	Une méthode approximative pour résoudre les chaînes de Markov à deux dimensions	32
2.5.1.	Les étapes de l'algorithme de la méthode approximative	34
2.6.	Une méthode approximative pour résoudre les chaînes de Markov multidimensionnelles.....	36
2.6.1.	Les étapes de l'algorithme de la méthode approximative	38
2.7.	Conclusion	41
3.	Modélisation des réseaux de capteurs.....	42
3.1.	Introduction.....	42
3.2.	Les Modèles formels et les réseaux de capteurs sans fil.....	42
3.3.	Description et analyse de la modélisation d'un nœud capteur	46
3.3.1.	Le Modèle avec N-vacance (modèle M0).....	47
3.3.2.	Modèle avec priorité (modèle M1).....	52
3.3.3.	Le modèle avec N-vacance et Rappel (M2).....	62
3.4.	Conclusion	76
4.	Implémentation et étude expérimentale	77
4.1.	Introduction.....	77
4.2.	Présentation de l'application.....	77

4.2.1.	Fenêtre d'accueil	77
4.2.2.	Fenêtre de choix multiples.....	79
4.2.3.	Fenêtre pour l'évaluation des indices de performances :.....	80
4.3.	Etude expérimentale	81
4.3.1.	Effet de la variation de taux d'arrivée des paquets λ	82
4.3.2.	Effet de la variation de taux de service μ	85
4.3.3.	Effet de la variation de seuil N	87
4.4.	Etude comparative	88
4.5.	Conclusion	90
	Conclusion générale	91
	Bibliographie	92

Liste des figures

1. Généralités sur les réseaux de capteurs sans fils.....	4
Figure 1.1: Un nœud de capteur [13].	5
Figure 1.2 : Les composants d'un capteur [14].	6
Figure 1.3: schéma général d'un réseau de capteur sans fil.....	7
Figure 1.4: Transmission directe [15]	8
Figure 1.5: Transmission en multi-sauts [15].	8
Figure 1.6: Clustering [15].	8
Figure 1.7: Consommation d'énergie en en acquisition, traitement et communication [16].	10
Figure 1.8: Techniques de conservation d'énergie [18].	12
Figure 1.9: Classification des réseaux de communication [15].	14
Figure 1.10: Pile protocolaire d'un RCSF [20].	17
2. Formalismes mathématiques.	20
Figure 2.1: Système de file d'attente [9].	21
Figure 2.2 : Le schéma général d'un système d'attente avec rappels [22].	24
Figure 2.3: Système de file d'attente avec vacance [12].	24
Figure 2.4: Diagramme de transition et la matrice correspondant de la CMTD.	29
Figure 2.5: Diagramme de transition.....	31
Figure 2.6: matrice de taux de transitions (générateur infinitésimal).	31
Figure 2.7: la matrice de taux de transition et son diagramme de transition correspondant.	32
Figure 2.8: la matrice P et le diagramme de transition correspondant.	32
Figure 2.9: Diagramme de taux de transition pour le modèle de chaîne de Markov à 2 dimensions. [11]	34
Figure 2.10: Diagramme des états fusionnés du modèle de chaîne de Markov. [11]	35
Figure 2.11: structure du réseau à base de file d'attente. [12]	36
Figure 2.12: L'espace d'états du modèle [12].	37
Figure 2.13: modèle divisé avec la phase d'espace SK, $K=0, \dots, R_0$ [12].	39
3. Modélisation des réseaux de capteurs.	42
Figure 3.1: Diagramme d'état transition de la politique N-vacance.	47
Figure 3.2: La chaîne de Markov du modèle M0.	48
Figure 3.3: La CMTC du M0 après le découpage.....	50
Figure 3.4: la chaîne de Markov du modèle M1.	53
Figure 3.5: la chaîne de Markov à 2D après la décomposition et la division par colonne.	58
Figure 3.6: la chaîne de Markov à 2D après la décomposition et la division par colonne.	60
Figure 3.7: la chaîne de Markov du modèle M2.	64
Figure 3.8: la chaîne de Markov à 2D après la décomposition et la division par colonne.	69
Figure 3.9 : la chaîne de Markov à 2D après la décomposition et la division par colonne.....	70
Figure 3.10: la chaîne de Markov à 2D après la décomposition et la division par colonne.	72
Figure 3.11: la chaîne de Markov à 2D après la décomposition et la division par colonne.	73
4. Implémentation et étude expérimentale	77
Figure 4.1: la fenêtre d'accueil de notre application.	78
Figure 4.2: Fenêtre du choix du modèle à évaluer.	79
Figure 4.3: fenêtre à choix multiple.....	80
Figure 4.4: la fenêtre des calculs des indices de performances.....	81
Figure 4.5 : La consommation moyenne d'énergie en fonction du taux d'arrivé.	82
Figure 4.6 : le délai d'attente dans le buffer en fonction du taux d'arrivé.	83

Figure 4.7: la probabilité de blocage en fonction de taux d'arrivée.	83
Figure 4.8 : le nombre de paquets dans le buffer en fonction de taux d'arrivée.	84
Figure 4.9: la variation de nombre de paquets et le taux d'attente dans l'orbite en fonction de taux d'arrivée.....	84
Figure 4.10: la consommation d'énergie en fonction de taux de service.	85
Figure 4.11: le taux d'attente en fonction de taux de service.	86
Figure 4.12 : la probabilité de blocage en fonction de taux de service.	86
Figure 4.13: la consommation d'énergie en fonction de seuil N.	87
Figure 4.14: le délai d'attente en fonction de seuil N.....	87
Figure 4.15: le nombre de paquets dans le buffer en fonction de seuil N.	88

Liste des tableaux

Tableau 3-1: Tableau qui résume les travaux connexes.	46
Tableau 3-2: Représentation des taux de transitions entre les états.	48
Tableau 3-3: Représentation des taux de transitions entre les états.	54
Tableau 3-4: Représentation des taux de transitions entre les états.	66
Tableau 4-1: les paramètres du système.	82
Tableau 4-2: les paramètres de système de l'étude comparative.	89
Tableau 4-3: les résultats comparatifs.....	89

Liste des abréviations :

ADC : Analog to Digital Converter.

CH : Cluster Head.

CM : Cluster Member.

CMI : Chaîne de Markov Incluse.

CMTD : Chaîne de Markov à Temps Discret.

CMTC : Chaîne de Markov à Temps Continu.

CSIP: Collaborative Signal and Information Processing.

DVS: Dynamic Voltage Scaling.

FPGA: Field programmable Gate Array.

HSN: Heterogeneous Sensor Networks.

IoT : Internet of Thing.

MEMS: Modified Energy Minimization Scheme.

PQM : Priority Queuing Model.

RAM : Random Access Memory.

RCSF : Réseaux de Capteur Sans Fil.

RdP : Réseau de Petri.

RdPS : Réseaux de Petri Stochastiques.

ROM : Read Only Memory.

SB : Station de Base.

SM : Station Mobile.

WSN : Wireless Sensor Network.

Introduction générale

Les progrès réalisés dans les technologies MEMS (Micro-Electro-Mechanical Systems) ont permis le développement des dispositifs électroniques miniatures dits nœuds micro capteurs ou simplement nœuds capteurs. Ces derniers ont la capacité de mesurer des grandeurs physiques, telles que la chaleur, l'humidité, le mouvement, la pression, etc., et de les convertir en information numérique. Ils ont aussi des capacités de communication sans fil leur permettant de communiquer entre eux. Le déploiement d'un certain nombre de nœuds capteurs avec un ou plusieurs nœuds spéciaux dits sinks dans des zones qu'on veut surveiller forme un réseau de capteurs sans fil (WSN : Wireless Sensor Network). Les nœuds détectent des événements qui se produisent dans leur entourage et envoient des messages correspondants au sink, généralement d'une manière multi-sauts, qui à son tour, peut les relayer à grande échelle. Ces réseaux sont simples et rapides à déployer même dans des milieux hostiles ou difficilement accessibles, car ils ne nécessitent aucune infrastructure. De plus, leur coût est très réduit et ils peuvent par conséquent être à la disposition d'une grande communauté.

Grâce à ces avantages, de nombreuses applications de réseaux de capteurs ont émergé des domaines aussi variés que la défense, la sécurité, la santé, l'agriculture, etc. En revanche, ces réseaux confrontent plusieurs défis, que les protocoles doivent prendre en compte, comme la tolérance aux pannes, le passage à l'échelle, la gestion de la topologie du réseau. D'autres défis sont liés directement aux nœuds capteurs eux-mêmes. Comme ces nœuds sont de petite taille, ils ont par conséquent, des unités d'émission et de réception avec une faible portée du signal. De plus, ils sont équipés par des microcontrôleurs possédant des vitesses de traitement limitées. Quant à l'énergie, ils utilisent le plus souvent des batteries irremplaçables, et donc la durée de vie du nœud et celle du réseau est déterminée par l'autonomie des batteries. En outre, la capacité de stockage de chaque nœud est très restreinte en la comparant avec celle que nous pouvons trouver sur les autres appareils informatiques actuels.

Vues ces contraintes, l'analyse des réseaux de capteurs sans fil avant leur déploiement est nécessaire et même indispensable. Ainsi, on aura une vision sur les performances et la consommation d'énergie du réseau. L'analyse et l'évaluation des réseaux de capteurs se font généralement par simulation grâce aux outils logiciels, tels que NS-3 [1], OMNet++ [2], OPNET [3], etc. Cependant, l'inconvénient de la simulation est qu'elle donne des résultats différents, en dépendance de l'outil utilisé et des données introduit. D'autre part, pour obtenir des résultats proches de la réalité, la simulation devrait prendre un temps d'exécution très long. Une autre alternative, qui reste encore jeune dans le domaine des réseaux de capteur, est l'utilisation des méthodes formelles basées sur des abstractions mathématiques, et qui ont l'avantage de fournir des résultats exacts.

Parmi les modèles formels, nous trouvons les files d'attente, les réseaux de Petri, les réseaux d'automates stochastiques. Une file d'attente consiste en un espace d'attente, un espace de service composé d'un ou de plusieurs serveurs, un processus qui décrit l'arrivée des clients sollicitant un service, et un processus décrivant leur service. Un système de file d'attente avec vacance est une extension d'un système de file d'attente classique où le serveur peut être non disponible pendant une durée limitée. Durant cette période d'absence, le serveur peut prendre une vacance d'une tâche principale pour s'occuper d'une ou plusieurs tâches secondaires ou simplement pour se reposer. Un comportement naturel des clients arrivant et ne pouvant pas octroyer le service immédiatement est de quitter le système et répéter la demande ultérieurement ou bien attendre dans un espace d'attente. Ceci est connu sous le nom de phénomène de rappel. Un système de file d'attente avec un tel phénomène est dit file d'attente avec rappel. La combinaison des vacances du serveur et du phénomène de rappel conduit à un système de file d'attente avec vacance et rappel. Cependant, ces modèles sont plus complexes à analyser et les résultats

Introduction générale.

analytiques n'existent que pour certains cas particuliers. Quand la source d'où les clients arrivent est finie, on parle de file d'attente à source finie. Dans un tel système, le taux d'arrivée des clients diminue quand le nombre de clients dans le système augmente. À cause de la complexité des systèmes caractérisés par le phénomène de rappel, les chercheurs dans ce domaine se sont orientés vers des méthodes numériques ou approximatives.

L'analyse des réseaux de capteurs sans fil à l'aide des méthodes formelles a commencé à attirer l'attention des chercheurs ces dernières années, mais elle reste encore jouvencelle. Parmi les travaux existants, la plupart d'entre eux se focalisent sur la conservation d'énergie des nœuds capteurs. C'est le cas des travaux cités dans [4], [5], [6], [7], [8], [9], [10]. Cependant, ces auteurs ignorent la limitation des tailles des buffers des nœuds capteurs, qui est une contrainte forte et qui ne devrait pas être écartée dans un réseau de capteur sans fil. Par contre, d'autres travaux étaient conscients de cette limitation [5], [6], [7], [8], [9], mais ils restent insuffisants, car ils ne prennent pas en considération le débordement des buffers des capteurs et les retransmissions des messages.

Au cours de ce mémoire, nous nous intéressons à la conservation d'énergie des capteurs sans fil en proposant une approche formelle de modélisation et d'analyse des réseaux de capteurs sans fil en prenant en compte la limitation des buffers des nœuds capteurs. Pour cela nous introduisons la notion de vacance qui consiste à mettre un capteur en état de veille et introduire la notion de priorité entre les différentes catégories de message. De plus, nous considérons une source de messages qui est infinie et les retransmissions dues à la limitation du buffer (dans le but d'éviter la perte de messages). Pour la modélisation, nous utilisons le formalisme des files d'attente avec vacance, rappel et avec priorité. Ainsi, notre modèle combine entre l'attente, la vacance, le rappel et la priorité. Aucune solution analytique ou numérique n'existe dans la littérature pour un tel modèle. Pour cela, nous avons développé un algorithme permettant facilement de générer le générateur infinitésimal de la chaîne de Markov à temps continu sous-jacente à chaque modèle proposé. Nous développons aussi les formules des principaux indices de performance relatifs à ce modèle. Afin de réduire le temps d'exécution, nous avons proposé l'adaptation de la méthode approximative présentée dans [11], [12].

Le reste de ce document est organisé comme suit :

- ✓ Le premier chapitre est consacré aux généralités sur les réseaux de capteurs sans fil comme par exemple la définition de ce domaine, la consommation d'énergie des capteurs, leurs domaines d'application, etc.
- ✓ Le deuxième chapitre est réservé pour le formalisme mathématique, en donnant quelques rappels des différents types des files d'attente, les chaînes de Markov, la loi exponentielle, la loi de poisson, ainsi une description de la méthode approximative pour le calcul du vecteur des probabilités à l'état stationnaire pour les CMTCs à plusieurs dimensions.
- ✓ Dans le troisième chapitre, nous abordons les différents types de files d'attente et quelques travaux connexes à notre problématique. Puis, nous présentons notre approche de modélisation pour les nœuds de capteur basée sur les files d'attente. Les algorithmes de calcul des matrices infinitésimales des CMTCs des modèles proposés ainsi que l'adaptation de la méthode approximative sont décrits. Les formules des principaux indices de performances, énergie consommée et délai d'attente sont en déduit.
- ✓ Le dernier chapitre est consacré à l'implémentation de notre approche et une étude expérimentale.

Introduction générale.

Enfin, nous terminons par une conclusion générale et quelques perspectives.

1. Généralités sur les réseaux de capteurs sans fils.

1.1. Introduction

Les réseaux de communication sans fil ont connu un succès sans cesse croissant au sein des communautés scientifiques et industrielles. Grâce à ses divers avantages qui peuvent être résumés en trois points : la facilité du déploiement, l'ubiquité de l'information et le coût réduit d'installation. Au cours de son évolution, le paradigme sans fil a vu naître diverses architectures dérivées, telles que : les réseaux cellulaires, les réseaux locaux sans fils et autres. Durant cette dernière décennie, une architecture nouvelle a vu le jour : les réseaux de capteurs sans fil. Ce type de réseaux résulte d'une fusion de deux pôles de l'informatique moderne : les systèmes embarqués et les communications sans fil. Un réseau de capteurs sans fil (RCSF), ou "Wireless Sensor Network" (WSN), est composé d'un ensemble d'unités de traitements embarquées, communiquant via des liens sans fil. Le but général d'un RCSF est la collecte d'un ensemble de paramètres de l'environnement, telles que la température ou la pression de l'atmosphère, afin de les acheminer vers des points de traitement.

Malheureusement, les RCSFs ne sont pas parfaits à cause de leur déploiement dans des zones parfois hostiles, les capteurs sont assez fragiles et vulnérables à diverses formes de défaillances : cassure, faible énergie, etc. Ces problèmes rendent les RCSF des systèmes à fragilité innée, qui doit être considérée comme une propriété normale du réseau.

Dans ce qui suit, nous allons présenter c'est quoi un capteur, ses composants, son architecture, les réseaux de capteurs sans fil et ses domaines d'application, etc.

1.2. Un nœud de capteur

Un nœud de capteur est un ensemble de dispositifs électroniques (voir la figure 1.1), très souvent autonome en terme d'énergie, équipé d'un dispositif d'acquisition, ce dispositif lui permet de récolter des données.

Un nœud de capteur peut jouer le rôle de capteur et de routeur au même temps. Dans ce dernier cas, un nœud peut recevoir des données venant des autres nœuds et peut les retransmettre vers d'autres nœuds, avec ses propres données donc chaque capteur également appelé "nœud" est doté d'un circuit radio lui permettant de transmettre et de recevoir des informations via un médium sans fil [13].



Figure 1.1: Un nœud de capteur [13].

1.3. Les composants d'un nœud capteur

Un nœud de capteur intègre plusieurs unités qui sont [14] :

1.3.1. Une unité d'acquisition

C'est une unité de capture ou d'acquisition des grandeurs physique tels que la température, humidité, vibrations, luminosité, etc. Cette unité contient un convertisseur (appelée CAN pour Convertisseur Analogique Numérique) qui transforme les données analogiques du capteur en grandeurs numériques compréhensibles par l'unité de traitement.

1.3.2. Une unité de traitement

Elle est composée de deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication. Cette dernière est le contrôleur principal du nœud capteur sans fil. Elle est généralement associée à des unités de stockage de type RAM (Random Access Memory), ROM (Read Only Memory) ou flash. Le choix de ces mémoires (taille et type) varie selon la plateforme utilisée et le type d'application du nœud capteur. Cette unité au point de vue matériel peut être de type microcontrôleur, microprocesseur, ou encore FPGA (Field programmable Gate Array) ou les deux à la fois et pour le point de vue logiciel l'unité de traitement intègre toutes les applications spécialement conçues pour le nœud capteur pour piloter les différents modules matériels [14].

1.3.3. Une unité de communication

Cette unité est responsable de la connexion du nœud capteur au réseau afin qu'il puisse communiquer avec les autres nœuds via un médium sans fil. Elle peut être de type optique, ultrasons ou radiofréquence. En effet, le rôle de cette unité est de recevoir et transmettre les données.

1.3.4. Une batterie

C'est une unité importante dans le nœud capteur pour but de stocker l'énergie et la fournir sous la forme requise. Chaque composant dans le nœud sans fil est alimenté par cette source, et sa capacité limitée exige un fonctionnement au rendement optimum pour les tâches effectuées par le composant.

On peut trouver d'autres composants additionnels qui peuvent être implantés dans un capteur, on cite parmi eux :

- **Un système de localisation :**

Le nœud capteur peut être équipé d'un système de localisation spécifique (GPS pour les réseaux extérieurs). Il peut dans ce cas se localiser de manière autonome. Un nœud peut aussi se localiser au sein de son réseau par rapport à des balises.

- **Un système reconstitution de l'énergie :**

Il sert à reconstitue l'énergie consommée par un réapprovisionnement (grâce à une source externe : cellules solaires, température, vibration, etc.). C'est une technique de récupération d'énergie par l'énergie thermique, cinétique, etc. cette technique est utilisée pour des applications de réseau qui nécessitent une longue vie.

- **Un système de mobilité de nœud :**

Pour qu'un nœud puisse se déplacer dans le réseau, si ce dernier n'est pas rattaché à un appareil mobile, il lui faut un système de déplacement. Ce dernier permet de déplacer les nœuds pour accomplir ses tâches.

La figure 1.2 ci-dessous représente l'architecture physique d'un nœud capteur.

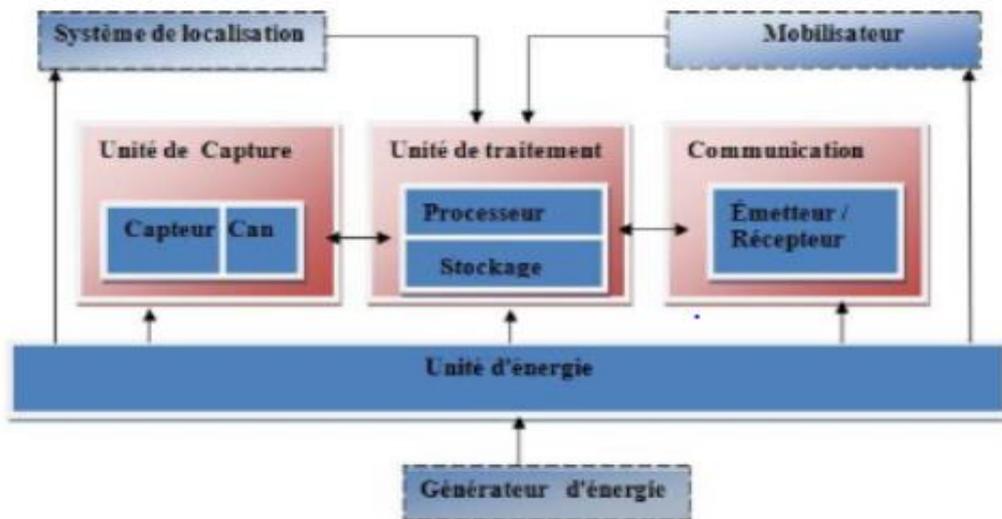


Figure 1.2 : Les composants d'un capteur [14].

1.4. Les réseaux de capteurs sans fils

Les réseaux de capteurs sans fils (représenté dans la figure 1.3) sont formés d'un très grand nombre de capteurs sans infrastructure préalable. Un capteur surveille son environnement, capte et propage les données récoltées des capteurs appartenant à sa zone de couverture. Chaque capteur relayant l'information sur sa propre zone de couverture [13].

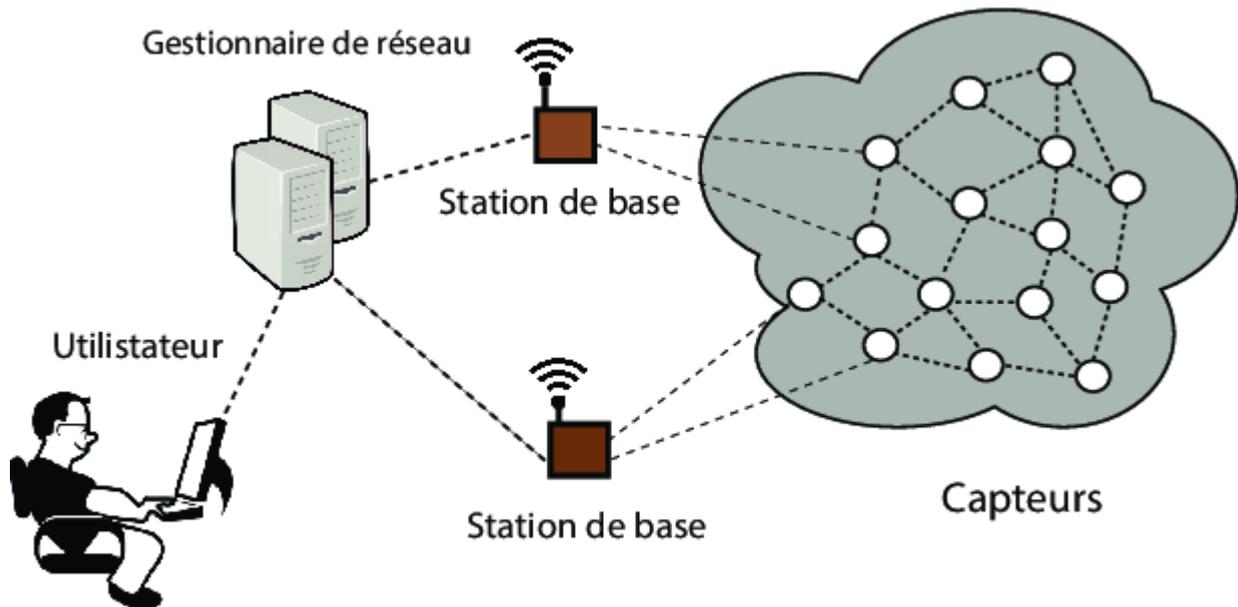


Figure 1.3: schéma général d'un réseau de capteur sans fil.

1.4.1. La durée de vie d'un réseau de capteurs

La durée de vie d'un réseau de capteurs est généralement définie par le temps durant lequel le réseau est capable de maintenir assez de connectivité, couvrir la zone de captage ou garder son taux de perte de nœud inférieur à un certain niveau. La durée de vie d'un réseau de capteurs est donc liée à la durée de vie des nœuds, qui dépend principalement de la durée de vie de sa batterie. Cette dernière dépend des énergies consommées par les différents modules qui composent le nœud, par la technologie de la batterie et la façon de s'en servir [14].

1.4.2. Architectures adoptées pour les RCSF

Un réseau de capteurs est généralement composé de nombreux nœuds de capteur déployés dans une zone. Ces nœuds sont reliés à une ou à plusieurs SB (Stations de Base) qui permettent l'interconnexion avec d'autres réseaux et la récupération des données. L'utilisateur du RCSF peut ainsi, adresser des requêtes aux autres nœuds capteurs du réseau en précisant le type de données requises et recueillir puis analyser les événements recensés par les capteurs sur la zone de couverture.

Les architectures dans les réseaux de capteurs dépendent essentiellement des applications et des techniques utilisées pour faire acheminer l'information des capteurs à la SB. Ce processus d'acheminement de l'information des capteurs vers la SB peut se faire selon deux formes [15] :

- **Une architecture à plat :**

Où les capteurs peuvent communiquer directement avec la SB (figure 1.4), ou en mode multi-sauts (figure 1.5).

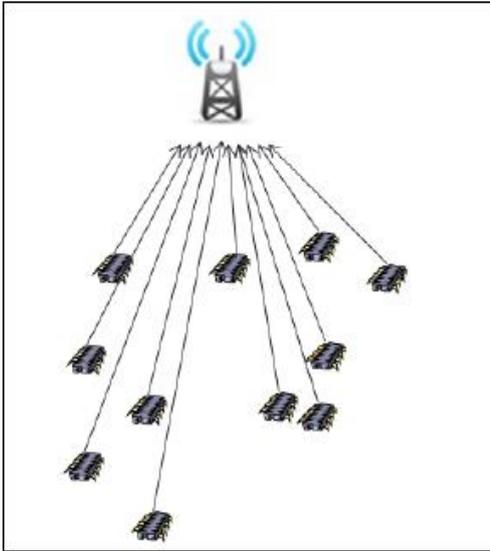


Figure 1.4: Transmission directe [15]

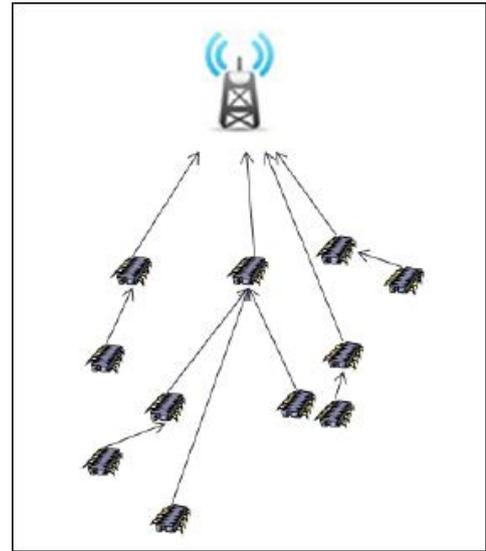


Figure 1.5: Transmission en multi-sauts [15].

- **Une architecture hiérarchisée :**

Qui consiste à regrouper les capteurs en groupes appelés clusters qui seront représentés par des clusters Head (CH) responsables de transmettre directement les données à la SB (Figure 1.6). Les réseaux de capteurs à base de cluster sont essentiellement séparés en deux types principaux ; homogènes et hétérogènes, dans les réseaux homogènes tous les nœuds de capteurs sont uniques en termes d'énergie dans la batterie et de la complexité du matériel et tous les nœuds peuvent être un CH au tour de rôle tandis que dans les réseaux de capteurs hétérogènes constitué de nœuds à ressources non égales c.à.d. Certains nœuds ont de haute performance peuvent être CH.

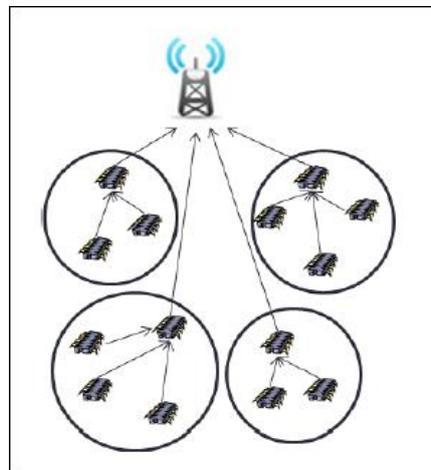


Figure 1.6: Clustering [15].

Il faut dire que l'architecture adoptée conditionne la répartition de la consommation d'énergie dans le réseau. En effet, dans le cas d'une transmission directe des données captées vers la SB, les capteurs

éloignés de celle-ci vont plus rapidement manquer d'énergie. Tandis que pour la transmission des données captées par sauts, les capteurs proches de la SB vont être vite manqué d'énergie car ils seront plus sollicités pour relayer les messages des autres. Au contraire, pour les architectures hiérarchisées, en changeant régulièrement les clusters-Head on obtient des réseaux où la consommation d'énergie est répartie équitablement.

1.5. Consommation d'énergie du capteur

L'énergie consommée par un nœud capteur est due essentiellement aux opérations suivantes : la capture, le traitement et la communication de données.

- **Energie de capture :**

Est dissipée pour accomplir l'échantillonnage, le traitement de signal, la conversion analogique/numérique et l'activation de la sonde de capture. En général, l'énergie de capture représente un faible pourcentage de l'énergie totale consommée par un nœud.

- **Energie de traitement :**

Est consommé dans la commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel). Par contre, l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à celle nécessaire pour la communication.

- **Energie de communication :**

Est consommé par la réception, l'émission et l'écoute du canal. Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance. Quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée. Notons que l'énergie de communication représente la portion la plus grande de l'énergie consommée par un nœud capteur. [16]

L'histogramme présenté par la figure ci-dessous représente les portions d'énergie consommée par les différentes unités.

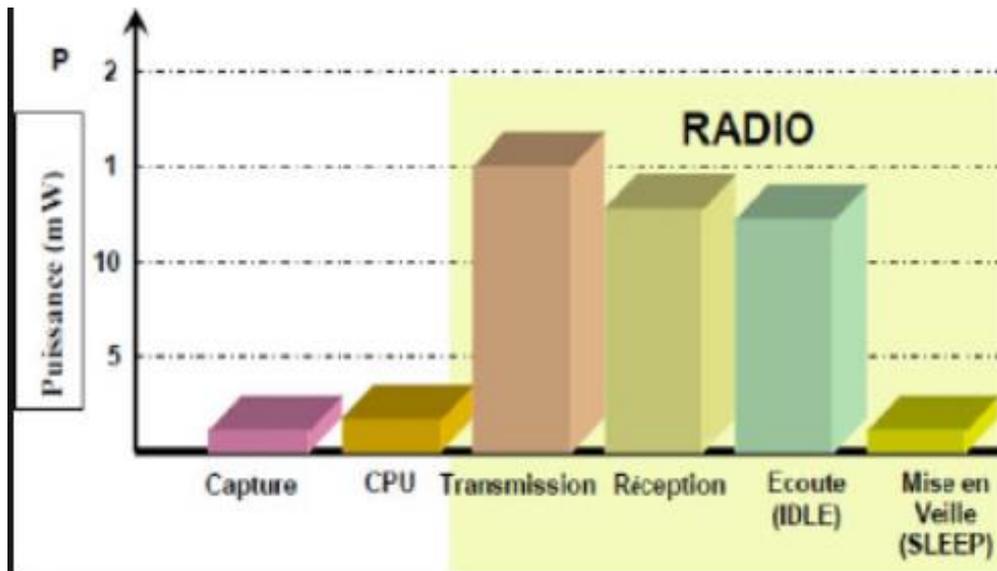


Figure 1.7: Consommation d'énergie en acquisition, traitement et communication [16].

1.5.1. Sources de surconsommation d'énergie

La conception d'un protocole MAC économisant l'énergie joue un rôle essentiel dans la réduction de la consommation d'énergie afin d'étendre la durée de vie des capteurs et du Réseau. Étant donné que les nœuds de capteurs sont supposés être disposés lorsqu'ils sont hors batterie, la conservation d'énergie est une exigence majeure dans la conception des protocoles de communication pour les réseaux de capteurs. Au cours de l'exploitation de RSCF, les principales sources de surconsommation d'énergie sont présentées dans les paragraphes suivants [17] :

État du module radio

Le module radio est le composant du nœud qui consomme le plus d'énergie car c'est lui qui assure la communication entre les nœuds de capteurs. On distingue quatre modes des composants radio (transmetteur et récepteur) : actif, réception, transmission et sommeil.

- **Etat actif :**

- ✓ **Etat de réception :**

La radio est en mode de réception d'un paquet.

- ✓ **Etat de transmission :** la radio est en mode de transmission d'un paquet.

Il est aussi à noter que le passage fréquent du mode actif au mode sommeil peut avoir comme effet une consommation d'énergie plus élevée que de laisser le module radio en mode actif. Ceci est dû à la puissance nécessaire pour la mise sous tension du module radio. Cette énergie est appelée l'énergie de transition. Ce changement d'état du module radio doit être géré par un protocole de la couche MAC.

- **Etat sommeil :** La radio est activée, mais elle est mise en veille. En d'autres termes, le nœud de capteur n'est ni en train de recevoir ni de transmettre. Cet état provoque une perte d'énergie suite à l'écoute inutile du canal.

