

Republique Algérienne Démocratique et Populaire
Ministère de L'enseignement Supérieur et de la Recherche
Scientifique
Université Saad Dahleb Blida
Faculté Des Sciences
Département D'informatique



Mémoire Présenté en vue de l'obtention du diplôme de Master.

En : INFORMATIQUE

Spécialité : SIR

Thème :

L'impact de la retransmission des paquets sur la
politique de conservation d'énergie dans les réseaux de
capteurs sans fil.

Présenté par :
TOUAHRI ZINEB
ALLAL IBTISSEM

Sous la direction de :
Mme Boutoumi Bachira
président jury : Derraf. H

Année Universitaire 2017/2018

Remerciements

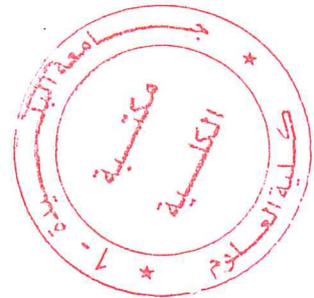
Tout d'abord, nous remercions Dieu le tout puissant qui nous à donné la force et la volonté pour réaliser ce modeste travail.

De suite, Un grand remerciement à madame Boutoumi Bachira, l'encadreuse de notre projet , pour sa disponibilité, ses orientations, ses précieux conseils et pour le temps qu'elle a bien voulu nous consacrer.

Nous exprimons notre profond respect aux membres de jury pour l'honneur d'examiner notre mémoire et l'intérêt qui lui ont porté.

Nos vifs remerciements sont adressés à nos parents pour leurs sacrifices, leurs encouragements et leurs patiences au cours de nos longues années d'études

Finalement nous remercions les gens qui nous ont aidés .





Dédicaces

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, a me donné le courage de réaliser mon rêve « el hamedolilah »

Je dédie ce travail à ma petite famille à mon père ma mère mon frère et ma soeur.

A tout mes chers amis : imane, sarah, zineb, sabrina,meriem benaziza a mon binôme ibtisssem .A ma copine Sofia, ma copine soumia et son marie Oussama .

A tout les personnes qui me souhaite cette succès.

TOUAHRI ZINEB.

Dédicaces

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, à me donné le courage et la santé " el hamedolilah "

je dédie ce travail à tout ma famille à mon père ma mère mon frère Walid et mes soeurs Sabrina Inesse.Amira et son marie Mohammed.

A tout mes chers amisà mon binome Zineb

à mon ami mourad que je le virement remerci à ces aide.

A tout les personnes qui me souhaite cette succès.

ALLAL IBTISSEM



Résumé

Les réseaux de capteurs sans fil (RCSF) sont composés de dispositifs électroniques conçus pour la collecte des grandeurs physiques de l'environnement dans lesquelles ils sont déployés. Ces derniers ont fait naître d'énorme problème de recherche, par les applications qu'ils permettent d'entrevoir et les nombreuses contraintes qu'ils supposent, 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 préserver l'énergie au niveau des nœuds capteurs dû à la fréquence de transition entre l'état oisif et l'état occupé, la notion de vacance issue des modèles de file d'attente avec vacance a été introduit dans le comportement de ces miniaturisé dispositif sans fil cependant, aucun des travaux n'a étudié l'impact de la retransmission des paquets sur cette amélioration. Dans le présent projet de fin d'étude nous proposons d'étudier l'impact de la retransmission des paquets sur la politique de conservation d'énergie dans les réseaux de capteurs sans fil nous allons travailler avec la notion de file d'attente avec rappel et avec vacance. le travail est modélisé par les chaînes de Markov à temps continus (CMTC) (des modèles N-policy avec et sans rappel) après l'analyse de ces derniers l'algorithme de construction du générateur infinitésimal est déduit. à partir de chaque modèle proposé les formules des principaux indices de performances sont déduites pour l'évaluation des performances d'un capteur sans fil.

Mots clés :

Réseaux de capteurs sans fil, file d'attente avec vacance, CMTC, évaluation des performances, N-Policy (Nvacance)

Abstract

The wireless sensor networks (WSN) are composed of electronic devices conceived to collect physical quantities of the environment in which they are spread. These latter have created enormous research problems, by allowing to foresee applications and supposing many constraints, the energy conservation in these networks is very important. The sensors that we consider are fed by non-refillable batteries. So, the conservation of this resource very limited at the sensor knots influences directly the life expectancy of the WSN. In order to preserve the energy at the sensor knots due to transition frequency between the idle state and the busy state, the notion of vacancy stemming from waiting line with vacancy models was introduced into the behavior of these miniature wireless device. However, none of the previous works has studied the impact of broadcasting packages on this improvement. In the present graduation project, we propose to study the impact of broadcasting packages on the conservation policy of energy in the wireless sensor networks. We are going to work with the waiting line with recall and with vacancy notion. The work is modelled by continuous time Markov channels (CTMC) (N-policy models with and without recall) after the analysis of this latter, the construction algorithm of the infinitesimal generator is deducted. From every proposed model the formulae of the performances main indications are deducted to evaluate the wireless sensor performances.

Keywords :

wireless sensor networks, queue with vacancy, CTMC, evaluation of the performances, N-Policy (N-vacancy)



Table des matières

Introduction Générale	1
1 Généralités sur les réseaux de capteurs sans fils	3
1.1 Introduction	3
1.2 Capteur sans fil	4
1.3 Architecture physique d'un capteur	4
1.4 Réseau sans fil	5
1.5 Réseau de capteur sans fil	6
1.6 Architecteur du réseau de capteur sans fil (RCSF)	6
1.7 Architecture protocolaire	7
1.8 Technologies des réseaux de capteurs	9
1.9 Domaines d'application des réseaux de capteurs	10
1.10 Facteurs et contraintes des RCSF	12
1.11 Caractéristiques des RCSF	15
1.12 Les principaux protocoles de routage dans les RCSF	16
1.13 Problèmes des réseaux de capteurs sans fils	19
1.14 Techniques de minimisation de la consommation d'énergie	21
2 Généralités sur les chaines de Markov	25
2.1 Introduction	25
2.2 Définition de chaine Markov	25
2.3 Définition d'un variable aléatoire	26
2.4 Loi exponentielle	26
2.5 Processus stochastique	27

2.6	Processus de Markov	27
2.7	Chaînes de Markov à temps discret	27
2.8	La chaîne de Markov a temps continu (CMTC)	29
2.9	Représentation d'une CMTC	30
2.10	Régime stationnaire	31
2.11	Les processus stochastique a temps continu	32
2.12	Conclusion	34
3	les files d'attente avec vacance et les files d'attente avec rappel	37
3.1	Introduction	37
3.2	La file d'attente classique	37
3.3	Classification des systèmes d'attente	38
3.4	Notation de Kendall	39
3.5	La discipline d'attente	39
3.6	Caractéristiques d'un système de files d'attente	40
3.7	Types de modèles	40
3.8	Analyse mathématique d'un système de files d'attente	42
3.9	Les file d'attente avec vacances	42
3.10	Systèmes d'attente avec rappels	43
3.11	Description du modèle d'attente avec rappels	43
3.12	les Politique FAR	44
3.13	Les travaux connexes	45
3.14	Conclusion	48
4	Modélisation et analyse des réseaux capteur sans fils	49
4.1	Introduction	49
4.2	Modélisation d'une RCSF	49

Table des matières	xiii
4.3 Les indices de performance	68
4.4 Conclusion	72
5 Implémentation et étude expérimentale	73
5.1 Introduction	73
5.2 Présentation de l'application	73
5.3 Etude expérimentale	75
Conclusion Générale	79
Bibliographie	82

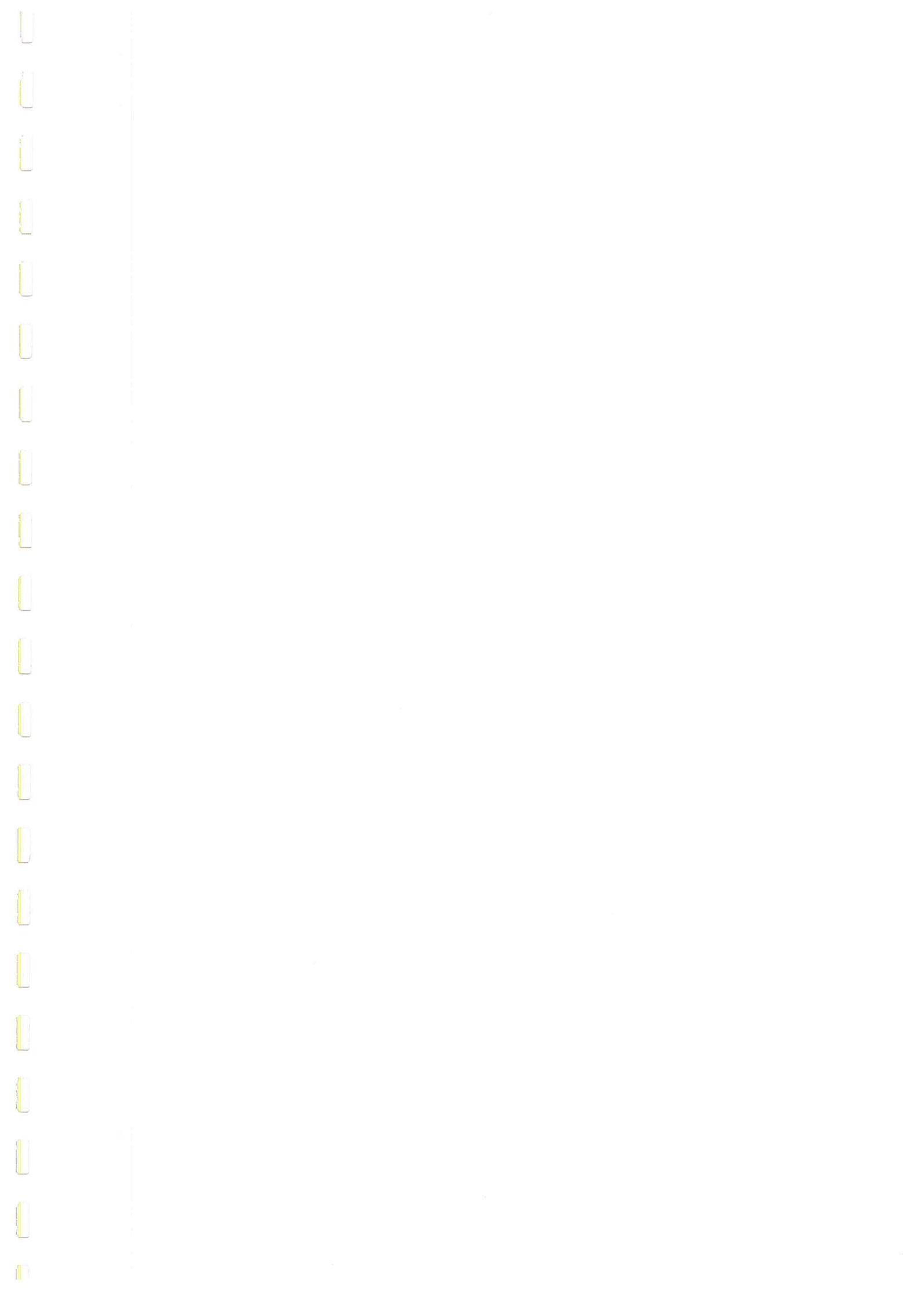
Table des figures

1.1	Exemple d'un capteur sans fil(MICAz 2.4 GHz)[25]	4
1.2	Architecture d'un nœud capteur.[2]	5
1.3	Exemple de réseaux de capteurs sans fil.[You10]	6
1.4	Architecture d'un réseau de capteur sans fil.[20]	7
1.5	la pile protocolaire.[26]	9
1.6	Les domaines d'application des réseaux de capteur sans fils.[27]	12
1.7	Les principaux protocoles de routages dans les RCSF. [You10]	16
1.8	Topologie hiérarchique.[You10]	17
2.1	Exemple d'un Chaîne de markov	26
2.2	Représentation de CMTD par matrice et par graphe	28
2.3	Exemple sur la méthode de calcul d'un vecteur stationnaire[28]	29
2.4	Diagramme de transition de la CMTC	31
2.5	La matrice de transition de la CMTC.	31
2.6	Représentation d'un CMTC avec le processus de poisson.	33
2.7	Graphe et matrice d'un processus de naissance et mort.[28]	34
3.1	système file d'attente.[MAMB+17]	38
3.2	Schéma illustratif d'un système avec rappels.	44
4.1	Diagramme de transition d'état d'un nœud de capteur avec N-policy sans l'état sleep .[BG17]	50
4.2	CMTC général de modèle N-Policy sans l'état sleep	53
4.3	CMTC général de modèle N-Policy avec l'état sleep	57
5.1	Fenêtre d'accueil de l'application	74

5.2	Menu principale des modèles	74
5.3	Interface d'analyse du modèle N-policy avec l'état sleep	75
5.4	La consommation moyenne d'énergie par rapport à seuil N, $\lambda : 0.5 - 1.25$.	76
5.5	La consommation moyenne d'énergie par rapport à le seuil N, $\lambda : 1-1.75$.	77

Liste des tableaux

3.1	Tableau comparatif des différents modèles existants	47
4.1	Représentation des taux de transitions entre les états	51
4.2	Représentation des taux de transitions entre les états	55
4.3	Représentation des taux de transitions entre les états	59
4.4	Représentation des taux de transitions entre les états	63
5.1	Paramètres du système.	76



Liste des abréviations

RCSF : réseaux de capteurs sans fil.
QoS : qualité de service.
WSNs :Wireless sensor networks .
ADC :Analog to Digital Converters
MAC : Media Access Control.
PC : computer personnel.
IEEE :Institut of Electrical and Electronics Engineers.
CSMA/CA :Carrier Sense Multiple-Access /with Collision Avoidance.
TDMA :Time-division multiple access.
TORA :Temporally-Ordered Routing Algorithm routing protocol .
AODV : Ad-hoc On-demand Distance Vector .
DSDV :destination sequenced distance vecteur.
GPS : Global Positioning System.
CMTD :Chaînes de Markov à temps discret.
CMTC :La chaine de Markov a temps continu.
FIFO :first in, first out.
FCFS :first come first served.
LIFO :last in, first out.
LCFS :last come, first served.
PNPN :Priority service.
PS :Processor Sharing.
SIRO :Served In Random Order.
FAR :File D'attente Avec Rappel.
UWB : ultra wideband .
PGF : fonctions de génération de probabilité .