Chapitre 1 : Généralités sur les réseaux de capteurs sans fils.

Les collisions

Elles sont la première source de perte d'énergie. Quand deux trames sont émises en même temps et se heurtent, elles deviennent inexploitables et doivent être abandonnées. Elles se manifestent alors lorsqu'un nœud reçoit plus d'un paquet simultanément, tous les paquets qui causent la collision doivent être jetés et retransmis, leur retransmission consomme plus d'énergie. Tous les protocoles Mac avec contention intègrent un processus d'évitement de collisions. Les nœuds capteurs sont composés essentiellement par une seule antenne radio et partagent le même canal de transmission. Par ailleurs, l'émission simultanée des données provenant de différents nœuds capteurs peut amener à l'apparition des collisions et ainsi une éventuelle perte de l'information transmise. La retransmission des paquets perdus peut causer une perte significative de l'énergie.

L'écoute active/ inactive

L'écoute inactive se produit lorsqu'un nœud écoute le canal pour une réception possible. C'est une source de gaspillage importante d'énergie dans les RCSF où il consomme habituellement 50 à 100% de l'énergie requise pour le processus de réception. L'écoute active du canal pour une telle réception de paquet qui ne sera pas reçue peut engendrer une grande perte de la capacité des nœuds en énergie. Il faut donc basculer les capteurs vers le mode sommeil le plus longtemps possible dans le but d'éviter ce problème.

La sur-écoute (Overhearing)

Cette source de dissipation d'énergie se produit lorsqu'un nœud capteur gaspille de l'énergie en recevant un paquet destiné à une destination différente. La sur-écoute apparaît donc quand un capteur reçoit des paquets qui ne lui sont pas destinés. La sur-écoute conduit à une perte d'énergie additionnelle à cause de l'implication des autres capteurs dans la réception des données.

La surcharge (Overhead)

Les paquets de données dans les RCSF sont généralement petits ; par conséquent, les en-têtes et les autres types de paquets (tels que les messages de contrôle, comme RTS, CTS, SYNC et ACK) impliquent un niveau élevé de gaspillage d'énergie. Plusieurs protocoles de la couche MAC fonctionnent par échange de messages de contrôle pour assurer différentes fonctionnalités : connectivité, signalisation, établissement de plan d'accès et évitement des collisions.

La surémission (Overemitting)

La surémission est causée lorsque la livraison du message échoue en raison de l'inactivité du nœud de destination. Ce phénomène apparaît aussi quand un nœud capteur envoie les données à un destinataire qui n'est pas prêt à les recevoir. En effet, les messages envoyés sont considérés inutiles et consomment une énergie additionnelle.

La taille des paquets

La taille des messages échangés dans le réseau a un effet sur la consommation d'énergie des nœuds émetteurs et récepteurs. Ainsi, la taille des paquets ne doit pas être ni trop élevée ni trop faible. En effet, si elle est petite, le nombre de paquets de contrôle (acquiescement) générés augmente l'overhead. Dans le cas contraire, une grande puissance de transmission est nécessaire pour des paquets de grande taille.

1.6. Techniques de minimisation d'énergie

Dans les réseaux de capteurs, la consommation d'énergie est très importante puisque généralement les capteurs sont déployés dans des zones inaccessibles. Ainsi, il est difficile de voir et impossible de remplacer les batteries après leur épuisement. De ce fait, la consommation d'énergie au niveau des capteurs a une grande influence sur la durée de vie du réseau. Nous présentons dans ce qui suit les différentes techniques utilisées pour minimiser cette consommation. Ces techniques sont appliquées soit au niveau de la couche liaison soit au niveau de la couche réseau. Le schéma présenté par (figure 1.8) donne un aperçu global de ces mécanismes [18].

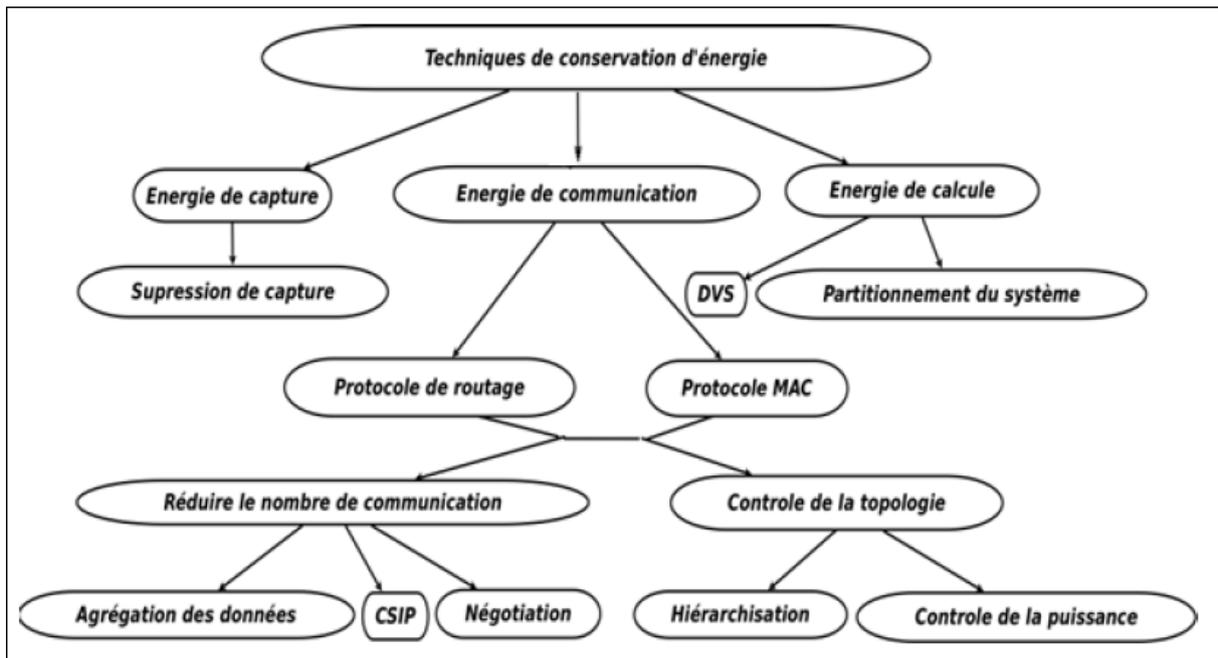


Figure 1.8: Techniques de conservation d'énergie [18].

1.6.1. L'énergie de capture

La seule solution apportée pour la minimisation de la consommation d'énergie au niveau de la capture consiste à réduire les fréquences et les durées de captures.

1.6.2. L'énergie de calcul

Peut-être optimisée en utilisant deux techniques :

- **L'approche DVS (Dynamique Voltage Scaling) :**

Elle consiste à ajuster de manière adaptative la tension d'alimentation et la fréquence du microprocesseur pour économiser la puissance de calcul sans dégradation des performances.

- **L'approche de partitionnement de système :**

Elle consiste à transférer un calcul prohibitif en temps de calcul vers une station de base qui n'a pas de contraintes énergétiques et qui possède une grande capacité de calcul.

1.6.3. L'énergie de communication

La minimisation de la consommation d'énergie pendant la communication est étroitement liée aux protocoles développés pour la couche réseau et la sous-couche MAC. Ces protocoles se basent sur plusieurs techniques : l'agrégation de données, la négociation et à la technique CSIP (Collaborative Signal and Information Processing). Cette dernière technique est utilisée pour le traitement collaboratif du signal et de l'information, elle est une discipline qui combine plusieurs domaines : la communication et le calcul à basse puissance, le traitement de signal, les algorithmes distribués, la tolérance aux fautes, les systèmes adaptatifs et la théorie de fusion des capteurs et des décisions. Ces techniques ont le but de réduire le nombre d'émission/réception des messages.

1.6.4. Technique de duty cycling

Cette technique est principalement utilisée dans l'activité réseau. Le moyen le plus efficace pour conserver l'énergie est de mettre la radio de l'émetteur en mode veille « low-power » à chaque fois que la communication n'est pas nécessaire. Idéalement, la radio doit être éteinte dès qu'il n'y a plus de données à envoyer et ou à recevoir, et devrait être prête dès qu'un nouveau paquet de données doit être envoyé ou reçu. Ainsi, les nœuds alternent entre période active et sommeil en fonction de l'activité du réseau. Ce comportement est généralement dénommé «Duty-cycling ». Ce dernier est défini comme étant la fraction de temps où les nœuds sont actifs. [17]

1.7. Contraintes et caractéristiques

Un RCSF est composé d'un grand nombre de nœuds capteurs qui sont déployés à forte densité en dedans du phénomène surveillé ou à proximité de celui-ci. Les positions des nœuds de capteurs ne doivent pas forcément être prédéterminées. Cette caractéristique unique permet aux RCSF d'être déployés dans les terrains inaccessibles. En outre, cela signifie aussi que les protocoles et les algorithmes dédiés aux réseaux de capteurs doivent posséder des capacités d'auto-organisation. Une autre caractéristique importante des réseaux de capteurs est la capacité de coopération des nœuds de capteurs.

En effet, les nœuds de capteurs sont équipés de microcontrôleur embarqué qui leurs permettent d'effectuer des traitements locaux des données et des calculs simples. Ainsi, ils ne transmettent que les données requises et partiellement transformées. Par conséquent, les réseaux de capteurs fourniront à l'utilisateur final une meilleure compréhension de l'environnement. Nous prévoyons que, à l'avenir, les RCSF feront partie intégrante de nos vies, plus que les ordinateurs personnels actuels.

Dans ce contexte, les RCSF sont souvent considérés comme étant les successeurs des réseaux Ad-hoc (figure 1.9).

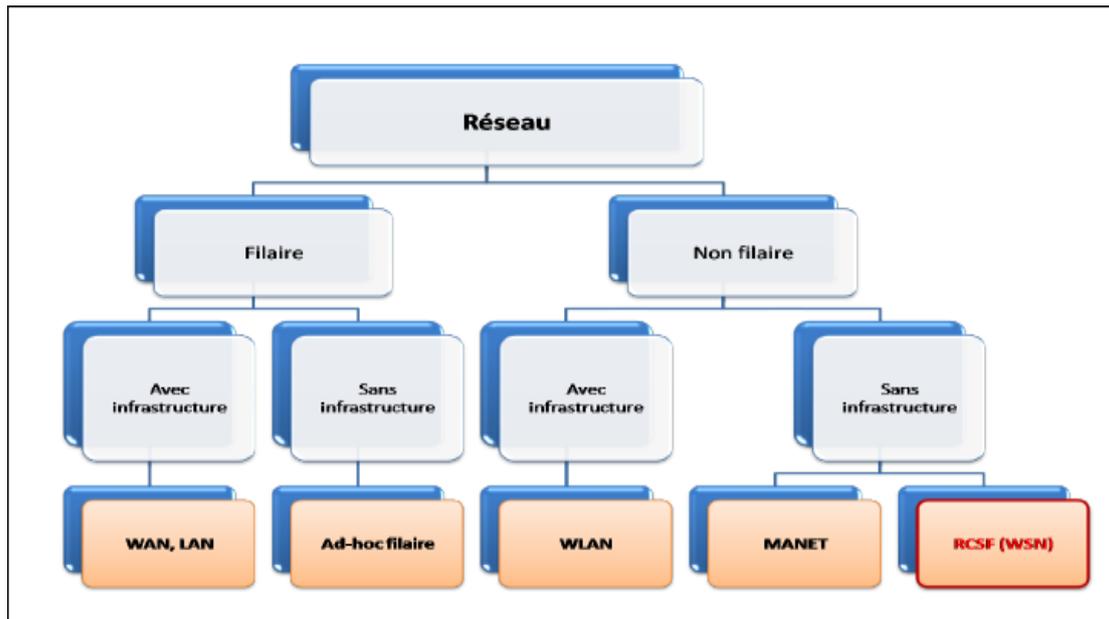


Figure 1.9: Classification des réseaux de communication [15].

En effet, les RCSF partagent avec les réseaux Ad-hoc plusieurs propriétés en commun, telles que l'absence d'infrastructure et les communications sans fil. Bien que de nombreux protocoles et algorithmes ont été proposés pour les réseaux Ad-hoc sans fil, ils ne sont pas bien adaptés pour les caractéristiques uniques et les exigences des réseaux de capteurs. Pour illustrer ce point, les différences entre les réseaux de capteurs et les réseaux Ad-hoc sont décrites ci-dessous :

- Dans les réseaux de capteurs, les nœuds sont déployés en grand nombre.
- Les nœuds de capteurs sont sujets à des défaillances.
- La topologie des réseaux de capteurs change très fréquemment.
- Les réseaux de capteurs utilisent principalement le paradigme de communication points tandis que les réseaux ad hoc sont basés sur le paradigme de communication point à point.
- Les capteurs sont limités en énergie, en capacités de calcul, et en mémoire.
- Les réseaux ad-hoc se focalisent plus sur la qualité de service que sur la consommation de l'énergie [15].

1.8. Les domaines d'application des réseaux de capteurs

La miniaturisation des capteurs, le coût de plus en plus faible et la large gamme de capteurs disponibles (thermique, optique, vibrations, etc.) ainsi que le support de communication sans fil utilisé, permettent l'utilisation des réseaux de capteurs dans plusieurs domaines parmi lesquels [19] :

- **Domaine militaire :**

Comme pour de nombreuses autres technologies, le domaine militaire a été le moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la

Chapitre 1 : Généralités sur les réseaux de capteurs sans fils.

tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Actuellement, les RCSF peuvent être une partie intégrante dans le commandement, le contrôle, la communication, la surveillance, la reconnaissance, etc.

- **Domaine médical :**

Les réseaux de capteurs sont également largement répandus dans le domaine médical. Cette classe inclut des applications comme : fournir une interface d'aide pour les handicapés, collecter des informations physiologiques humaines de meilleure qualité, facilitant ainsi le diagnostic de certaines maladies, surveiller en permanence les malades et les médecins à l'intérieur de l'hôpital.

- **Domaine agricole :**

Questionner le réseau de capteurs sur l'état du champ (déterminer par exemple les secteurs les plus secs afin de les arroser en priorité). On peut aussi imaginer équiper des troupeaux de bétail de capteurs pour connaître en tout temps, leur position ce qui éviterait aux éleveurs d'avoir recours à des chiens de berger.

- **Domaine environnemental :**

Dans ce domaine, les capteurs peuvent être exploités pour détecter les catastrophes naturelles (feux de forêts, tremblements de terre, etc.), détecter des émanations de produits toxiques (gaz, produits chimiques, pétrole, etc.) dans des sites industriels tels que les centrales nucléaires ou pétrolières.

- **Domaine commercial :**

Parmi les domaines dans lesquels les réseaux de capteur sont aussi prouvés leur utilité, on trouve le domaine commercial. Dans ce secteur nous pouvons énumérer plusieurs applications comme : la surveillance de l'état du matériel, le contrôle et l'automatisation des processus d'usinage, etc.

- **Applications domestiques :**

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, magnétoscopes...etc. Ces capteurs embarqués peuvent contrôler les appareils domestiques localement ou à distance. D'où on peut économiser l'énergie en gérant l'éclairage ou le chauffage en fonction de la localisation des personnes si cela est nécessaire.

1.9. Facteurs influençant la conception des RCSF

La conception et la réalisation des réseaux de capteurs sans fils, leurs protocoles et algorithmes sont influencées par de nombreux facteurs et contraintes [19] :

1.9.1. Contraintes Conceptuelles

On distingue plusieurs contraintes conceptuelles, elles sont définies comme suit :

- **La tolérance aux pannes :**

La défaillance ou le blocage de certains nœuds dans un réseau de capteurs peut être engendrés par plusieurs causes, notamment l'épuisement d'énergie, l'endommagement physique ou les interférences

Chapitre 1 : Généralités sur les réseaux de capteurs sans fils.

liées à l'environnement. Ces problèmes ne devraient pas affecter le reste du réseau. C'est le principe de la tolérance aux pannes.

- **L'extensibilité (passage à l'échelle) :**

L'une des caractéristiques des RCSF est qu'ils peuvent contenir des centaines voire des milliers de nœuds capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs. Les nouveaux schémas doivent pouvoir garantir un bon fonctionnement avec ce nombre élevé de capteurs. Ils doivent aussi exploiter la nature fortement dense des réseaux de capteurs.

- **L'environnement :**

Les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : forte chaleur, pluie, humidité, etc.

- **Le média de transmission :**

Les nœuds communicants sont reliés sans fil. Ce lien peut être réalisé par radio, signal infrarouge ou un média optique.

- **L'énergie de stockage et de calcul :**

La caractéristique la plus critique dans les réseaux de capteurs est la modestie de ses ressources énergétiques car chaque capteur du réseau possède de faibles ressources en termes d'énergie, de calcul et de stockage. Afin de prolonger la durée de vie du réseau, une minimisation des dépenses énergétiques est exigée au niveau de chaque nœud.

- **L'agrégation de données :**

Dans les RCSF, les données produites par les nœuds capteurs voisins sont corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'information redondante. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau ainsi d'améliorer sa durée de vie. L'une des techniques utilisées pour réduire la transmission d'informations redondantes est l'agrégation des données, appelée aussi fusion des données.

1.9.2. Les contraintes matérielles

Parmi les contraintes matérielles liées aux RCSF, on peut citer :

- **La dimension :**

La taille réduite des capteurs peut présenter de nombreux avantages, elle permet un déploiement flexible et simple du réseau. Cependant, la puissance des batteries utilisées pour alimenter les nœuds capteurs est limitée par la petite taille de ces derniers.

- **La puissance de calcul :**

Les processeurs des réseaux de capteurs sont différents de ceux d'une machine classique car ils utilisent souvent des micros contrôleurs de faibles fréquences.

1.10. Modèle de communication des RCSF

Le modèle de communication peut être fait selon l'approche traditionnelle de la pile protocolaire qui comprend plusieurs couches ou bien en utilisant l'approche inter couches, également appelée multicouches telle décrite dans la figure 1.10. Cette approche est connue sous sa terminologie anglophone "Cross-layer" où le modèle combine trois plans de gestion verticaux pour assurer les fonctionnalités inter-couches, à savoir : la gestion de mobilité du réseau, la coordination des tâches entre les capteurs et le partage des ressources en utilisant judicieusement l'énergie disponible [18].

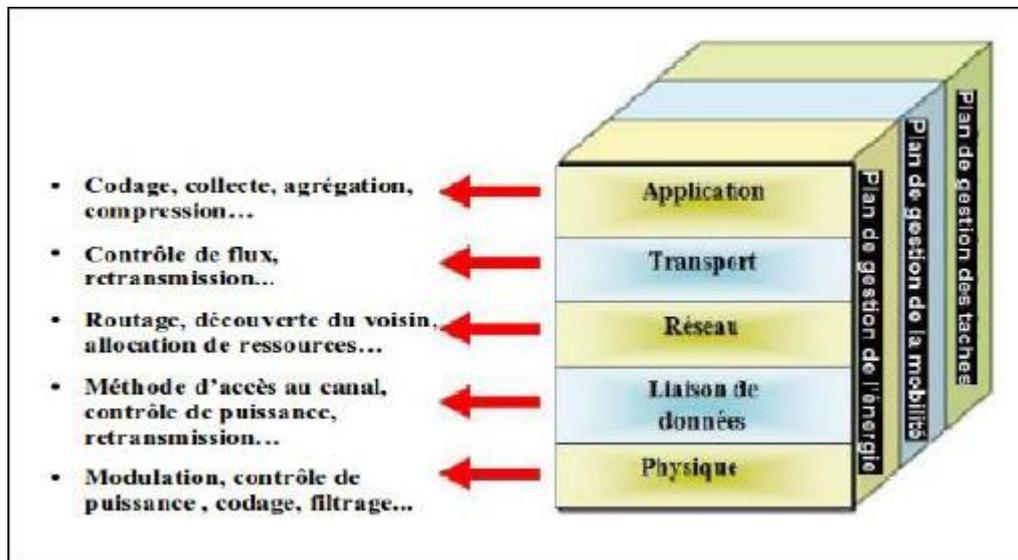


Figure 1.10: Pile protocolaire d'un RCSF [20].

1.10.1. Couches de communication

Cette pile comprend cinq couches de communications qui sont [14] :

- **La couche application** : représente les différentes applications et pilotes : codage, collecte, agrégation, compression, etc. Elle assure l'interface avec les applications. C'est le niveau le plus proche des utilisateurs, géré directement par les logiciels.
- **La couche transport** : est chargée du transport des données, contribue au maintien du flux si c'est nécessaire (contrôle de flux, gestion des éventuelles erreurs de transmission non corrigée au niveau 2, etc.), au découpage des données en paquets, ainsi que la conservation de l'ordre des paquets.
- **La couche réseau** : s'occupe de l'acheminement des données fournies par la couche transport (routage, relais, etc.) Le protocole de routage est l'acteur principal dans cette couche, il permet de trouver une route et une transmission performante des données. Plusieurs nouveaux algorithmes de routage ont été conçus pour les problèmes de routage dans les réseaux capteurs (optimisation de l'utilisation de l'énergie des capteurs, gestion des trafics, etc.).
- **La couche liaison de données** : spécifie comment les données sont expédiées entre deux nœuds dans une distance d'un saut. Cette couche s'occupe du multiplexage des données, elle s'occupe aussi de la méthode d'accès au canal, du contrôle de puissance, de la retransmission, et du contrôle d'erreurs. Cependant, elle assure la liaison point à point et multipoint dans un réseau de

Chapitre 1 : Généralités sur les réseaux de capteurs sans fils.

communication. Le protocole MAC dans cette couche liaison doit tenir en compte de la consommation d'énergie et doit être en mesure de réduire les collisions.

- **La couche physique** : est relative à la technologie ou l'architecture utilisée, la modulation, la technique de transmission et de réception, le contrôle de puissance, le codage, le filtrage, etc.

1.10.2. Plans de gestion

La conception des protocoles dans les réseaux classiques est basée principalement sur la pile protocolaire traditionnelle où chaque couche de la pile est conçue pour fonctionner séparément l'une de l'autre. Les interfaces entre les couches sont statiques et insuffisante en termes d'optimalité et de flexibilité, ce qui ne s'adapte pas avec la limitation des ressources des capteurs.

C'est pourquoi une nouvelle vision s'impose, celle du "Cross-layer design", qui peut être vu comme un moyen d'assouplir les limites de la pile traditionnelle. Son objectif est d'être flexible et efficace pour la mise en œuvre des protocoles de communication. L'idée consiste à rompre le modèle monolithique en plusieurs plans associés et hiérarchiques, où il n'y a pas de frontières strictes entre les différentes couches. Les messages de contrôle échangés comportent des informations de statut sur chaque couche, de sorte que le système tire profit de cette indépendance.

Les plans de gestion de tâches, de mobilité et d'énergie permettent ainsi aux capteurs de se coordonner et de réaliser leurs missions en étroite collaboration. Cette synergie entre les plans permet d'acheminer les données recueillies et partager les ressources en utilisant efficacement l'énergie disponible, ce qui prolonge positivement la durée de vie du réseau [20].

- **Plan de gestion d'énergie** : contrôle l'utilisation de la batterie. Après la réception d'un message, le capteur éteint son récepteur afin d'éviter la duplication des messages déjà reçus. En outre, si le niveau d'énergie devient bas, le capteur diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage et par conséquent, l'énergie restante est réservée au captage.
- **Plan de gestion de mobilité** : détecte et enregistre le mouvement du capteur. Ainsi, un retour arrière vers l'utilisateur est toujours maintenu et le capteur peut garder la trace de ses voisins et peut balancer l'utilisation de son énergie et la réalisation de tâche.
- **Plan de gestion de tâches** : balance et ordonnance les différentes tâches de collecte de données dans une région spécifique. Il n'est pas nécessaire que tous les capteurs d'une région effectuent le captage au même temps. En revanche, certains d'entre eux exécutent cette tâche plus que d'autres selon leur niveau de batterie.

1.11. Conclusion

Dans ce chapitre nous avons procédé à l'étude des réseaux de capteurs sans fil. Nous avons posé les briques de base et fédéré quelques concepts nécessaires à la compréhension de nos problématiques dans la suite de ce manuscrit.

Cela fait des années que les réseaux de capteurs suscitent un engouement important dans la recherche. Nous avons remarqué à travers nos lectures que « minimiser la consommation d'énergie d'un nœud de capteur » est « le cheval de bataille » de toutes les solutions et protocoles proposés.

Nous aborderons dans la suite de ce mémoire l'application des modèles formels pour l'obtention de résultats exactes. Nous nous intéressons particulièrement à l'analyse des RCSFs en utilisant les formalismes de files d'attente avec vacance et rappel et chaînes de Markov pour déterminer ses performances. De ce fait, nous allons présenter dans le chapitre suivant, les ingrédients nécessaires à la compréhension de ces formalismes

2. Formalismes mathématiques.

2.1. Introduction

Les files d'attente peuvent être considérées comme un phénomène caractéristique de la vie contemporaine, on les rencontre dans divers domaines d'activité. La théorie des files d'attente est une théorie mathématique relevant du domaine des probabilités, qui étudie les solutions optimales de gestion des phénomènes d'attente. L'étude mathématique des phénomènes d'attente constitue un champ d'application important des processus stochastiques.

Une façon pour analyser un système est de représenter son fonctionnement de manière plus ou moins précise. Pour cela, on s'appuie sur des outils (descriptifs, mathématiques ou autres) permettant d'approcher le comportement du système, afin d'évaluer les performances des systèmes.

En effet, le mot performance signifie accomplissement, réalisation et résultats réels. La notion de performance est complexe. Elle est multicritère et s'articule autour de plusieurs enjeux à savoir le cout, l'amélioration continue, la qualité de service et les délais des services.

Dans ce chapitre, nous allons présenter les formalismes mathématiques : les files d'attente avec rappel, les files d'attente avec vacance et les files d'attente avec priorité. Cependant, l'analyse de ces modèles nécessite la maîtrise des processus stochastiques et plus particulièrement des chaînes de Markov, qui feront l'objet d'une section. La méthode approximative connue sous le nom Phase Merging Algorithm pour le calcul des chaînes de Markov multidimensionnel sera présentée aussi. Nous terminons le chapitre par une conclusion.

2.2. Les files d'attente

2.2.1. Définition

Une file d'attente ou queue est un système composé d'un espace d'attente, d'un ou plusieurs serveurs et des clients qui arrivent, attendent, se font servir selon des règles de priorité données et quittent le système, Une représentation graphique d'une file d'attente classique est donnée par la figure (2.1) sous-dessous.

La théorie des files d'attente consiste à modéliser et à analyser de nombreuses situations en apparence très diverses. Elle fut développée pour fournir des modèles permettant de prévoir le comportement de systèmes répondant à des demandes aléatoires.

Les files d'attente sont généralement considérées comme outil très puissant d'évaluation des mesures de performance de divers systèmes réels [21].



Figure 2.1: Système de file d'attente [9].

2.2.2. Caractéristiques des files d'attente

Un système de files d'attente est l'abstraction mathématique d'un sujet qu'on peut décrire par les éléments suivants [22] :

- _ Le flux des arrivées des clients,
- _ La source des clients,
- _ La loi de la durée de service de chaque client,
- _ La discipline de service,
- _ Le nombre de serveurs,
- _ La capacité de la file.

- **Le flux des arrivées des clients :**

Il définit la distribution des intervalles séparant deux arrivées consécutives. D'habitude, on suppose que les temps inter-arrivées sont indépendants et identiquement distribués.

- **La source des clients :** La source peut être finie ou infinie selon le modèle réel qu'elle modélise. Cependant, les modèles réels sont souvent à source limitée mais on suppose souvent qu'elle est infinie quand sa capacité est assez grande.
- **La durée de service :** Elle décrit la durée du temps de service des clients. En général, on suppose que les durées de service sont indépendantes, identiquement distribuées et indépendantes des temps des inters arrivées.
- **La discipline de service :** Les clients peuvent être servis individuellement ou par groupe. Cependant, plusieurs possibilités existent quant à l'ordre selon lequel ils seront servis.

Les principales disciplines de service sont :

FIFO (First In First Out)

Les entités sortent dans l'ordre suivant lequel elles sont entrées, cette discipline est la plus utilisée.

LIFO (Last In First Out)

La dernière entité dans la file est la première à être servie.

Random

Toutes les entités ont la même probabilité d'être servies en premier.

Prioritaire

Les entités sont servies suivant un attribut qui leur est associé, par exemple l'entité ayant le plus court temps de traitement d'abord.

- **Le nombre de serveurs** : Il peut être égal à l'unité ou plus selon la nature du service à fournir.
- **La capacité de la file** : Dans pas mal de cas, la file est supposée infinie. Cependant, il n'est pas rare de rencontrer des situations dans lesquelles elle est finie.

2.2.3. Notation de Kendall

Pour décrire un système de files d'attente, il suffit de donner ses paramètres par la notation de Kendall suivante [23] :

A/B/C[/D/E] où :

*A : Processus d'arrivée des clients dans la file :

M pour Markovian ou memoryless, correspondant à un processus d'arrivée Poisson ;

D pour determinist ;

G pour generally distributed ;

*B Distribution du temps de service :

M pour Markovian ou memoryless, correspondant à un temps de service distribué selon la loi exponentielle ;

D pour determinist ;

G (ou GI) pour generally distributed (et independent) ;

*C Nombre de serveurs :

*D Nombre de places dans la file : en comptant les clients en service sur un serveur

Valeur par défaut : 1.

*E Discipline de service :

Valeur par défaut : FIFO.

2.3. Types des systèmes d'attente

2.3.1. Les systèmes de files d'attente avec priorité

Il n'est pas rare, dans la vie courante de rencontrer des systèmes d'attente avec plusieurs types de clients, où certains sont prioritaires par rapport à d'autres. On distingue deux types de priorités : priorité relative et priorité absolue.

- **Priorité relative :**

La priorité relative se caractérise par le fait qu'un client de priorité supérieure ne peut en aucun cas interrompre le service d'un client priorité inférieure. Il doit attendre (s'il le désire bien sûr) jusqu'à ce que le service du client de priorité inférieure soit fini avant d'être servi.

Le système d'attente avec priorité relative peut aussi être considéré comme un système d'attente à serveur non fiable, la panne est prise en considération après que le client en service ait terminé son service. Ce modèle est connu sous le nom " Breakdown postponible interruption ".

- **Priorité absolue :**

Contrairement au cas précédent, ici, un client de priorité supérieure a le droit d'interrompre le service d'un client de priorité inférieure pour se faire servir. Le service interrompu sera alors repris à partir du point où il était suspendu.

2.3.2. Les systèmes de file d'attente avec rappels

Un système d'attente avec rappels est un système composé de c ($c \geq 1$) serveurs identiques et indépendants, d'un buffer de capacité $K - c$ ($K \geq c$) et d'une orbite de capacité N . À l'arrivée d'un client, s'il y a un ou plusieurs serveurs libres et en bon état, le client sera servi immédiatement et quittera le système à la fin de son service. Sinon, s'il y a une position d'attente libre dans le buffer, le client la rejoindra. Par ailleurs, si un client arrive et trouve tous les serveurs et toutes les positions d'attente du buffer occupés, il quittera le système définitivement avec la probabilité $1 - H_0$, ou bien entre en orbite avec la probabilité H_0 et devient une source d'appels répétés et tentera sa chance après une durée de temps aléatoire. Les clients qui reviendront et rappelleront pour le service sont dits en "orbite". Cette dernière peut être finie ou infinie. Dans le cas d'une orbite à capacité finie, si elle est pleine, un client qui trouve tous les serveurs et les positions d'attente du buffer occupés, sera obligé de quitter le système définitivement sans être servi. Chaque client en orbite appelé aussi « client secondaire », est supposé rappeler pour le service à des intervalles de temps suivant une loi de probabilité et une intensité de rappels bien définie (rappels constants, rappels classiques, ou bien rappels linéaires, ...). Chacun de ces clients secondaires est traité comme un client primaire c'est-à-dire un nouveau client qui arrive de l'extérieur du système. S'il trouve un serveur libre, il sera servi immédiatement puis quittera le système. Sinon, s'il y a des positions d'attente disponibles dans le buffer, il le rejoindra. Par contre, si tous les serveurs et les positions d'attente sont encore occupés, le client quittera le système pour toujours avec la probabilité $1 - H_K$ (si c'est le k me rappel sans succès) ou bien entre en orbite avec la probabilité H_K si l'orbite n'est pas pleine. Le schéma général d'un système d'attente avec rappels est donné par la figure (2.2) [22].

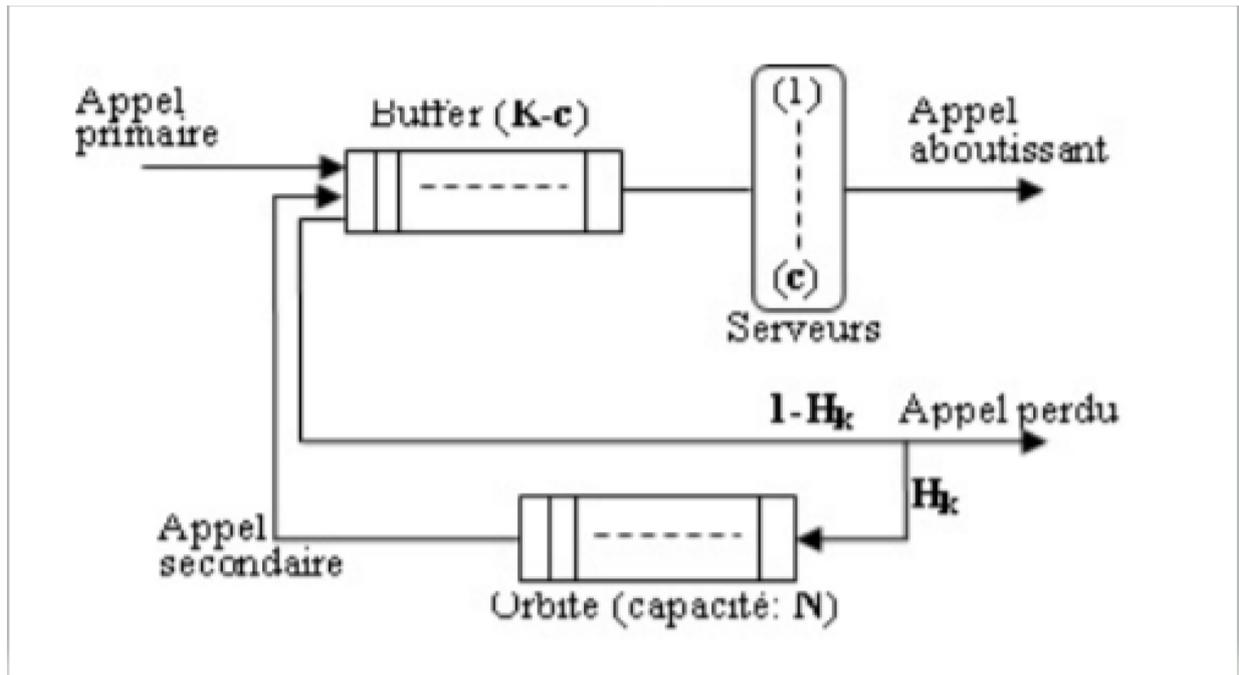


Figure 2.2 : Le schéma général d'un système d'attente avec rappels [22].

2.3.3. Système de file d'attente avec vacances

2.3.3.1. Définition

Les files d'attente avec vacance sont des files d'attente dans lesquels le serveur prend des vacances d'une tâche primaire pour s'occuper d'une ou plusieurs tâches secondaires ou pour se reposer ou pour accomplir quelques tâches de maintenance préventive, pendant ce temps les clients attendent d'être servis. Cette classe de files d'attente peut se trouver dans plusieurs applications [24].

La représentation graphique d'un système de files d'attente avec vacance est donnée dans le schéma suivant :

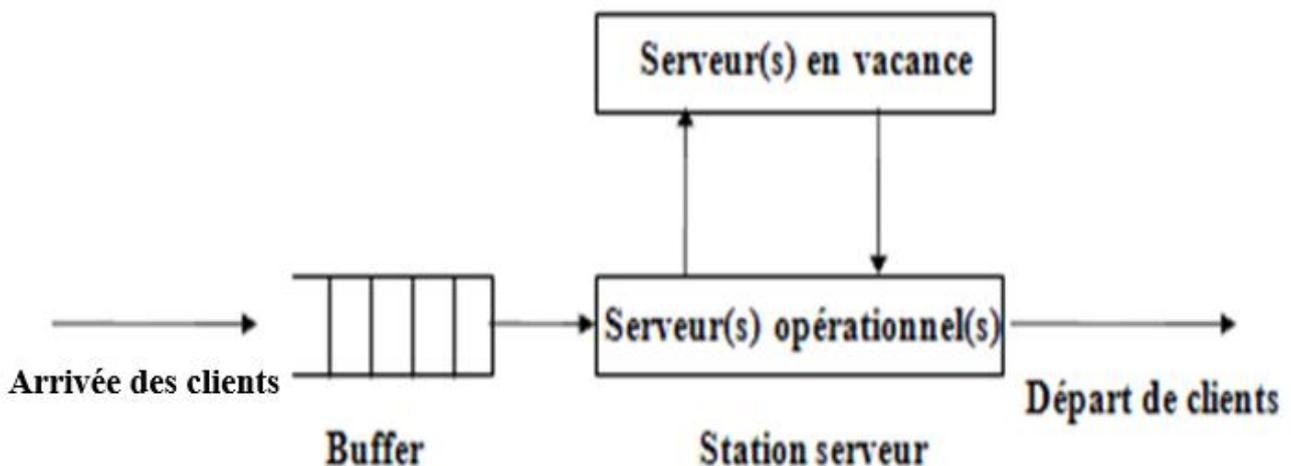


Figure 2.3: Système de file d'attente avec vacance [12].

2.3.3.2. Notation

Les files d'attente avec vacance sont notées par une séquence de symboles :

$$A/B/S/L/N/K_v$$

Où

A : distribution des inter-arrivées.

B : distribution de service.

S : nombre de serveurs.

L : capacité du système.

N : taille de la source de clients.

K_v : discipline de service.

Où K peut être :

E : service exhaustif.

L : service limité.

A : service avec barrière.

B : service de Bernoulli.

N : service n-vacances.

Et « v » peut correspondre à :

S: vacance unique.

M: vacance multiple.

H : vacance hybride

2.3.3.3. Aspects des politiques de vacances

Une politique de vacance est caractérisée par trois aspects : règle de début de vacances, règle de fin de vacances et la durée de vacances [24].

- **Règle de début de vacance :**

Connue aussi sous le nom discipline de service, cette règle détermine deux disciplines de service majeurs : service exhaustif et service non exhaustif.

- ✓ **Un service exhaustif :**

Chapitre 2 : Formalismes mathématiques.

Le serveur doit servir tous les clients dans sa file d'attente et ne peut prendre de vacance qu'une fois que tous les clients sont servis ou que la file d'attente soit vide.

✓ **Un service non exhaustif :**

Un serveur peut prendre une vacance même si la file n'est pas vide. Dans ce dernier cas on distingue deux politiques un service limité et service avec barrière.

❖ **Avec un service limité :**

Dans les systèmes de vacance à service limité, quand le serveur retourne d'une vacance il sert un nombre k de clients au maximum puis débutera une autre vacance. C'est-à-dire le serveur sert jusqu'à ce que la file d'attente soit vide ou les k clients seront servis.

❖ **Avec un serveur avec barrière :**

Dans les systèmes de vacance à service avec barrière, quand le serveur retourne des vacances, il sert seulement les clients qui étaient dans la file d'attente à son arrivée et ne reprend une autre vacance que si tous ces clients sont servis.

• **Règle de fin de vacance :**

Cette règle détermine le moment où le serveur reprend son service. Il existe deux politiques populaires, nous citons : vacances uniques et vacances multiples.

✓ **Une vacance multiple :**

Représente le fait que le serveur prend des vacances successives jusqu'à trouver au moins un client en attente à la fin d'une vacance.

✓ **Un service avec vacance unique :**

Le serveur à son retour de vacance sert les clients présents ou reste en attente si la file est libre jusqu'à servir au moins un client.

Quant à la durée de la vacance il existe des règles générales telles que :

✓ **La politique N-vacance (N-Policy) :**

Où la reprise du service se fait si le nombre de client dans la file atteint un seuil N ,

✓ **La politique T-vacance :**

Le serveur arrête sa vacance et reprend son fonctionnement quand une durée de temps T s'écoule.

• **Durée d'une vacance :**

Les vacances du serveur sont généralement considérées comme des variables aléatoires indépendantes et identiquement distribuées avec une certaine loi de probabilité, selon les caractéristiques du système modélisé.

2.4. Chaines de Markov

Les chaînes de Markov correspondent à une première approche mathématique pour modéliser des systèmes de contrôle/commande en réseau via l'étude des transitions entre différents états à partir de

Chapitre 2 : Formalismes mathématiques.

probabilités bien définies. Une chaîne de Markov repose ainsi sur une suite de variables aléatoires ($X_n, n \in \mathbb{N}$) qui permet de modéliser l'évolution dynamique d'un système aléatoire : X_n représente l'état discret du système à l'instant n . La propriété fondamentale des chaînes de Markov, dite propriété de Markov, est qu'il y a plusieurs évolutions possibles à partir d'une situation, chacune d'entre elles ayant une probabilité de se réaliser sans considérer l'histoire passée.

$$P(X_{t+1} = x_j / X_t = x_i, X_{t-1} = x_k, X_{t-2} = x_l, \dots) = P(X_{t+1} = x_j / X_t = x_i).$$

C'est cette probabilité $P(X_{t+1} = b / X_t = a)$ qui détermine la chance de passer d'un état X_t à un nouvel état X_{t+1} à l'instant suivant, on peut interpréter cette définition en disant que c'est un processus à temps discret ou le futur ne dépend que du présent, pas du passé. C'est cette incertitude qui introduit la dynamique aléatoire ou stochastique. Les applications des chaînes de Markov sont très nombreuses (réseaux, génétique des populations, mathématiques financières, gestion de stock, algorithmes stochastiques d'optimisation, simulation, . . .).

Donc une chaîne de Markov est un processus stochastique markovien $\{X_t, t \in T\}$ à espace d'état discret E ($E \subset \mathbb{N}$). Selon les valeurs que prend l'index t , une chaîne de Markov peut être à temps discret (CMTD : Chaîne de Markov à Temps Discret), ou bien à temps continu (CMTC : Chaîne de Markov à Temps Continu) [25] [26].

2.4.1. La loi exponentielle

Une variable X suit une loi exponentielle de paramètre $\lambda > 0$, dont la densité de probabilité est définie par [27] :

$$f(t) = \begin{cases} \lambda e^{-\lambda t} & \text{si } t \geq 0 \\ 0 & \text{sinon} \end{cases}$$

Sa fonction de distribution est donnée par :

$$F(t) = \int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t}, t \geq 0$$

Pour une variable qui suit la loi exponentielle sa moyenne est donné par :

$$E(x) = \frac{1}{\lambda}$$

2.4.2. Variable aléatoire

Une variable aléatoire est une application X d'un espace de probabilité Ω dans un espace d'états E qui est défini comme un ensemble de valeurs que pourrait prendre la variable aléatoire X . [27]

$$X : \Omega \rightarrow E$$

$$\omega \rightarrow X(\omega)$$

On distingue deux types de variables aléatoires :

- ✓ Lorsque $E \subset \mathbb{Z}$, X est dite une variable aléatoire discrète qui est défini par ses probabilités d'états : $p(n) = P[X=n]$, $n = -\infty, \dots, +\infty$, avec :

$$\sum_{n=-\infty}^{+\infty} p(n) = 1$$

- ✓ Lorsque $E \subset \mathbb{R}$, X est dite une variable aléatoire continue qui est définie par sa fonction densité de probabilité $f_x(x)$ pour $x \in]-\infty, +\infty [$, telle que :

$$\int_{-\infty}^{+\infty} f_x(x) dx = 1$$

2.4.3. Processus stochastique

On peut définir un processus stochastique comme étant une famille $\{X_t\}_{t \in T}$ de variables aléatoires indexées par le temps t . Les mots processus et stochastique signifient respectivement fonction et aléatoire. Alors qu'une variable aléatoire X associe à chaque $\omega \in \Omega$ une réalisation $X(\omega)$, un processus stochastique $\{X_t\}_{t \in T}$ associe à chaque ω une fonction (ou trajectoire) $\{X_t(\omega)_{t \in T}\}$:

$$T \rightarrow S ;$$

$$t \rightarrow X_t(\omega) ;$$

Où S est l'espace d'arrivée des variables aléatoires X_t . Passer des variables aléatoires aux processus stochastiques revient à passer en analyse des points aux fonctions. Lorsque l'ensemble des temps T est en plus dénombrable (par exemple $T = \mathbb{N}$), on parle de processus stochastiques à temps discret. Lorsqu'il est continu (i.e. $T = [0 ; t_0]$ ou $T = \mathbb{R}^+$), on parle de processus stochastiques à temps continu [9].

2.4.4. Chaîne de Markov à temps discret

2.4.4.1. Définition

Une chaîne de Markov à temps discret est un processus stochastique $\{X_t\}_{t \geq 0}$ satisfait les restrictions suivantes [21] :

1. Le processus est à temps discret ;
2. L'espace des états E est fini ou dénombrable ;
3. Le processus satisfait la propriété de Markov.
 - On s'intéresse ici uniquement aux CMTD à espace d'état fini (E possède N états) : $E = \{1, 2, \dots, N\}$.
 - Une chaîne de Markov est homogène si la probabilité de transition d'un état i vers un état j entre 2 étapes consécutives ($n-1$ et n) :

$$p_{ij} = P [X_n = j \mid X_{n-1} = i] \quad \forall n \in \mathbb{N}$$

$$\text{Telles que } \sum_j p_{ij} = 1 \quad \forall i \in E$$

Matrice de transition $P = [p_{ij}]_{i, j \in E}$.

Exemple :

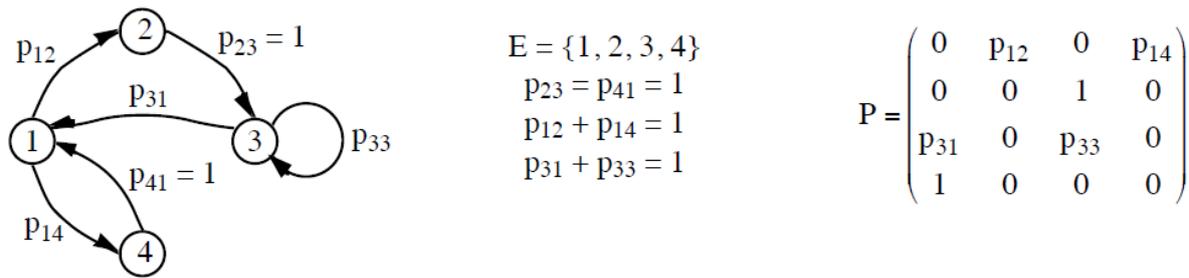


Figure 2.4: Diagramme de transition et la matrice correspondant de la CMTD.

2.4.4.2. Régime transitoire (au bout d'un nombre fini d'étapes)

Consiste à déterminer le vecteur $\pi^{(n)}$ des probabilités d'état $\pi_j^{(n)} = P [X_n = j]$, $j \in E$ pour que le processus $\{X_n\}$ $n \in \mathbb{N}$ se trouve dans l'état j à la $n^{\text{ième}}$ étape du processus :

$$\pi^{(n)} = [\pi_1^{(n)}, \pi_2^{(n)}, \pi_3^{(n)} \dots \pi_n^{(n)}]$$

- Ce vecteur dépend de :
 - La matrice des transitions P .
 - L'état initial $\pi^{(0)}$

Equation du régime transitoire : $\pi^{(n)} = \pi^{(n-1)} P$

2.4.4.3. Nature des états d'une CMTD

État accessible : j est accessible à partir d'un état i si la probabilité de transition de i en j en un certain nombre d'étapes est positive.

État communiquant : On dit que l'état j communique avec l'état i , si chacun d'eux est accessible à partir de l'autre.

État persistant (ou récurrent) : i est un état récurrent si en partant de l'état i on repassera presque sûrement par cet état.

État récurrents non nuls : Un état i est dit récurrent non nul, si la chaîne partant de i repassera par i au bout d'un temps fini.

État récurrents nuls : Un état i est dit récurrent nul, si la chaîne partant de i repassera par i au bout d'un temps infini.

État transitoire : i est un état transitoire si en partant de l'état i on risque de ne pas repasser par cet état.

État périodique : Un état est dit périodique de période k ($k > 1$), si on ne peut y revenir (après l'avoir quitté) qu'en un nombre d'étapes multiples de k .

2.4.4.4. Classification des Chaines de Markov discrètes

- **CMTD irréductible** : Une CMTD est dite irréductible, si de tout état i on peut atteindre tout état j (en un nombre fini d'étapes).
- **CMTD apériodique** : Une CMTD est dite apériodique si sa période est égale à 1 (La période d'une CMTD est définie comme le PGCD de la période de tous ses états).
- **CMTD ergodique** : On dit qu'une chaîne de Markov à temps discret est ergodique si elle est finie, irréductible et apériodique.

2.4.5. Chaines de Markov à temps continu

2.4.5.1. Définition

La chaîne de Markov à temps continu est une variante à temps continu $\{X(t) ; t \geq 0\}$ du processus de Markov. Plus précisément, c'est un modèle mathématique à valeur dans un ensemble dénombrable dans lequel le temps passé dans chacun des états est une variable aléatoire réelle positive, suivant une loi exponentielle [28].

Cet objet est utilisé pour modéliser l'évolution de certains systèmes, comme les files d'attente.

Considérons trois points consécutifs dans le temps où il y a eu changement d'états :

r ($r > 0$) temps passé.

s ($s > 0$) temps courant (actuel).

$(s + t)$ t unités de temps dans le futur.

Un processus stochastique en temps continu $\{X(t) ; t \geq 0\}$ à la propriété de Markov si :

$$P(X(s+t) = j \mid X(s) = i, X(r) = l) = P(X(s+t) = j \mid X(s) = i).$$

2.4.5.2. La représentation d'une CMTC

Les CMTC peuvent être décrites soit par un graphe de transition d'état, soit par une matrice de taux de transitions appelée générateur infinitésimal.

Les chaînes de Markov à temps continu sont représentées par leur diagramme de transition, qui est un graphe orienté, dont les sommets sont les états, et les arcs les transitions qui ont un taux de transition non nul dans Q . Autrement dit, il y aura un arc de l'état i à l'état j si $q_{(i,j)} \neq 0$. De plus on ne représente jamais de transition entre i et i et éléments de la diagonale q_{ii} sont, par définition, égaux à l'opposé de la somme des autres éléments de la ligne.

Exemple :

Un modèle où un processeur doit traiter des requêtes, qui s'accumulent dans une file d'attente M/M/1. Les requêtes arrivent suivant un processus de poisson de taux λ et le processeur les traite avec une loi exponentielle de taux μ . Le diagramme de transition et la matrice de taux de transitions est représenté par les figures (2.5 et 2.6) :

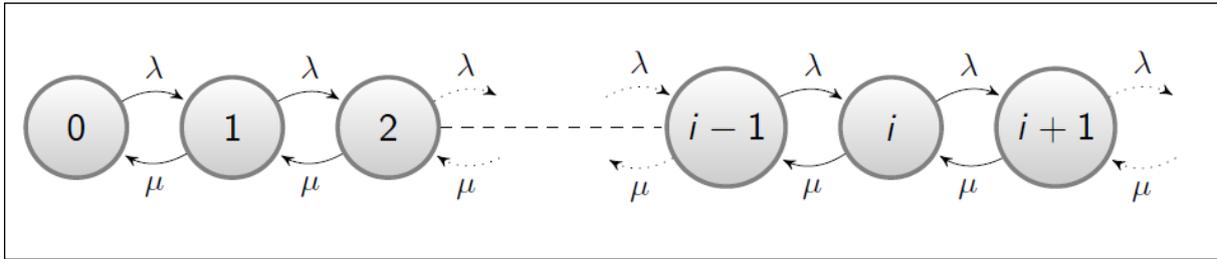


Figure 2.5: Diagramme de transition.

$$Q = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & \dots \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \end{matrix} & \begin{pmatrix} -\lambda & \lambda & & & \\ \mu & -(\lambda + \mu) & \lambda & & \\ & \mu & -(\lambda + \mu) & \lambda & \\ & & \ddots & \ddots & \ddots \end{pmatrix} \end{matrix}$$

Figure 2.6: matrice de taux de transitions (générateur infinitésimal).

2.4.5.3. Étude des chaînes de Markov à temps continu

Il existe une relation étroite entre les CMTC et les CMTD. En effet, pour chaque CMTC définie par son générateur infinitésimal Q, on peut définir une CMTD dite chaîne de Markov incluse (CMI). Cette dernière est définie par sa matrice de probabilités de transitions P, les termes de la matrice P sont obtenus comme suit :

On sait que : $q_{i,j} = q_i \cdot P_{i,j}$.

On a donc :

$$P_{i,j} = \begin{cases} q_{i,j} / q_i & \text{si } i \neq j \text{ et } q_i \neq 0, \\ 0 & \text{si } i \neq j \text{ et } q_i = 0; \end{cases} \quad \text{et } P_{i,i} = \begin{cases} 0 & \text{si } q_i \neq 0, \\ 1 & \text{si } q_i = 0. \end{cases}$$

Et que : $q_i = \sum_{j \neq i, j=0}^M q_{ij} \cdot \quad i \text{ et } j \in \{0,1,\dots,M\}$

Exemple :

On reprend l'exemple ou $E = \{a ; b ; c\}$ et Q la matrice de taux de transition (figure 2.7)

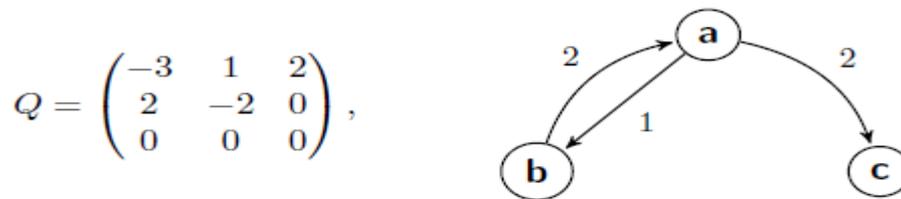


Figure 2.7: la matrice de taux de transition et son diagramme de transition correspondant.

Alors la matrice P et le diagramme de transition de la CMTD incluse correspondante sont donnés par (figure 2.8) :

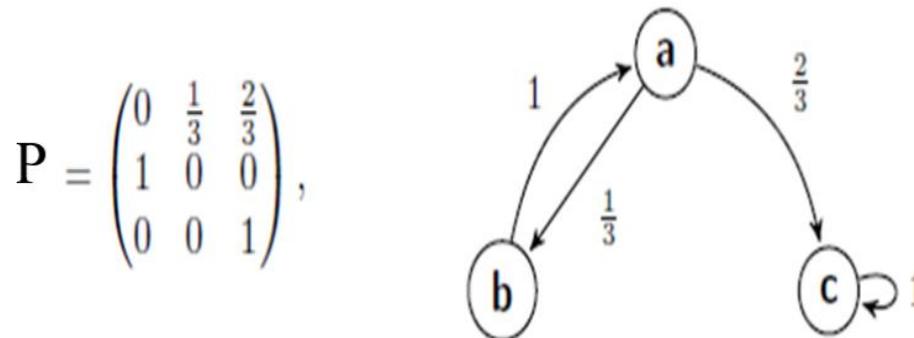


Figure 2.8: la matrice P et le diagramme de transition correspondant.

Grâce à ce résultat, l'étude des CMTC est énormément facilitée. En effet, pour garantir l'existence d'un régime stationnaire, la CMTC doit être irréductible à partir des résultats suivants :

Résultat 1 : Une CMTC est irréductible si et seulement si sa CMTD incluse est irréductible.

Résultat 2 : Une CMTC finie et irréductible est ergodique.

Résultat 3 : Par conséquent, une CMTC finie et irréductible, tend vers une distribution stable π après l'écoulement d'un temps infini. Le vecteur π est l'unique solution du système d'équations matricielles suivant [29] :

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

Où : E est l'espace des états de la chaîne de Markov.

2.5. Une méthode approximative pour résoudre les chaînes de Markov à deux dimensions

A cause des difficultés rencontrées dans le calcul de la distribution stationnaire des chaînes de Markov multidimensionnelles (la complexité, le nombre énorme des états de la CMTC, etc.). Plusieurs chercheurs ont pensé à développer une méthode numérique efficace pour résoudre ce problème, parmi eux on cite **B. CHOI et al** [11] qui ont modélisé un système de files d'attente avec rappel en appliquant une méthode

Chapitre 2 : Formalismes mathématiques.

approximative sur une chaîne de Markov à 2 dimensions pour calculer la distribution stationnaire afin d'évaluer par la suite les mesures de performance de leur modèle. Enfin, les résultats numériques montrent la précision de leur algorithme nommé « phase merging algorithm ».

Ces chercheurs ont modélisé leur système comme suit :

- La station de service contient $(n + m)$ serveurs.
- Les arrivées des clients suivent une loi de poisson avec un taux d'arrivée λ_h pour les clients de haute priorité et λ_o désigne le taux d'arrivée des clients de basse priorité.
- A l'arrivée d'un nouveau client avec un taux λ_h , il rejoint le service s'il y a au moins un serveur libre, sinon il est forcément rejeté.
- A l'arrivée d'un client non prioritaire, s'il y a au moins (n) serveurs occupés, le client sera bloqué et rejoint l'orbite avec probabilité H_o ou quitte le système avec une probabilité $(1-H_o)$. Les clients bloqués essaient de rejoindre la file d'attente après une durée exponentiellement distribué avec une moyenne $1/\eta$.
- Un client en orbite fait une tentative de rejoindre la file d'attente du système et en cas d'échec, il rejoint une autre fois l'orbite et sera retransmis avec une probabilité H_1 au quitte définitivement le système avec une probabilité $(1-H_1)$ Soit $X_1(t)$ une variable aléatoire représentant le nombre des paquets bloqués dans l'orbite, $X_2(t)$ est une variable aléatoire représentant le nombre des paquets occupant le service à l'instant t .

Alors $X_1(t), X_2(t)$ est une chaîne de Markov à deux dimensions, illustrée dans la figure 2.9 (avec $\lambda = \lambda_o + \lambda_h$) dont son diagramme de taux de transition est donné par :

$$q((i,j),(j',j')) = \begin{cases} \lambda & \text{si } i'=i, & j < n-1, & j'=j+1. \\ \lambda_h & \text{si } i'=i, & j \geq n, & j'=j+1. \\ i\eta & \text{si } i'=i-1, & j < n-1, & j'=j+1. \\ i\eta(1-H_1) & \text{si } i'=i-1, & j \geq n, & j'=j'. \\ \lambda_o H_o & \text{si } i'=i+1, & j \geq n, & j'=j'. \\ j\mu & \text{si } i'=i, & & j'=j-1. \\ 0 & \text{Autrement} & & \end{cases}$$

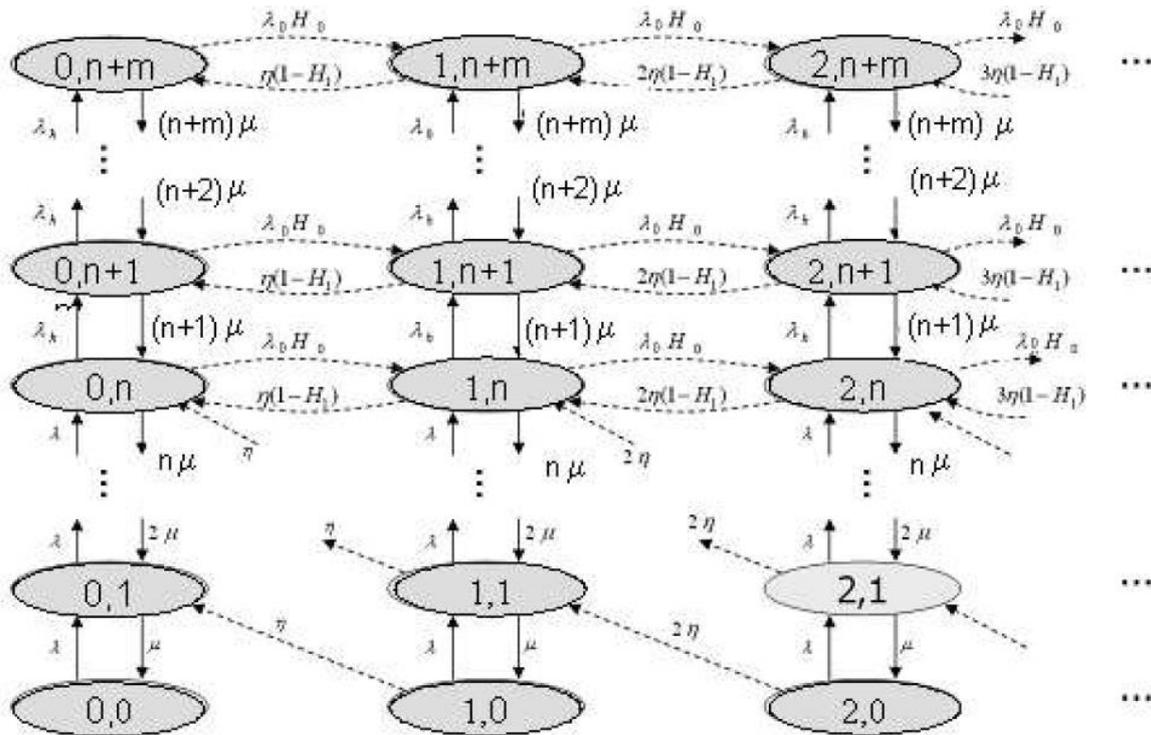


Figure 2.9: Diagramme de taux de transition pour le modèle de chaîne de Markov à 2 dimensions. [11]

2.5.1. Les étapes de l'algorithme de la méthode approximative

Soit $p(i, j)$ la probabilité stationnaire d'état (i, j) .

Etape1 :

La division de l'espace d'état de la chaîne de Markov originale à 2 dimensions comme suit :

$$S = \bigcup_{i=0}^R S_i, S_i \cap S_{i'} = \emptyset, i \neq i'$$

D'où : $S_i = \{(i, j) \in S, 0 \leq j \leq n + m\}$

Etape2 :

La distribution stationnaire conditionnelle $p(j/i)$ de chaque classe S_i est identique au système de file d'attente $M / M / m + n / m + n$ où les taux d'arrivée sont :

$$\begin{cases} \lambda_0 + \lambda_n & \text{si } j < n. \\ \lambda & \text{si } j \geq n. \end{cases}$$

Par conséquent $p(j/i), 0 \leq j \leq n + m$ est obtenu comme suit :

$$p(j/i) = \begin{cases} \frac{p^j}{j!} p(0/i) & \text{si } 1 \leq j \leq n \\ \frac{p^n}{j!} \rho_h^{j-n} \cdot p(0/i) & \text{si } n < j \leq n + m \dots (1) \end{cases}$$

Chapitre 2 : Formalismes mathématiques.

D'où $\rho = \frac{\lambda_0 + \lambda_h}{\mu}$, $\rho_h = \frac{\lambda_h}{\mu}$ et

$$p(0/i) = \left(\sum_{j=0}^n \frac{p^j}{j!} + \rho_h \sum_{j=n+1}^{n+m} \frac{p^j}{j!} \rho_h^{j-n} \right)^{-1} \dots (2)$$

D'après les formules (1), (2), on déduit que les probabilités $p(j/i)$ ne dépendent pas de i .
Par conséquent $p(j/i) \equiv p(j)$.

Étape 3 :

Tous les états du sous-ensemble S_i se combinent en un seul état fusionné $\langle i \rangle$ (voir la figure 2.10).
Le modèle fusionné est un processus de naissance et de mort et ses taux de transition sont donnés par :

$$q(\langle i \rangle, \langle i' \rangle) = \begin{cases} \lambda_0 H_0 \sum_{j=0}^{n+m} p(j) & \text{si } i > 0, i' = i + 1. \\ i \eta \left(\sum_{j=0}^{n-1} p(j) + (1 - H_1) \sum_{j=n}^{n+m} p(j) \right) & \text{si } i > 1, i' = i - 1. \\ 0 & \text{autrement} \end{cases}$$

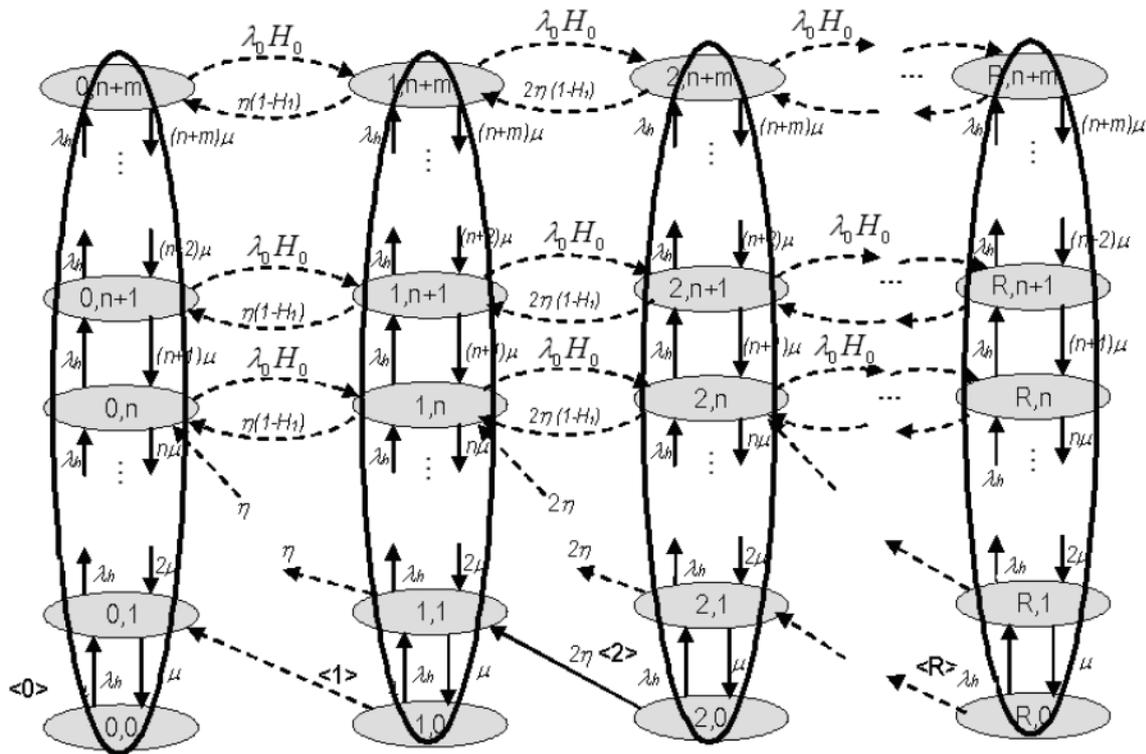


Figure 2.10: Diagramme des états fusionnés du modèle de chaîne de Markov. [11]

Ainsi, la distribution stationnaire du modèle fusionné $\pi(\langle i \rangle)$ ou $\langle i \rangle \in S$.

$$\pi(\langle i \rangle) = \frac{t^i}{i!} \pi(\langle 0 \rangle)$$

D'où :

$$\pi(\langle 0 \rangle) = \left(\sum_{i=0}^R \frac{t^i}{i!} \right)^{-1}$$

$$t = \frac{\lambda_0 H_0 \sum_{j=0}^{n+m} p(j)}{\eta \left(\sum_{j=0}^{n-1} p(j) + (1-H_1) \sum_{j=n}^{n+m} p(j) \right)}$$

Etape4 :

Enfin l'obtention d'une approximation de $p(i, j)$ comme suit :

$$P(j/i) \cong p(j) \cdot \pi(\langle i \rangle).$$

2.6. Une méthode approximative pour résoudre les chaînes de Markov multidimensionnelles

L'étude des chaînes de Markov multidimensionnelles avec la méthode exacte est très compliquée, car le nombre d'états tend vers l'infini. A cause de ces contraintes **A.Z. Melikov et al** [12] ont proposé une méthode approximative nommée « Hierarchical space merging algorithm », pour le calcul de la distribution à l'état stationnaire à partir d'une chaîne de Markov à 3 dimensions. La figure 2.11 ci-dessous montre la structure générale de réseau étudié, ce dernier contient deux nœuds est chacun entre eux est modélisé par une file d'attente avec capacité finie et un seul serveur, le taux de service suit une distribution exponentielle $1/\mu_i$, avec $i=1, 2$, $\mu_1 \neq \mu_2$.