Introduction Générale

En conséquence des progrès de la communication sans fil et de la technologie de l'électronique, les réseaux de capteurs sans fil (RCSFs) ont été un champ de recherche émergeant pour une large gamme d'applications potentielles, telles que la surveillance de l'environnement, la gestion des catastrophes ainsi que la détection des incendies. En raison de leur portabilité et capacité de déploiement, la plupart des nœuds de capteurs de RCSF sont équipés de batteries non rechargeables qui ont une durée de vie limitée. Typiquement, les nœuds de capteur peuvent être déployés dans un environnement distant, sans surveillance avec de grandes quantités. La taille physique d'un nœud de capteur est réduite au minimum pour des missions furtives et des économies de coûts, il est généralement difficile de recharger ou de remplacer leurs batteries. L'un des principaux défis de conception pour les RCSF est d'augmenter la durée de vie opérationnelle des nœuds de capteurs aussi longtemps que possible. [Jia+12b]

Il existe plusieurs recherches dans ce domaine de réseau de capteur, la majorité des chercheurs, sans utiliser la politique N-Policy (N-vacance) pour diminuer la consommation d'énergie dans les RCSFs. mais n'a aucun de ces chercheurs qui a utilisé le principe de retransmission de paquet. Donc de ce projet nous allons toucher cette technique pour voir dans quelle mesure la retransmission des paquets conserve l'énergie dans les réseaux de capteur sans fil.

Pour réaliser notre objectif, nous allons travailler avec les files d'attente avec vacance et les files d'attente avec rappel. Pour le principe de file d'attente avec rappel, lorsque le paquet arrive et trouve le capteur dans l'état sleep, ou le buffer plein soit il quitte le système définitivement, ou bien il atteint l'orbite. Ce dernier qui fait le rappel (retransmettre) des paquets. Nous allons voir est-ce que cette retransmission consomme l'énergie?. Pour pouvoir déduire les indices de performance qui nous aident à réaliser cet objectif. Nous allons proposer des modèles présentés par les CMTC.

Ce rapport est structuré en cinq chapitres comme suit :

- Le premier chapitre est consacré aux réseaux de capteurs sans fils. On décrit en premier lieu, les réseaux de capteurs sans fils, ensuite nous présenterons les domaines d'application de RCSF, les caractéristiques, et les facteurs de RCSF. On verra finalement les problèmes d'un réseau de capteur et les techniques proposées pour résoudre ces problèmes.
- Le second chapitre consiste à présenter les chaînes de Markov qui sont l'outil utilisé pour présenter notre approche de modélisation et d'analyse de performances d'un RCSF. Tout d'abord nous allons voir les chaînes de Markov en général,

leur représentation graphique, et matricielle, nous passerons ensuite à les étudier à temps discret, et à temps continu pour ce dernier nous allons présenter ces processus stochastique.

- Le troisième chapitre est dédié à la théorie des files d'attente avec rappel et avec vacance. Un rappel sur les modèles de file d'attente classiques sera d'abord présenté suivi du modèle de file d'attente avec rappel ainsi que les files d'attente avec vacance et enfin terminé par les travaux connexes.
- Le quatrième chapitre permet de présenter notre approche de modélisation et d'analyse de performance d'un RCSF. Nous montrons les modèles proposés, ainsi que les formules des principaux indices de performances.
- Et dans le cinquième chapitre, nous présenterons notre approche algorithmique pour la modélisation et l'évaluation des performances des systèmes avec vacance, et avec rappel source finie de paquet. Sachant que nous avons implémenté cette approche, en développant un outil en java. Qui se concrétise par une application développée pour tester nos calculs. Nous allons aussi effectuer une étude expérimentale pour établir l'effet de la variation des paramètres.

Nous terminerons notre rapport par une conclusion générale.

Généralités sur les réseaux de capteurs sans fils

1.1 Introduction

Les avancées technologiques récentes confortent la présence de l'informatique et de l'électronique au cœur du monde réel. De plus en plus d'objets se voient ainsi équipés de processeurs et de moyens de communication sans fil et mobiles, leur permettant de traiter des informations mais également de les transmettre. Les réseaux de capteurs sans fil (RCSF) entrent dans ce cadre.[Sah] L'avantage principal de l'utilisation des RCSFs est l'absence de câblage ce qui réduit considérablement le coût d'installation. Un autre avantage est la flexibilité du déploiement et la facilité d'entretien. En effet, les capteurs sont autonomes et nécessitent très peu d'intervention humaine.[BBC11]

Les RCSFs ont suscité beaucoup d'engouement dans la recherche scientifique en raison notamment des nouveaux problèmes de routage sous forte contrainte de durée de vie du réseau et de faibles capacités des nœuds. Malgré la miniaturisation et la réduction du coût de fabrication, ces capteurs sont généralement dotés de ressources limitées en termes de puissance de transmission, de capacité de traitement et de stockage des données et d'énergie.

Dans ce chapitre, nous allons présenter les réseaux de capteurs sans fils avec un plan méthodologique que nous avons adopté. Nous commençons par une définition d'un capteur, son architecture, définition d'un réseau de capteur, ensuite, les domaines d'application, les facteurs, et les caractéristiques d'un RCSF .Finalement nous allons parler des problèmes de ces dernières et les techniques proposées pour les résoudre.

1.2 Capteur sans fil

Un capteur est un petit appareil autonome capable de capter des mesures simples sur son environnement [BB], capable de traiter des informations et de les transmettre, via les ondes radio, sur une distance limitée à quelques mètres.[Abd08]

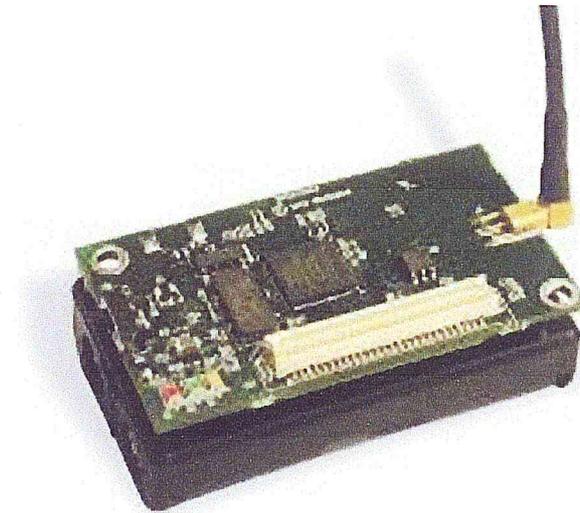


FIGURE 1.1 – Exemple d'un capteur sans fil(MICAz 2.4 GHz)[25]

1.3 Architecture physique d'un capteur

Un nœud capteur sans fil est composé de quatre composants de base (voir figure 1.2) à savoir : l'unité d'acquisition, l'unité de traitement, l'unité de communication et l'unité source d'énergie.

1.3.1 Unité d'acquisition

L'unité d'acquisition comprend deux sous-unités : le capteur et le convertisseur analogique/numérique (ADC pour Analog to Digital Converter). Le capteur est chargé de mesurer une grandeur physique. Le convertisseur analogique/numérique convertit les signaux analogiques en données numériques et les envoie à l'unité de traitement. [Aby16]

1.3.2 Unité de traitement

L'unité de traitement comporte l'unité de stockage et l'unité de calcul généralement un micro-contrôleur. L'unité de stockage permet de stocker les données environnemen-

tales recueillies. Ces données sont ensuite traitées par le microprocesseur.[Aby16]

1.3.3 Unité de communication

Elle a pour fonction de transmettre et recevoir l'information. Elle est équipée d'un couple émetteur/récepteur. Elle permet la communication au sein du réseau, dans le cas qui nous intéresse par radiofréquence (ondes radios). Il existe cependant d'autres possibilités de transmission (optique, infrarouge, etc. . .).[BB]

1.3.4 Unité d'énergie

Batterie : un micro-capteur est muni d'une ressource énergétique pour alimenter tous ses composants. Cependant, en conséquence de sa taille réduite, la ressource énergétique dont il dispose est limitée et généralement irremplaçable. Cette unité peut aussi gérer des systèmes de recharge d'énergie à partir de l'environnement observé telles que les cellules solaires, afin d'étendre la durée de vie totale du réseau.[23]

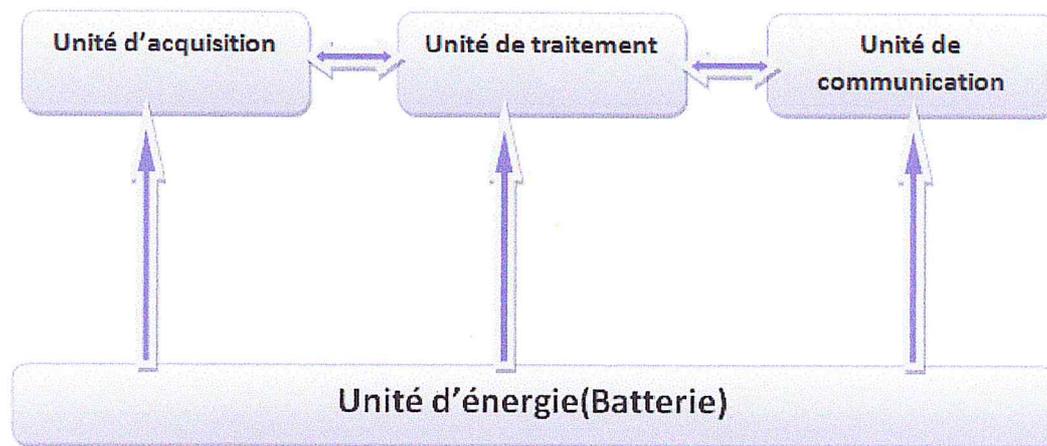


FIGURE 1.2 – Architecture d'un nœud capteur.[2]

1.4 Réseau sans fil

Un réseau sans fil est un réseau qui permet de connecter différents nœuds sans l'aide d'une connexion physique, mais qui établit la communication par des ondes électromagnétiques. La norme la plus utilisée actuellement pour les réseaux sans fils est la norme IEEE802.11.[17]

1.5 Réseau de capteur sans fil

Les réseaux de capteurs sans fils sont forme d'un très grand nombre de capteurs, pour former un réseau sans infrastructure. Un capteur capter son environnement, et propage les données récoltées aux capteurs appartenant à sa zone de couverture. Chaque capteur relayant l'information sur sa propre zone de couverture.[Abd08]

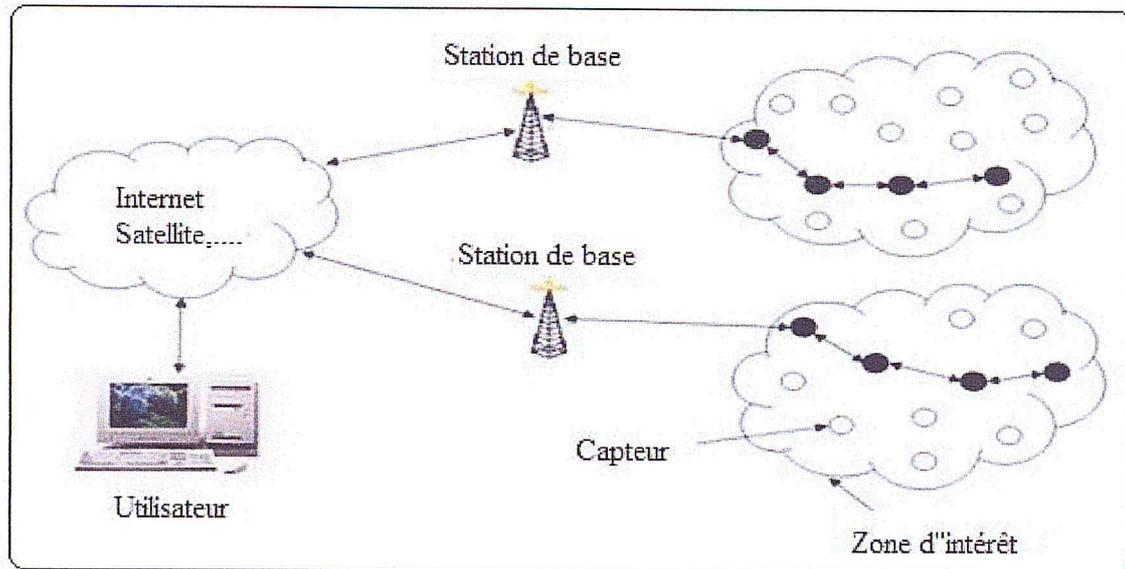


FIGURE 1.3 – Exemple de réseaux de capteurs sans fil.[You10]

1.6 Architecteur du réseau de capteur sans fil (RCSF)

Un RCSF est composé d'un ensemble de nœuds capteurs. Ces nœuds capteurs sont organisés en champs « sensor fields ». Chacun de ces nœuds a la capacité de collecter des données et de les transférer au nœud passerelle (dit "sink" en anglais ou puits) par l'intermédiaire d'une architecture multi-sauts. Le puits transmet ensuite ces données par internet ou par satellite à l'ordinateur central pour analyser ces données et prendre des décisions. [20]