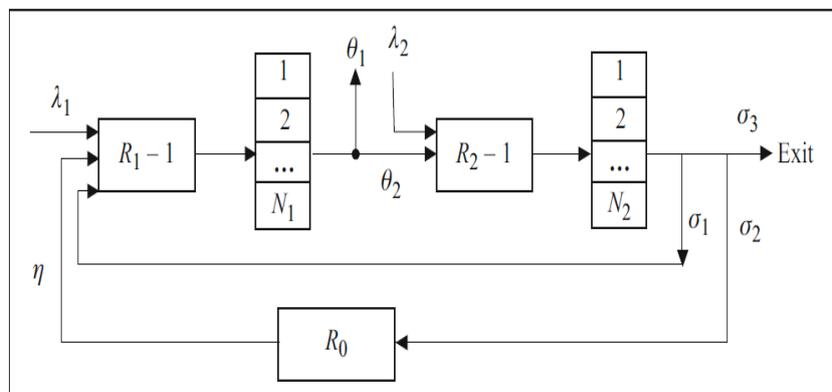


Figure 2.11: structure du réseau à base de file d'attente. [12]

Les caractéristiques du modèle étudié sont comme suit :

- Les flux arrivent aux deux nœuds de l'extérieur suivent la loi de poisson. Une partie des appels déjà traités au premier nœud entrent instantanément au deuxième nœud (s'il y a de l'espace libre ici) pendant que les autres appels quittent le réseau. Une fois le service termine sur le deuxième nœud, trois possibilités s'offrent :
 - (i) l'appel quitte le réseau.
 - (ii) il renvoie instantanément au premier nœud (s'il y a de l'espace libre).
 - (iii) il sera retransmis au premier nœud après le passage à l'orbite.
- La capacité totale du nœud i (c'est-à-dire le nombre total d'appels qu'il contient, y compris les appels dans le canal) est R_i , avec $i=1, 2$, si on atteint la capacité du nœud alors la probabilité de perdre du paquet sera égale à 1.
- Si le premier nœud serve un paquet, ce dernier abandonne définitivement le réseau avec une probabilité de θ_1 . Ou arrive au deuxième nœud avec probabilité $\theta_2 = 1 - \theta_1$.

Chapitre 2 : Formalismes mathématiques.

- Lorsque le service d'un paquet spécifique est terminé sur le deuxième nœud, l'une des décisions suivantes peuvent être prises : retourne instantanément au premier nœud avec probabilité σ_1 , Arrive à l'orbite avec une probabilité σ_2 ou ce paquet sera rejeté et abandonne le réseau avec probabilité $\sigma_3 = 1 - \sigma_2 + \sigma_1$.
- La taille de l'orbite est R_0 , cela signifie que l'appel arrivé du deuxième nœud est amené sur l'orbite si le nombre des paquets est actuellement inférieur à R_0 , sinon il est perdu avec une probabilité 1.
- Le temps de séjour des appels sur l'orbite sont indépendantes distribuées de manière exponentielle avec la valeur moyenne $1/\eta$. Pour être précis, nous supposons que les appels de l'orbite ne sont pas persistants, c'est-à-dire si, au moment de l'arrivée d'un appel depuis l'orbite, il n'y a pas de place sur le premier nœud, l'appel arrivé abandonne finalement le réseau.
- Le système étudié est modélisé par une chaîne de Markov à trois dimensions (3D) (voir figure 2.12), cette dernière est représentée par le vecteur $n = (n_1, n_2, n_3)$, d'où (n_1) et (n_2) représente le nombre des paquets dans le premier et le deuxième nœud respectivement, (n_3) indique le nombre des paquets dans l'orbite. L'espace d'états étudié est défini comme suit :

$$S = \{n : n_1 = 0, \dots, R_1, n_2 = 0, \dots, R_2, n_3 = 0, \dots, R_0\}.$$

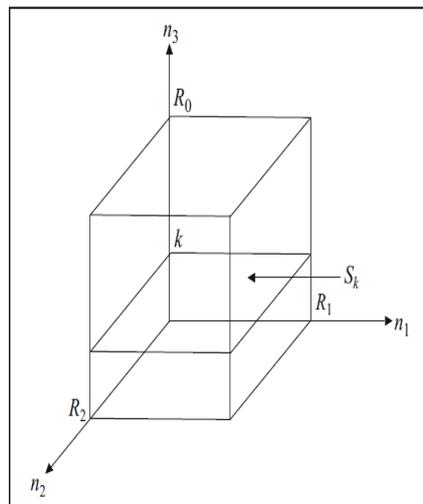


Figure 2.12: L'espace d'états du modèle [12].

- Le taux de transition de l'état n à l'état n' est désigné par $q(n, n')$, telle que $n, n' \in S$. Ces quantités constituent la matrice de taux de transition (Q) de la chaîne de Markov à 3D, définie par les relations suivantes :

$$q(n, n') = \begin{cases} \lambda_1 & \text{si } n' = n + e_1. \\ \mu_{101} & \text{si } n' = n - e_1. \\ \mu_{102} & \text{si } n' = n - e_1 + e_2. \\ \lambda_2 & \text{si } n' = n + e_2. \\ \mu_{2\sigma_3} & \text{si } n' = n - e_2. \\ \mu_{2\sigma_1} & \text{si } n' = n - e_2 + e_1. \\ \mu_{2\sigma_2} & \text{si } n' = n - e_2 + e_3. \end{cases} \quad (*)$$

2.6.1. Les étapes de l'algorithme de la méthode approximative

Soit $p(n)$ la probabilité de l'état stationnaire $n \in S$ et $\rho^K(n_1, n_2)$ la probabilité de l'état (n_1, n_2) à l'intérieur dans l'espace d'état S_k et $\pi_1(\langle K \rangle)$ la probabilité d'état fusionné.

Pour faciliter la résolution de la chaîne de Markov à 3 dimensions, les chercheurs ont proposé la division de l'espace d'état comme suit : une chaîne de Markov à 2 dimension $n(n_1, n_2)$ avec l'espace d'état S_K , $K=0, \dots, R_0$ et autre unidimensionnelle $n(n_3)$ avec l'état fusionné $K=0, \dots, R_0$.

La méthode proposée permet de trouver des valeurs approximatives des probabilités d'états $\rho^K(n_1, n_2)$ sous charges asymétriques d'appels sur différents nœuds, c'est-à-dire que nous distinguerons deux conditions :

(i) $\lambda_1 \gg \lambda_2$, $\mu_1 \gg \mu_2$.

(ii) $\lambda_1 \ll \lambda_2$, $\mu_1 \ll \mu_2$.

Partiel :

Si $\lambda_1 \gg \lambda_2$, $\mu_1 \gg \mu_2$

Etape1 :

Si $\lambda_1 \gg \lambda_2$, $\mu_1 \gg \mu_2$ ils ont divisé la chaîne de 2 dimensions par colonne comme suit :

$$S_K = \bigcup_{i=0}^{R_2} S_K^i, S_K^i \cap S_K^j = \emptyset, \text{ si } i \neq j.$$

Ou : $S_K^i = \{n \in S_K, n_2 = i\}, i = 0, \dots, R_2$.

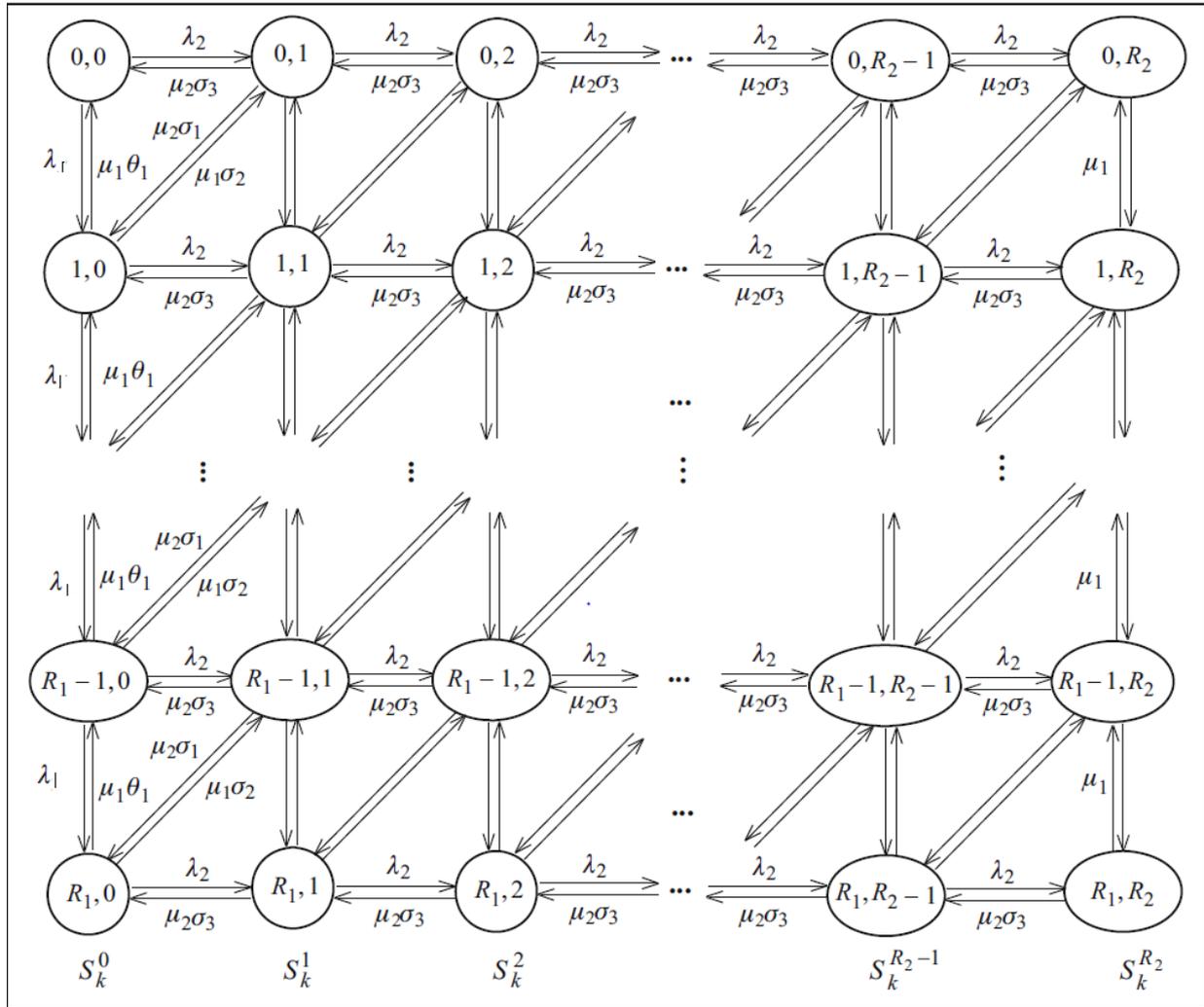


Figure 2.13: modèle divisé avec la phase d'espace SK, $K=0, \dots, R_0$ [12].

Etape2 :

Pour la décomposition la fonction d'intégration suivante est construite :

$$U_2(n) = \langle i \rangle \text{ if } n \in S_k^i \quad (**)$$

Où i désigne l'ensemble de tous les états fusionnés obtenu en (**).

Selon la théorie HSMA, nous obtenons :

$$\rho^K(n_1, n_2) \approx \rho_{n_2}^k(n_1) \pi_2^K(\langle n_2 \rangle)$$

Où $\rho_{n_2}^k$ la probabilité de l'état (n_1, n_2) à l'intérieur du modèle divisé avec l'espace d'état $S_k^{n_2}$ et π_2^K est la probabilité d'état fusionné.

Les valeurs de $n_2=0, \dots, R_2-1$.

On trouve aussi que les probabilités $\rho_{n_2}(n_1)$ pour les valeurs de n_2 coïncide avec les probabilités d'états de $M(\lambda_1) / M(\mu_1 \theta_1) / 1/R_1$ et pour $n_2=R_2$ ces probabilités coïncident avec les probabilités d'états de $M(\lambda_1) / M(\mu_1) / 1/R_1$. Par conséquent, les probabilités requises des états $\rho_i(j), j = 0, \dots, R_1$ peuvent être calculés comme suit :

Chapitre 2 : Formalismes mathématiques.

$$\rho_i(j) = \begin{cases} \frac{1 - v_1}{1 - v_1^{1+R_1}} v_1^j & \text{si } i = 0, \dots, R_2 - 1. \\ \frac{1 - v_2}{1 - v_2^{R_1+1}} v_2^j & \text{si } i = R_2. \end{cases}$$

Ou $v_1 = \lambda_1 / \mu_1$ et $v_2 = \lambda_1 / \mu_1$.

Remarque :

Lors du calcul des probabilités d'états du modèle fractionné avec l'espace d'états $S_{R_0}^i$ on devrait prendre en compte que $\sigma_2 = 0$.

Après certaines transformations algébriques, nous trouvons que les intensités de transition entre états fusionnés $\langle i \rangle \langle j \rangle$. Peut être défini comme suit :

$$q(\langle i \rangle, \langle j \rangle) = \begin{cases} \lambda_2 + \mu_1 \theta_2 ((1 - \rho_i(0))) & \text{si } i = 0, \dots, R_2 - 1. \\ \mu_2 (\sigma_3 + \sigma_1 ((1 - \rho_i(R_1))) & \text{si } i = 1, \dots, R_2, j = i - 1. \quad (***) \end{cases}$$

Par conséquent, à partir des relations (***), ils ont obtenu la formule pour calculer les probabilités des états fusionnés ($\pi_2 \langle n_2 \rangle$).

$$(\pi_2 \langle n_2 \rangle) = \prod_{i=0}^{n_2-1} \frac{q(\langle i \rangle, \langle i+1 \rangle)}{q(\langle i+1 \rangle, \langle i \rangle)} \quad (\pi_2 \langle 0 \rangle), n_2 = 1, \dots, R_2.$$

Ou ($\pi_2 \langle 0 \rangle$) peut être obtenu avec la normalisation c'est-à-dire $\sum_{n_2=0}^{R_2} (\pi_2 \langle n_2 \rangle) = 1$.

Etape 3 :

A partir de (*), (**) et (***) après certaines transformations algébriques ils ont constaté que les intensités des transitions entre les classes S_K et $S_{K'}$, peuvent être trouvées comme suit :

$$q(S_K, S_{K'}) = \begin{cases} \Lambda_1 & \text{si } K' = K + 1 \\ K\eta & \text{si } K' = K - 1. \\ 0 & \text{Autres} \end{cases} \quad (+)$$

Avec $\Lambda_1 = \mu_2 \sigma_2 (1 - (\pi_2 \langle 0 \rangle))$.

Etape 4 :

A partir de (+) ils ont obtenu la probabilité de l'état fusionné $\pi_1(\langle K \rangle)$ peut être trouvé à partir des formules classiques d'Erlang pour M (Λ_1)/M(η)/R₀/0 c'est-à-dire :

Chapitre 2 : Formalismes mathématiques.

$$\pi_1 (<K>) = \frac{\varphi^K}{K!} \left(\sum_{i=0}^{R0} \frac{\varphi^i}{i!} \right), K=1, \dots, R0.$$

Avec : $\varphi = \Lambda 1/\eta$.

Etape 5 :

Au final ils ont obtenu les probabilités d'état stationnaire de la chaîne de Markov 3D comme suit :

$$p(n_1, n_2, n_3) = \rho_{n_2}(n_1) \pi_2 (<n_2>) \pi_1 (<n_3>).$$

Partie 2 :

1. Pour la condition (ii) c'est similaire quand $\lambda_1 \gg \lambda_2$, $\mu_1 \gg \mu_2$ par contre ils ont divisé la chaîne par lignes et voilà les formules qui se change.

$$S_K = \bigcup_{i=0}^{R1} S_K^i, S_K^i \cap S_K^j = \emptyset, \text{ si } i \neq j.$$

Ou : $S_K^i = \{n \in S_K, n_1 = i\}, i = 0, \dots, R1$.

$$2. q(<i>, <j>) = \begin{cases} \lambda_1 + \mu_2 \sigma_3 (1 - \rho(0)) & \text{si } j = i + 1. \\ \mu_1 (\rho(R2) + \theta(1 - \rho(R2))) & \text{si } j = j - 1. \\ 0 & \text{autres} \end{cases}$$

$$3. q(S_K, S_{K'}) = \begin{cases} \Lambda 2 & \text{si } K' = K + 1 \\ K\eta & \text{si } K' = K - 1. \\ 0 & \text{autres} \end{cases}$$

Avec $\Lambda 2 = \mu_2 \sigma_2 (1 - \rho(0))$.

2.7. Conclusion

Au cours de ce chapitre, nous avons présenté des notions mathématiques nécessaires à la compréhension des modèles de files d'attente. Nous avons abordé la théorie des files d'attente classiques, et nous avons vu que les modèles de files d'attente standards ne permettent pas de décrire le comportement réel des clients et des serveurs, d'où vient la nécessité de d'utilisation des files d'attente avec vacance, avec priorité et rappel pour la modélisation des systèmes d'attente. Nous avons vu que le phénomène de vacance, les classes de priorité et le rappel rend l'analyse des files d'attente complexe. Ainsi les résultats analytiques n'existent que pour quelques modèles et avec des hypothèses contraignantes sur certains paramètres, et par conséquent, les chercheurs dans ce domaine se sont orientés vers des méthodes numériques, des approximations ou des simulations. La méthode approximative Phase Merging Algorithm pour le calcul des chaînes de Markov multidimensionnel a été aussi abordé dans de ce chapitre.

Dans le chapitre suivant, nous allons proposer dans un premier temps une modélisation de capteurs sans fil avec les files d'attente avec vacance, priorité et rappel, ainsi qu'une approche d'analyse des performances. Dans un second temps, nous allons étudier le même système en utilisant la méthode approximative Phase Merging Algorithm.

3. Modélisation des réseaux de capteurs.

3.1. Introduction

Les réseaux de capteurs sans fil évoluent sans cesse et continuent à gagner leur place dans plusieurs domaines, aussi bien militaires que civils. Nous avons vu dans le premier chapitre qu'ils sont des réseaux très contraints en ressources. Leur portée du radio est très restreinte, ce qui exige une communication multi-sauts au lieu d'envoyer l'information directement au sink. De plus, dans la majorité des RCSFs, les nœuds capteurs opèrent avec des batteries irremplaçables. De ce fait, leurs sources d'énergie sont très limitées, et la durée de vie devient très courte à cause de l'épuisement d'énergie des nœuds capteurs. Une autre limitation concerne les buffers des nœuds capteurs. Chacun d'entre eux dispose d'un buffer de petite capacité, et par conséquent, des messages reçus par ce nœud risque d'être perdus à cause de l'overflow de ce buffer.

Vue l'ampleur des applications émergentes de ces réseaux, l'évaluation de leurs performances est nécessaire. Une approche très importante pour l'analyse et l'évaluation de ces réseaux est d'utiliser des techniques de modélisation formelles. Ces dernières ont l'avantage de fournir des résultats exacts. L'application du modèle de file d'attente avec vacance pour la conservation d'énergie des réseaux de capteur sans fil a fait l'objet de plusieurs travaux. Cependant, la plupart des travaux ont proposés pour un nœud capteur des modèles avec buffer infinie [4], [10]. C'est la raison pour laquelle nous nous intéressons à la modélisation d'un nœud capteur par une file d'attentes avec buffer finie.

Afin d'optimiser la consommation d'énergie dû à la fréquence de transition entre l'état oisif (idle) et l'état occupé des nœuds capteur, nous proposons trois modèles pour un nœud capteur basés sur la théorie des files d'attente avec vacance des serveurs, classes de priorité et rappel. Un premier modèle est basé sur l'utilisation de la politique N-vacance (N-policy) avec service exhaustif. Dans le deuxième modèle, afin de minimiser le délai d'attente des paquets prioritaire nous avons considéré des flux de différentes priorité. Dans le dernier modèle, nous avons introduit les rappels pour remédier au problème de perte de messages.

Dans la section suivante, nous présentons les travaux qui concernent l'utilisation des méthodes formels pour l'évaluation des performances des RCSFs. Nous présentons par la suite, une description détaillée des différents modèles proposés, la solution exacte et la solution approximative. Nous terminerons notre chapitre par une conclusion.

3.2. Les Modèles formels et les réseaux de capteurs sans fil

Vue que la simulation a l'inconvénient de ne pas pouvoir trouver une relation entre un ensemble de données en entrée et le résultat en sortie, l'application des modèles mathématiques formels s'avère indispensable lors de la conception des réseaux de capteurs. Ce type de

Chapitre 3 : Modélisation des réseaux de capteurs.

modélisation vise à avoir des résultats exacts en appliquant des formules analytiques ce qui rend le modèle facilement analysable ou bien en appliquant des méthodes numériques ou algorithmiques pour des modèles plus complexes.

Ces analyses peuvent s'intéresser au cas moyen comme le font les simulations mais elles permettent aussi d'exhiber le pire cas du modèle par rapport à la propriété à valider. Ces modèles permettent ainsi une meilleure compréhension du système. Dans ce qui suit, nous avons abordé des travaux dans ce domaine, appliquant des modèles de files d'attente avec vacance ou des réseaux de Petri.

Maheswar et Jayaparvathy [4] ont utilisé la théorie des files d'attente standards M/G/1 avec une politique de retour des vacances de type N-vacance pour optimiser la consommation d'énergie. Dans leur modèle, ils ont considéré des nœuds capteurs avec des buffers à capacité illimitée, et un schéma de communication plusieurs-a-un, dans lequel un groupe de nœuds de type CM, communiquent seulement avec un seul nœud spécifique qui est le CH, avec un seul saut. Les nœuds CM peuvent uniquement capté et envoient les données au nœud CH. Les CH captent et collectent les données des CMs et les transmettent à la station de base du RCSF. Durant le mode active, les CM et les CH peuvent être dans l'état oisif ou occupé. Le nœud se trouve initialement dans l'état oisif et reste dans cet état jusqu'à ce que le nombre de paquet en attente atteint le seuil N (la politique N-vacance). Le nœud passe de l'état occupé à l'état oisif lorsque le buffer devient vide. Les auteurs ont analysé les performances du système en fonction des paramètres de performance qui sont l'énergie consommé et le délai moyen d'attente.

Dae-Eun Lim et al [5] ont modélisé leur système par une file d'attente rarement étudié de type Geo/ G/1 avec N-vacance à temps discret cette politique est très utile pour atténuer le nombre de changements entre les états oisif et occupés dans le but d'optimiser la consommation d'énergie. Cette étude permet d'obtenir les fonctions des probabilités de la longueur de la file et le temps de séjour. Ils ont appliqué leurs résultats sur un nœud de capteur dans les RCSFs dans le cas d'une connexion non fiable soumise à la perte des paquets causés par des catastrophes. Plusieurs expérimentations numériques ont mené à prolonger la durée d'un nœud de capteur.

Dans un RCSF les nœuds de capteurs proches de sink consomment plus d'énergie due à la communication (many to one) qui causent le problème d'EHP (Energy Hole problem). **Der-Chen et al** [6] ont proposé un nouveau schéma de conservation d'énergie pour résoudre EHP, la modélisation de leur système était par une file d'attente M / M / 1 / k avec la politique N-vacance, capacité du buffer finie (k), l'arrivée des paquets suit la loi de poisson de paramétré λ , le service suit une distribution exponentielle avec un taux μ . Ils ont résolu un système Markovien à partir de la matrice de taux de transitions Q afin de déduire la formule de l'énergie consommée. Les auteurs ont validé l'efficacité de leur étude par des formules mathématiques ainsi la simulation valide que l'approche proposée est rentable en termes d'optimisation de la consommation d'énergie.

B. Boutoumi et N. Gharbi [7] ont opté pour la modélisation d'un nœud capteur par une file d'attentes avec buffer et source finie et différentes politiques de vacances, et ce en se basant sur le modèle des réseaux de Pétri Stochastiques Généralisés. Ils ont considéré un capteur sans fil qui alterne entre deux principaux modes : sommeil et actif. Lorsqu'il est en

Chapitre 3 : Modélisation des réseaux de capteurs.

mode sommeil, le capteur ne peut pas interagir avec son environnement par conséquent, l'énergie consommé est très basse. Durant le mode actif, le capteur écoute le canal, génère des paquets, reçoit et transmet les paquets de données de ces voisins donc l'énergie consommé est considérable. Ainsi, pendant la période actif le capteur transite entre l'état oisif ou il peut générer et recevoir des paquets et l'état occupé où il peut générer, recevoir et émettre. Il est à noter également que l'énergie dissipée pour la transition entre l'état oisif et l'état occupé, génère une surconsommation d'énergie importante nommé l'énergie de transition (the switching energy).

Afin d'optimiser la consommation d'énergie due à la fréquence de transition entre oisif et l'état occupé des nœuds capteur, ils ont proposé un premier modèle basé sur l'utilisation de la politique N-vacance avec service exhaustif. Pour montrer l'impact de cette politique sur la conservation de l'énergie dans les réseaux de capteurs sans fil, une étude détaillée est argumentée par les résultats numérique a été établie.

La politique N-vacance, qui représente une approche de conception efficace pour diminuer la consommation d'énergie du nœud capteur, introduit un délai d'attente supplémentaires pour les paquets de données en attente dans le cas où le taux d'arriver est petit. Pour remédier à cet inconvénient, dans le deuxième modèle ils ont considéré une règle de fin de vacance hybride. Ils ont optés pour une fusion entre la politique N-vacance et une vacance d'une durée aléatoire dont le but est de réduire le délai d'attente des paquets dans le buffer avant d'être envoyé. Dans ce modèle, un capteur en sommeil peut devenir actif et reprendre son service de deux manières différentes. La première par l'écoulement d'un certain temps aléatoire tandis que la deuxième sera un réveil immédiat si N paquets sont en attente dans son buffer. Le modèle hybride est particulièrement intéressant dans les réseaux de capteur sans fil ou le taux d'arriver est variable. Ceci permet d'éviter la saturation rapide du buffer et la dissipation de l'énergie due à la retransmission des paquets perdus dans le cas où le taux d'arriver est élevé. De plus dans le cas où le taux des arrives est petit, la vacance aléatoire permet de réduire le délai d'attentes des paquets.

Afin d'optimiser la consommation d'énergie dû à la fréquence de transition entre oisif et l'état occupé des nœuds capteur **Boutoumi et Gharbi** [8] ont proposé un premier modèle basé sur l'utilisation de la politique N-vacance avec service exhaustif. Pour montrer l'impact de cette politique sur la conservation de l'énergie dans les réseaux de capteurs sans fil, une étude détaillée et argumenté par les résultats numériques a été établie. Dans le deuxième modèle, afin de minimiser le délai d'attente des paquets, ils ont opté d'introduire la notion de vacances de travail dans le comportement des capteurs. Les auteurs ont modélisé un seul capteur en full-duplex durant sa période actif. Le capteur a été modélisé par une file d'attente M/M/1/K avec la politique "two thresholds working vacation policy" qu'ils ont proposés. Dans ce modèle, un capteur en sommeil peut devenir semi-busy si le nombre de paquets en attente atteint le seuil N2 et commence le service avec le taux de vacance de travail μ_2 . Dès que le nombre de paquets en attente atteint le seuil N1 ($N_2 < N_1$) le nœud passe à l'état occupé et commence le service avec le taux normal μ_1 ($\mu_2 < \mu_1$). Les Réseaux de Pétri Stochastique Généralisé ont été utilisé comme outil de modélisation et d'évaluation des performances.

Chapitre 3 : Modélisation des réseaux de capteurs.

Pour prolonger la durée de vie d'un nœud de capteur sans fil il existe une technique très importante qui est le schéma de minimisation d'énergie (EMS) avec N-vacance mais malheureusement elle augmente le délai moyen. Pour cette raison **P. Jayarajan et al** [9] ont proposé un schéma de minimisation d'énergie modifié (MEMS) pour but de réduire le délai moyen et minimiser l'énergie consommée d'un nœud de capteur. Ils ont considéré un RCSF hétérogène à base de cluster, avec des nœuds capteurs de haute priorité et autres de basse priorité.

La modélisation de ce système a été faite en utilisant les files d'attente avec priorité M/D/1 ou l'arrivée des paquets suit un processus de Poisson de taux d'arrivée moyen ($\lambda = \lambda_1 + \lambda_2$) ou λ_1 et λ_2 sont les taux d'arrivées des paquets de haute priorité et de basse priorité respectivement, le service avec le taux ($1 / \mu$). Les auteurs ont validé la précision de leur approche par simulation afin de trouver des résultats similaires aux résultats analytique.

P. Jayarajan et al [10] ont encore proposé une autre technique de minimisation de l'énergie basée sur la file d'attente M / M / 1, avec la politique N-vacance modèle d'un nœud de capteur, la capacité du buffer est infinie. Ils ont considéré un RCSF homogène à base de cluster, un nœud de capteur peut agir comme un chef de cluster (CH) ou un membre (CM). Dans chaque cluster le nœud de capteur ayant un niveau d'énergie le plus élevé est considéré comme (CH). Un capteur peut être à l'état oisif ou occupé, pendant l'état occupé, les capteurs délivrent les paquets de CM au nœud CH, de son tour un CH fait le relaye jusqu'à atteindre la station de base. L'arrivée des paquets suit la loi de Poisson avec un taux d'arrivée moyen λ_1 et λ_2 pour un capteur de types CH et CM respectivement. Lorsque le nœud de capteur agit comme un CM, il génère des données captées et il ne fait que le relai, contrairement à un CH qui capte et fait le relaye des paquets reçus à partir des voisins. Ils ont supposé que le taux d'arrivé $\lambda_2 \gg \lambda_1$ et les paquets sont transmet avec un temps de service moyen ($1 / c$). A la fin de cet article une étude expérimentale est faite pour étudier l'influence des paramètres sur la consommation d'énergie.

Ce tableau résume les travaux connexes cité auparavant :

Les références	Formalisme mathématique	Paramètre analysé	Teste et validation.
[4]	Les files d'attente standards M/G/1 avec la politique de N-vacance.	La consommation d'énergie et le délai moyen d'attente des nœuds capteurs.	La méthode analytique et simulation.
[5]	Les files d'attente Geo/G/1 avec N-vacance.	La consommation d'énergie.	La méthode analytique.
[6]	les files d'attente M / M / 1 / k avec la politique N-vacance,	La consommation d'énergie.	La méthode analytique et simulation.

Chapitre 3 : Modélisation des réseaux de capteurs.

	et les chaînes de Markov.		
[7]	Les files d'attente avec vacance et le modèle des réseaux de Pétri Stochastiques Généralisés.	La consommation d'énergie et le délai d'attente.	La méthode analytique.
[8]	File d'attente M/M/1/K avec la politique "two thresholds working vacation policy" et le modèle des réseaux de Pétri Stochastiques Généralisés.	La consommation d'énergie et le délai d'attente.	La méthode analytique.
[9]	Les files d'attente avec priorité M/D/1	La consommation d'énergie et le délai moyen.	La Méthode analytique et par simulation.
[10]	Les files d'attente M/M1 avec N-vacance.	La consommation d'énergie et le délai moyen.	La Méthode analytique et par simulation.

Tableau 3-1: Tableau qui résume les travaux connexes.

3.3. Description et analyse de la modélisation d'un nœud capteur

Afin de conserver l'énergie dissipée par un nœud capteur durant la période active dû à la fréquence de transition entre l'état oisif et l'état occupé, nous avons proposé de modéliser un nœud capteur par une file d'attente à un seul serveur avec vacance et buffer finie. L'état oisif représente la période de vacance du serveur et l'état occupé représente la période de service. Nous avons opté pour la politique N-vacance (N-policy) pour le passage du serveur de l'état oisif à l'état occupé. Initialement, le capteur est dans l'état oisif, il peut uniquement recevoir des paquets. Dès que le nombre de paquet en attente dans le buffer atteint le seuil N, le capteur passe à l'état occupé, ou il peut en plus transmettre les paquets en attente. La politique N-vacance (illustré par la figure 3.1) est connue comme une technique efficace pour optimiser la consommation de l'énergie dans les RCSFs. Dans la suite de cette section, nous allons présenter la modélisation correspondante à un nœud capteur.

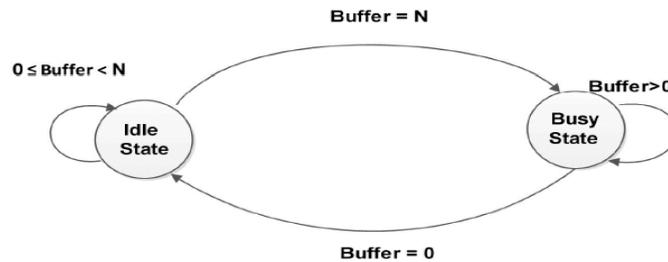


Figure 3.1: Diagramme d'état transition de la politique N-vacance.

3.3.1. Le Modèle avec N-vacance (modèle M0)

Comme indiqué dans le paragraphe précédent, nous avons opté de modéliser un nœud capteur par une file d'attente à un seul serveur avec vacance. Le buffer est fini de capacité K . Les paquets arrivent suivant le processus de poisson avec paramètre λ et le service est exponentiel avec le taux μ . Le capteur possède deux états de fonctionnement : oisif et occupé. Le passage de l'état oisif à l'état occupé est contrôlé par la politique N-vacance illustré par figure 3.1 ci-dessus qui montre le comportement du modèle proposé pour le capteur.

3.3.1.1. La CMTC du modèle de nœud capteur avec N-vacance

Considérant le modèle proposé pour un nœud de capteur avec vacance et on se basant sur le fait qu'un nœud de capteur peut alterner uniquement entre les états oisif et occupée, ce système de file d'attente avec vacance peut être modélisé par une CMTC à deux dimensions ou chaque état est décrit par deux variables aléatoires $(X(t), Y(t) : t \geq 0)$. Tel que $X(t)$ représente le nombre de paquet au niveau de buffer et $Y(t)$ représente l'état de système.

Où :

$Y(t)=0$ le serveur est à l'état oisif

$Y(t)=1$ le serveur est à l'état occupée

L'espace d'état de la CMTC est donnée par :

$$\Omega = \{(i, j) : 0 \leq i \leq K, 0 \leq j \leq 1\}$$

Le diagramme d'état transition correspondant est présenté dans la figure (3.2) et les taux de transition entre les états adjacents présenté dans le tableau suivant (3.1) :

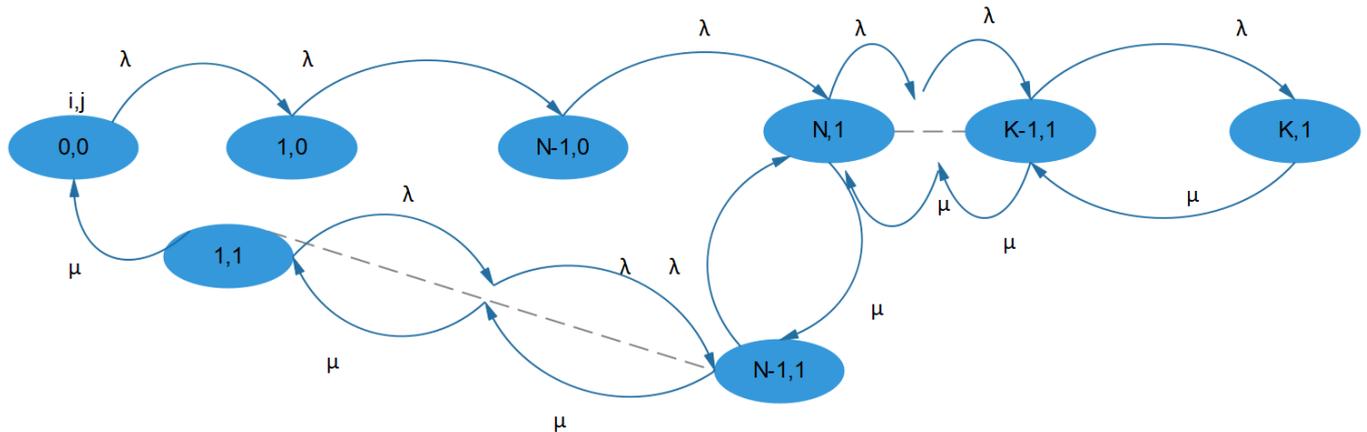


Figure 3.2: La chaîne de Markov du modèle M0.

Description	Etat sortant	Taux de transition	Etat entrant
$0 \leq i < N, j = 0$	(i, j)	λ	$(i+1, j)$
$N \leq i < K, j=1$	(i, j)	λ	$(i+1, j)$
$N < i \leq k, j=1$	(i, j)	μ	$(i-1, j)$
$0 < i < N, j = 1$	(i, j)	μ	$(i-1, j)$
$0 < i < N, j = 1$	(i, j)	λ	$(i+1, j)$

Tableau 3-2: Représentation des taux de transitions entre les états.

Par conséquent, le générateur infinitésimal Q est une matrice M^*M ($M=K+N$) et peut-être calculé par l'algorithme (Algorithm I) :

Algorithme I : l'algorithme de construction du générateur infinitésimal du M0 :

```

for  $j \leftarrow 0, 1$  do
  if  $j = 0$  then
    for  $i \leftarrow 0, N - 1$  do
       $Q[(i, j), (i + 1, j)] \leftarrow \lambda$ 
    end for
  else  $\triangleright j = 1$ 
     $i \leftarrow N$ 
     $Q[(i, j), (i + 1, j)] \leftarrow \lambda$ 
     $Q[(i, j), (i - 1, j)] \leftarrow \mu$ 
    for  $i \leftarrow N + 1, K - 1$  do
       $Q[(i, j), (i + 1, j)] \leftarrow \lambda$ 
       $Q[(i, j), (i - 1, j)] \leftarrow \mu$ 
    end for
     $i \leftarrow K$ 
     $Q[(i, j), (i - 1, j)] \leftarrow \mu$ 
    for  $i \leftarrow 2, N - 1$  do
       $Q[(i, j), (i + 1, j)] \leftarrow \lambda$ 
       $Q[(i + 1, j), (i, j)] \leftarrow \mu$ 
    end for
     $i \leftarrow 1$ 
     $Q[(i, j), (i + 1, j)] \leftarrow \lambda$ 
     $Q[(i + 1, j), (i, j)] \leftarrow \mu$ 
     $Q[(i, j), (i - 1, j - 1)] \leftarrow \mu$ 
  end if
end for

```

Comme la CMTC proposé est ergodique, l'espace d'état est fini et irréductible, le vecteur des probabilités $\pi = \pi_1, \pi_2, \dots, \pi_M$ stationnaire existe et il est unique. Il peut être calculé en résolvant le système d'équation suivant :

$$\begin{cases} \Pi * Q = 0 \\ \sum \pi_i = 1 \end{cases}$$

3.3.1.2. La description de la méthode approximative du M0

Vue la complexité du calcul matricielle des matrices volumineuse, nous avons opté pour l'adaptation de la méthode approximative « Phase Merging Algorithm » [11]. Sachant que notre capteur a été décrit par une CMTC à deux variables aléatoires $(X(t), Y(t) : t \geq 0)$. Tel que $X(t)$ représente le nombre de paquet au niveau de buffer et $Y(t)$ représente- l'état de système.

Chapitre 3 : Modélisation des réseaux de capteurs.

Où :

$$\left\{ \begin{array}{l} Y(t)=0 \text{ le serveur est à l'état oisif.} \\ Y(t)=1 \text{ le serveur est à l'état occupée.} \end{array} \right.$$

Pour ce faire nous avons abordé les étapes suivantes :

1. La décomposition de la CMTC illustré dans la figure 3.3 en respectant l'état du serveur, on obtient deux espaces d'états S_0 et S_1 . L'espace S_0 représente les états de la CMTC ou le serveur est oisif. A l'opposé, l'espace S_1 contient les états ou le serveur est occupé.

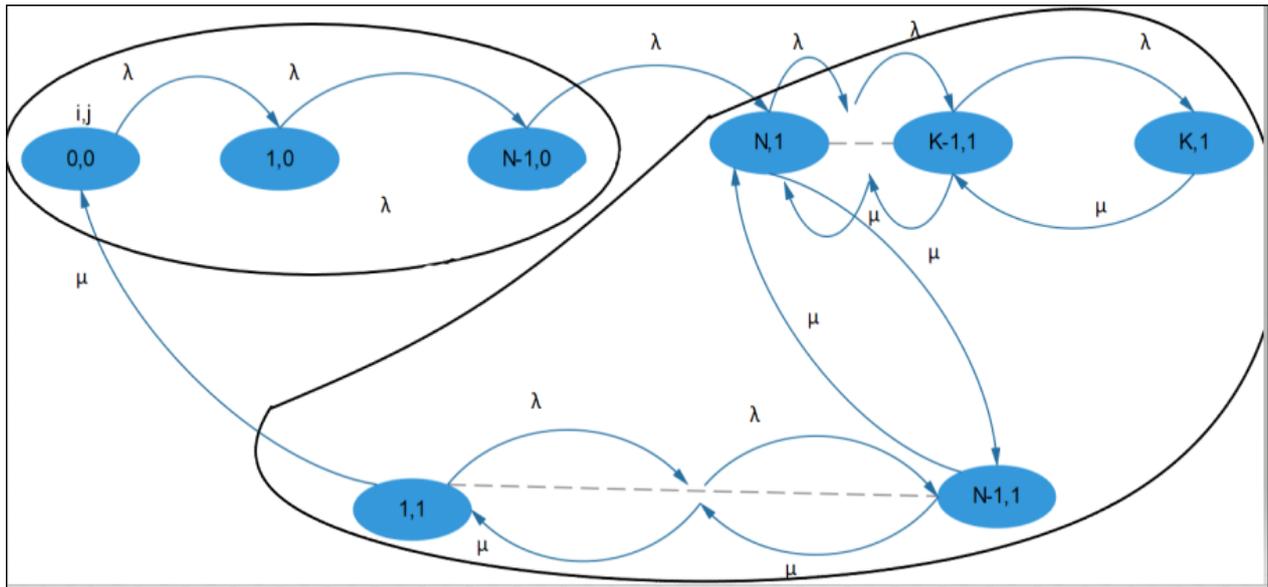


Figure 3.3: La CMTC du MO après le découpage.

2. Nous procédant au calcul de la probabilité $p(j/i)$ pour chaque classe S_j , ($j = \{0,1\}$)

$$p(0/i) = p(0) = \left(\frac{\lambda}{0!} + \frac{\rho}{1!}\right)^{-1} = (\lambda + \rho)^{-1} \quad (\text{avec } \rho = \frac{\lambda}{\mu})$$

$$p(j/i) = p(1) = \rho * p(0/i)$$

3. Puis on calcule $q\langle i, i' \rangle$ qui désigne la probabilité de passage entre la classe S_i et $S_{i'}$

$$q\langle i, i' \rangle = \begin{cases} \lambda * p(0) & \text{si } i' = i + 1 \\ \mu * p(1) & \text{si } i' = i - 1 \end{cases}$$

4. $\Pi\langle i \rangle = \frac{t^i}{i!} * \pi_0$

Avec : $t = \frac{\lambda * p(0)}{\mu * p(1)}$

Et $\Pi\langle 0 \rangle = \left(\sum_0^{N+K} \frac{t^i}{i!}\right)^{-1}$

5. Enfin on obtient le vecteur approximatif de la distribution à l'état stationnaire

$$P(i,j) \cong p(j) * \pi < i >$$

3.3.1.3. Les indices de performances du modèle M0

- La durée moyenne d'une période oisive :

$$\bar{T} = \frac{N}{\lambda}$$

- Le nombre moyen de paquets dans le capteur :

$$\bar{Q} = \sum_{i=0}^{K-1} \sum_{j=0}^1 i * \pi_{i,j} + \sum_{i=0}^{N-1} i * \pi_{i,1}$$

- La durée moyenne d'une période occupée :

$$\bar{B} = \frac{\bar{Q}}{\mu}$$

- La durée moyenne d'une période du cycle :

$$\bar{C} = \bar{T} + \bar{B}$$

- La probabilité d'état oisif :

$$p_i = \sum_{i=0}^{N-1} \pi_{i,0}$$

- La probabilité d'état occupée :

$$p_b = 1 - p_i$$

- La probabilité de blocage :

$$p_{\text{blocage}} = \pi_{K,1}$$

- Le débit de la réception de paquets :

$$\bar{\lambda} = \lambda \left(\sum_{i=0}^{K-1} \sum_{j=0}^1 i * \pi_{i,j} + \sum_{i=0}^{N-1} i * \pi_{i,1} \right)$$

- La durée moyenne d'attente :

$$\bar{W} = \frac{\bar{Q}}{\bar{\lambda}}$$

- L'énergie consommée :

$$EC = p_i * EC_i + p_b * EC_b + EC_s / \bar{C} + \bar{Q} * EC_h .$$

EC_i : La consommation d'énergie de l'unité de transmission à l'état idle.

EC_b : La consommation d'énergie de l'unité de transmission à l'état busy.

EC_s : La consommation d'énergie du changement de l'état de capteur entre réception et transmission.

EC_h : La consommation d'énergie de transmission des paquets.

3.3.2. Modèle avec priorité (modèle M1)

Dans un réseau de capteur à base de cluster, on peut avoir deux types de capteurs : un chef de cluster CH et les membres du cluster CM, ces derniers transmettent les paquets captés au voisin et faire le relai des paquets jusqu'à atteindre le CH. A son tour le CH transmet ces paquets captés et les paquets reçu et les transmet en multi-sauts à la station de base. Chaque nœud (CH ou CM) génère deux types de paquets : les paquets captés et les paquets relayés. Suivant le rôle du nœud dans le cluster, le taux des paquets reçu selon leur nature est différent. Un nœud qui joue le rôle de CM et qui se trouve à la frontière d'un cluster capte plus qu'il fait le relai. A l'opposé un CM qui se trouve proche du sink fait le relai plus qu'il capte. Le CH généralement il reçoit tous les paquets captés dans le cluster et capte aussi. Ainsi, chaque nœud reçoit deux flux, les paquets captés et les paquets relayés. Suivant le rôle du nœud dans le cluster, peut en préciser la priorité pour chaque type de flux.

Dans le but de minimiser le délai d'attente dans le buffer des paquets prioritaires, nous avons proposé de modéliser un nœud capteur par une file d'attente à serveur unique avec vacance et priorité entre les clients (les paquets). Nous opté pour deux classes de priorité, La classe 1 représente les paquets les plus prioritaire. En outre, la classe 2 représente les paquets de faible priorité. Le buffer est de capacité fini K. Les paquets arrivent suivant le processus de poisson de paramètre $\lambda = \lambda_1 + \lambda_2$. Sachant que λ_1 et λ_2 représentent les taux d'arriver des paquets de classes 1 et 2 respectivement. De même, le capteur possède deux états principaux oisif et occupé. Le passage de l'état oisif à l'état occupé est contrôlé par la politique N-vacance. Durant l'état occupé, le capteur commence par servir les paquets de la classe 1 avant ceux de la classe 2 avec le même taux μ .

3.3.2.1. La CMTC du modèle de nœud capteur avec N-vacance et priorité

Considérant le modèle proposé pour un nœud de capteur avec vacance et priorité et on se basant sur le fait qu'un nœud de capteur peut alterner uniquement entre les états oisif et occupée, ce système de file d'attente avec vacance peut être modélisé par une CMTC à trois dimensions ou chaque état est décrit par trois variables aléatoires $(X(t), Y(t), Z(t) : t \geq 0)$. Tel que $X(t)$ représente le nombre de paquets au niveau de premier sous buffer, $Y(t)$: le nombre de paquets au niveau de deuxième sous buffer et $Z(t)$ représente l'état de système.

Où :

$$\left\{ \begin{array}{l} Z(t)=0 \text{ le serveur est à l'état oisif.} \\ Z(t)=1 \text{ le serveur est à l'état occupée.} \end{array} \right.$$

L'espace d'état de la CMTC est donnée par :

$$\Omega = \{(i, j, m) : 0 \leq i \leq K, 0 \leq j \leq K, 0 \leq m \leq 1\}$$

Le diagramme d'état transition correspondant est présenté dans la figure (3.4) et les taux de transition entre les états adjacents présenté dans le tableau suivant (3.2) :

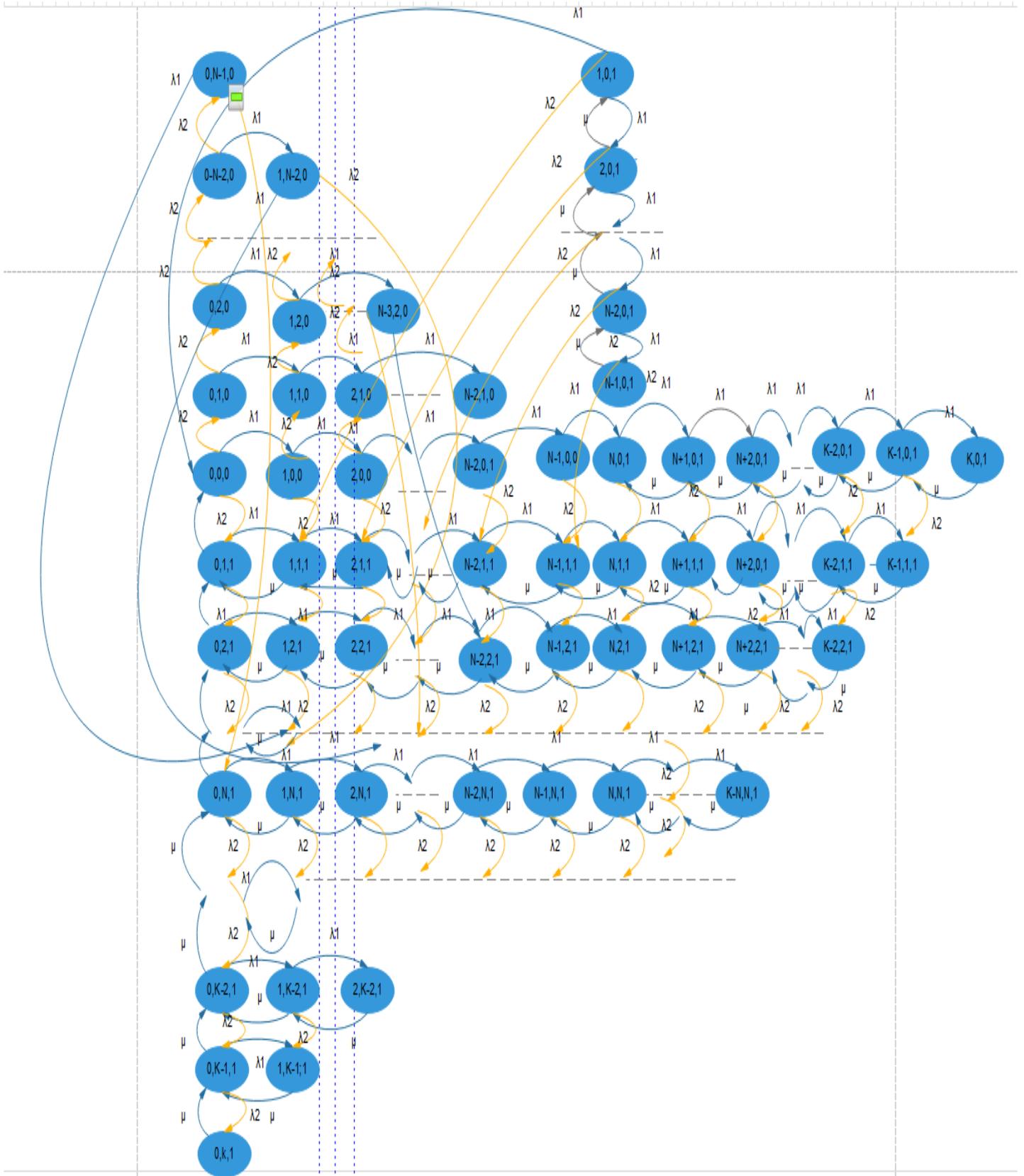


Figure 3.4: la chaine de Markov du modèle M1.

Chapitre 3 : Modélisation des réseaux de capteurs.

Description	Etat sortant	Taux de transition	Etat entrant
$i+j < N-1, m=0$	(i, j, m)	$\lambda 1$	$(i+1, j, 0)$
$i+j < N-1, m=0$	(i, j, m)	$\lambda 2$	$(i, j+1, 0)$
$i+j = N-1, m=0$	(i, j, m)	$\lambda 1$	$(i+1, j, 1)$
$i+j = N-1, m=0$	(i, j, m)	$\lambda 2$	$(i, j+1, 1)$
$N \leq i+j < K, m=1$	(i, j, m)	$\lambda 1$	$(i+1, j, 1)$
$N \leq i+j < K, m=1$	(i, j, m)	$\lambda 2$	$(i, j+1, 1)$
$i=N-1, j=0, m=0$	(i, j, m)	$\lambda 1$	$(N, 0, 1)$
$j=N-1, i=0, m=0$	(i, j, m)	$\lambda 2$	$(0, N, 1)$
$N \leq i \leq K, j=0, m=1$	(i, j, m)	μ	$(i-1, 0, 1)$
$N \leq j \leq K, i=0, m=1$	(i, j, m)	μ	$(0, j-1, 1)$
$N \leq i+j \leq K, m=1$	(i, j, m)	μ	$(i-1, j, 1)$
$0 < i < N, j=0, m=1$	(i, j, m)	μ	$(i-1, 0, 1)$
$0 < j < N, i=0, m=1$	(i, j, m)	μ	$(0, j-1, 1)$
$0 < i+j < N, m=1$	(i, j, m)	μ	$(i-1, j, 1)$
$i=1, j=0, m=1$	(i, j, m)	μ	$(0, 0, 0)$
$i=0, j=1, m=1$	(i, j, m)	μ	$(0, 0, 0)$
$N \leq i < K, j=0, m=1$	(i, j, m)	$\lambda 1$	$(i+1, 0, 1)$
$N \leq j < K, i=0, m=1$	(i, j, m)	$\lambda 2$	$(0, j+1, 1)$

Tableau 3-3: Représentation des taux de transitions entre les états.

Par conséquent, le générateur infinitésimal Q est une matrice M^*M ($M = \sum_{i=1}^{K+1} i + N - 1 + \sum_{i=1}^{N-1} N - i$) et peut-être calculé par l'algorithme (Algorithm II) :

AlgorithmeII : l'algorithme de construction du générateur infinitésimal du M1 :

For $m \leftarrow 0, 1$ **do**

If $m=0$ **then**

$j \leftarrow 0$

For $i \leftarrow 0, N-1$ **do**

$Q [(i, j, m), (i+1, j, m)] \leftarrow \lambda 1$

$Q [(i, j, m), (i, j+1, m)] \leftarrow \lambda_2$

End for

$Q [(N-1, j, m), (N, m+1)] \leftarrow \lambda_1$

$Q [(N-1, j, m), (N-1, j+1, m+1)] \leftarrow \lambda_2$

For $i \leftarrow 0, N-3$ **do**

For $j \leftarrow 1, N-2$ **do**

$Q [(i, j, m), (i+1, j, m)] \leftarrow \lambda_1$

$Q [(i, j, m), (i, j+1, m)] \leftarrow \lambda_2$

End for

End for

For $i \leftarrow 0, N-2$ **do**

$Q [(i, N-i-1, m), (i+1, N-i-1, m+1)] \leftarrow \lambda_1$

$Q [(i, N-i-1, m), (i, N-i-1, m+1)] \leftarrow \lambda_2$

End for

Else

$Q [N, 0, m), (N+1, 0, m)] \leftarrow \lambda_1$

$Q [N, 0, m), (N, 1, m)] \leftarrow \lambda_2$

$Q [N, 0, m), (N-1, 0, m)] \leftarrow \mu$

For $i \leftarrow N+1, K$ **do**

$Q [i, 0, m), (i+1, 0, m)] \leftarrow \lambda_1$

$Q [i, 0, m), (i+1, j+0, m)] \leftarrow \lambda_2$

$Q [i, 0, m), (i-1, 0, m)] \leftarrow \mu$

End for

$Q [K, j, m), (K-1, j, m)] \leftarrow \mu$

For $i \leftarrow 2, N-1$ **do**

$Q [i, 0, m), (i+1, 0, m)] \leftarrow \lambda_1$

$Q [i, 0, m), (i+1, 1, m)] \leftarrow \lambda_2$

$Q [i+1, 0, m), (i, 0, m)] \leftarrow \mu$

End for

$Q [1, 0, m), (2, 0, m)] \leftarrow \lambda_1$

Chapitre 3 : Modélisation des réseaux de capteurs.

$$Q [1, 0, m), (1, 1, m)] \leftarrow \lambda_2$$

$$Q [2, 0, m), (1, 0, m)] \leftarrow \mu$$

$$Q [1, 0, m), (0, 0, m-1)] \leftarrow \mu$$

For $i \leftarrow 1, K-2$ **do**

For $j \leftarrow 2, K-i-1$ **do**

$$Q [(i, j, m), (i+1, j, m)] \leftarrow \lambda_1$$

$$Q [(i, j, m), (i, j+1, m)] \leftarrow \lambda_2$$

$$Q [(i, j, m), (i-1, j, m)] \leftarrow \mu$$

End for

End for

For $j \leftarrow 2, K-1$ **do**

$$Q [1, 0, m), (1, j, m)] \leftarrow \lambda_1$$

$$Q [0, j, m), (1, j+1, m)] \leftarrow \lambda_2$$

$$Q [0, j, m), (0, j-1, m)] \leftarrow \mu$$

End for

$$Q [0, 1, m), (0, 2, 1)] \leftarrow \lambda_2$$

$$Q [0, 1, m), (0, 0, m-1)] \leftarrow \mu$$

$$Q [0, K, m), (0, K-1, m)] \leftarrow \mu$$

For $i \leftarrow 1, K-1$ **do**

$$Q [i, K-i, m), (i-1, K-i, m)] \leftarrow \mu$$

End for

End if

End for

Comme la CMTC proposé est ergodique, l'espace d'état est fini et irréductible, le vecteur des probabilités stationnaire $\pi = \pi_1, \pi_2, \dots, \pi_M$ existe et il est unique. Il peut être calculé, on résoudra le système d'équation suivant de même principe dans La CMTC du modèle de nœud capteur avec N-vacance :

$$\left\{ \begin{array}{l} \Pi * Q = 0 \\ \sum \pi_i = 1 \end{array} \right.$$

3.3.2.2. La description de la méthode approximative du M1

Le fonctionnement de nœud de capteur à base de fils d'attente avec vacance est décrit par une chaîne de Markov à trois dimensions. Sachant que $(X(t), Y(t), Z(t) : t \geq 0)$. Tel que $X(t)$ représente le nombre de paquets de classe 1 dans le buffer ($X(t) = (0, 1, \dots, K)$), $Y(t)$ représente le nombre de paquets de la classe 2 ($Y(t) = (0, 1, \dots, K)$) et $Z(t)$ représente l'état de système.

Où :

$$\left[\begin{array}{l} Z(t)=0 \text{ le serveur est à l'état oisif} \\ Z(t)=1 \text{ le serveur est à l'état occupée} \end{array} \right.$$

En appliquant la méthode approximative sur les chaînes de Markov multidimensionnelles « **Hierarchical space merging algorithm** » proposé par Melikov [12], on aura deux cas le premier si $\lambda_1 \ll \lambda_2$ et l'autre dans le cas où $\lambda_1 \gg \lambda_2$ pour ce faire on suit les étapes suivantes :

- Si $\lambda_2 \gg \lambda_1$ (division par colonne)
 1. La décomposition de la CMTC de 3D à une chaîne de 2D illustré dans (la figure 3.5) décrite par $X(t)$ et $Y(t)$, et une autre unidimensionnelle décrite par la variable $Z(t)$, dans ce cas on fait le découpage des espaces d'états S_K^X par colonne par rapport le nombre des paquets dans le premier buffer :

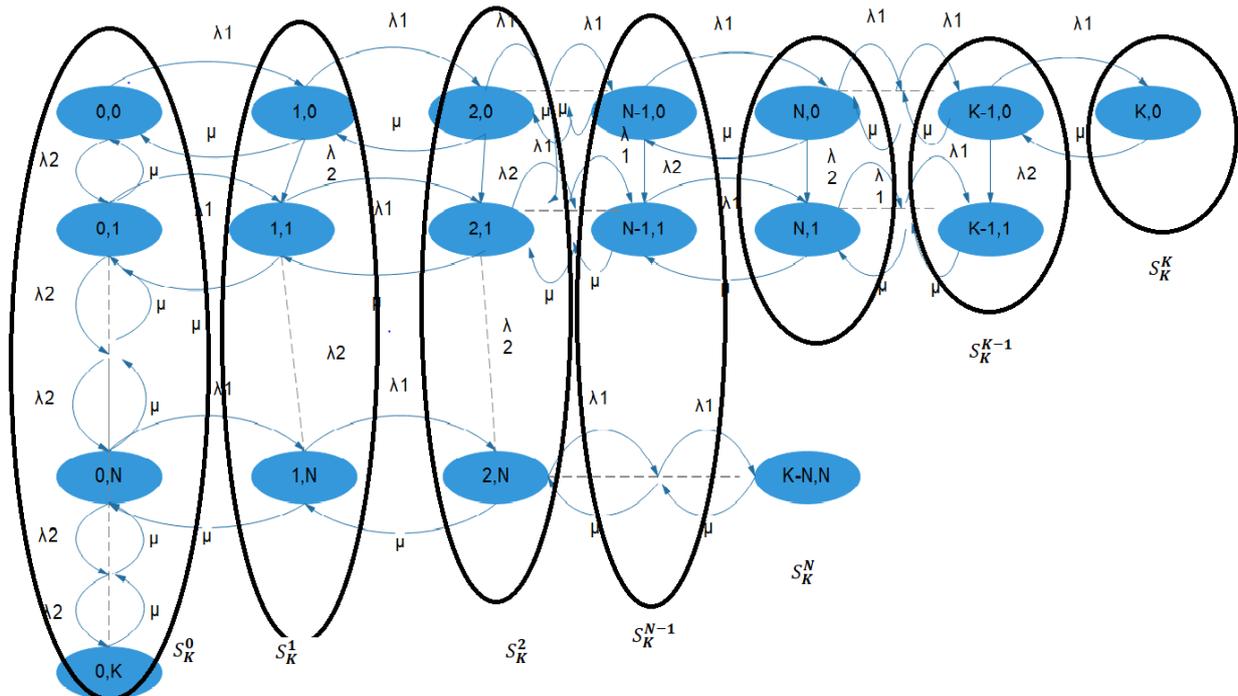


Figure 3.5: la chaine de Markov à 2D après la décomposition et la division par colonne.

2. Nous procédons au calcul de $\pi(i,j)$ qui désigne la probabilité de l'état (i,j) dans l'espace d'états $S_K^i, i=0, \dots, k$ et $j=0, \dots, K$

$$\pi(i,j) = \begin{cases} \frac{1-V_1}{1-V_1^{(K+1-i)}} V_1^j & \text{Si } i=0 \\ \frac{1-\lambda_2}{1-\lambda_2^{(K+1-i)}} \lambda_2^j & \text{Si } 0 < i \leq N-2 \\ \frac{1-\lambda_2}{1-\lambda_2^{(K-i+2)}} \lambda_2^j & \text{Si } i=N-1 \\ \frac{1-\lambda_2}{1-\lambda_2^{(K-i+1)}} \lambda_2^j & \text{Si } N \leq i \leq K \end{cases}$$

Avec : $V_1 = \frac{\lambda_2}{\mu}$

3. Puis on calcule $q(\langle i \rangle, \langle j \rangle)$ qui représente la probabilité de passage de l'espace d'états S_K^i au S_K^j , elle est donné par :

$$q(\langle i \rangle, \langle j \rangle) = \begin{cases} \lambda_1(1-\pi_0(K-i)) & \text{si } j=i+1 \\ \mu(1-\pi_0(i)) & \text{si } j=i-1 \\ 0 & \text{autrement} \end{cases}$$

Chapitre 3 : Modélisation des réseaux de capteurs.