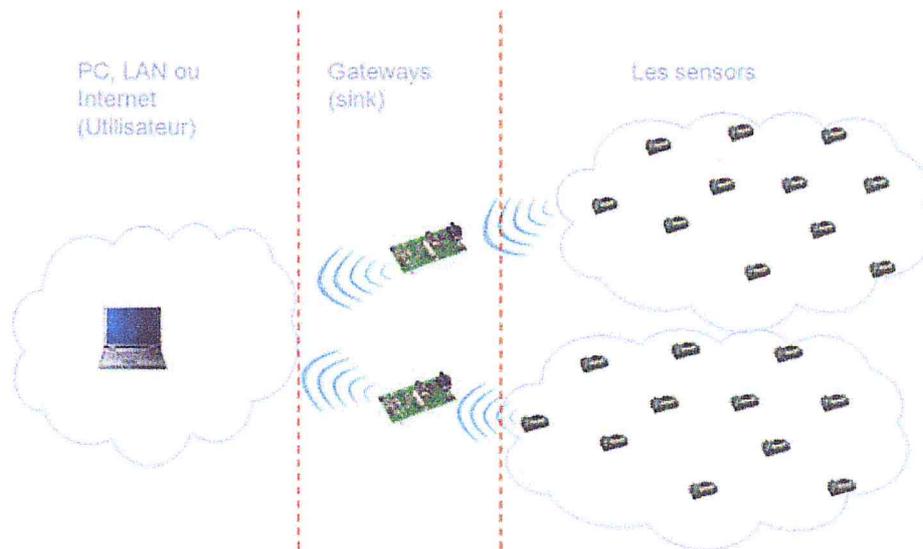


FIGURE 1.4 – Architecture d'un réseau de capteur sans fil.[20]

1.7 Architecture protocolaire

La pile de protocoles utilisée par le puits (Sink) ainsi que par tous les nœuds capteurs. Cette pile de protocoles combine le routage la gestion d'énergie et l'intégrité des les données avec les protocoles réseau. Elle communique de manière efficace à travers le support sans fil et favorise les efforts de coopération entre les nœuds capteurs [Kac09]. La pile protocolaire comprend la couche application, la couche transport, la couche réseau, la couche liaison de données, la couche physique, le plan de gestion de l'énergie, le plan de gestion de la mobilité et le plan de gestion des tâches.[2]

1.7.1 La Couche physique

La couche physique est responsable du choix de la fréquence, de la génération de la fréquence porteuse, de la détection du signal, de la modulation et du chiffage des données. Dans un réseau de capteurs multi-sauts, les nœuds communicants sont liés par un médium sans fil. Ces liens peuvent être constitués par les ondes radio ou des signaux infrarouges.[BB]

1.7.2 La couche liaison de données

Beaucoup de recherches dans le domaine des réseaux de capteurs entrent dans le cadre de l'optimisation de la couche de liaison. Elle manipule toutes les issues de commu-

nication entre les noeuds voisins. Dans les réseaux sans fil, l'accès au médium commun (la fréquence) doit être contrôlé. Ceci est appelé le contrôle d'accès au Media (MAC : Media Access Control). La tâche principale de cette couche est d'interdire l'accès simultané au canal dans la même marge de fréquence radio. Si un récepteur reçoit deux signaux simultanément, c'est malheureusement une collision qui détruit toute l'information reçue par le récepteur. La plupart des protocoles MAC de la couche liaison tentent d'éliminer entièrement les collisions ou de réduire au minimum la capacité de canal qu'ils exigent.[BB]

1.7.3 La couche réseau

La couche réseau en ce qui concerne les réseaux de capteurs est habituellement conçue selon les principes suivants :[BB]

- L'efficacité d'énergie est toujours une considération importante.
- Les réseaux de capteurs sont la plupart du temps data-centric (centré sur les données.).
- L'agrégation de données est une fonctionnalité majeure.

1.7.4 La couche application

Selon les tâches de capture, il existe différents types de logiciels qui peuvent être installés et employés pour la couche application.[BB]

1.7.5 La gestion d'énergie

La couche de gestion d'énergie contrôle la manière d'utiliser l'énergie par le noeud capteur, et gère la consommation d'énergie selon le mode de fonctionnement employé (capture, calcul, et communication par radio). [BB]

1.7.6 La gestion de mobilité

La couche de gestion de mobilité détecte et enregistre le mouvement/mobilité des noeuds capteurs. En utilisant ces positions, les noeuds capteurs peuvent connaître qui sont leurs voisins. Parfois une auto organisation des noeuds est nécessaire à cause de la destruction de quelques noeuds. Dans ce cas, la couche de gestion de mobilité doit être capable de faire changer la position des noeuds.[BB]

1.7.7 La gestion des tâches

La couche de gestion des tâches assure la coopération des efforts des nœuds capteurs, elle ordonnance les événements captés, et les tâches détectées dans une zone de capture spécifique. Par conséquent, les nœuds capteurs qui appartiennent à la même zone de capture ne sont pas obligés d'effectuer les tâches de capture en même temps. Selon leur niveau d'énergie, quelques nœuds capteurs peuvent accomplir des tâches de capture mieux que d'autres .[BB]

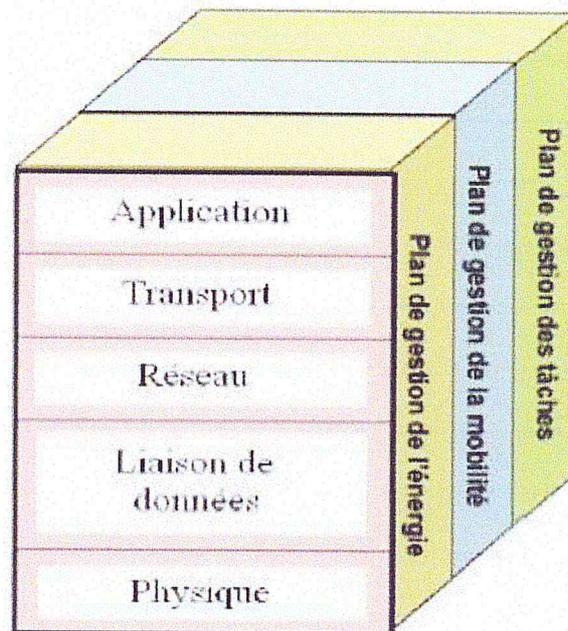


FIGURE 1.5 – la pile protocolaire.[26]

1.8 Technologies des réseaux de capteurs

Parmi les technologies radio utilisées pour permettre les communications entre capteurs, nous pouvons citer : Bluetooth (IEEE 802.15.1), UWB (ou IEEE 802.15.3) et Zigbee (IEEE 802.15.4)... [Ram12].

1.8.1 Bluetooth

La technologie bluetooth, dont "Ericsson", a pour but principal de remplacer les câbles sur les petites distances. Elle est utilisée dans la plupart des téléphones portables comme interface de connexion pour accéder à un PC. Malheureusement, le grand défaut

cette technologie est sa trop grande consommation d'énergie. Donc elle ne peut pas être utilisée par des capteurs qui sont alimentés par une batterie, et qui idéalement devraient fonctionner durant plusieurs années .[Ram12]

1.8.2 Zigbee

Le standard zigbee, combiné avec IEEE 802.15.4, offre des caractéristiques qui répondent encore mieux aux besoins des réseaux de capteurs. Zigbee offre des débits de données moindres, mais il consomme également nettement moins que bluetooth. Un petit débit de données (250Kbit/s max) n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas soutenues et conséquentes.[Ram12]

1.8.3 UWB

L'UWB (ultra wideband) peut être utilisé en tant que technique de communication sans fil, qui fournit des taux de transfert réseaux très élevés sur des distances relativement courtes et à faible puissance. De plus, l'atténuation du signal engendré par des obstacles est moindre qu'avec les systèmes radio à bande étroite conventionnels.[Ram12]

1.8.4 Infrarouge

Les ondes infrarouges sont utilisées pour la communication à courte distance, ainsi elles s'adaptent au mode de communication entre les nœuds d'un réseau de capteur. Les ondes infrarouges sont préférées aux ondes radio, car elles n'interfèrent pas avec les autres signaux électromagnétiques. Les émetteurs/récepteurs utilisant la technologie infrarouge ne sont pas chers, et ils sont faciles à fabriquer. [Ram12]

1.9 Domaines d'application des réseaux de capteurs

Comme beaucoup de technologie, le développement des RCSFs a été suscité par des besoins militaires. L'absence de câbles entre les nœuds, leur faible taille, le nombre élevé de mottes déplorables pour couvrir une zone étendue répondent à ces critères. Puis, la diminution des coûts de fabrication des capteurs, ainsi que la réduction de leur taille a entraîné une utilisation dans des applications civiles. [Abd08]

Parmi elles, nous citons :



FIGURE 1.6 – Les domaines d'application des réseaux de capteur sans fils.[27]

1.10 Facteurs et contraintes des RCSF

La conception et la réalisation des réseaux de capteurs sans fils sont influencées par plusieurs paramètres :

1.10.1 Durée de vie du réseau

C'est l'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où l'énergie du premier nœud s'épuise. Selon l'application, la durée de vie exigée pour un réseau peut varier entre quelques heures et plusieurs années.[Sah]

1.10.2 Ressources limitées

Compte-tenu de leurs petites tailles et leurs faibles coûts, les nœuds capteurs ont plusieurs limitations, telles que des capacités énergétiques de stockage et de traitements limitées, de faibles capacités de détection et de communication. En effet, les rayons de détection et de communication des capteurs varient généralement entre des dizaines de mètres et des centaines de mètres. [Sof07]

1.10.3 Bande passante limitée

Afin de minimiser l'énergie consommée lors de transfert de données entre les nœuds de capteurs opèrent à bas débit. Typiquement, le débit utilisé est de quelques dizaines

de Kb/s. Un débit de transmission réduit n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes. [Sah]

1.10.4 Facteur d'échelle

Le nombre de nœuds déployés pour une application peut atteindre des milliers. Dans ce cas, le réseau doit fonctionner avec des densités de capteurs très grandes. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que la station de base soit équipée de mémoire suffisante pour stocker les informations reçues.[Sah]

1.10.5 Topologie dynamique

La topologie des réseaux de capteurs peut changer au cours du temps pour les raisons suivantes [Sah] :

- Les nœuds capteurs peuvent être déployés dans des environnements hostiles (champ de bataille par exemple), la défaillance d'un nœud capteur est, donc très probable.
- Un nœud capteur peut devenir non opérationnel à cause de l'expiration de son énergie.
- Dans certaines applications, les nœuds capteurs et les stations de base sont mobiles.

1.10.6 Agrégation de donnée

Dans les réseaux de capteurs, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie. L'une des techniques utilisées pour réduire la transmission d'informations redondantes est l'agrégation des données. Avec cette technique, les nœuds intermédiaires agrègent l'information reçue de plusieurs sources. Cette technique est connue aussi sous le nom de fusion de données.[Sah]

1.10.7 Méthodes d'accès au support

La couche MAC joue un rôle fondamental dans la pile de communication des RCSF. Elle est chargée entre autre des fonctions de gestion du contrôle d'accès au support

de communication, de synchronisation entre les différents nœuds, de détection d'erreurs et des mécanismes de retransmission en cas d'erreurs. Dans l'environnement de communication des RCSF, tous les nœuds à portée partagent le même média de communication qui est le support sans fil. En plus, les protocoles MAC d'accès au support sont généralement basés sur des techniques aléatoires utilisant le protocole d'accès au médium CSMA-CA. Ainsi, des collisions fréquentes peuvent se produire si plusieurs nœuds tentent d'accéder au support en même temps, ce qui va accroître la consommation énergétique des nœuds capteurs qui retransmettent leurs données. Par conséquent, des mécanismes de minimisation de la consommation énergétique doivent être intégrés au niveau de cette couche. Pour ce faire, certaines approches, par exemple dans essayent de minimiser les collisions entre les nœuds capteurs et permettent ainsi de réduire la consommation énergétique dans le réseau. En plus des retransmissions dues aux collisions, d'autres aspects liés à l'accès au média de transmission peuvent également avoir des incidences néfastes sur la consommation énergétique dans les RCSF. Ces différents mécanismes sont entre autres l'écoute active, la sur-écoute, la taille des paquets, la surcharge et la surémission. Ces différents aspects sont décrits dans les paragraphes ci-dessous. [Sof07]

1.10.7.1 L'écoute active du canal

L'écoute active du canal consiste pour un nœud donné d'écouter le canal dans l'espérance de recevoir un paquet qui ne sera pas finalement pas reçu. Ainsi, cette forme d'écoute peut entraîner des dépenses inutiles d'énergie. L'une des techniques les plus utilisées pour remédier à ce problème consiste à mettre les nœuds le plus longtemps possible dans l'état sleep. [Sof07]

1.10.7.2 La surécoute

Se produit quand un nœud reçoit des paquets qui ne lui sont pas destinés. Ce phénomène engendre aussi des dépenses énergétiques inutiles pour ces nœuds en question. [Sof07]

1.10.7.3 La taille des paquets

La taille des paquets échangés dans les RCSF peut influencer sur la consommation énergétique des nœuds, et par conséquent sur la durée de vie du réseau. Cette taille doit être raisonnable pour le réseau et elle doit être aussi conforme pour le type d'application. [Sof07]

1.10.7.4 La surémission

Dans les RCSF se produit lorsqu'une source envoie des données à un nœud destinataire qui n'est pas prêt à recevoir des données. Ainsi, les données envoyées peuvent être perdues alors que la source a dépensé inutilement de l'énergie durant sa transmission. [Sof07]

1.11 Caractéristiques des RCSF

Un ensemble de caractéristiques sont importantes pour l'accomplissement des tâches assignées aux applications. Les plus importants sont :

1.11.1 Le type de service

On s'attend à ce que le RCSF (réseaux de capteurs sans fil) offre à l'utilisateur, des informations significatives sur l'objet d'intérêt.[BB]

1.11.2 La qualité de service (QoS)

C'est une métrique de la qualité de service qui va être offerte par un RCSF à ses utilisateurs applications. Le niveau de QoS est défini par un ensemble d'attributs comme le temps d'attente, la largeur de bande, et la perte de paquets qu'on relie directement avec le type de service du réseau. Dans la QoS pour les RCSFs, la quantité et la qualité d'information extraites à partir des puits deviennent appropriées.[BB]

1.11.3 Tolérance aux fautes

Il est important que le RCSF soit capable de traiter l'échec des nœuds capteurs. Une manière reconnue de satisfaire cette contrainte est de réaliser un déploiement redondant des nœuds capteurs.[BB]

1.11.4 La durée de vie

C'est la durée pendant laquelle le réseau reste opérationnel. On s'attend à ce que le RCSF puisse fonctionner au moins pendant le temps requis pour accomplir la tâche donnée. Néanmoins, la définition de la durée de vie dépend de l'application du RCSF et elle est en relation directe avec le fonctionnement efficace du réseau.[BB]

1.11.5 Scalabilité

Cette caractéristique traduit la capacité de maintenir la performance indépendamment de la taille du réseau. Comme un grand nombre de noeuds de capteur peuvent être employés dans les applications de RCSF, les architectures et les protocoles doivent fournir le support approprié pour maintenir efficacement les services fournis par le réseau.[BB]

1.11.6 Maintenance

Les changements dans l'environnement du réseau, par exemple, l'apparition de noeuds de capteurs avec des batteries épuisées, exigent une solution permettant l'adaptation et le maintien des services du RCSF.[BB]

1.11.7 Programmation flexible

C'est la capacité des noeuds de capteur à modifier les options de traitement des données acquises ,et à effectuer des changements et des ajustements de leurs tâches.[BB]

1.12 Les principaux protocoles de routage dans les RCSF

Dans les réseaux de capteurs sans fil, plusieurs protocoles de routage ont été proposés pour trouver le chemin vers la station de base et minimiser les couts de communication, afin de réduire la consommation énergétique.