4. $\pi_2\langle n_r \rangle$ est la probabilité de l'état fusionné, décrit par :

$$\pi_2\langle n_r \rangle = \prod_{i=0}^{K-1} \frac{q(\langle i \rangle, \langle i+1 \rangle)}{q(\langle i+1 \rangle, \langle i \rangle)} * \pi_2\langle 0 \rangle \quad (n_r = 1, \dots, K)$$

On obtient la valeur de $\pi_2\langle 0 \rangle$ à partir de la condition de normalisation suivante :

$$\sum_{n_1=0}^K \pi_2(\langle n_1 \rangle) = 1.$$

5. $q(S_K, S_{K'})$ décrit la chaîne unidimensionnelle lors de passage de l'espace S_K à l'espace

$$q(S_K, S_{K'}) = \begin{cases} \lambda_1 & \text{si } K' = K+1 \\ \mu & \text{si } K' = K-1 \\ 0 & \text{autrement} \end{cases}$$

6. $\pi_1\langle 1 \rangle = \varphi (\sum_{i=0}^1 \frac{\varphi^i}{i!})^{-1}$

Avec : $\varphi = \lambda_1 / \mu$

7. On déduit que le vecteur de probabilité à l'état stationnaire de la CMTC à 3D est donné par :

$$p(n_1, n_2, n_3) = \rho n_1(n_2) * \pi_2\langle n_2 \rangle * \pi_1\langle 1 \rangle$$

- Si $\lambda_2 \ll \lambda_1$ (Division par ligne)

Dans ce cas la CMTC à 2D est décomposé par ligne à un ensemble d'espace d'états $S_K^{n_2}$ par rapport à la valeur prise par la variable $Y(t)$ (voir figure 3.6) :

Chapitre 3 : Modélisation des réseaux de capteurs.

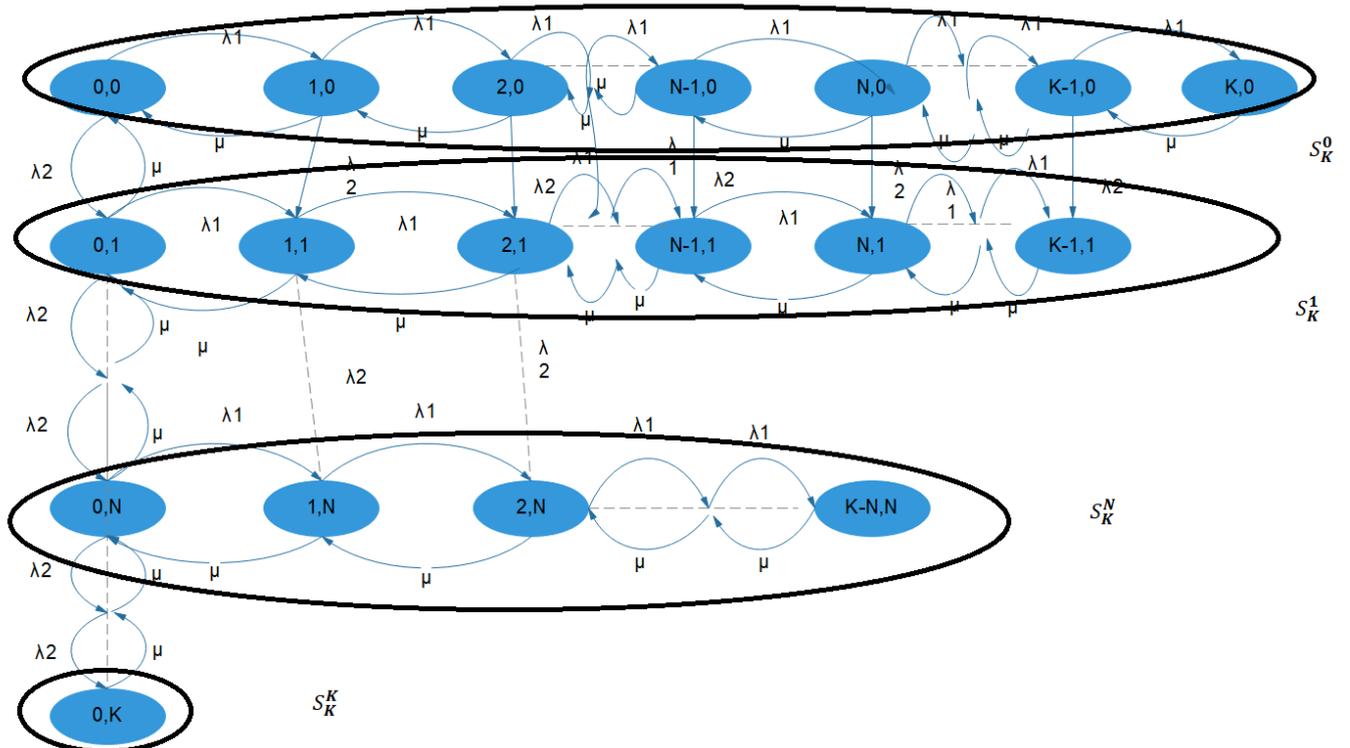


Figure 3.6: la chaîne de Markov à 2D après la décomposition et la division par colonne.

En suivant les mêmes étapes citées dans le premier cas, on obtient les équations suivantes :

$$1. \quad \pi_i(j) = \begin{cases} \frac{1-V_1}{1-V_1^{(K+N)}} V_1^j & \text{si } i=0 \\ \frac{1-V_1}{1-V_1^{(K+N+1-2i)}} V_1^j & \text{si } 0 < i \leq N-1 \\ \frac{1-V_1}{1-V_1^{(K+1-i)}} V_1^j & \text{si } N \leq i \leq K \end{cases}$$

Avec : $V_1 = \lambda_1 / \mu$.

$$2. \quad q(\langle i \rangle, \langle j \rangle) = \begin{cases} \lambda_2(1-\rho_i-(K-i)) & \text{si } j=i+1 \\ \mu(\rho_0(i)) & \text{si } j=i-1 \\ 0 & \text{autrement} \end{cases}$$

$$3. \quad \pi_2\langle n_i \rangle = \prod_{i=0}^{K-1} \frac{q(\langle i \rangle, \langle i+1 \rangle)}{q(\langle i+1 \rangle, \langle i \rangle)} * \pi_2\langle 0 \rangle$$

$$4. \quad q(S_K, S_{K'}) = \begin{cases} \lambda_2 & \text{si } K' = K+1 \\ \mu & \text{si } K' = K-1 \\ 0 & \text{autrement} \end{cases}$$

$$5. \quad \pi_{1 < 1 >} = \varphi \left(\sum_{i=0}^1 \frac{\varphi^i}{i!} \right)^{-1}$$

$$\text{Avec : } \varphi = \lambda_1 / \mu$$

6. On obtient par la suite le vecteur des probabilités à l'état stationnaire donné par :

$$p(n_1, n_2, n_3) = \rho n_1(n_2) * \pi_{2 < n_2 >} * \pi_{1 < 1 >}$$

3.3.2.3. Les indices de performances du modèle M1

- La durée moyenne d'une période oisive :

$$\bar{I} = \frac{N}{\lambda_1 + \lambda_2}$$

- Le nombre moyen des paquets de haute priorité en attente :

$$\bar{Q}_1 = \sum_{m=0}^1 \sum_{i=1}^K i * \pi_{i,0,m} + \sum_{i=1}^{N-1} i * \pi_{i,0,1} + \sum_{i=0}^{K-1} \sum_{j=1}^{K-i} i * \pi_{i,0,1} + \sum_{i=0}^{N-2} i * \pi_{i,j,0}$$

- Le nombre moyen des paquets de faible priorité en attente :

$$\bar{Q}_2 = \sum_{i=0}^{K-1} \sum_{j=1}^{K-i} j * \pi_{i,j,1} + \sum_{i=0}^{N-2} \sum_{j=1}^{N-1-i} j * \pi_{i,j,0}$$

- Le nombre moyen de paquets en attente :

$$\bar{Q} = \bar{Q}_1 + \bar{Q}_2$$

- La durée moyenne d'une période occupée :

$$\bar{B} = \frac{\bar{Q}}{\mu}$$

- La durée moyenne d'une période du cycle :

$$\bar{C} = \bar{I} + \bar{B}$$

- La probabilité d'état oisif :

$$p_i = \sum_{i=0}^{N-1} \pi_{i,0,0} + \sum_{i=0}^{N-1} \sum_{j=1}^{N-i-1} \pi_{i,j,0}$$

- La probabilité d'état occupé :

$$p_b = 1 - p_i$$

- La probabilité de blocage :

$$p_{\text{blocage}} = \pi_{K,0,1} + \sum_{i=0}^{K-1} \pi_{i,K-i,1}$$

- **Le débit de la réception de paquets :**

$$\bar{\lambda} = \bar{\lambda}_1 + \bar{\lambda}_2$$

Tel que :

$$\bar{\lambda}_1 = \lambda_1 \left(\sum_{m=0}^1 \sum_{i=1}^{K-1} i * \pi_{i,0,m} + \sum_{i=1}^{N-1} i * \pi_{i,0,1} + \sum_{i=0}^{k-1} \sum_{j=1}^{K-i} i * \pi_{i,0,1} + \sum_{i=0}^{K-2} \sum_{j=1}^{K-i-1} i * \pi_{i,0,1} + \sum_{i=0}^{N-2} \sum_{j=1}^{N-i-1} i * \pi_{i,j,0} \right)$$

$$\bar{\lambda}_2 = \lambda_2 \left(\sum_{m=0}^1 \sum_{i=1}^{K-1} j * \pi_{i,0,m} + \sum_{i=1}^{N-1} j * \pi_{i,0,1} + \sum_{i=0}^{k-2} \sum_{j=1}^{K-i-1} j * \pi_{i,j,1} + \sum_{i=0}^{N-2} \sum_{j=1}^{N-i-1} j * \pi_{i,j,0} \right)$$

- **La durée moyenne d'attente :**

$$\bar{W} = \bar{W}_1 + \bar{W}_2 = \frac{\bar{Q}_1}{\bar{\lambda}_1} + \frac{\bar{Q}_2}{\bar{\lambda}_2}$$

- **L'énergie consommée :**

$$EC = p_i * EC_i + p_b * EC_b + EC_s / \bar{C} + \bar{Q} * EC_h .$$

EC_i : La consommation d'énergie de l'unité de transmission à l'état idle.

EC_b : La consommation d'énergie de l'unité de transmission à l'état busy.

EC_s : La consommation d'énergie du changement de l'état de capteur entre réception et transmission.

EC_h : La consommation d'énergie de transmission des paquets.

3.3.3. Le modèle avec N-vacance et Rappel (M2)

Dans les modèles M0 et M1, en cas de saturation du buffer, les paquets seront perdus. Pour remédier à ce problème dans ce modèle, nous prenons en considération la retransmission des paquets (pour les deux classes de priorités) en cas d'overflow. Pour ce faire, nous avons modélisé, un capteur par une file d'attente finie avec vacance, priorité et rappel. Le capteur de la même façon possède deux états de fonctionnement oisif et occupé. La différence par rapport aux modèles M0 et M1 réside dans le fait qu'en cas de saturation du buffer, chaque paquet qui arrive a la possibilité d'être retransmis avec la probabilité p ou d'être perdu avec la probabilité $1-p$. Les paquets qui seront retransmis rejoignent un espace d'attente imaginaire qui est l'orbite. Ainsi, les paquets qui se trouvent dans l'orbite seront retransmis avec le taux de rappel θ_1 pour les paquets de la classe de priorité 1 et θ_2 pour les paquets de la classe de priorité 2.

3.3.3.1. La CMTC du modèle M2 avec N-vacance et Rappel

Considérant le modèle proposé pour un nœud de capteur avec retransmission et on se basant sur le fait qu'un nœud de capteur peut alterner uniquement entre les états oisif et occupé, ce système de file d'attente avec rappel peut être modélisé par une CMTC à cinq dimensions ou chaque état est décrit par cinq variables aléatoires (I(t), J(t), X(t), Y(t), Z(t)) :

Chapitre 3 : Modélisation des réseaux de capteurs.

$t \geq 0$). Tel que $I(t)$ représente le nombre de paquets au niveau du buffer de classe 1, $J(t)$ représente le nombre de paquets au niveau du buffer de classe 2, $X(t)$ représente le nombre de paquets de classe 1 dans l'orbite, $Y(t)$ représente le nombre de paquet au niveau du buffer de classe 1, $Z(t)$ représente l'état de système.

Où :

$$\left\{ \begin{array}{l} Z(t)=0 \text{ le serveur est à l'état oisif} \\ Z(t)=1 \text{ le serveur est à l'état occupée} \end{array} \right.$$

L'espace d'état de la CMTC est donnée par :

$$\Omega = \{(i, j, x, y, z) : 0 \leq i \leq K, 0 \leq x \leq 1, 0 \leq y \leq R, 0 \leq z \leq R\}$$

Le diagramme d'état transition correspondant est présenté dans la figure (3.7) et les taux de transition entre les états adjacents présenté dans le tableau suivant (3.4) :

Chapitre 3 : Modélisation des réseaux de capteurs.

Description	Etat sortant	Taux de transition	Etat entrant
$i+j < N-1, x=0, y=0, z=0$	(i, j, x, y, z)	$\lambda 1$	$(i+1, j, x, y, z)$
$i+j < N-1, x=0, y=0, z=0$	(i, j, x, y, z)	$\lambda 2$	$(i, j+1, x, y, z)$
$i+j = N-1, x=0, y=0, z=0$	(i, j, x, y, z)	$\lambda 1$	$(N, j, x, y, 1)$
$i+j = N-1, x=0, y=0, z=0$	(i, j, x, y, z)	$\lambda 2$	$(i, j+1, x, y, 1)$
$N \leq i+j < K, x=0, y=0, z=1$	(i, j, x, y, z)	$\lambda 1$	$(i+1, j, x, y, 1)$
$N \leq i+j < K, x=0, y=0, z=1$	(i, j, x, y, z)	$\lambda 2$	$(i, j+1, x, y, 1)$
$i=K, j=0, x=0, y=0, z=1$	(i, j, x, y, z)	$\lambda 1 * p$	$(K, j, 1, y, 1)$
$j=K, i=0, x=0, y=0, z=1$	(i, j, x, y, z)	$\lambda 2 * p$	$(i, K, x, 1, 1)$
$i+j=K, x+y < R, z=1$	(i, j, x, y, z)	$\lambda 1 * p$	$(i, j, x+1, y, 1)$
$i+j=K, x+y < R, z=1$	(i, j, x, y, z)	$\lambda 2 * p$	$(i, j, x, y+1, 1)$
$N \leq i \leq K, j=0, z=1$	(i, j, x, y, z)	μ	$(i-1, j, x, y, 1)$
$N \leq j \leq K, i=0, z=1$	(i, j, x, y, z)	μ	$(i, j-1, x, y, 1)$
$N \leq i+j \leq K, z=1$	(i, j, x, y, z)	μ	$(i-1, j, x, y, 1)$
$0 < i < N, j=0, z=1$	(i, j, x, y, z)	μ	$(i-1, j, x, y, 1)$
$0 < j < N, i=0, z=1$	(i, j, x, y, z)	μ	$(i, j-1, x, y, 1)$
$0 < i+j < N, z=1$	(i, j, x, y, z)	μ	$(i-1, j, x, y, 1)$
$i=1, j=0, x=0, y=0, z=1$	(i, j, x, y, z)	μ	$(0, 0, 0, 0, 0)$
$i=0, j=1, x=0, y=0, z=1$	(i, j, x, y, z)	μ	$(0, 0, 0, 0, 0)$
$0 < i < N, j=0, z=1$	(i, j, x, y, z)	$\lambda 1$	$(i+1, j, x, y, 1)$
$0 < j < N, i=0, z=1$	(i, j, x, y, z)	$\lambda 2$	$(i, j+1, x, y, 1)$

Chapitre 3 : Modélisation des réseaux de capteurs.

$i+j < K, x \neq 0$	(i, j, x, y, z)	$\theta 1 * x$	$(i+1, j, x-1, y, z)$
$i+j < K, j \neq 0$	(i, j, x, y, z)	$\theta 2 * y$	$(i, j+1, x, y-1, z)$

Tableau 3-4: Représentation des taux de transitions entre les états.

Par conséquent, le générateur infinitésimal Q est une matrice M^*M ($M = \sum_{i=1}^{K+R+1} i + N - 1 + \sum_{i=1}^{N-1} N - i + ((\sum_{i=1}^{K+1} i + \sum_{i=1}^N i) - 2) * \sum_{i=1}^R i + (\sum_{i=N+1}^K i + N * (N + 1) - 1) * R$) et peut-être calculé par l'algorithme (Algorithme III) :

Algorithme III : l'algorithme de construction du générateur infinitésimal du M2 :

For $z \leftarrow 0, 1$ **do**

If $z=0$ **then**

$j \leftarrow 0$

For $i \leftarrow 0, N-1$ **do**

$Q [(i, 0, x, y, z), (i+1, j, x, y, z)] \leftarrow \lambda 1$

$Q [(i, 0, x, y, z), (i, j+1, x, y, z)] \leftarrow \lambda 2$

End for

$Q [(N-1, 0, x, y, z), (N, j, x, y, z+1)] \leftarrow \lambda 1$

$Q [(N-1, 0, x, y, z), (N-1, j+1, x, y, z+1)] \leftarrow \lambda 2$

Else

For $i \leftarrow 0, N-1$ **do**

$Q [(i, N-i-1, x, y, z), (i+1, N-i-1, x, y, z+1)] \leftarrow \lambda 1$

$Q [(i, N-i-1, x, y, z), (i, N-i-1, x, y, z+1)] \leftarrow \lambda 2$

End for

$Q [N, 0, x, y, z), (N+1, 0, x, y, z)] \leftarrow \lambda 1$

$Q [N, 0, x, y, z), (N, 1, x, y, z)] \leftarrow \lambda 2$

$Q [N, 0, x, y, z), (N-1, 0, x, y, z)] \leftarrow \mu$

For $i \leftarrow N+1, K$ **do**

$Q [i, 0, x, y, z), (i+1, 0, x, y, z)] \leftarrow \lambda 1$

$Q [i, 0, x, y, z), (i, j+1, x, y, z)] \leftarrow \lambda 2$

$Q [i, 0, x, y, z), (i-1, 0, x, y, z)] \leftarrow \mu$

End for

$Q [K, j, x, y, z), (K-1, j, x, y, z)] \leftarrow \mu$

End if

For $i \leftarrow 2, N-1$ **do**

$Q [i, 0, x, y, z), (i+1, 0, x, y, z)] \leftarrow \lambda_1$

$Q [i, 0, x, y, z), (i, 1, x, y, z)] \leftarrow \lambda_2$

$Q [i+1, 0, x, y, z), (i, 0, x, y, z)] \leftarrow \mu$

End for

$Q [1, 0, x, y, z), (2, 0, x, y, z)] \leftarrow \lambda_1$

$Q [1, 0, x, y, z), (1, 1, x, y, z)] \leftarrow \lambda_2$

$Q [2, 0, x, y, z), (1, 0, x, y, z)] \leftarrow \mu$

$Q [1, 0, x, y, z), (0, 0, x, y, z-1)] \leftarrow \mu$

For $i \leftarrow 1, K-2$ **do**

For $j \leftarrow 2, K-i-1$ **do**

$Q [(i, j, x, y, z), (i+1, j, x, y, z)] \leftarrow \lambda_1$

$Q [(i, j, x, y, z), (i, j+1, x, y, z)] \leftarrow \lambda_2$

$Q [(i, j, x, y, z), (i-1, j, x, y, z)] \leftarrow \mu$

End for

End for

For $j \leftarrow 2, K-1$ **do**

$Q [1, 0, x, y, z), (2, j, x, y, z)] \leftarrow \lambda_1$

$Q [0, j, x, y, z), (1, j+1, x, y, z)] \leftarrow \lambda_2$

$Q [0, j, x, y, z), (0, j-1, x, y, z)] \leftarrow \mu$

End for

For $x \leftarrow 0, R$ **do**

For $j \leftarrow 0, R-x$ **do**

If $(i+j = K \text{ and } x+y < R)$ **then**

$Q [i, j, x, y, z), (i, j, x+1, y, z)] \leftarrow \lambda_1 * p$

$Q [i, j, x, y, z), (i, j, x, y+1, z)] \leftarrow \lambda_2 * p$

End if

If (i+ j<K) then

$Q [i, j, x, y, z), (i+1, j, x-1, y, z)] \leftarrow \theta 1 * x$

$Q [i, j, x, y, z), (i, j+1, x, y-1, z)] \leftarrow \theta 2 * y$

End if

End for

End for

Comme la CMTC proposé est ergodique, l'espace d'état est fini et irréductible, le vecteur des probabilités stationnaire $\pi = \pi_1, \pi_2, \dots, \pi_M$ existe et il est unique. Il peut être calculé, on résoudre le système d'équation suivant de même principe dans La CMTC du modèle de nœud capteur avec N-vacance :

$$\begin{cases} \Pi * Q = 0 \\ \sum \pi_i = 1 \end{cases}$$

3.3.3.2. La description de la méthode approximative du M2

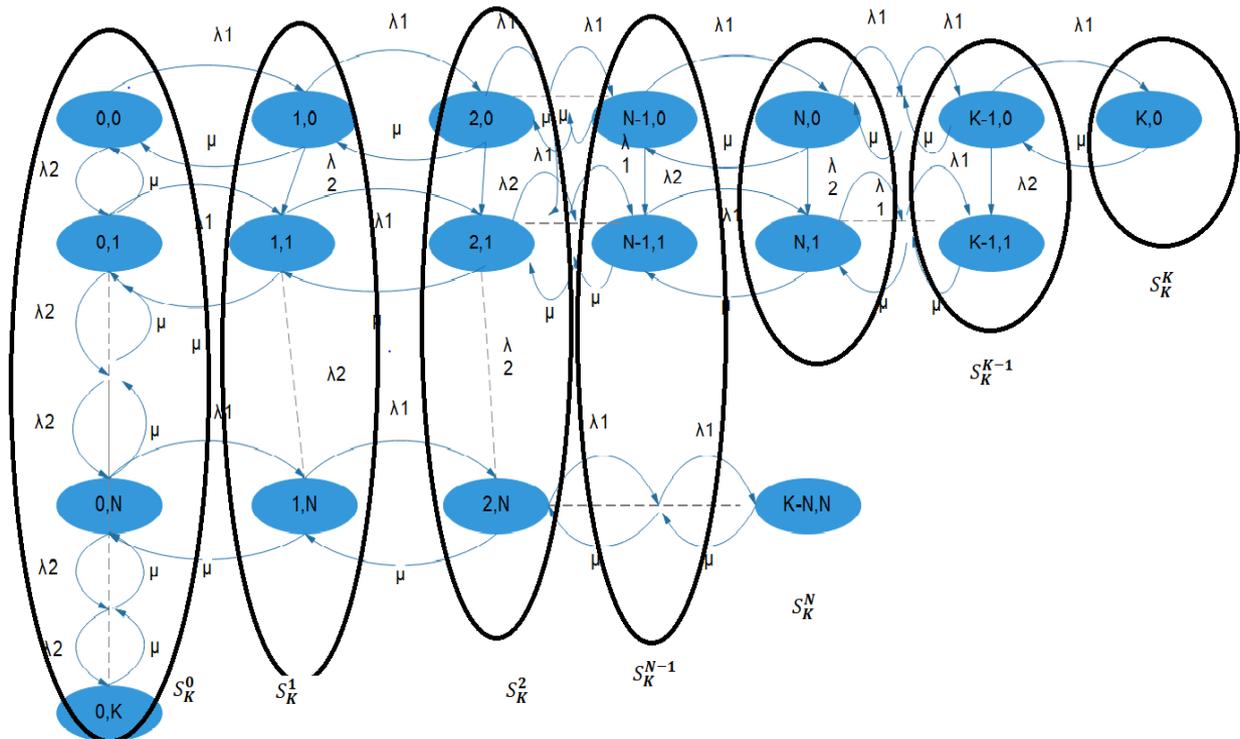
Le fonctionnement de nœud de capteur à base de fils d'attente avec vacance est décrit par une chaîne de Markov à cinq dimensions ou chaque état est décrit par cinq variables aléatoires (I(t), J(t), X(t), Y(t), Z(t) : t ≥ 0). Tel que I(t) représente le nombre de paquets au niveau du buffer de classe 1, J(t) représente le nombre de paquets au niveau du buffer de classe 2, X(t) représente le nombre de paquets de classe 1 dans l'orbite, Y(t) représente le nombre de paquet au niveau du buffer de classe 1, Z(t) représente l'état de système.

Où :

$$\begin{cases} Z(t) = 0 \text{ le serveur est à l'état oisif} \\ Z(t) = 1 \text{ le serveur est à l'état occupée} \end{cases}$$

En appliquant la méthode approximative sur les chaînes de Markov multidimensionnelles « **Hierarchical space merging algorithm** » proposé par Melikov [12], notre chaîne de 5D est décrite par le vecteur $n = (n_1, n_2, n_3, n_4, n_5)$, qui représente les variables aléatoires : (I(t), J(t), X(t), Y(t) et Z(t) respectivement. Nous avons opté à la décomposition de la CMTC de 5D à deux chaînes de 2D qui représentent à la fois le nombre de paquets dans les deux buffers I(t) et J(t) et les deux orbites X(t), Y(t) et une autre unidimensionnelle qui représente l'état de serveur, dans ce cas on fait le on aura deux cas le premier si $\lambda_1 \ll \lambda_2$ et l'autre dans le cas ou $\lambda_1 \gg \lambda_2$ pour ce faire on suit les étapes suivantes :

- Si $\lambda_1 \ll \lambda_2$ (division par colonne)
 - ✓ La chaîne 2D (buffer 1 et buffer 2)



1. Comme il est montré par la figure 3.8, on fait le découpage des espaces d'états S_K^l par colonne par rapport au nombre des paquets de la classe 1 :

Figure 3.8: la chaîne de Markov à 2D après la décomposition et la division par colonne.

2. Nous procédons au calcul de $\pi(j)$ désigne la probabilité de l'état (i,j) dans l'espace d'états S_K^i , $i=0, \dots, k$ et $j=0, \dots, K$

$$\pi(j) = \left\{ \begin{array}{ll} \frac{1-V_1}{1-V_1^{((R+1)((R+2)(K+N))/2)}} V_2^j & \text{Si } i=0 \\ \frac{1-\lambda_2}{1-\lambda_2^{(K+R+N+1-2i+(K+N-1)\lceil \frac{R(R+3)}{2} \rceil)}} \lambda_2^j & \text{Si } 0 < i \leq N-1 \\ \frac{1-\lambda_2}{1-\lambda_2^{(K-N+1)\left(\frac{2+R(R+1)}{2}+R\right)}} \lambda_2^j & \text{Si } i=N \\ \frac{1-\lambda_2}{1-\lambda_2^{(R+1)\lceil \frac{2+R}{2} \rceil}} \lambda_2^j & \text{Si } N < i \leq K \end{array} \right.$$

Avec : $V_2 = \frac{\lambda_2}{\mu}$

Chapitre 3 : Modélisation des réseaux de capteurs.

3. Puis, on calcule $q(\langle i \rangle, \langle j \rangle)$ désigne la probabilité de passage de l'espace d'états S_K^i au S_K^j , elle est donné par :

$$q(\langle i \rangle, \langle j \rangle) = \begin{cases} \lambda_1(1-\rho_1(K-i)) & \text{si } j=i+1 \\ \mu(1-\rho_0(i)) & \text{si } j=i-1 \end{cases}$$

4. $\pi_2 \langle n_1 \rangle$ est la probabilité de l'état fusionné, décrit par :

$$\pi_2 \langle n_1 \rangle = \prod_{i=0}^{K-1} \frac{q(\langle i \rangle, \langle i+1 \rangle)}{q(\langle i+1 \rangle, \langle i \rangle)} \pi_2 \langle 0 \rangle \quad (n_1 = 1, \dots, K)$$

On obtient la valeur de $\pi_2 \langle 0 \rangle$ à partir de la condition de normalisation suivante :

$$\sum_{n_1=0}^K \pi_2(\langle n_1 \rangle) = 1.$$

✓ La chaîne 2D (orbite 1 et orbite 2)

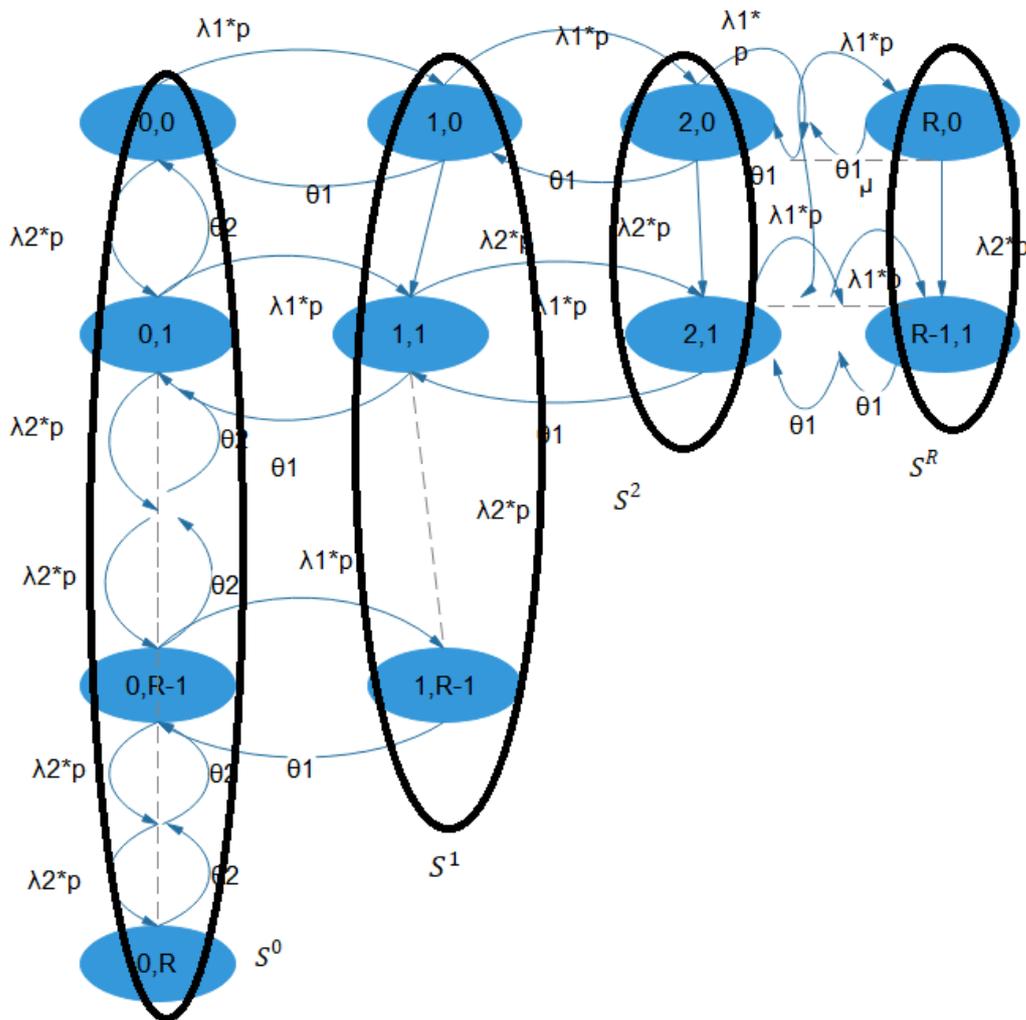


Figure 3.9 : la chaîne de Markov à 2D après la décomposition et la division par colonne.

On décompose la chaîne par colonne et on suit les mêmes étapes citées dans le premier cas, on obtient les équations suivantes :

$$1. P_x(y) = \begin{cases} \frac{1-V_1}{1-V_1 \left(\frac{(N-1)(N+2)(K+1)(K+2)+R(K+1)(K+2)+N(N+1)-4}{2} \right)} V_1^y & \text{si } x=0 \\ \frac{1-V_1}{1-V_1 \left(\frac{(K+1)(K+2)+N(N+1)-4}{2} \right)^{(R-i+1)}} V_1^y & \text{si } 1 \leq x \leq R \end{cases}$$

Avec : $V_1 = \lambda_2 * p / \theta_2$.

$$2. q(\langle x \rangle, \langle y \rangle) = \begin{cases} \lambda_1 * p(1-\rho_i - (R-i)) & \text{si } y=x+1 \\ \theta_1 (1-\rho_0(i)) & \text{si } y=x-1 \\ 0 & \text{autrement} \end{cases}$$

$$3. \pi_2 \langle n_3 \rangle = \prod_{i=0}^{K-1} \frac{q(\langle i \rangle, \langle i+1 \rangle)}{q(\langle i+1 \rangle, \langle i \rangle)} * \pi_2 \langle 0 \rangle$$

✓ $q(S_K, S_{K'})$ et $\pi_1 \langle 1 \rangle$ décrivent la chaîne unidimensionnelle lors de passage de l'espace S_K à l'espace $S_{K'}$:

$$q(S_K, S_{K'}) = \begin{cases} \lambda_1 & \text{si } K'=K+1 \\ \mu & \text{si } K'=K-1 \\ 0 & \text{autrement} \end{cases}$$

$$\pi_1 \langle 1 \rangle = \varphi \left(\sum_{i=0}^1 \frac{\varphi^i}{i!} \right) - 1$$

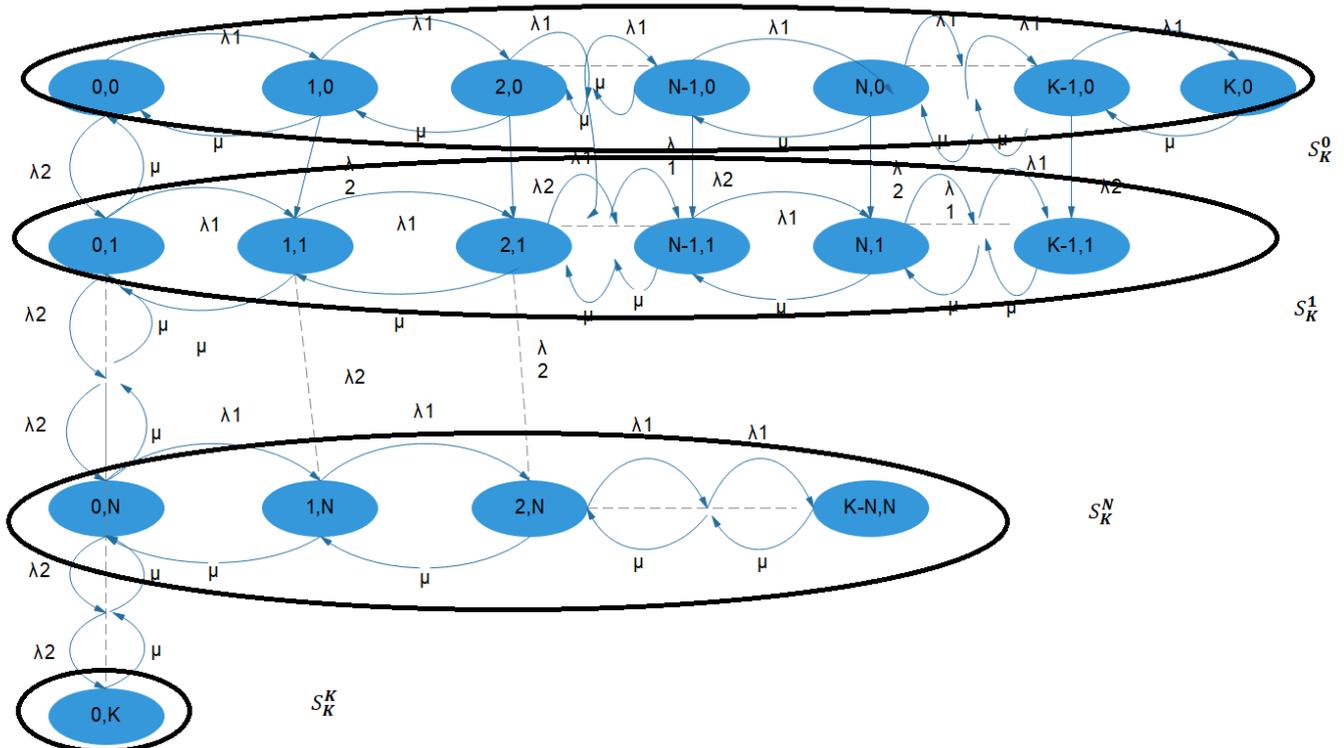
Avec : $\varphi = \lambda_1 / \mu$

On calcule le vecteur de probabilité à l'état stationnaire de la CMTC à 5D à partir de l'équation suivante :

$$P(n_1, n_2, n_3, n_4, n_5) = \rho n_1(n_2) * \pi_2 \langle n_1 \rangle * \rho n_3(n_4) * \pi_2 \langle n_3 \rangle * \pi_1 \langle 1 \rangle$$

- Si $\lambda_1 \gg \lambda_2$ (Division par ligne)
 - ✓ La chaîne 2D (buffer1, buffer2)

Chapitre 3 : Modélisation des réseaux de capteurs.



Dans ce cas la CMTC à 2D est décomposé par ligne à un ensemble d'espace d'états S_K^{n2} par rapport à la valeur prise par la variable $J(t)$ (voir figure 3.10) :

Figure 3.10: la chaîne de Markov à 2D après la décomposition et la division par colonne.

En suivant les mêmes étapes citées dans le premier cas, on obtient les équations suivantes :

$$1. P_j(i) = \begin{cases} \frac{1-V_1}{1-V_1^{(K+1)(R+1)+\frac{R*(R+1)+2}{2}}} V_1^i & \text{Si } j=0 \\ \frac{1-V_1}{1-V_1^{(K-i+1)((R+1)*\frac{R(R+1)}{2})}} V_1^i & \text{Si } 0 < j < K \\ \frac{1-\lambda_2}{1-V_1^{(R+1)*(\frac{R+2}{2})}} V_1^i & \text{Si } j=K \end{cases}$$

Avec : $V_1 = \frac{\lambda_1}{\mu}$

1. Puis, on calcule $q(\langle j \rangle, \langle i \rangle)$ désigne la probabilité de passage de l'espace d'états S_K^j au S_K^i , elle est donné par :

$$q(\langle j \rangle, \langle i \rangle) = \begin{cases} \lambda_2(1-p_i(K-i)) & \text{si } i=j+1 \end{cases}$$

Chapitre 3 : Modélisation des réseaux de capteurs.

$$\mu (1-\rho_0(i)) \quad \text{si } i=j-1$$

2. $\pi_2 \langle n_2 \rangle$ est la probabilité de l'état fusionné, décrit par :

$$\pi_2 \langle n_2 \rangle = \prod_{i=0}^{K-1} \frac{q(\langle i \rangle, \langle i+1 \rangle)}{q(\langle i+1 \rangle, \langle i \rangle)} * \pi_2 \langle 0 \rangle \quad (n_2 = 1, \dots, K)$$

On obtient la valeur de $\pi_2 \langle 0 \rangle$ à partir de la condition de normalisation suivante :

$$\sum_{n_1=0}^K \pi_2(\langle n_1 \rangle) = 1.$$

✓ La chaîne 2D (orbite 1, orbite 2)

Dans ce cas la CMTC à 2D est décomposé par ligne à un ensemble d'espace d'états $S_K^{n_4}$ par rapport le nombre dans le deuxième buffer (voir figure 3.11) :

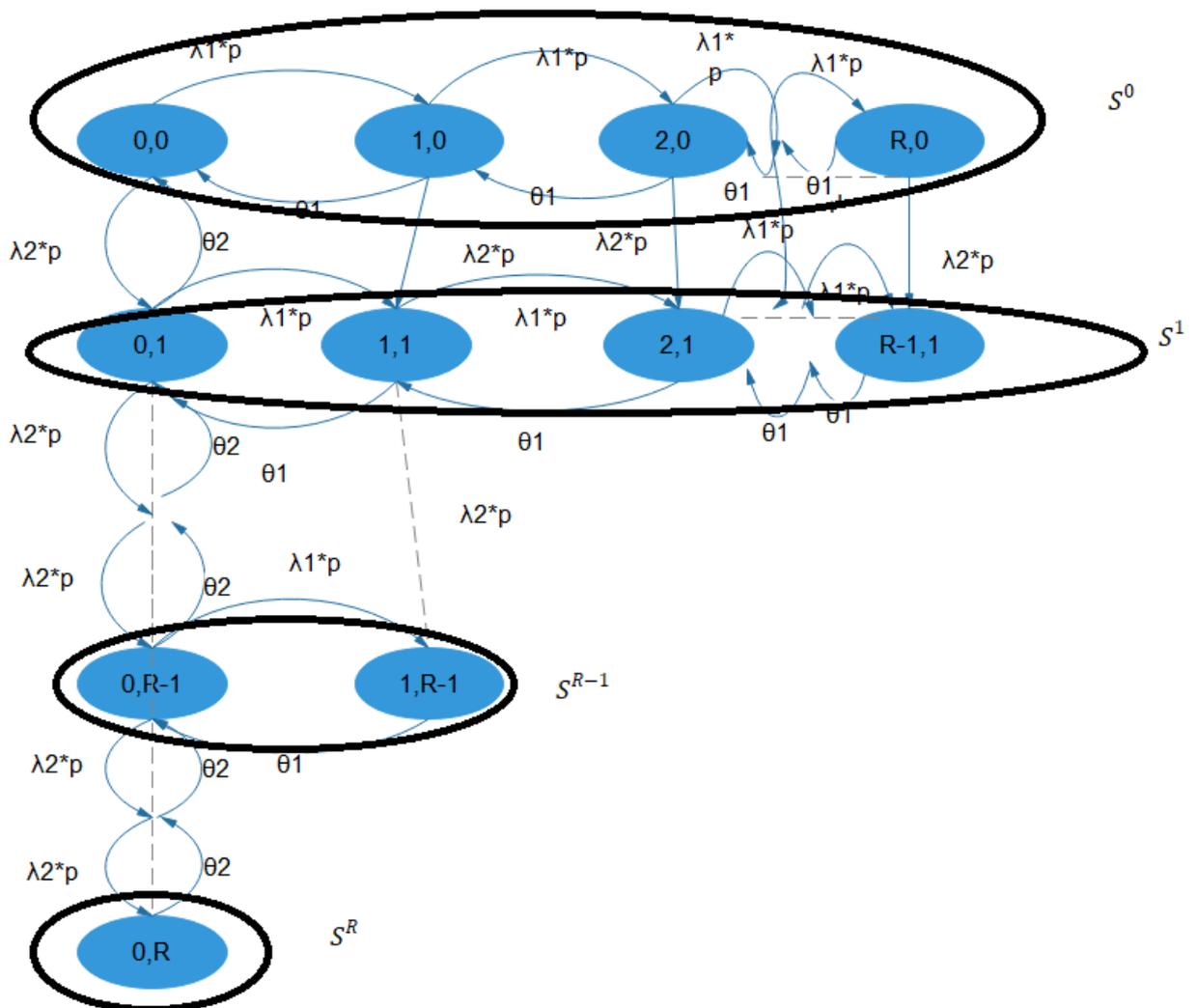


Figure 3.11: la chaîne de Markov à 2D après la décomposition et la division par colonne.

En suivant les mêmes étapes citées dans le premier cas, on obtient les équations suivantes :

$$1. P_y(x) = \begin{cases} \frac{1-V_2}{1-V_2 \binom{(K+R+1)(K+R+2)-R(R+1)+\frac{K(K+1)R}{2}}{}} V_1^x & \text{si } y=0 \\ \frac{1-V_1}{1-V_1 \binom{(K+1)(K+2)+2}{2}} V_1^x & \text{si } 1 \leq y \leq R \end{cases}$$

Avec : $V_1 = (\lambda_1 * p) / \theta_1$

$$2. q(\langle y \rangle, \langle x \rangle) = \begin{cases} \lambda_2 * p(1-\rho_i - (R-i)) & \text{si } x=y+1 \\ \theta_2 (1-\rho_0(i)) & \text{si } x=y-1 \\ 0 & \text{autrement} \end{cases}$$

$$3. \pi_2 \langle n_4 \rangle = \prod_{i=0}^{K-1} \frac{q(\langle i \rangle, \langle i+1 \rangle)}{q(\langle i+1 \rangle, \langle i \rangle)} * \pi_2 \langle 0 \rangle$$

✓ $q(S_K, S_{K'})$ et $\pi_1 \langle 1 \rangle$ décrivent la chaîne unidimensionnelle lors de passage de l'espace S_K à l'espace $S_{K'}$:

$$q(S_K, S_{K'}) = \begin{cases} \lambda_1 & \text{si } K'=K+1 \\ \mu & \text{si } K'=K-1 \\ 0 & \text{autrement} \end{cases}$$

$$\pi_1 \langle 1 \rangle = \varphi \left(\sum_{i=0}^1 \frac{\varphi^i}{i!} \right) - 1$$

Avec : $\varphi = \lambda_1 / \mu$

On calcule le vecteur de probabilité à l'état stationnaire de la CMTC à 5D à partir de l'équation suivante :

$$P(n_1, n_2, n_3, n_4, n_5) = \rho n_2(n_1) * \pi_2 \langle n_2 \rangle * \rho n_4(n_3) * \pi_2 \langle n_4 \rangle * \pi_1 \langle 1 \rangle$$

3.3.3.3. Les indices de performances du modèle M2

- La durée moyenne d'une période oisive :

$$\bar{T} = \frac{N}{\lambda_1 + \lambda_2}$$

- Le nombre moyen des paquets de haute priorité en attente :

$$\bar{Q}_1 = \sum_{z=0}^1 \sum_{i=1}^K \sum_{x=0}^R i * \pi_{i,0,x,0,z} + \sum_{i=1}^{N-1} i * \pi_{i,0,0,0,1} + \sum_{i=0}^{k-1} \sum_{j=1}^{K-i} \sum_{x=0}^R \sum_{y=1}^{R-x} i * \pi_{i,j,x,y,1} + \sum_{y=0}^R \sum_{x=1}^{R-y} i * \pi_{0,K,x,y,1}$$

- **Le nombre moyen des paquets de faible priorité en attente :**

$$\bar{Q}_2 = \sum_{i=0}^{K-1} \sum_{j=1}^{K-i} \sum_{x=0}^R \sum_{y=1}^{R-x} j * \pi_{i,j,x,y,1} + \sum_{i=0}^{N-2} \sum_{j=1}^{N-1-i} j * \pi_{i,j,0,0,0}$$

- **Le nombre moyen de paquets en attente :**

$$\bar{Q} = \bar{Q}_1 + \bar{Q}_2$$

- **Le nombre moyen des paquets de haute priorité dans l'orbite :**

$$\overline{Q_{orbite1}} = \sum_{i=0}^K \sum_{x=0}^R i * \pi_{i,0,x,0,z} x + \sum_{i=0}^{k-1} \sum_{x=0}^R \sum_{y=1}^{R-x} x * \pi_{i,K-i,x,y,1} + \sum_{y=0}^R \sum_{x=1}^{R-y} x * \pi_{0,K,x,y,1}$$

- **Le nombre moyen des paquets de faible priorité dans l'orbite :**

$$\overline{Q_{orbite2}} = \sum_{i=0}^{k-1} \sum_{x=0}^R \sum_{y=1}^{R-x} y * \pi_{i,K-i,x,y,1} + \sum_{y=0}^R \sum_{x=1}^{R-y} y * \pi_{0,K,x,y,1}$$

- **Le nombre moyen de paquets dans l'orbite :**

$$\overline{Q_{orbite}} = \overline{Q_{orbite1}} + \overline{Q_{orbite2}}$$

- **La durée moyenne d'une période occupée :**

$$\bar{B} = \frac{\bar{Q}}{\mu}$$

- **La durée moyenne d'une période du cycle :**

$$\bar{C} = \bar{I} + \bar{B}$$

- **La probabilité d'état oisif :**

$$p_i = \sum_{i=0}^{N-1} \pi_{i,0,0,0,0} + \sum_{i=0}^{N-1} \sum_{j=1}^{N-i-1} \pi_{i,j,0,0,0}$$

- **La probabilité d'état occupé :**

$$p_b = 1 - p_i$$

- **La probabilité de blocage :**

$$p_{blocage} = \sum_{x=0}^R \pi_{K,0,x,0,1} + \sum_{i=0}^{K-1} \sum_{x=0}^R \sum_{y=1}^{R-x} \pi_{i,K-i,x,y,1}$$

- **Le débit de la réception de paquets :**

$$\bar{\lambda} = \bar{\lambda}_1 + \bar{\lambda}_2$$

Tel que :

$$\bar{\lambda}_1 = \lambda_1 \left(\sum_{z=0}^1 \sum_{i=1}^{K-1} \sum_{x=0}^{R-1} \sum_{y=1}^{R-x} i * \pi_{i,0,x,y,z} + \sum_{i=1}^{N-1} i * \pi_{i,0,0,0,1} + \sum_{i=0}^{k-1} \sum_{j=1}^{K-i} \sum_{x=0}^{R-1} \sum_{y=1}^{R-x} i * \pi_{i,j,x,y,1} + \sum_{i=0}^{N-2} \sum_{j=1}^{N-i-1} i * \pi_{i,j,0,0,0} \right)$$

$$\bar{\lambda}_2 = \lambda_2 \left(\sum_{z=0}^1 \sum_{i=1}^{K-1} \sum_{x=0}^{R-1} \sum_{y=1}^{R-x} j * \pi_{i,0,x,y,z} + \sum_{i=1}^{N-1} j * \pi_{i,0,0,0,1} + \sum_{i=0}^{k-1} \sum_{j=1}^{K-i} \sum_{x=0}^{R-1} \sum_{y=1}^{R-x} j * \pi_{i,j,x,y,1} + \sum_{i=0}^{N-2} \sum_{j=1}^{N-i-1} j * \pi_{i,j,0,0,0} \right)$$

- **La durée moyenne d'attente :**

$$\bar{W} = \bar{W}_1 + \bar{W}_2 = \frac{\bar{Q}_1}{\lambda_1} + \frac{\bar{Q}_2}{\lambda_2}$$

- **L'énergie consommée :**

$$EC = p_i * EC_i + p_b * EC_b + EC_s / \bar{C} + \bar{Q} * EC_h .$$

EC_i : La consommation d'énergie de l'unité de transmission à l'état idle.

EC_b : La consommation d'énergie de l'unité de transmission à l'état busy.

EC_s : La consommation d'énergie du changement de l'état de capteur entre réception et transmission.

EC_h : La consommation d'énergie de transmission des paquets.

3.4. Conclusion

L'objectif de ce chapitre était la présentation d'une approche de modélisation et d'évaluation des performances des réseaux de capteurs sans fil. Nous avons présenté dans un premier temps, une modélisation avec les files d'attente avec vacance, classes de priorité et rappel, ce qui a rendu le modèle plus complexe. Pour l'analyse du modèle à l'état stationnaire, nous avons utilisé le processus sous-jacent des chaînes de Markov, et nous avons développé une méthode numérique qui nous a permis de calculer les probabilités stationnaires, et ensuite les indices de performances les plus importants du réseau.

Puis, afin de valider les résultats obtenus et minimiser le temps d'exécution, nous avons proposé l'adaptation de la méthode approximative pour chacun des modèles proposés pour le calcul des probabilités stationnaires. Les formules des principaux indices de performance, l'énergie consommée et le délai d'attente ont été dérivé.

Dans le prochain chapitre, nous nous intéressons à la mise en œuvre de notre algorithme et de nos modèles et aux résultats expérimentaux qui montent l'impact des différents paramètres des modèles proposés sur les performances du réseau

4. Implémentation et étude expérimentale

4.1. Introduction

L'objectif principal de ce chapitre est l'implémentation d'une application pour l'évaluation des performances des RCSFs en utilisant les files d'attente finies avec vacance, classes de priorités et le rappel afin de réduire la consommation d'énergie dans réseaux de capteurs sans fil.

Cette application implémente les algorithmes proposés dans le chapitre précédant pour la génération automatique du générateur infinitésimal Q de la CMTC sous-jacente aux différents modèles en fonction des paramètres du système. De plus, l'implémentation du calcul du vecteur des probabilités à l'état stationnaire a été implémenté en utilisant la méthode exacte et l'adaptation proposé pour la méthode approximative. Ainsi, les différents indices de performances, la consommation d'énergie et le délai d'attente d'un nœud capteur pour chaque modèle proposé précédemment peuvent être en déduit.

Nous commençons ce chapitre par la présentation des différentes fonctionnalités de notre application. Suivie d'une étude expérimentale pour montrer l'influence de certains paramètres de système comme le taux de réception des paquets, le taux de service et le seuil N sur la consommation d'énergie et le délai d'attente d'un nœud capteur. De plus, une étude comparative entre les méthodes exacte et approximative pour le calcul des probabilités stationnaires sera présentée. Nous terminons par une conclusion.

4.2. Présentation de l'application

L'implémentation de notre application est réalisée sous le système Windows, en utilisant l'outil C# de Microsoft Visual Studio 2013. Ce dernier est un langage simple, performant, orienté objet, à typage fort et offre des fonctionnalités utiles au développement d'un environnement convivial et interactif.

Notre application contient plusieurs fenêtres et chaque fenêtre a ses propres fonctionnalités la description des fenêtres est la suivante :

4.2.1. Fenêtre d'accueil

La fenêtre d'accueil présenté dans la figure 4.1 est la première fenêtre qui apparait lors du lancement de l'application elle contient deux fonctionnalités :

- **Démarrer** : pour lancer la fenêtre des choix du modèle a évalué. Cette dernière, nous offre la possibilité de choisir un modèle parmi les trois modèles implémentés pour un capteur sans fil.
- **Quitter** : pour quitter l'application.



Figure 4.1: la fenêtre d'accueil de notre application.

La fenêtre du choix du modèle à évaluer est illustrée dans la figure 4.2. Cette fenêtre permet à l'utilisateur de sélectionner un modèle parmi les modèles implémentés et préciser la méthode de calcul des probabilités à l'état stationnaire (méthode exacte ou approximative) pour lancer la fenêtre de choix multiples.

Chapitre 4 : Implémentation et étude expérimentale.



Figure 4.2: Fenêtre du choix du modèle à évaluer.

Cette fenêtre contient deux boutons :

- Retour : pour revenir à la page d'accueil.
- Quitter : pour sortir du système.

4.2.2. Fenêtre de choix multiples

La fenêtre du choix multiple est illustrée dans la figure 4.3, elle donne à l'utilisateur la possibilité de choisir l'un des paramètres afin d'étudier l'influence de la variation de ce dernier sur les indices de performances.

Chapitre 4 : Implémentation et étude expérimentale.

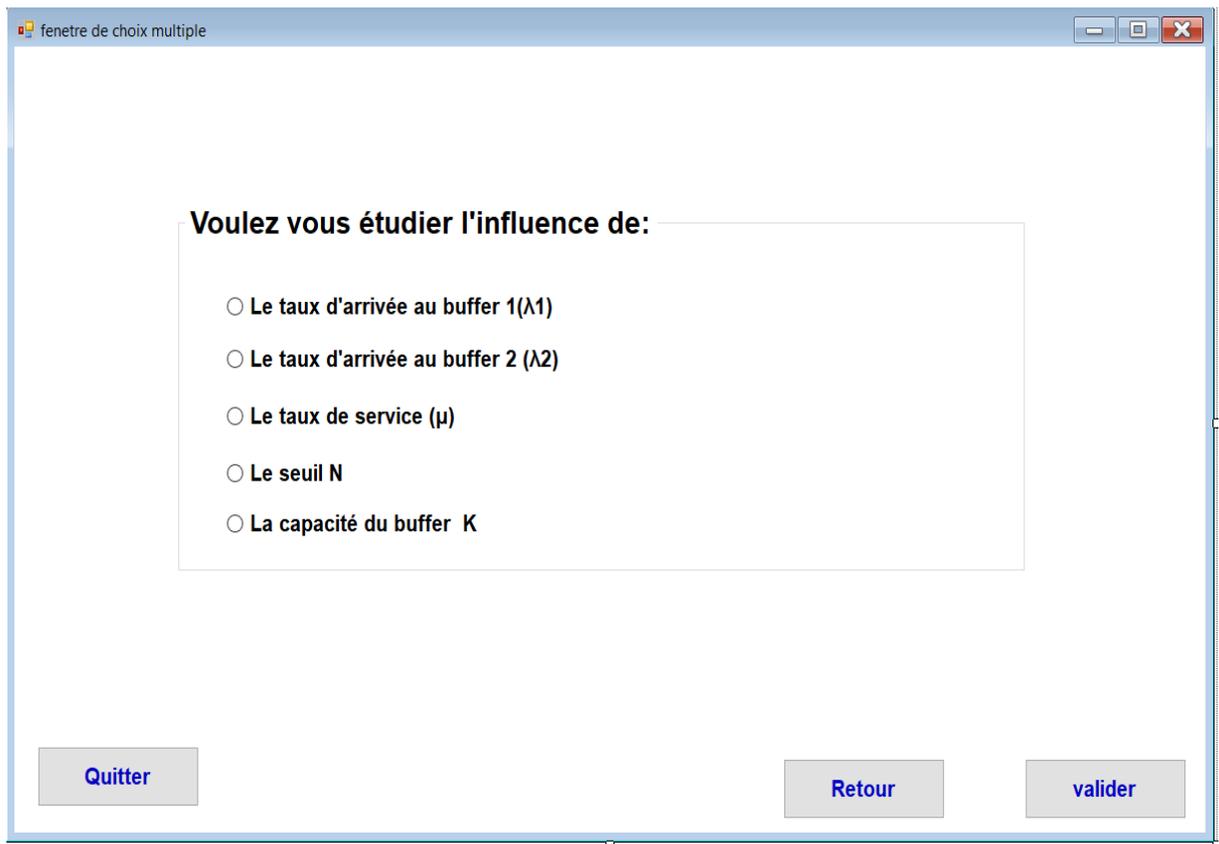


Figure 4.3: fenêtre à choix multiple.

Aussi d'autres boutons :

- ✚ Retour : pour revenir à la page d'accueil.
- ✚ Quitter : pour sortir de l'application.

4.2.3. Fenêtre pour l'évaluation des indices de performances :

L'utilisateur doit choisir le type du nœud de capteur CH ou CM par la sélection de l'un des deux Radio buttons, puis valider le choix par le bouton « choisir ». Il doit remplir les champs vides de la fenêtre (figure 4.4) tel que le taux d'arrivée, le taux de service, le seuil N, la capacité du buffer K, les énergies EC_i , EC_b , EC_s et EC_h , pour obtenir les valeurs des différents indices de performances, ainsi le graphe de la variation de la consommation d'énergie en fonction du paramètre étudié.

Chapitre 4 : Implémentation et étude expérimentale.

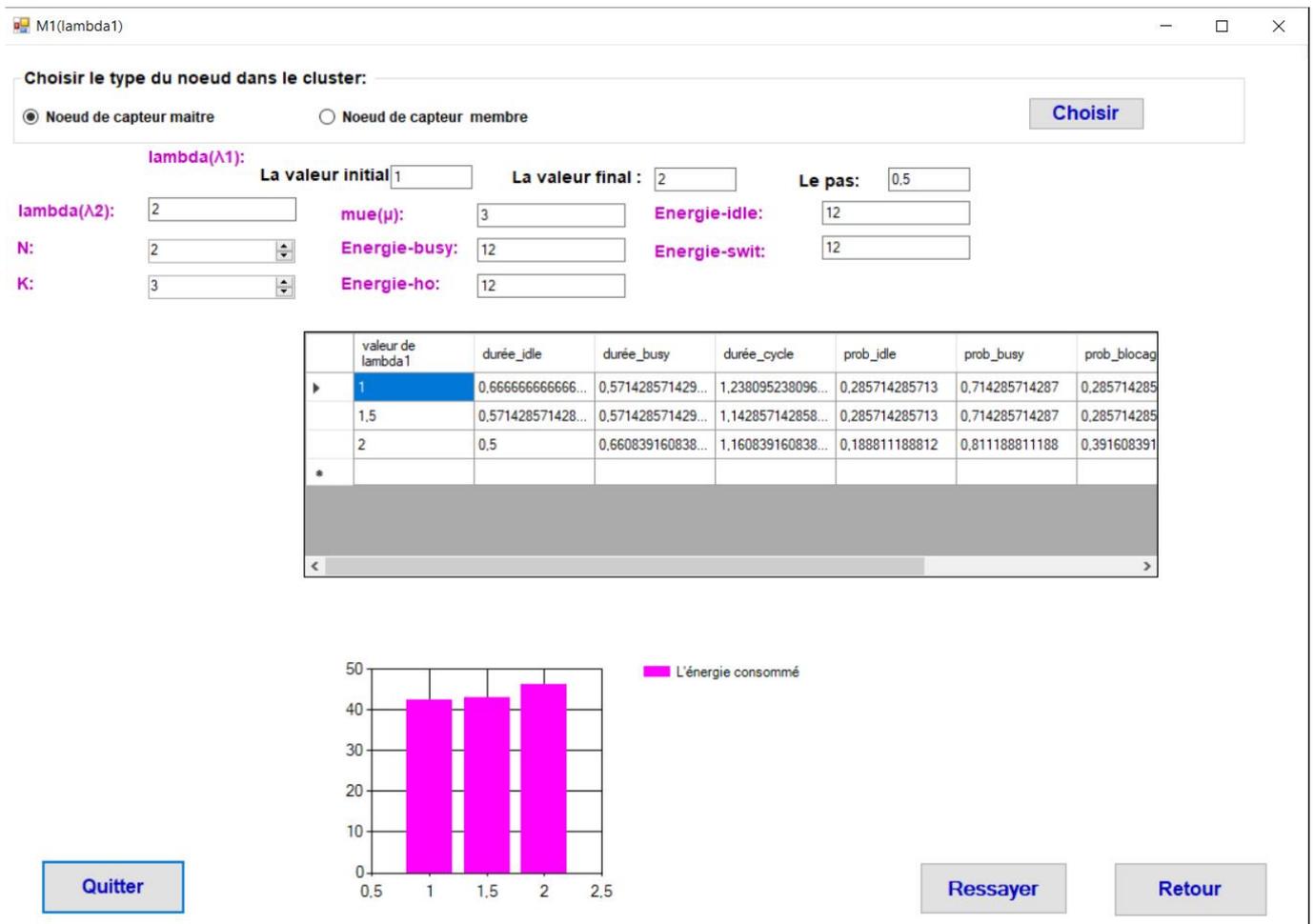


Figure 4.4: la fenêtre des calculs des indices de performances.

Cette fenetre contient les boutons :

- **Ressayer** : pour recommencer une nouvelle étude.
- **Retour** : pour revenir à la page d'accueil.
- **Quitter** : pour sortir de l'application.

4.3. Etude expérimentale

Nous allons étudier l'effet de la variation des paramètres de notre système sur la consommation moyenne d'énergie, la durée moyenne d'attente et la saturation du buffer pour les différents modèles. Les paramètres considérés sont le taux d'arrivée λ , le taux de service μ et le seuil N . Le tableau suivant (voir Tableau 4.1) résume les différentes valeurs utilisées dans l'étude expérimental des paramètres du système.

Chapitre 4 : Implémentation et étude expérimentale.

λ	μ	N	K	R (modèle M2)	$\theta 1$ (modèle M2)	$\theta 2$ (modèle M2)	EC _i	EC _b	EC _s	EC _h
(0.5- 2.5)	2	10	10 (3 pour M2)	6	0.2	0.1	50	500	300	5
1.5	(0.5- 2.5)	2	10 (3 pour M2)	6	0.2	0.1	50	500	300	5
1.5	2	(1-3)	10 (3 pour M2)	6	0.2	0.1	50	500	300	5

Tableau 4-1: les paramètres du système.

4.3.1. Effet de la variation de taux d'arrivée des paquets λ

En premier, nous avons étudié l'impact de variation de taux d'arrivée sur la consommation moyenne de l'énergie, la durée d'attente, la probabilité de blocage, le nombre de paquets dans le buffer et le nombre de paquets en attente.

- La consommation moyenne d'énergie :

La figure 4.5 ci-dessous représente l'influence de la variation de taux d'arrivée des paquets λ sur la consommation moyenne de l'énergie dans les trois modèles proposés.

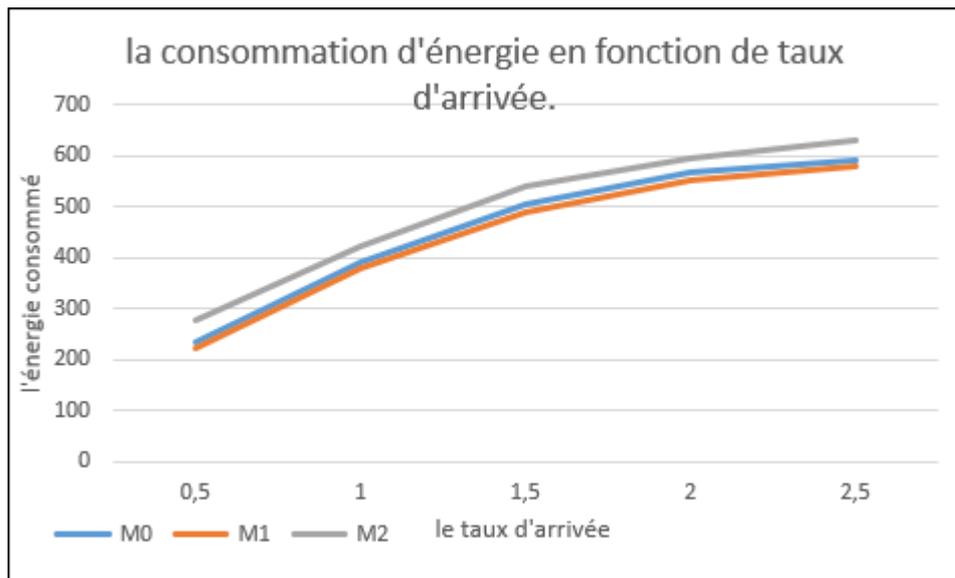


Figure 4.5 : La consommation moyenne d'énergie en fonction du taux d'arrivé.

Analyse : on remarque que la consommation d'énergie dans le modèle M0 est similaire à celle du M1 pour les mêmes valeurs de N et K, en la comparant par le modèle M2. L'augmentation de taux d'arrivée dû à l'augmentation de l'énergie consommé car plus le taux d'arrivée augmente plus le nœud consomme ses ressources en énergie pour traiter les paquets arrivants dans le capteur.

Chapitre 4 : Implémentation et étude expérimentale.

- Le délai d'attente dans le buffer :

La figure 4.6 ci-dessous représente l'influence de la variation du taux d'arrivée λ sur la durée moyenne d'attente dans le buffer pour les trois modèles.