La figure suivante résume les principaux protocoles de routage .[You10]

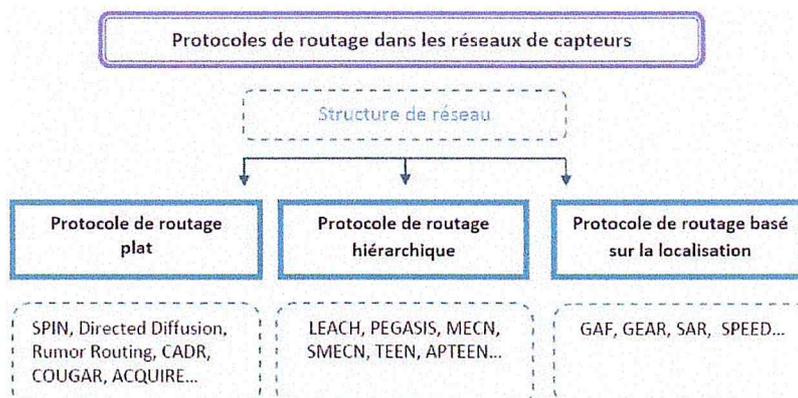


FIGURE 1.7 – Les principaux protocoles de routages dans les RCSF. [You10]

1.12.1 Protocoles centralisés

1.12.1.1 Les Protocoles hiérarchiques

L'objectif principal du routage hiérarchique est de maintenir efficacement la consommation d'énergie de nœuds de capteurs, en les impliquant dans la communication multi-hop au sein d'un cluster, et en effectuant l'agrégation et la fusion des données afin de diminuer le nombre de messages transmis à la destination. La formation de clusters est généralement fondée sur la réserve d'énergie des capteurs et sur les capteurs qui sont à proximité de cluster-head.[You10]

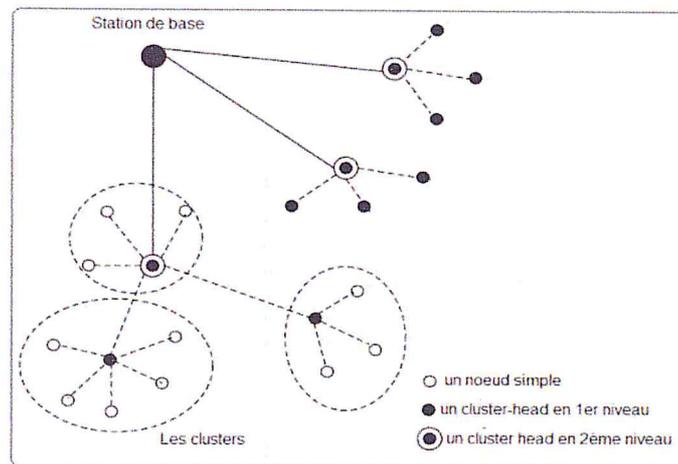


FIGURE 1.8 – Topologie hiérarchique.[You10]

1.12.1.2 Les protocoles de routage à plat

Ces protocoles de routage sont des protocoles non hiérarchiques dans lesquels les noeuds jouent le même rôle et ont la même tâche[12]. Et peuvent communiquer entre eux sans devoir passer par un noeud particulier ou une passerelle. Seul un noeud particulier, le sink, est chargé de la collecte des données issues des différents noeuds capteurs afin de les transmettre vers les centres de traitement.[Sah]

1.12.1.3 Les protocoles de routage basés sur la localisation

Les protocoles de routage basés sur la localisation utilisent les informations d'emplacement, pour guider la découverte de routage et la transmission des données. Ils permettent la transmission directionnelle de l'information, en évitant l'inondation d'information dans l'ensemble du réseau. Par conséquent, le coût de contrôle de l'algorithme

est réduit et le routage est optimisé. De plus, avec la topologie réseau basée sur des informations de localisation de nœuds, la gestion du réseau devient simple. L'inconvénient de ces protocoles de routage est que chaque nœud doit connaître les emplacements des autres nœuds.[You10]

1.12.2 Protocoles proactifs

Les protocoles de routages proactifs constituent une catégorie de routage ad hoc, dans laquelle chaque élément du réseau cherche à établir des tables de routages valides en permanence. Pour ce faire, ces protocoles proactifs échangent en permanence même si aucun transfert de données n'est demandé. Pour résumer, les performances du routage proactif se caractérisent par [Ram12] :

- Les latences les plus faibles, puisque les applications souhaitant émettre peuvent supposer l'existence de routes à jour et valides.[35]
- Une consommation d'énergie importante et une utilisation importante de la bande passante, due au fait que le protocole de mise à jour des routes est employé en permanence, même si les routes ne sont jamais exploitées. [Ram12]
- Une utilisation importante de mémoire vive, due à la nécessité de mémoriser sur le long terme les tables de routages, y compris pour les routes peu ou pas utilisées. [Ram12]

1.12.3 Routage réactif

La famille des routages réactifs réunit les techniques de routages ,pour lesquelles les routes sont déterminées uniquement au moment ou une transmission de données doit être réalisée. La transmission se déroule en deux phases. Dans la première phase a lieu la recherche de route et dans la deuxième phase a lieu le transfert de données. Dans ce cas, ce qu'on entend par le terme de "route" est l'intégralité des nœuds intermédiaires qui vont constituer les différentes étapes des paquets de données. [Car06] Les principales caractéristiques de performance des routages réactifs sont :

- Une latence plus élevée que dans le cas d'un routage proactif, due au fait que chaque transmission est précédée de la phase de recherche de route.
- Une consommation d'énergie contrôlée et qui correspond aux routes effectivement utilisées par les informations échangées.
- Une utilisation de la mémoire évoluant avec l'utilisation réelle des informations échangées.[Car06]

1.12.4 Algorithmes de routage

Il existe plusieurs algorithmes de routage voici quelque exemple :

- Algorithme de DIJKSTRA.
- Algorithme de ASTAR A.
- Algorithme de Bellman-Ford.

1.13 Problèmes des réseaux de capteurs sans fils

Un ensemble de métriques permet de déterminer le design d'un réseau de capteurs. Ces facteurs influencent sur l'architecture des réseaux de capteurs et le choix des protocoles à implémenter.[BB]

1.13.1 Consommation d'énergie

L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner. En effet, un réseau de capteurs ne peut pas survivre si la perte de noeuds est très importante car ceci engendre des pertes de communications dues à une très grande distance entre les noeuds restants. Les réseaux de capteurs fonctionnant selon un mode de routage par saut, chaque noeud du réseau joue un rôle important dans la transmission de données. Le mauvais fonctionnement d'un noeud implique un changement dans la topologie et impose une réorganisation du réseau.[BB]

1.13.2 Routage

En réseaux ad hoc, protocoles de routage sont censés appliquer trois fonctions principales : la détermination et la détection des changements de la topologie du réseau ; le maintien de la connectivité réseau ; et le calcul et la détection des bon itinéraires. En réseaux de capteurs, moins d'effort à été donnée aux protocoles de routage, même si c'est clair que les protocoles de routage ad hoc tels que DSDV (destination sequenced distance vecteur), TORA (temporally-ordered routing algorithm), DSR (dynamic source routing), et AODV (ad hoc on demand distance vector) ne sont pas adaptées pour le réseaux capteur pour la cause du type de trafic qui est "plusieurs à un" et que tous les noeuds typiquement transmettent à une seule station de base ou centre de fusion. Néanmoins, certains mérites de ces protocoles se rapportent aux caractéristiques des réseaux de capteurs, comme la communication multi-sauts et le routage QoS. Le routage peut être associé à la compression des données pour améliorer l'évolutivité du réseau.[BB]

1.13.3 Localisation

Un système de localisation existe déjà, qui est disponible sur toute la surface du globe le GPS. Pourtant, il n'est pas satisfaisant pour l'usage nécessaire, car il cumule les handicaps. Il est disponible seulement en extérieur, et encore si aucun obstacle ne vient obstruer le champ de vue des récepteurs : le fonctionnement sous un feuillage dense, ou dans des villes aux rues étroites, n'est pas possible, où seulement dans de très mauvaises conditions. De plus il est particulièrement coûteux, tant en ce qui concerne le matériel qui est dupliqué en nombreux exemplaires dans un réseau à forte densité de capteurs. De plus, la réception du signal est très gourmande en énergie, ce qui n'est pas compatible avec les problématiques de gestion de durée de vie des batteries. La localisation par moyens propres est donc indispensable. Elle se fait en deux étapes : premièrement l'estimation de la distance aux autres noeuds, et ensuite la triangulation. Le développement de nouvelles techniques de localisation est devenu un grand souci pour les réseaux sans fil de capteurs.[BB]

1.13.4 La sécurité

En fonction de l'application, la sécurité peut être critique. Le réseau devrait permettre la détection des intrusions et la tolérance, ainsi qu'un fonctionnement robuste dans le cas de défaillance parce que, souvent, les noeuds capteur ne sont pas protégés contre les mauvaises manipulation ou attaques. L'écoute, le brouillage, et les attaques de retransmission peuvent entraver ou empêcher l'opération. Par conséquent, le contrôle d'accès, l'intégrité des messages, et la confidentialité doit être garanti.[BB]

1.13.5 Environnement

Les capteurs sont souvent déployés en masse dans des endroits hostiles. Ils sont soumis à différentes conditions d'environnement, ils peuvent fonctionner sous haute pression au fond de l'océan, dans un environnement dur tel que les champs de bataille, dans des champs biologiquement ou chimiquement souillés ou même dans des milieux extrêmement froids. Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiquement éloignées ou inaccessibles .[BB]

1.13.6 Agrégation des données

Dans les réseaux de capteurs, les données produites par les noeuds capteurs sont très reliées, ce qui implique l'existence de redondances de données. Une approche répandue consiste à agréger les données au niveau des noeuds intermédiaires afin de réduire la consommation d'énergie lors de la transmission de ces données.[BB]

1.13.7 Topologie dynamique

La dynamique du réseau découle des défaillances des noeuds ou des cassures des liens entre ceux-ci. La disparition d'un nombre de capteurs dans le réseau, ainsi que le déploiement de nouveaux capteurs, rend la topologie du réseau fréquemment instable. La maintenance d'un réseau est d'autant plus importante que le changement de sa topologie. On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau.[BB]

1.14 Techniques de minimisation de la consommation 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 voire 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.[You10] La durée de vie d'un réseau de capteurs peut être prolongée par l'application conjointe de différentes techniques [Kac09]. Ces techniques sont appliquées soit au niveau de la couche liaison soit au niveau de la couche réseau. [You10] :

1.14.1 Techniques du Duty-cycling

Cette technique est principalement utilisée dans l'activité réseau. Le moyen le plus efficace pour conserver l'énergie est de mettre le noeud de capteur en mode veille à chaque fois que la communication n'est pas nécessaire. Idéalement, le noeud doit être éteint dès qu'il n'y a plus de données à envoyer et ou à recevoir, et devrait être prêt dès qu'un nouveau paquet de données doit être envoyé ou reçu. Ainsi, les noeuds alternent entre périodes actives et sommeil en fonction de l'activité du réseau. Ce comportement est généralement dénommé Duty-cycling. Un Duty-cycle est défini comme étant la fraction de temps où les noeuds sont actifs.[Kac09]

1.14.1.1 Protocoles Sleep/Wakeup

Un régime sleep/wakeup peut être défini pour un composant donné (i.e. le module Radio) du noeud capteur. On peut relever les principaux plans sleep/wakeup implantés sous forme de protocoles indépendants au-dessus du protocole MAC. Les protocoles sleep/wakeup sont divisés en trois grandes catégories : à la demande, rendez-vous programmés, régimes asynchrones.[Dia14]

1.14.1.2 Protocoles du niveau MAC

Plusieurs protocoles MAC pour les réseaux de capteurs sans fil ont été proposés, et de nombreux états de l'art et introductions aux protocoles MAC sont disponibles dans la littérature. Nous nous concentrons principalement sur les questions de gestion d'énergie plutôt que sur les méthodes d'accès au canal. La plupart d'entre eux mettent en œuvre un régime avec un faible duty-cycle pour gérer la consommation d'énergie. Nous avons recensés les protocoles MAC les plus communs en les classant en trois catégories : les protocoles fondés sur TDMA, les protocoles utilisant la contention et les protocoles hybrides. [Kac09]

1.14.2 Techniques orientées données

Généralement, les plans Duty-cycling ne tiennent pas compte des données prélevées par les nœuds. Par conséquent, des approches orientées données peuvent être utiles pour améliorer l'efficacité en énergie. En fait, la détection (ou prélèvement de données) affecte la consommation d'énergie de deux Manières :

- Des échantillons inutiles : les données échantillonnées ont souvent de fortes corrélations spatiales et/ou temporelle il est donc inutile de communiquer les informations redondantes à la station de base. Un échantillonnage inutile implique une consommation d'énergie à son tour inutile. En effet, même si le coût de l'échantillonnage est négligeable, cela induit aussi des communications tout le long du chemin qu'emprunte le message.
- La consommation électrique du module de détection : réduire la communication ne suffit pas lorsque le capteur est lui même très consommateur. Des techniques orientées données sont conçues pour réduire la quantité d'échantillonnage de données en garantissant un niveau de précision acceptable dans la détection pour l'application.[Kac09]

1.14.2.1 Réduction des données

Réduire les données en termes de volume ou de nombre de paquets, dans les réseaux peut avoir un impact majeur sur la consommation d'énergie, due à la communication. Parmi les méthodes de réductions de données, nous trouvons le In-network processing qui consiste à réaliser de l'agrégation de données au niveau des nœuds intermédiaires entre la source et le sink. Ainsi, la quantité de données est réduite tout en parcourant le réseau vers le sink. Une agrégation de données appropriée est spécifique à l'application. Un état de l'art sur les techniques In-network processing est présenté dans le document. La compressions de données peut être appliquée également pour réduire la quantité d'informations transmises par les nœuds sources. Ce régime implique l'encodage d'informations au niveau des nœuds qui engendrent des données.[Kac09]

1.14.3 Conclusion

Les réseaux de capteurs sans fils présentent un intérêt considérable et une nouvelle étape dans l'évolution des technologies de l'information et de la communication. Cette nouvelle technologie suscite un intérêt croissant vu la diversité de ces applications santé, environnement, . . . ect

Dans ce chapitre, nous avons présenté les réseaux de capteurs sans fils, leurs architectures, leurs applications. Nous avons cité également les principaux facteurs, les problèmes, et finalement les solutions qui ont été déjà proposé pour diminuer la consommation d'énergie.

Notre objectif dans ce projet est de voir l'impact de la retransmission des paquets dans la politique de conservation d'énergie. Suite à ça nous allons voir dans les prochaines chapitres les méthodes et les formules qui nous aiderons à réaliser notre but de projet, qui est représenté dans les chaines de Markov et les fils d'attente (FAR/FAV).

Généralités sur les chaînes de Markov

2.1 Introduction

Une chaîne de Markov est une suite de variables aléatoires, définie sur un espace de probabilité E appelé espace d'état, qui peut être fini ou dénombrable. Dans notre projet nous allons travailler sur les chaînes de Markov à espace d'états fini car les modèles que nous allons proposer et sous la base de système file d'attente fermé. Dans ce présent chapitre nous allons voir les chaînes de Markov en générale. Ensuite les chaînes de Markov à temps continu, et à temps discret. Dans ce rapport nous allons utiliser les CMTC comme un outil de modélisation, cette présentation qui nous permet a déduit l'algorithme, et les indice de performance, pour obtenir les résultats final qui nous aide à faire les comparaisons et voir est ce que la retransmission des paquets consomme énergie.

2.2 Définition de chaîne Markov

2.2.1 En mathématique

En mathématique la chaîne de Markov est un outil qui permet de prédire un état futur à partir d'un état présent, et cette traduction est fait à travers des probabilités, c'est-à-dire que concrètement une chaîne de Markov est une liste d'éléments qui sont associés à d'autres éléments suivant une probabilité qui sont apparaitre après. [17]

2.2.2 Algorithmique

Une chaîne de Markov est 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 du système à l'instant n . La propriété fondamentale des chaînes de Markov, dite propriété

de Markov, est que son évolution futur ne dépend du passé qu'au travers de sa valeur actuelle. Autrement dit, conditionnellement à X_n , (X_0, \dots, X_n) et $(X_{n+k}, k \in \mathbb{N})$ sont indépendants. Les applications des chaînes de Markov sont très nombreuses (réseaux, génétique des populations, mathématiques financières, gestion de stock, algorithmes stochastiques d'optimisation, simulation, ...). [4]

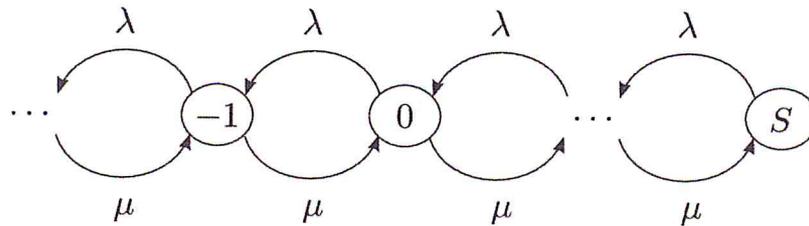


FIGURE 2.1 – Exemple d'un Chaîne de markov

2.3 Définition d'un variable aléatoire

Une variable aléatoire est une fonction définie sur l'ensemble des résultats possibles d'une expérience aléatoire, telle qu'il soit possible de déterminer la probabilité pour qu'elle prenne une valeur donnée, ou qu'elle prenne une valeur dans un intervalle donné. Et à partir de là une variable aléatoire est une fonction qui associe à chaque résultat d'une expérience aléatoire un nombre réel.[5]

$$\begin{cases} X : \Omega \rightarrow E \\ \omega \rightarrow X(\omega) \end{cases}$$

2.4 Loi exponentielle

La loi exponentielle est une loi essentielle dans la théorie des chaînes de Markov. Soit X une variable aléatoire de paramètre λ , rappelons que la densité de probabilité f_X d'une variable aléatoire est [11] :

$$f_x(t) = \begin{cases} \lambda \exp(-\lambda t) & \text{si } t = 0 \\ 0 & \text{si } t \geq 0 \end{cases}$$

2.5 Processus stochastique

Un processus stochastique (ou un processus aléatoire) est une famille des variables aléatoires $\{X(t) | t \in T\}$ défini sur un espace de probabilité donné, indexé par le paramètre t (le temps) qui varie dans un ensemble des indices T . Les valeurs de la variable continue $X(t)$ s'appellent les états, et l'ensemble de toutes les valeurs possibles représente l'espace d'état d'un processus.

Si l'ensemble des indices T est discret alors nous parlons d'un processus en temps discret ; sinon le processus est dit en temps continu. [Mih12]

2.6 Processus de Markov

Un processus stochastique $\{X(t) | t \in T\}$ est un processus de Markov si pour tout $t_0 < t_1 < \dots < t_n < t$ la distribution conditionnelle de $X(t)$ pour certaines valeurs de $X(t_0), X(t_1), \dots, X(t_n)$ dépend seulement de $X(t_n)$:

$$P[X(t) \leq x | X(t_n) = x_n, X(t_n - 1) = x_{n-1}, \dots, X(t_0) = x_0] = P[X(t) \leq x | X(t_n) = x_n].$$

Dans la plupart des cas cette fonction de distribution a la propriété d'invariance par rapport à t_n : $P[X(t) \leq x | X(t_n) = x_n] = P[X(t - t_n) \leq x(0) = x_n]$. Dans ce cas, le processus de Markov est dit homogène et il est caractérisé par le fait que l'histoire du processus est comprise dans l'état courant. [Mih12]

2.7 Chaînes de Markov à temps discret

2.7.1 Définition

Un processus stochastique à espace d'état et de temps discrets $\{X_n, n > 0\}$ est une chaîne de Markov à temps discret (CMTD) si et seulement si la propriété de Markov suivante est vérifiée : $P[X_{n+1} = j_{n+1} | X_n = j_n, \dots, X_0 = j_0] = P[X_{n+1} = j_{n+1} | X_n = j_n], \forall n \in \mathbb{N}, j_k \in E$. [20]

- Dans une CMTD, le comportement futur ne dépend que de l'état actuel.
- $P_{ij}(n)$ est appelé probabilité de transition de l'état i à l'état j à l'instant n .
- Une CMTD est un processus stochastique à événements discrets, temps discret et sans mémoire.

Une "chaîne de Markov homogène" est une chaîne telle que la probabilité pour passer dans un certain état à la n -ème est soit indépendante du temps. et en tout temps

la loi de probabilité à tout moment de la chaîne est toujours la même pour caractériser la transition à l'étape en cours.[8] Nous pouvons alors définir la loi de "probabilité de transition" d'un état i vers un état j par : $P_{ij} = P(X_n = j | X_{n-1} = i)$.

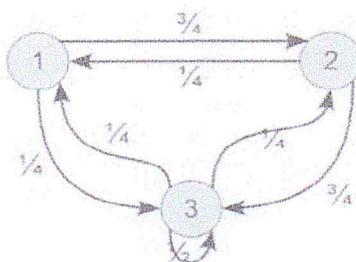
Puis il est facile de définir la "matrice de transition" ou "matrice stochastique" :

$$\begin{bmatrix} P_{11} & P_{12} & \cdot \\ \cdot & P_{22} & P_{23} \\ P_{21} & P_{22} & P_{23} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}$$

Dans la matrice stochastique CMTD il faut :

- Tous ses termes sont positifs ou nuls.
- La somme des termes de chaque ligne vaut 1.

Les chaînes de Markov peuvent être représentées graphiquement sous la forme d'un graphe orienté ayant pour sommet les points i et pour arcs les couples orientés (i, j) . Nous associons alors à chaque composante un arc orienté et sa probabilité de transition.[8].Voici l'exemple dans la figure suivant :



On a la matrice stochastique : $P = [P_{ij}]$, $P^{(n)} = [P_{ij}^{(n)}]$.

$$P = \begin{bmatrix} 0 & 3/4 & 1/4 \\ 1 & 0 & 0 \\ 1/4 & 1/4 & 3/4 \end{bmatrix}$$

FIGURE 2.2 – Représentation de CMTD par matrice et par graphe

2.7.2 Distributions stationnaires

Une distribution de probabilité discrète (π) est dite stationnaire par rapport à la matrice stochastique P si $\pi P = \pi$. Autrement dit π est un vecteur propre normalisé π de P associé à la valeur propre 1 .[9]

$$\begin{cases} \pi.P = \pi \\ \sum \pi_i = 1 \end{cases}$$

Une chaîne de Markov est appelée stationnaire si la distribution $\pi (n)$ de la variable aléatoire X_n est indépendante du temps n , ce qui signifie que $\pi(0)$ est une distribution stationnaire du processus pour déterminer les vecteurs stationnaires d'une chaîne de Markov finie, on a les deux approches suivantes :

- On détermine un vecteur propre π normalisé de P , relatif à la valeur propre 1.
- On utilise le graphe des transitions en interprétant les probabilités π_k comme des masses associées aux états $k \in S$ et les produits $\pi_k P_{kj}$ comme des flux de masse entre les états k et j . La répartition des masses π_k est stationnaire si, lors d'une transition, le flux d'entrée est égal au flux de sortie pour chacun des états. Les m équations ainsi déterminées sont appelées équations de balance.[9]

voici l'exemple dans la figure suivant :

Exemple $P = \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix}$

$$\begin{cases} \pi_0 = 0.8\pi_0 + 0.5\pi_1 \\ \pi_1 = 0.2\pi_0 + 0.5\pi_1 \end{cases} \longrightarrow \begin{cases} \pi_0 = 0.8\pi_0 + 0.5\pi_1 \\ \pi_0 + \pi_1 = 1 \end{cases} \Leftrightarrow \begin{cases} \pi_0 = 5/7 \\ \pi_1 = 2/7 \end{cases}$$

FIGURE 2.3 – Exemple sur la méthode de calcul d'un vecteur stationnaire[28]

2.8 La chaîne de Markov a temps continu (CMTC)

Les chaînes de Markov permettent d'étudier des phénomènes aléatoires, pour lesquels les changements d'état se produisent à des instants fixés à l'avance. Dans beaucoup de cas il est nécessaire d'avoir recours à un modèle stochastique, faisant intervenir tous les instants appartenant à un intervalle donné, c'est à dire à un processus à temps continu.[8]

2.8.1 Définition d'une CMTC

Un processus stochastique $X(t)$, $t > 0$ à espace d'état discret et à temps continu est une chaîne de Markov à temps continu (CMTC) si et seulement si [10] :

$P[X(t_n + 1) = j_{n+1} | X(t_n) = j_n, \dots, X(t_0) = j_0] = P[X(t_n + 1) = j_{n+1} | X(t_n) = j_n]$ pour tout $n \in \mathbb{N}$.

- Dans une CMTC, le comportement futur ne dépend que de l'état actuel.
- Une CMTC est homogène si et seulement si : $P_{ij}(t) = P[X(s + t) = j | X(s) = i]$ pour tout $s \geq 0$.

2.9 Représentation d'une CMTC

Une chaîne de Markov à temps continu peut être décrite soit par un diagramme de transition d'état, ou bien par une matrice des taux de transition dite générateur Infinitésimal. [10]

2.9.1 Le diagramme de transition

Est un graphe orienté, les sommets correspondent aux états de la chaîne de Markov et les arcs sont étiquetés par les taux de la distribution associés à la transition d'un état à un autre. [10]

2.9.2 Le générateur infinitésimal Q

Soit $\{X(t), t_0\}$ un processus tel que : quand il se trouve dans l'état i , il peut en sortir pour aller dans l'état $j \neq i$ avec P_{ij} , après un temps exponentiellement distribué de paramètre μ_{ij} . On suppose que $\sum \mu_{ij} < \infty$. Le processus reste effectivement jusqu'à la date $\min_j \{Y_{ij}\}$ puis saute vers l'état k qui réalise le minimum. Alors le processus $\{X(t), t \geq 0\}$ est une chaîne de Markov en temps continu, dont le générateur infinitésimal Q est donné par [11] :

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

En vertu de cette construction, on peut interpréter chaque $Q(i, j)$ comme le paramètre d'une variable aléatoire exponentielle. Pour cela, on dit que $Q(i, j)$ désigne le

taux de transition est de l'état i vers l'état j . [11] et P_{ij} la probabilité de passé de l'état i vers l'état j . Exemple : Considérons une chaîne de Markov à trois états. Autrement dit, soit $E = \{ A, B, C \}$

Supposons que diagramme de transition de cette chaîne soit :

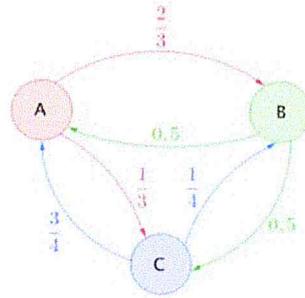


FIGURE 2.4 – Diagramme de transition de la CMTC

Donc le générateur infinitésimal de cette chaîne soit :

$$\begin{array}{c}
 \begin{array}{ccc}
 & A & B & C \\
 A & \left(\begin{array}{ccc}
 -1 & 2/3 & 1/3 \\
 0.5 & -1 & 0.5 \\
 3/4 & 1/4 & -1
 \end{array} \right) \\
 B \\
 C
 \end{array}
 \end{array}$$

FIGURE 2.5 – La matrice de transition de la CMTC.