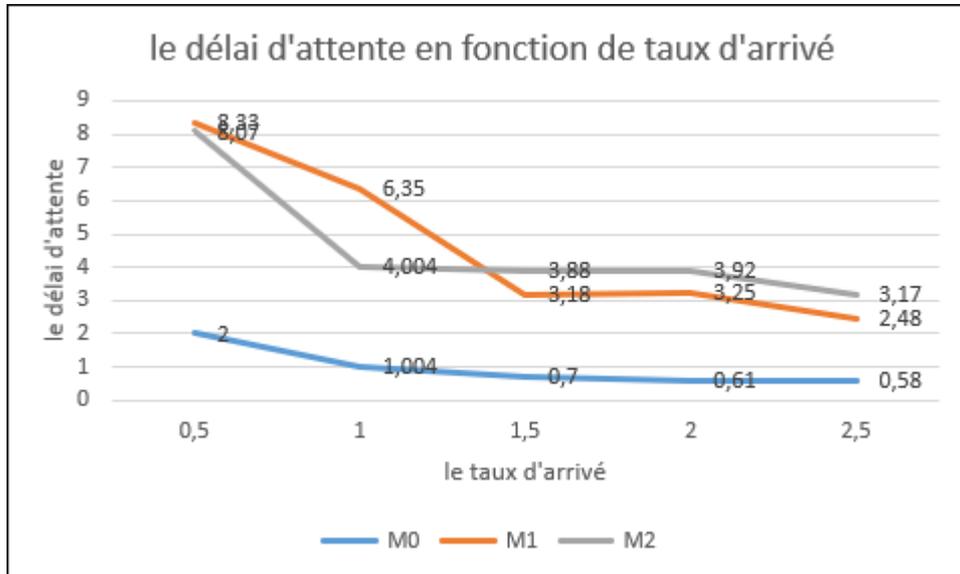


Figure 4.6 : le délai d'attente dans le buffer en fonction du taux d'arrivé.

Analyse : la figure 4.6 illustre que l'augmentation de taux d'arrivée λ diminue le délai d'attente des paquets dans le buffer dans les trois modèles, car le seuil N est atteint avec les valeurs les plus élevées de λ plus rapide que les valeurs inférieures.

- La probabilité de blocage :

La figure 4.7 ci-dessous représente l'influence de la variation du taux d'arrivée λ sur la probabilité de blocage.

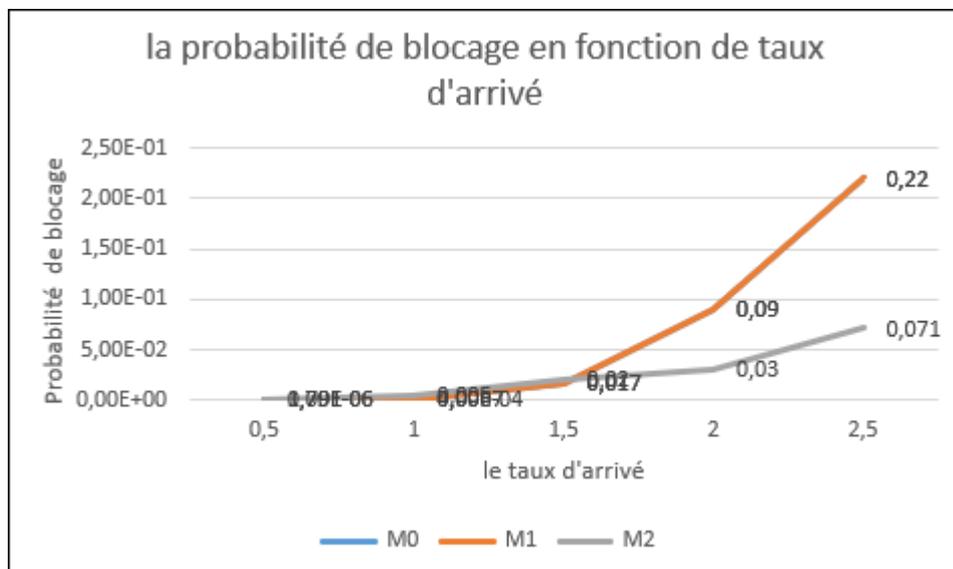


Figure 4.7: la probabilité de blocage en fonction de taux d'arrivé.

Chapitre 4 : Implémentation et étude expérimentale.

Analyse : on déduit que l'augmentation de taux d'arrivée λ augmente la probabilité de blocage des paquets dans les trois modèles, car la capacité K est atteinte rapidement. La probabilité de blocage dans le modèle M2 est inférieure à celle de M1, puisque le modèle M2 traite la retransmission des paquets lorsqu'on atteint la capacité K .

- Le nombre de paquets :

Les figures 4.8 et 4.9 ci-dessous représentent l'influence de la variation de taux d'arrivée λ sur le nombre de paquets dans le buffer pour les trois modèles, ainsi nombre de paquets et la durée d'attente dans l'orbite pour le modèle M2.

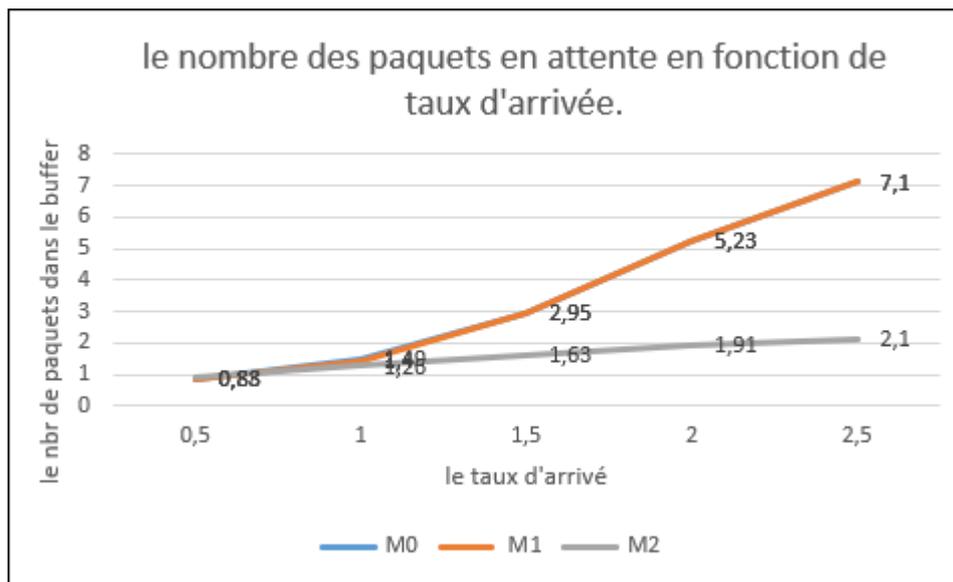


Figure 4.8 : le nombre de paquets dans le buffer en fonction de taux d'arrivée.

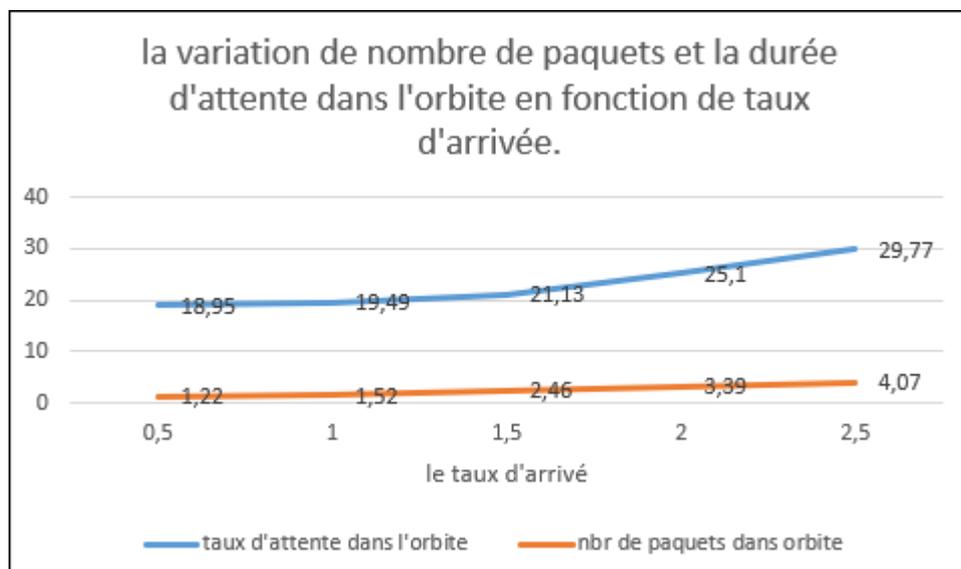


Figure 4.9: la variation de nombre de paquets et le taux d'attente dans l'orbite en fonction de taux d'arrivée.

Chapitre 4 : Implémentation et étude expérimentale.

Analyse : on déduit que l'augmentation de taux d'arrivée λ influe à la fois sur le nombre de paquets dans le buffer pour les trois modèles et sur le nombre de paquets retransmis et le délai d'attente dans l'orbite pour M2 (voir figures 4.8 et 4.9), car l'augmentation de taux d'arrivée des paquets augmente le nombre de paquets dans le buffer et l'orbite.

4.3.2. Effet de la variation de taux de service μ

Dans ce qui suit, nous avons étudié l'impact de variation de taux de service μ sur : la consommation moyenne de l'énergie, la durée d'attente et la probabilité de blocage.

- La consommation d'énergie et la durée d'attente :

Les courbes représentés par les figures 4.10 et 4.11 ci-dessous donnent les résultats de la consommation moyenne d'énergie et la durée moyenne d'attente en fonction du taux de service avec les trois modèles.

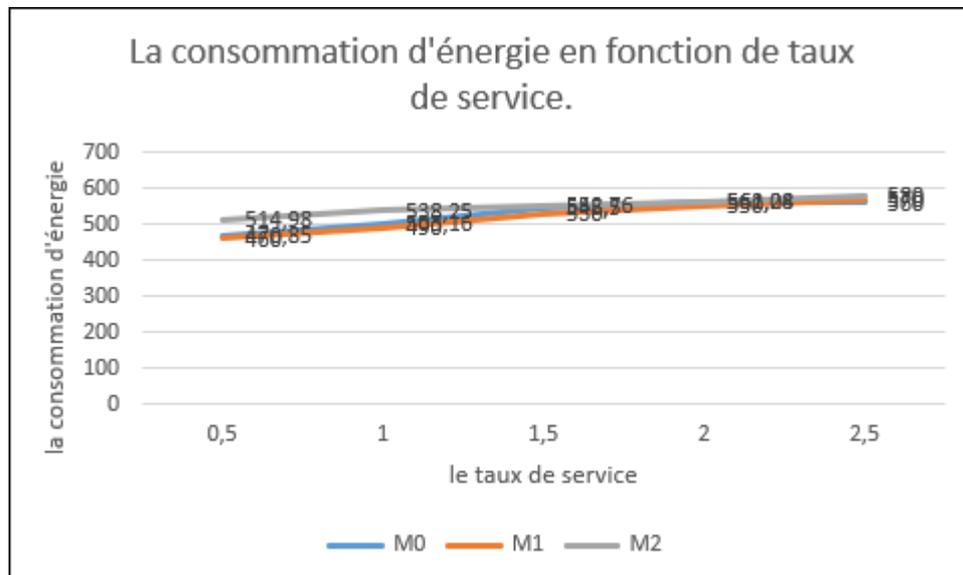


Figure 4.10: la consommation d'énergie en fonction de taux de service.

Chapitre 4 : Implémentation et étude expérimentale.

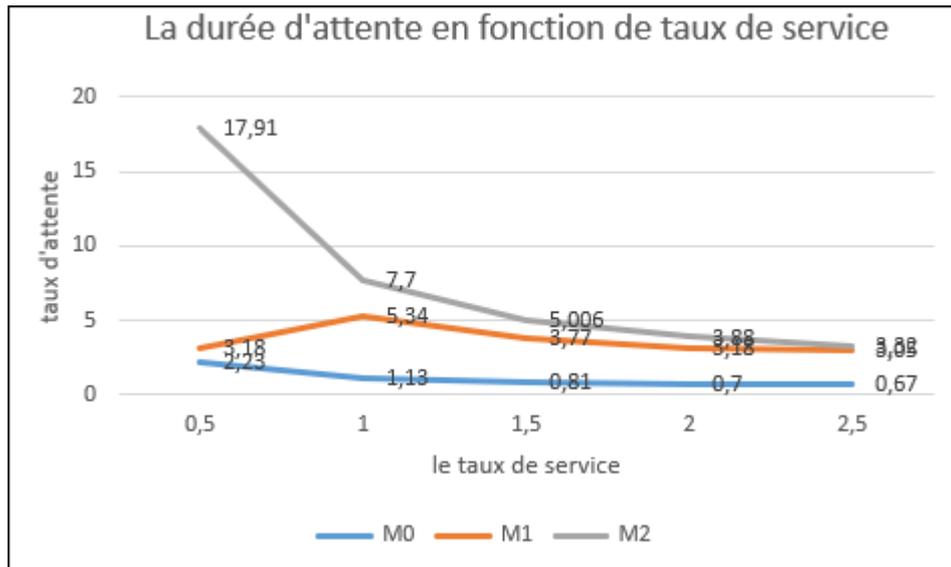


Figure 4.11: le taux d'attente en fonction de taux de service.

Analyse : nous remarquons que la consommation d'énergie est similaire pour les trois les modèles. L'augmentation du taux de service par le nœud capteur influe directement sur la consommation d'énergie et la durée d'attente, car plus le capteur fournit des services plus l'énergie est consommée et plus le délai d'attente diminue.

- La probabilité de blocage :

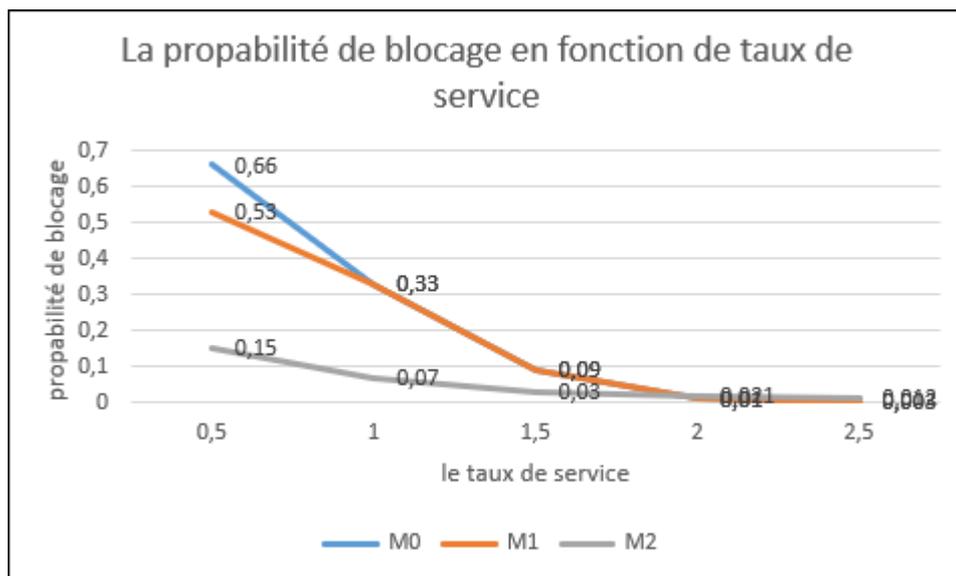


Figure 4.12 : la probabilité de blocage en fonction de taux de service.

Analyse : la figure 4.12 illustre que l'augmentation de taux service μ influe sur la probabilité de blocage dans les trois modèles. L'augmentation de taux de service implique la décroissance de la probabilité de blocage, car les paquets servira rapidement donc on prend plus de temps pour atteint la capacité K du buffer.

4.3.3. Effet de la variation de seuil N

Nous avons évalué l'impact de variation de seuil N sur la consommation moyenne de l'énergie, la durée d'attente et le nombre de paquets dans le buffer.

- La consommation d'énergie et le délai d'attente :

Les courbes représentés par les figures 4.13 et 4.14 ci-dessous donnent les résultats de l'énergie consommée et la durée moyenne d'attente en fonction de taux de service avec les trois modèles.

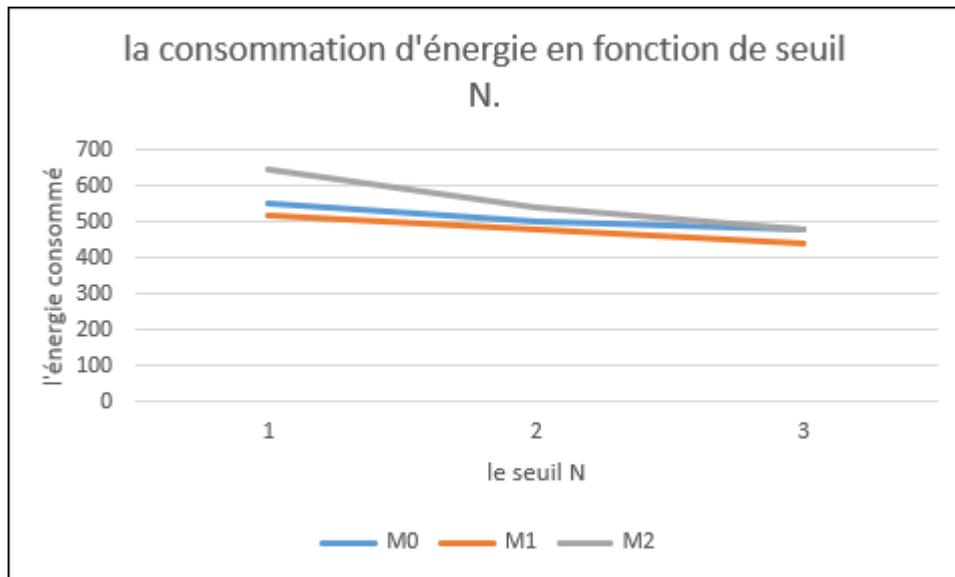


Figure 4.13: la consommation d'énergie en fonction de seuil N.

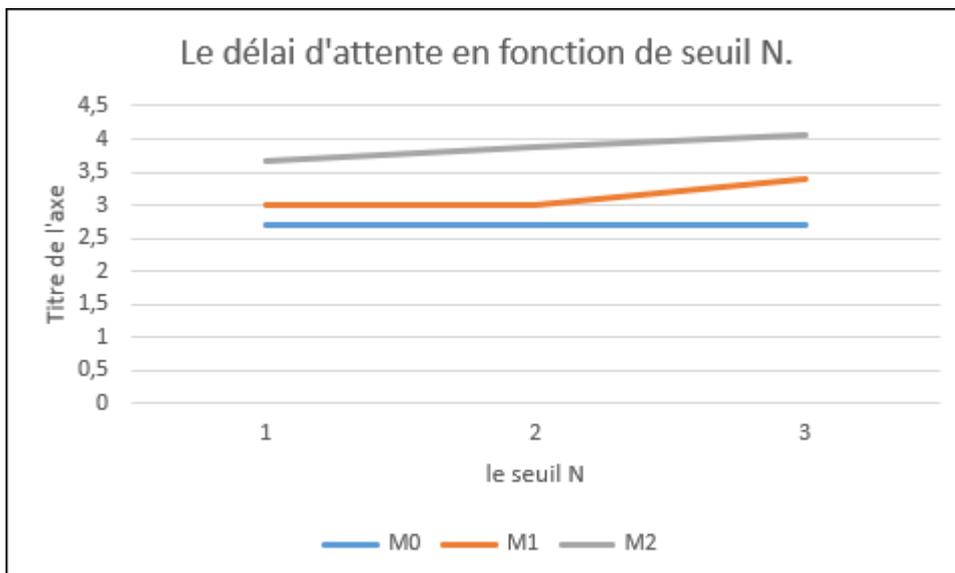


Figure 4.14: le délai d'attente en fonction de seuil N.

Analyse : on constate que la consommation d'énergie diminue dans les trois modèles et elle est similaire dans les deux premiers modèles, le délai d'attente reste stable dans M0 en

Chapitre 4 : Implémentation et étude expérimentale.

augmentant le seuil N. l'augmentation de seuil implique la prolongation de la durée de vacance du capteur, donc la consommation moyenne de l'énergie sera diminuée. Et d'autre part, l'augmentation de nombre de paquets en attente.

- Le nombre de paquets en attente

La figure 4.15 ci-dessous représente l'influence de la variation de seuil N sur le nombre de paquets dans le buffer pour les trois modèles.

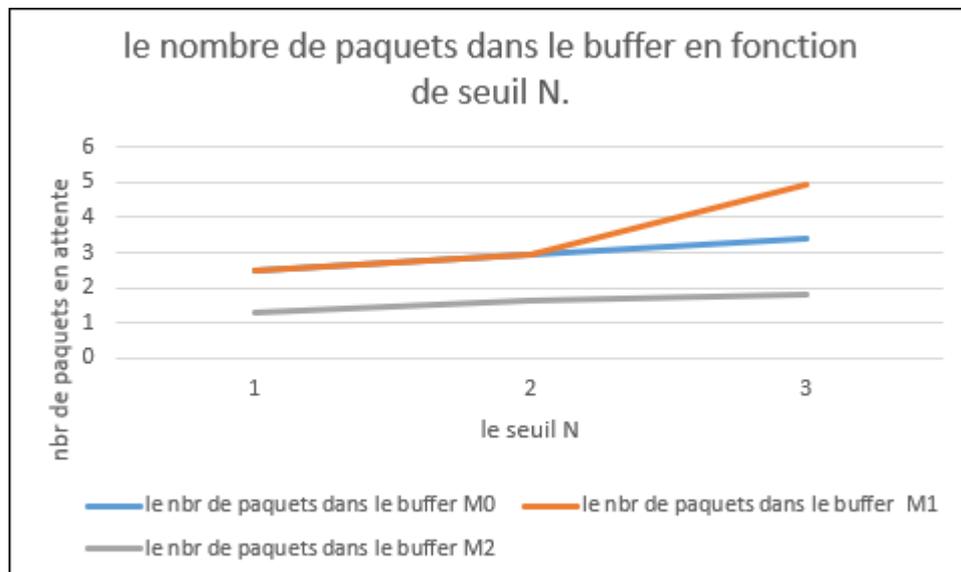


Figure 4.15: le nombre de paquets dans le buffer en fonction de seuil N.

Analyse : on constate que l'augmentation de seuil N implique l'augmentation de nombre de paquets en attente, car les paquets prennent plus de temps dans le buffer avant d'être servis.

4.4. Etude comparative

Afin de faire une étude comparative entre les méthodes exacte et approximative, nous calculons les métriques de comparaison citée dans [30], ainsi le temps d'exécution dans les deux méthodes. Les métriques sont les suivants :

$$\|N\|1 = \max_{n \in E} |p(n) - \tilde{p}(n)|.$$

$$\|N\|2 = \frac{\sum_{n \in E} p(n) \tilde{p}(n)}{(\sum_{n \in E} (p(n))^2)^{\frac{1}{2}} (\sum_{n \in E} (\tilde{p}(n))^2)^{\frac{1}{2}}}$$

$$\|N\|3 = \frac{\sum_{n \in E} \min\{p(n), \tilde{p}(n)\}}{\sum_{n \in E} \max\{p(n), \tilde{p}(n)\}}$$

Ou :

$P(n), n \in E$: représente les probabilités d'état stationnaire obtenues par la méthode exacte.

$\tilde{p}(n), n \in E$: représente les probabilités d'état stationnaires obtenues par la méthode approximative.

Chapitre 4 : Implémentation et étude expérimentale.

D'après Jaccard, on aura une meilleure précision dans le calcul des probabilités à l'état stationnaire si N_1 rapproche de 0, tandis que N_2 et N_3 rapproche de 1.

Les valeurs prise par les paramètres de système sont donnée par le tableau ci- dessous (voir Tableau 4-2)

Variable	λ	μ	N	K	R (modèle M2)	θ_1 (modèle M2)	θ_2 (modèle M2)	EC _i	EC _b	EC _s	EC _h
Valeur	0.5	2	2	3	6	0.2	0.1	50	500	300	5

Tableau 4-2: les paramètres de système de l'étude comparative.

Les résultats des trois modèles sont résumés par le tableau suivant :

	M0	M1	M2
N1	0.1710	0.0321	0.2509
N2	0.5613	0.5973	0.4732
N3	0.6901	0.6583	0.5482
Le temps d'exécution de la méthode exacte (s)	5.9947908	5.091618	56.2539731
Le temps d'exécution de la méthode approximative (s)	4.4987422	4.9235792	9.5368265

Tableau 4-3: les résultats comparatifs.

L'étude comparative montre la précision de notre adaptation de la méthode approximative vue que les résultats de N_1 , N_2 et N_3 sont correcte.

La méthode approximative donne des résultats durant un temps d'exécution très raisonnable par apport à l'exacte, vu que cette méthode se base sur des formules mathématiques on peut avoir des exceptions dans les paramètres d'entrée (λ_1 , λ_2) parce que on aura une division par 0. Afin de résoudre ces exceptions on a défini un domaine de définition pour éviter ce conflit qui est $\lambda_1, \lambda_2 \neq 1$ dans le modèle M1 et $\lambda_1, \lambda_2 \neq 1,2$ dans le modèle M2.

4.5. Conclusion

Dans ce chapitre nous avons implémenté notre approche qui est basée sur trois différents algorithmes, pour but de minimiser la consommation d'énergie dans les RCSFs et évaluer les performances de ces réseaux. Ainsi, nous avons adapté la méthode approximative sur les modèles étudiés.

Tout d'abord nous avons présenté notre application avec les différentes fonctionnalités des fenêtres. Puis on a entamé une étude expérimentale qui nous a permis de voir l'influence de la variation de certains paramètres comme le taux d'attente des paquets, le taux de service et le seuil N sur la consommation d'énergie et le délai d'attente d'un nœud capteur.

A la fin une étude comparative a été faite entre la méthode exacte et approximative pour déduire la précision qui montre l'efficacité de notre adaptation approximative.

Chapitre 4 : Implémentation et étude expérimentale.

Conclusion générale

Vu la révolution qu'a subi la recherche dans le domaine des réseaux de capteurs, ces derniers ont ouvert de nouvelles perspectives d'impacts significatifs dans de nombreux domaines d'applications (sécurité, santé, environnement, sécurité alimentaire, télécommunications, robotique, etc.). Dans ce rapport nous nous sommes intéressés, en particulier, aux réseaux de capteurs sans fil.

Les RCSFs présentent certaine faiblesse de point de vue défis de conception compte tenu de leurs capacités de calcul limitée et leurs batteries irremplaçables et surtout leurs stockages restreints. Le fameux accroc dans la conception des RCSFs est la consommation énergétique qui présente une garantie pour un bon fonctionnement d'un tel réseau. De ce fait, l'objectif de notre travail est de minimiser la consommation d'énergie dans les RCSFs et d'évaluer les performances de ces derniers avec deux méthodes exacte et approximative et distinguer à la fin laquelle de ces méthodes est-elle la plus efficace. Pour ce faire nous proposons un formalisme mathématique basé sur les files d'attente avec vacance, rappel et priorité en développant un algorithme qui permet de générer le générateur infinitésimal de la CMTC dans chaque modèle étudié.

Dans le premier chapitre, nous avons présenté une description synthétique des réseaux de capteurs, leurs architectures, leurs caractéristiques et contraintes ainsi que leurs domaines d'applications.

En effet, dans le deuxième chapitre nous avons cités les méthodes formelles utilisées pour évaluer notre système en citant les types des files d'attente avec vacance, rappel et priorité, les chaînes de Markov à temps continu et la description de la méthode approximative.

Le troisième chapitre a été consacré à présentons les travaux qui concernent l'utilisation des méthodes formels pour l'évaluation des performances des RCSFs. Par la suite, une description détaillée des différents modèles proposés, la solution exacte et la solution approximative.

Dans le dernier chapitre nous avons présenté les différentes fonctionnalités des interfaces de notre application, Nous enchainons la fin par une étude expérimentale illustrant l'influence de certains paramètres du système sur les performances des réseaux de capteurs sans fil. Ainsi une étude comparative entre la méthode exacte et approximative qu'il nous a permis de déduire que le premier modèle sert à minimiser la consommation d'énergie dans les RCSFs, Le deuxième modèle diminue la durée d'attente dans le buffer du nœud et le troisième modèle

Enfin comme perspective à ce travail, il serait intéressant d'appliquer notre approche dans plusieurs domaines différents et de proposer d'autres méthodes formelles afin d'évaluer les performances des nœuds de de capteurs et les rendre plus efficaces.

Bibliographie

- [1] Carneiro, G. (2010, April). NS-3: Network simulator 3. In *UTM Lab Meeting April* (Vol. 20, pp. 4-5).
- [2] Varga, A., & Hornig, R. (2008, March). An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops* (p. 60). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [3] Chang, X. (1999, December). Network simulations with OPNET. In *WSC'99. 1999 Winter Simulation Conference Proceedings. 'Simulation-A Bridge to the Future'* (Cat. No. 99CH37038) (Vol. 1, pp. 307-314). IEEE.
- [4] Maheswar, R., & Jayaparvathy, R. (2011). Optimal power control technique for a wireless sensor node: a new approach. *International Journal of Computer and Electrical Engineering*, 3(1), 1793-8163.
- [5] Lim, D. E., Lee, D. H., Yang, W. S., & Chae, K. C. (2013). Analysis of the GI/Geo/1 queue with N-policy. *Applied Mathematical Modelling*, 37(7), 4643-4652.
- [6] Huang, D. C., Tseng, H. C., Deng, D. J., & Chao, H. C. (2012). A queue-based prolong lifetime methods for wireless sensor node. *Computer Communications*, 35(9), 1098-1106.
- [7] Boutoumi, B., & Gharbi, N. (2017, April). An energy saving and latency delay efficiency scheme for wireless sensor networks based on GSPNs. In *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 0645-0650). IEEE.
- [8] Boutoumi, B., & Gharbi, N. (2018, July). Two Thresholds Working Vacation Policy for Improving Energy Consumption and Latency in WSNs. In *International Conference on Queueing Theory and Network Applications* (pp. 168-181). Springer, Cham.
- [9] Jayarajan, P., Maheswar, R., & Kanagachidambaresan, G. R. (2017). Modified energy minimization scheme using queue threshold based on priority queueing model. *Cluster Computing*, 1-8.
- [10] Nageswari, D., Maheswar, R., & Kanagachidambaresan, G. R. (2018). Performance analysis of cluster based homogeneous sensor network using energy efficient N-policy (EENP) model. *Cluster Computing*, 1-8.
- [11] Choi, B. D., Melikov, A., & Velibekov, A. (2008). A simple numerical approximation of joint probabilities of calls in service and calls in the retrial group in a picocell. *Appl. Comput. Math*, 7(1), 21-30.
- [12] Melikov, A. Z., Ponomarenko, L. A., & Rustamov, A. M. (2016). Hierarchical space merging algorithm for the analysis of open tandem queueing networks. *Cybernetics and Systems Analysis*, 52(6), 867-877.
- [13] Dessales, D. (2011). Conception d'un réseau de capteurs sans fil, faible consommation, dédié au diagnostic in-situ des performances des bâtiments en exploitation (Doctoral dissertation, Poitiers).
- [14] Randrianarisaina, A. (2015). Modélisation de la consommation d'énergie En vue de la conception conjointe (matériel/logiciel) des applications embarquées. Application aux réseaux de capteurs sans fil (wsn) (Doctoral dissertation, UNIVERSITE DE NANTES).

- [15] Jorio, A. (2015). Le Clustering basé sur la Classification Spectrale pour l'Optimisation d'Energie dans les Réseaux de Capteurs Sans Fil Homogènes.
- [16] Sofiane, M. O. A. D. (2007). Optimisation de la consommation d'énergie dans les réseaux de capteurs sans fil. Master en informatique, Université: IFSIC-Rennes1, 2008.
- [17] BOUAZZI, I. (2018). Optimisation d'accès au medium afin d'économiser de l'énergie dans les réseaux de capteurs sans fils(Doctoral dissertation, Université de Monastir).
- [18] Yousef, Y. (2010). Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil (Doctoral dissertation, Mulhouse).
- [19] Boucheneb, S., Tahir, O., & Nafi, M. (2016). Gestion de clés basée sur des clusters dans les réseaux de capteurs sans fil(Doctoral dissertation, Université A/Mira de Bejaia).
- [20] Bouatit, M. N. (2018). Optimisation du routage, tolérance aux pannes et gestion de l'énergie et de l'interférence pour la transmission des flux multimédia temps réel dans les réseaux de capteur sans fil (Doctoral dissertation, Conservatoire national des arts et metiers-CNAM).
- [21] Bazizi, L., Touazi, A., & Abbas, K. (2017). Développement de quelques algorithmes d'analyse numérique des files d'attente (Doctoral dissertation, Université Abderrahmane Mira).
- [22] Belkacemi, H., Bouzidi, K., & Lekadir, O. (2018). Analyse et simulation de la communication radio-fréquence dans les réseaux de capteurs via les d'attente avec priorité (Doctoral dissertation, Université abderrahmane mira).
- [23] Vergne, A., & Comte, C. (2018). Files d'attente.
- [24] Naishuo, T., & Zhang, Z. G. (2006). Vacation Queueing Models: Theory and Applications.
- [25] Benaïm, M., & El Karoui, N. (2005). Promenade aléatoire: chaînes de Markov et simulations: martingales et stratégies. Editions Ecole Polytechnique.
- [26] Ndiaye, M. A. (2017). Évaluation de performance d'architecture de contrôle-commande en réseau dans un contexte incertain d'avant-vente (Doctoral dissertation, Université de Lorraine).
- [27] Solaiman, B. (2006). Processus stochastiques pour l'ingénieur. PPUR presses polytechniques.
- [28] Valette, R. (2000). Les réseaux de Petri. LAASCNRS Toulouse, Septembre.
- [29] Vergne, A., & Comte, C. (2018). Files d'attente.
- [30] Jaccard P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin del la Société Vaudoise des Sciences Naturelles. 1901. Vol. 37. P. 547-579.