2.10 Régime stationnaire

Dans une chaîne de Markov CMTC, le vecteur des probabilités stationnaires $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ existe et est l'unique solution du système matriciel suivant :

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

Où E est l'espace des états de chaîne de Markov.[10]

2.11 Les processus stochastique a temps continu

Les chaînes de Markov permettent d'étudier des phénomènes aléatoires pour lesquels les changements d'état se produisent à des instants fixés à l'avance. Dans beaucoup de cas il est nécessaire d'avoir recours à un modèle stochastique faisant intervenir tous les instants appartenant à un intervalle donné, c'est à dire à un processus à temps continu. Parmi ceux-ci [9] :

2.11.1 Processus de poisson

Un processus de poisson est un modèle mathématique modélisant des événements aléatoires qui se reproduisent au cours du temps : naissances, pannes, désintégration radioactive.[12]

Par définition, un processus de poisson satisfait trois conditions :

- le processus est stationnaire : cela signifie que le nombre d'arrivées dans un intervalle $[t, t+s]$, noté $N(t, t+s)$, ne dépend que de la durée de la période (s) et pas de sa date de début (t). On notera donc $N(s) = N(t, t+s)$
- le processus est sans mémoire : la probabilité d'avoir i arrivées dans un intervalle $[t, t+s]$ ne dépend pas des arrivées ayant précédé l'instant t .
- le processus est ordonné : les arrivées se produisent une par une .

A partir de ces hypothèses, nous allons établir un certain nombre de résultats permettant de démontrer la fonction de distribution correspondant à un processus de Poisson.

2.11.1.1 Caractérisation du processus de Poisson

1er résultat : La probabilité qu'aucun événement ne se produise dans un intervalle de longueur t est égale à $\exp^{-\lambda t}$ (la signification de λ sera donnée plus loin).

2ème résultat : La probabilité d'avoir i arrivées dans un intervalle de longueur t est égale à $P_i(t) = \frac{\exp^{-\lambda t} (\lambda t)^i}{i!}$

- Le paramètre λ : doit être interprété comme le nombre moyen d'arrivées par unité de temps.
- Un processus de Poisson est une CMTC réductible voici la figure :

2.11.4 Un processus de naissance et mort

Un processus de naissance et mort est un processus markovien de saut à valeurs dans \mathbb{N} dont les amplitudes des sauts sont égales à ± 1 . Ses taux de transition sont donnés par [13] :

$$\begin{cases} i \rightarrow i + 1 & \text{au taux } \lambda_i \\ i \rightarrow i - 1 & \text{au taux } \mu_i \end{cases}$$

Avec $(\lambda_i)_i$ et $(\mu_i)_i$ deux suites de réels positifs ou nuls, pour $i \in \mathbb{N}$ Le générateur infinitésimal vaut : $Q_{i,i+1} = \lambda_i$, $Q_{i,i-1} = \mu_i$, $Q_{i,j} = 0$ sinon. Le taux global de saut pour une population de taille i vaut $\lambda_i + \mu_i$. Ainsi après un temps de loi exponentielle de paramètre $\lambda_i + \mu_i$, le processus augmente de 1 avec probabilité $\lambda_i / (\lambda_i + \mu_i)$ et décroît de -1 avec probabilité $\mu_i / (\lambda_i + \mu_i)$. [13]

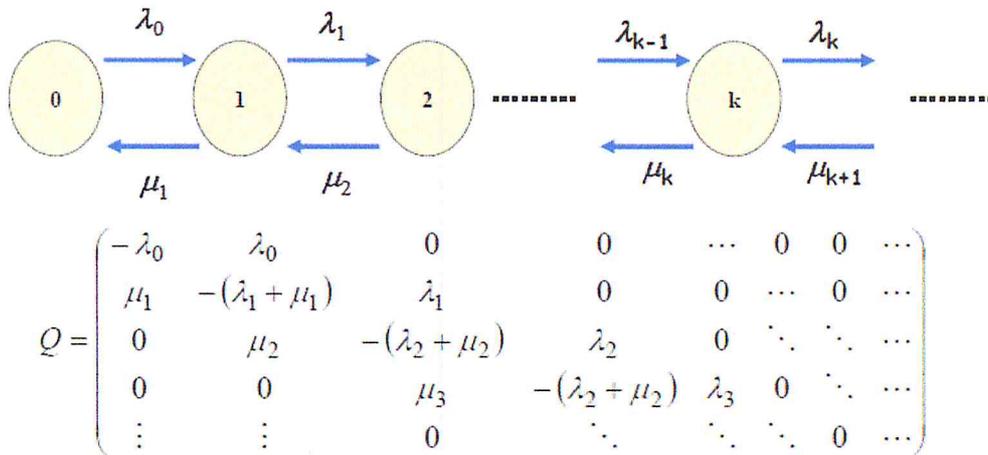


FIGURE 2.7 – Graphe et matrice d'un processus de naissance et mort. [28]

Dans notre thèse nous allons travailler avec les chaînes de Markov en temps continu (CMTC) avec les processus naissance et mort.

2.12 Conclusion

Pour atteindre notre but de ce projet, nous allons baser sur le modèle de file d'attente markovienne. Dans ce chapitre nous avons vu tout ce qui concerne les chaînes de Markov à temps discret (CMTD) et à temps continu (CMTC). Ainsi que la présentation graphique et matricielle de chaque chaîne. Nous avons étudié le principe de chaîne de

Markov car nous allons modéliser nous proposition par ce dernier (les chaînes de Markov à temps continu). lorsque nous analysons le CMTC on déduit

l'algorithme de générateur infinitésimal Q , et les différents indices de performances et faire les calculs, on peut voir dans quelle mesure la retransmission des paquets conserve l'énergie.

Les files d'attente avec vacance et les files d'attente avec rappel

3.1 Introduction

Les modèles d'attente avec rappels et vacances se distinguent des modèles classiques par l'existence de deux paramètres supplémentaires. Le premier paramètre décrit le phénomène de rappel bien connu dans les applications liées aux systèmes téléphoniques et informatiques. Le second paramètre décrit l'activité de vacances introduite en général pour exploiter l'oisiveté du serveur. Les modèles des files d'attente avec rappels et vacances sont caractérisés par la présence simultanée des phénomènes de répétition de demandes et de vacances à la fois. Dans ces modèles, durant la période de vacances, le serveur n'est pas disponible aux nouvelles arrivées de paquets primaires ni secondaires. Dans ce cas, tout paquet qui trouve le serveur non disponible (occupé ou en vacance) est bloqué, alors il quitte la zone de service et atteint un espace d'attente orbite, après un certain temps aléatoire, il renouvelle sa tentative d'entrer en service, une fois, deux fois, ..., jusqu'à ce qu'il le trouve disponible. [24]

Donc L'analyse des systèmes de files d'attente s'appuie sur les outils de la théorie des chaînes de Markov à temps continu au niveau de ce chapitre nous allons toucher la vue des files d'attente en générale. Ensuite les files d'attente avec vacance et les files d'attente avec rappel. Pour la suite de ce projet nous allons utiliser le système de file d'attente à source fermé. [Kha12]

3.2 La file d'attente classique

Une file d'attente est un système stochastique composé d'un certain nombre (fini ou non) de places d'attente, d'un ou plusieurs serveurs, et bien sûr de paquet qui arrivent attendent se font servir selon des règles de priorité données, et quittent le système.

La description précédente d'une file d'attente, dont une représentation schématique est donnée en figure ci-dessous, ne saurait capturer toutes les caractéristiques des différents modèles que comptent la littérature, mais elle identifie les éléments principaux permettant la classification de la grande majorité des files d'attente simples. [MAMB+17]

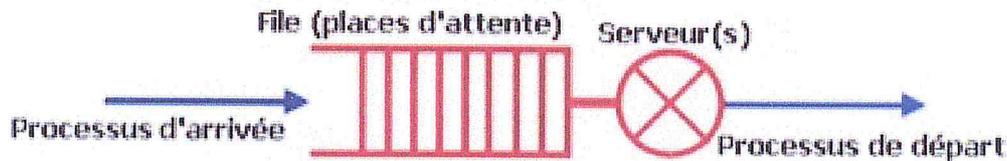


FIGURE 3.1 – système file d'attente.[MAMB+17]

3.3 Classification des systèmes d'attente

Afin de spécifier un système de file d'attente, on se base sur trois éléments :

- **Le processus d'arrivée**

Pour le processus d'arrivée, on s'intéresse aux instants d'arrivés des paquets dans la file. Ils sont en général aléatoires. Certaines hypothèses sont faites sur leurs lois. Tout d'abord, il n'arrive qu'un paquet à la fois. La deuxième hypothèse est l'homogénéité dans le temps. Cela se traduit par le fait que les temps d'inter arrivés des paquets sont des variables aléatoires. De même loi. Ils sont également supposés indépendants. Enfin, la loi des temps d'inter arrivée est supposée connue. Le cas le plus courant est celui où cette loi est exponentielle. Dans ce cas, le modèle des temps d'arrivée des clients est un processus de Poisson. Évidemment d'autres cas peuvent se présenter : temps d'inter arrivés constants, de loi uniforme, loi normale ou encore gamma. [15]

- **Le processus de service :**

Qui compris le nombre de serveurs et la loi probabiliste décrivant la durée des services [15].

- **Structure et discipline de la file :**

La discipline de service détermine l'ordre dans lequel les clients sont rangés dans la file et y sont retirés pour recevoir un service. [15]

3.4 Notation de Kendall

La notation suivante, appelée la notation de Kendall, est largement utilisée pour classer les différents systèmes de files d'attente comprenant six symboles rangés dans : L'ordre A/B/c/m/P -D, où : [Kha12]

- A décrit la nature du processus d'arrivée.
- B décrit la nature du processus de service.
- c est le nombre de serveurs.
- m est la capacité d'accueil d'un système.
- P est la taille de la population.
- D est la discipline de la file.

3.5 La discipline d'attente

La discipline d'attente, est la règle de priorité déterminant l'ordre dans lequel les paquets vont accéder à la ressource modélisé par le serveur. Les disciplines d'attente classiques, ainsi que leurs acronymes, sont : [Kha12]

- FIFO (first in, first out) ou FCFS (first come first served) : C'est la file standard dans laquelle les clients sont servis dans leur ordre d'arrivée. Notons que les disciplines FIFO et FCFS ne sont pas équivalentes lorsque la file contient plusieurs serveurs. Dans la première, le premier client arrivé sera le premier à quitter la file alors que dans la deuxième, il sera le premier à commencer son service. Rien n'empêche alors qu'un client qui commence son service après lui, dans un autre serveur, termine avant lui. [15]
- LIFO (last in, first out) ou LCFS (last come, first served) : Cela correspond à une pile, dans laquelle le dernier client arrivé (donc posé sur la pile) sera le premier traité (retiré de la pile). À nouveau, les disciplines LIFO et LCFS ne sont équivalentes que pour une file monoserveur. [15]
- PNP (Priority service) : Les clients sont servis à leur priorité. Tous les clients de la plus haute priorité sont servis premier, puis les clients de moindre priorité sont servis. [15]
- PS (Processor Sharing) :
Les clients sont servis de manière égale. La capacité du système est partagée entre les clients. [15]
- SIRO (Served In Random Order) : Les clients sont servis au aléatoirement. [15]

3.6 Caractéristiques d'un système de files d'attente

On note λ le taux d'arrivée des clients. Cela signifie que l'espérance mathématique de la durée séparant deux arrivées successives est $E[A] = \frac{1}{\lambda}$. On note μ le taux de service des clients. Cela signifie que l'espérance de la durée de service est $E[Se] = \frac{1}{\mu}$. La distribution stationnaire du processus stochastique introduit permet d'obtenir les caractéristiques d'exploitation du système, telles que : le temps d'attente d'un client (le temps qu'un client passe dans la file d'attente), le temps de séjour d'un client dans le système (composé du temps d'attente et de la durée de service), le taux d'occupation des dispositifs de service, la durée de la période d'activité (l'intervalle de temps pendant lequel il y a toujours au moins un client dans le système) [Kha12].

Nous allons voir tous ces caractéristiques en détail dans le chapitre suivant (les indices de performances) .

3.7 Types de modèles

3.7.1 Modèles markoviens

Les modèles markoviens caractérisent les systèmes dans lesquels les deux quantités stochastiques principales, qui sont le temps inter-arrivées et la durée de service, sont des variables aléatoires indépendantes et exponentiellement distribuées. La propriété d'absence de mémoire de la loi exponentielle facilite l'étude de ces modèles. L'étude mathématique de tels systèmes se fait par l'introduction d'un processus stochastique approprié. Ce processus est souvent le processus $\{N(t), t \geq 0\}$ défini comme étant le nombre de clients dans le système à l'instant t . L'évolution temporelle du processus markovien est complètement définie grâce à la propriété d'absence de mémoire.[Kha12].

3.7.1.1 Système M/M/1

Le système de file d'attente M/M/1 est le système le plus élémentaire de la théorie des files d'attente. Le flot des arrivées est poissonnier de paramètre λ et la durée de service est exponentielle de paramètre μ , la discipline d'attente est FIFO, la file d'attente est de capacité infinie. [15]

3.7.1.2 La file M/M/1/K

On considère un système à serveur simple identique à la file M/M/1 excepté que la capacité de la file d'attente est finie. On a donc toujours les hypothèses suivantes : le processus d'arrivée des clients dans la file est un processus de poisson de taux λ et le temps de service d'un client est une variable aléatoire exponentielle de taux μ . Soit K la capacité de la file d'attente : c'est le nombre maximal de clients qui peuvent être présents dans le système, soit en attente, soit en service. Quand un client arrive alors qu'il y a déjà K clients présents dans le système, il est perdu. Ce système est connu sous le nom de file M/M/1/K. L'espace d'états E est maintenant fini : $E = \{0, 1, 2, \dots, K\}$. La capacité de la file étant limitée, même si les clients arrivent en moyenne beaucoup plus vite que ce que le serveur de la file est capable de traiter, dès que celle-ci est pleine, les clients qui se présentent sont rejetés. Le nombre de clients dans la file ne peut donc jamais "partir" à l'infini. De plus, dès qu'un client est autorisé à entrer, il sortira un jour et son temps de séjour dans la file est fini, puisqu'il correspond au temps de service de tous les clients devant lui et que ce nombre est limité par K . Sur un temps très long, le débit de sortie sera donc bien égal au débit d'entrée, ce qui correspond bien à la stabilité inconditionnelle du système.[15]

3.7.2 La file M/M/C

On considère un système identique à la file M/M/1 excepté qu'il comporte C serveurs identiques et indépendants les uns des autres. On conserve les hypothèses : processus d'arrivée des clients poissonnier de taux λ et temps de service exponentiel de taux μ (pour chacun des serveurs). Ce système est connu sous le nom de file M/M/C. L'espace d'états E est, comme pour la M/M/1 infini : $E = \{0, 1, 2, \dots\}$. La file d'attente est de capacité infini. Si l'un des serveurs est libre, le client qui arrive se dirige immédiatement vers ce serveur. Dans le cas contraire, le client prend sa place dans une file d'attente commune pour tous les serveurs. Lorsqu'un serveur se libère, le client en tête de la file occupe ce serveur. Par conséquent, la discipline d'attente est FIFO.[15]

3.7.2.1 La file M/M/ ∞

On considère un système composé d'un nombre illimité de serveurs identiques et indépendants les uns des autres. Dès qu'un client arrive, il rentre donc instantanément en service. Dans cette file particulière, il n'y a donc pas d'attente. On suppose toujours que le processus d'arrivée des clients est poissonien de taux λ et que les temps de service sont exponentiels de taux μ (pour tous les serveurs). Ce système est connu sous le nom de file M/M/ ∞ . [15]. Dans notre thèse nous allons travailler avec les file d'attente M/M/1/K ou la file et limité et avec un seul serveur .

3.7.3 Modèles non markoviens

En l'absence de l'exponentialité ou lorsque l'on s'écarte de l'hypothèse d'exponentialité de l'une des deux quantités stochastiques : le temps des inter-arrivées et la durée de service, ou en prenant en compte certaines spécificités des problèmes par introduction de paramètres supplémentaires, on aboutit à un modèle non markovien [Kha12].

3.8 Analyse mathématique d'un système de files d'attente

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

- Les probabilités d'état $\pi_n(t) = P(N(t) = n)$, qui définissent le régime transitoire du processus stochastique $\{N(t), t \geq 0\}$. Il est évident que les fonctions $\pi_n(t)$ dépendent de l'état initial ou de la distribution initiale du processus.[15]

3.9 Les file d'attente avec vacances

Le modèle de file d'attente avec vacances est similaire à la file d'attente standard. La seule différence avec une file d'attente standard c'est que le serveur peut devenir indisponible pendant une période aléatoire c'est à dire il est en vacances.

Le serveur passe "en vacances" à chaque fois que le système devient vide (pas de client (paquet) en attente. (buffer vide)) .

Il existe différents types de systèmes de files d'attente de vacances. Dans le système de vacances unique, le serveur prend des vacances d'une durée aléatoire lorsque la file d'attente est vide. À la fin des vacances, le serveur retourne à la file d'attente. S'il y a au moins un client en attente lorsque le serveur revient de vacances, le serveur effectue l'une des actions suivantes en fonction de la politique de service.

(a) Selon la politique de service exhaustive, le serveur servira tous les clients en attente ainsi que ceux qui arrivent alors qu'il est encore en service à la gare. Il prend d'autres vacances quand la file d'attente devient vide.

(b) En vertu de la politique de service sécurisé, le serveur servira uniquement les clients qu'il trouve dans la file d'attente à son retour de vacances. À la fin de leur service, le serveur commencera à prendre d'autres vacances et tous les clients qui arriveront alors que le serveur servait déjà à la station seront servis lorsque le serveur

reviendra des vacances.

(c) En vertu de la politique de service limité, le serveur servira seulement un nombre maximum prédéfini des clients et commencera alors un autre congé. Le schéma de service unique dans lequel exactement un client est servi est un type spécial de cette politique. Si la file d'attente est vide au retour du serveur, le serveur attend une période occupée en utilisant l'une des stratégies de service avant de prendre un autre congé.

Dans le plan de vacances multiples, si le serveur revient d'un congé et trouve la file d'attente vide, il commence immédiatement un autre congé. S'il y a au moins un client en attente, il commencera le service conformément à la politique de service en vigueur.[II14]

3.10 Systèmes d'attente avec rappels

Dans la théorie des files d'attente classique, il est supposé qu'un client qui ne peut pas obtenir son service immédiatement dès son arrivée, rejoint la file d'attente ou quitte le système définitivement. Les systèmes de files d'attente développés tentent de prendre en considération des phénomènes de répétition de demandes de service, et ceci après une durée du temps aléatoire. Un tel système est connu comme «système de files d'attente avec rappels». Pour identifier un système de files d'attente avec rappels, on a besoin des spécifications suivantes : la nature stochastique du processus des arrivées, la distribution du temps de service, le nombre de serveurs qui composent l'espace de service, la capacité et discipline d'attente ainsi que la spécification concernant le processus de répétition d'appels.[15]

3.11 Description du modèle d'attente avec rappels

Un système d'attente avec rappels (Retrial Queue) est un système composé de c serveurs (modèle est dit mono serveur si $c=1$ et il est multiserveurs si $c \geq 2$) identiques et indépendants, d'un buffer de capacité K , et d'une orbite (espace d'attente imaginaire), et par un nombre R de clients (paquets) représentant la taille de la source du système. Si le nombre R est fini le système est dit à source finie (fermé) sinon le système est dit à source infinie (ouvert).

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 $1-p$, ou bien entre en orbite avec la probabilité p 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. 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 $1 - p$ ou bien entre en orbite avec la probabilité p si l'orbite n'est pas pleine.

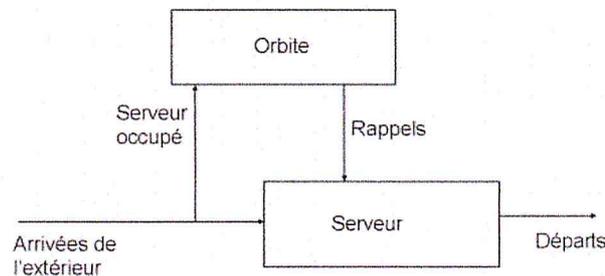


FIGURE 3.2 – Schéma illustratif d'un système avec rappels.

Dans notre recherche nous allons utiliser les files d'attente avec rappel de source fini et un seul serveur.

3.12 les Politique FAR

3.12.1 Politique linéaire classique

La politique de rappels linéaire classique est caractérisée par le fait que chaque client (paquet) en orbite engendre sa propre tentative de joindre le serveur indépendamment des autres clients en orbite.[Kha12]

3.12.2 Politique de rappels constante

Cette politique été introduite initialement par Fayolle et est caractérisée par le fait que l'orbite engendre un seul processus de poisson, et des que ce processus trouve le serveur libre, un seul client en orbite (le premier en file un autre choisi aléatoirement) reçoit son service et quitte le système. Ce qui veut dire que le taux de rappels ne dépend pas du nombre de clients en orbite. [Kha12]

3.12.3 Politique de rappels versatile

Cette politique est une combinaison des deux précédentes et a été introduite par Artalejo et Gomez-Corral. L'orbite tente de trouver le serveur libre selon un processus de poisson de taux indépendamment des clients en orbite et chaque client en orbite effectue sa propre tentative selon un processus de poisson de taux μ . [Kha12]

3.13 Les travaux connexes

Il existe une littérature abondante sur l'économie d'énergie dans WSN, car de nombreuses méthodes ont été proposées dans ces dernières années, et il y a encore beaucoup de recherches en cours sur la façon d'optimiser la consommation d'énergie. Dans ce projet en utilisant la théorie des files d'attente donc nous allons citer quelques méthodes qui ont déjà été réalisées dans cette recherche. Parmi ces méthodes :

R. Maheswar et al., considèrent un réseau de capteur composé d'un certain nœud statique identiques, qui sont uniformément répartis et le nœud sink, collecte toutes les informations recueillies par les capteurs est situé au centre dans le domaine. Ils considèrent pour un capteur deux états opérationnelle qui sont l'état idle et l'état busy. Initialement le nœud est dans l'état idle il peut uniquement recevoir le paquet de nœud voisin. Lorsque le nœud atteint un certain seuil, il passe à l'état busy dans cet état il peut transmettre les paquets. Ils ont proposé un modèle analytique à partir de laquelle ils ont pu déduire le délai d'attente et la consommation d'énergie d'un nœud de capteur. [MJ11]

F.C Jiang et al., ont proposé un nouveau schéma d'économie d'énergie, pour éviter la surconsommation d'énergie due au passage fréquent entre les différents états de capteur. Ils sont basés sur une file d'attente avec vacance M/M/1 avec la politique N-vacance. De la même façon ils ont opté pour l'état idle et busy. Durant l'état idle il reçoit et il génère les paquets jusqu'à ce que le nombre de paquets dans le buffer atteigne un seuil N, pendant l'état busy peut recevoir et transmettre les paquets. Généralement le nœud dans l'état idle conserve la consommation la plus faible d'énergie. [Jia+12b]

De mêmes façons, les auteurs de [LY13] ont opté pour la politique N-Policy avec la variation $Geo / G / 1$ pour la conservation d'énergie dans les réseaux de capteur sans fil.

Dans les travaux présente la file d'attente a été considéré de la taille infinie ce qui fausser de les résultats. Les auteurs Jiang et all [Jia+12a] Boutoumi Bachira et all [BG17] ont considéré des variante de la file d'attente avec vacance et N-Policy avec buffer finie.

Le tableau ci-dessous illustre les travaux connexes cités précédemment :

TABLE 3.1 – Tableau comparatif des différents modèles existants

Référence	Modèle	Méthode de résolution	Indice de performance
[MJ11]	Le modèle analytique pour analyser les performances du système	La consommation d'énergie des nœuds dans un réseau de capteurs pendant transmission de paquet basée sur le seuil de file d'attente	La consommation d'énergie moyenne et délai moyen
[Jia+12a]	Le modèle M / G / 1 avec le Min (N, T).	Pour réduire La consommation d'énergie avoir un compteur N et un temporisateur T	la consommation d'énergie et le délai de latence
[Jia+12b]	Le modèle M/M/1 . la politique N.	La technique d'économie d'énergie basée sur M/M/1	le taux d'arrivée, d'un paquet λ , Le taux de service μ , la probabilité, la consommation d'énergie .
[LY13]	Le modèle Geo/G/1 la politique N.	La probabilité de la distribution de la longueur de file d'attente et la distribution de temps de séjour	Le schéma d'économie l'énergie. le taux d'arrivé de paquet (catastrophe)
[HL13]	Le modèle M/G/1/K avec la politique N	Ils ont proposé une méthode du seuil optimal dynamique pour assure la consommation minimal d'énergie qui pourrait être obtenue en considération défférent taux arrivé	La consommation d'énergie minimale avec un seuil N et un taux d'arrivé λ
[BG17]	La politique N-vacance et la politique Hybride.	Elles ont travaillé avec les réseaux de petri pour modéliser la soution proposé	Elles ont utilisé plusieurs indice de performance comme la consommation d'énergie et le délai d'attente.

3.14 Conclusion

L'étude d'une file d'attente a pour but de calculer ou d'estimer les performances d'un système, dans des conditions de fonctionnements données. Ce calcul se fait le plus souvent durant le régime stationnaire. Alors dans ce chapitre nous avons commencé par présenter les files d'attente classique, avec vacance et avec rappel. Puis nous avons abordé les travaux qui ont opté par l'utilisation des files d'attente avec vacance pour la conservation de l'énergie des RCSF. La politique N-Policy a été largement utilisé. Cette discipline efficace a minimise la consommation d'énergie et introduite le délai d'attente supplémentaire. De même dans l'étude de cette politique aucun n'a été considère la retransmission des paquets. Sachant que ce phénomène influe largement sur la consommation d'énergie .dans le chapitre suivant nous allons étudier l'impact de la retransmission des paquets sur la politique d'économie d'énergie à savoir N-Policy.

Modélisation et analyse des réseaux capteur sans fils

4.1 Introduction

Pour étudier l'impact de la retransmission des paquets dans la politique de conservation d'énergie dans les RCSF, nous allons utiliser la notion des files d'attente. Dans ce chapitre nous allons proposer quatre modèles, basés sur le modèle de file d'attente avec vacance et avec rappel. Les chaînes de Markov à temps continu (CMTC) vont être utilisées comme méthode de modélisation des performances. Un algorithme sera déduit pour chaque modèle proposé à partir de la CMTC correspondant, qui permet d'avoir le générateur infinitésimal Q , par la suite les indices de performance et la consommation d'énergie.

4.2 Modélisation d'une RCSF

Nous allons proposer quatre modèles affecter pour un nœud capteur sans fil, pour voir l'impact de la retransmission des paquets sur la conservation d'énergie dans les RCSFs, qui sont :

4.2.1 Le modèle N-policy sans l'état sleep

Dans cette section, nous décrivons le modèle correspondant à un nœud de capteur dans sa période active. Nous considérons un nœud de capteur en tant que système de file d'attente avec vacances à buffer fini, dans lequel les paquets arrivent selon le processus quasirandom, avec paramètre λ et les temps de service sont distribués exponentiellement avec la moyenne $1/\mu$. Tout d'abord, nous considérons la politique N-Policy comme un politique pour passer de l'état idle à l'état busy. La figure 4.1 représente le fonctionnement de capteur. Le capteur possède deux états principaux : l'état idle et l'état busy. Si le nombre de paquets dans le buffer est inférieur à le seuil N , donc notre système est

idle, une fois le nombre de paquet est égal à N , le système passe à l'état busy. Dans ce cas les paquets au niveau du buffer peuvent servir avec un taux de transition μ . Le plus important que le buffer reçoit un nombre limité de paquet K , une fois le nombre de paquet au niveau de buffer égal à zéro, le système retourne à l'état idle.

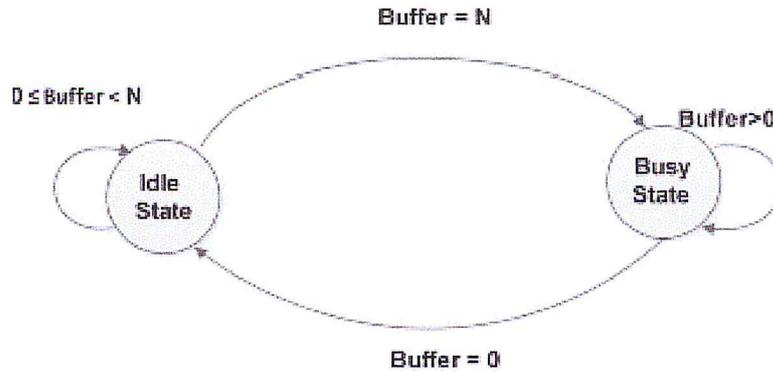


FIGURE 4.1 – Diagramme de transition d'état d'un nœud de capteur avec N-policy sans l'état sleep .[BG17]

4.2.2 Forme générale de la CMTC correspondante

4.2.2.1 Description de la chaîne générale

En observant notre chaîne de Markov nous avons constaté qu'elle peut être décrite par deux variables (i, j) tel que :

$X(i)$: Représente le nombre de paquet au niveau de buffer.

$X(j)$: Représente l'état de système. Si $X(j)=1$ il est dans l'état busy sinon l'état idle.

Le générateur infinitésimal Q est construit comme suit :

$$Q[(i, j), (X, Y)] = \begin{cases} Q[(i, j), (X, Y)] & \text{si } (i, j) \neq (X, Y) \\ -\sum_{\{(i, j) \neq (X, Y)\}} Q[(i, j), (X, Y)] & \text{si } (i, j) = (X, Y) \end{cases}$$

Où $Q[(i, j), (X, Y)]$ est la transition de l'état (i, j) vers l'état (X, Y) . En analysant notre CMTC nous obtenons les transitions suivantes :

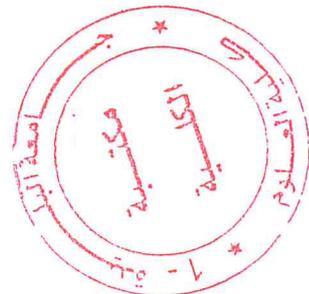


TABLE 4.1 – Représentation des taux de transitions entre les états

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

4.2.2.2 Formule générale du nombre d'états de la CMTC :

Soient N et K les deux paramètres de notre modèle tel que k représente la capacité de buffer et N le seuil de buffer à partir de ces deux paramètres nous pouvons déterminer le nombre total d'états de la CMTC est : $N+K$.

Algorithme de construction du générateur infinitésimal de la CMTC :

Algorithm 1 Algorithme de Modèle 1

```

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

```

Explication : Notre système avec une source fini donc a chaque fois que nous avons passé d'un état i a un état $i+1$ la taille de la source diminuer par 1 en cas générale le nombre de paquet dans la source il devient égal $R-i$ telle que :

R : la taille de la source .

i : le nombre de paquet au niveau de buffer.

4.2.2.3 La CMTC général du modèle N-policy sans l'état sleep

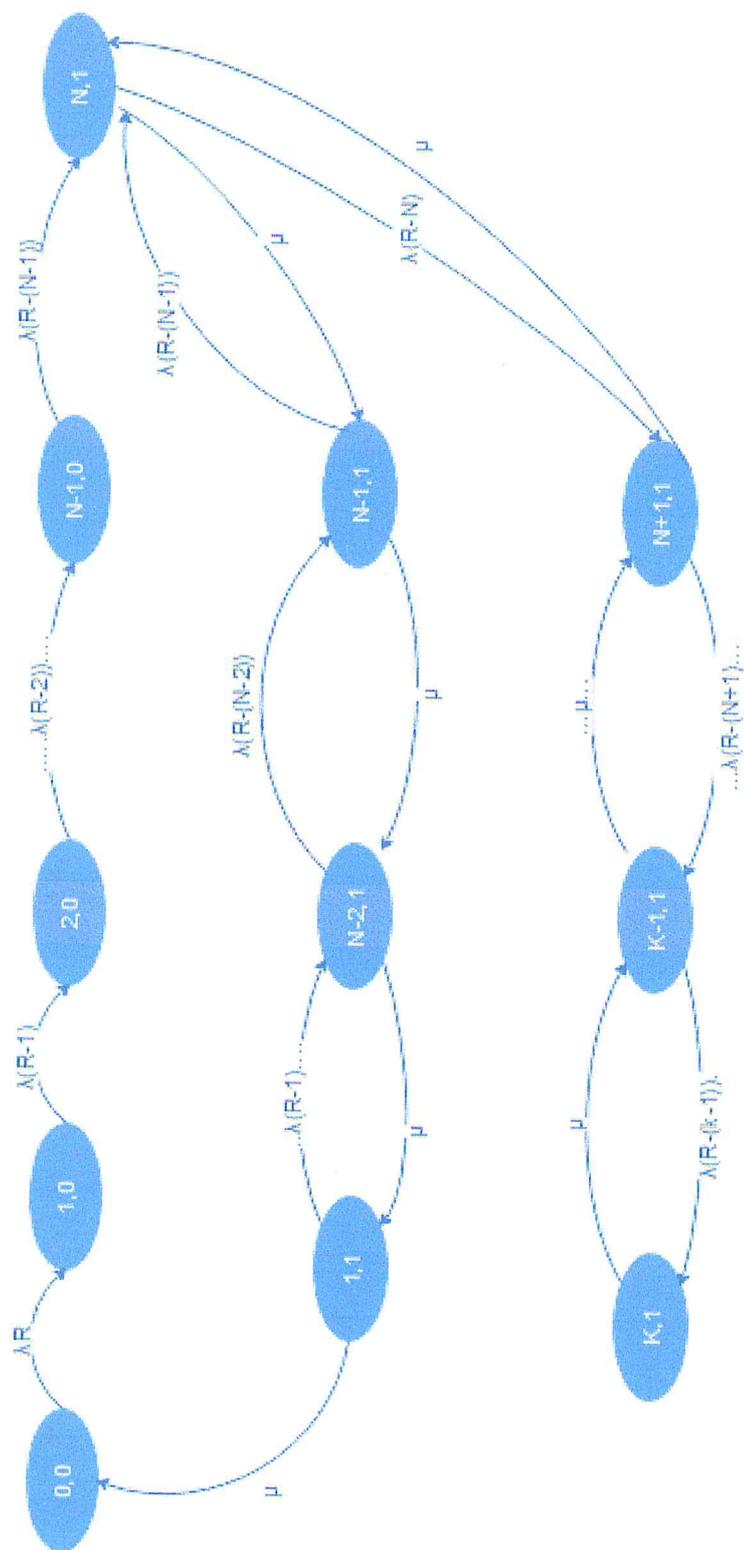


FIGURE 4.2 – CMTG général de modèle N-Policy sans l'état sleep .

4.2.3 Le modèle N-policy avec l'état sleep

Dans cette section, nous décrivons le modèle correspondant à un nœud de capteur dans sa période active et avec sleep. Nous considérons un nœud de capteur en tant que système de file d'attente avec vacances a buffer fini avec un seul serveur, dans lequel les paquets arrivent selon un processus quasirandom avec paramètre λ et les temps de service sont distribuée exponentiellement avec la moyenne $1/\mu$.

Au début notre système et dans l'état sleep (il est en vacance), une fois il reçoit un paquet il se réveille et passe à l'état idle. Le capteur reçoit les paquets qui transmettent avec un taux d'arrivée λ , une fois le nombre de paquet dans le buffer atteint un certain seuil N , le système passe à l'état busy, dans ce cas le capteur peut transmettre les paquets avec le taux de service μ , une fois tous les paquets au niveau de buffer servis le capteur revient à l'état sleep.

4.2.4 Forme générale de la CMTC correspondante

4.2.4.1 Description de la chaîne générale

En observant notre chaîne de Markov nous avons constaté qu'elle peut être décrite par trois variables (i, j, k) tel que :

$X(i)$: Représente le nombre de paquet au niveau de buffer.

$X(j)$: Représente l'état idle.

$X(k)$: Représente l'état busy.

Si $(X(j)=0 \text{ et } X(k)=0)$ le système est dans l'état sleep.

Si $(X(j)=1 \text{ et } X(k)=0)$ le système est dans l'état idle.

Si $(X(j)=0 \text{ et } X(k)=1)$ le système est dans l'état busy.

Le générateur infinitésimal Q est construit comme suit :

$$Q[(i, j, k), (X, Y, Z)] = \begin{cases} Q[(i, j, k), (X, Y, Z)] & \text{si } (i, j, k) \neq (X, Y, Z) \\ -\sum_{\{(i, j, k) \neq (X, Y, Z)\}} Q[(i, j, k), (X, Y, Z)] & \text{si } (i, j, k) = (X, Y, Z) \end{cases}$$

Où $Q[(i, j, k), (X, Y, Z)]$ est la transition de l'état (i, j, k) vers l'état (X, Y, Z) . En analysant notre CMTC nous obtenons les transitions suivantes :

TABLE 4.2 – Représentation des taux de transitions entre les états

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

4.2.4.2 Formule générale du nombre d'états de la CMTC :

Soient N et K les deux paramètres de notre modèle tel que k représente la capacité de buffer et N le seuil de buffer à partir de ces deux paramètres nous pouvons Déterminer le nombre total d'états de la CMTC par la formule suivante : $N+K$.

Algorithme de construction du générateur infinitésimal de la CMTC

Algorithm 2 Algorithme de Modèle 2

```

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

```

4.2.4.3 CMTC général de modèle N-Policy avec l'état sleep

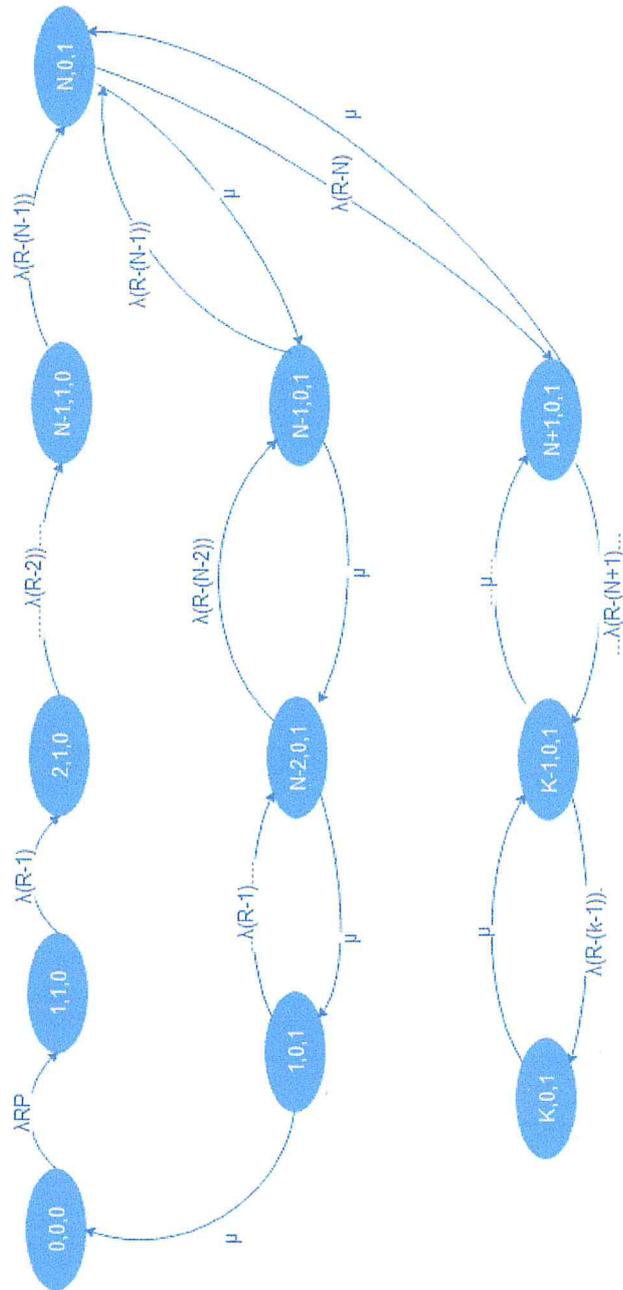


FIGURE 4.3 – CMTC général de modèle N-Policy avec l'état sleep .

4.2.5 Le modèle N-policy avec rappel

Nous considérons un capteur comme une file d'attente avec vacance et rappel .Les paquets arrivent dans le système avec un taux λ .

Au début le système est dans l'état idle, une fois le nombre de paquet au niveau de buffer atteindre un certain seuil N, le système passe à l'état busy, dans ce cas le capteur peut transmettre les paquets avec le taux de service μ . Si le buffer est plein et le système reçoit un nouveau paquet nous avons deux probabilités, telle que avec la probabilité $1-p$ le paquet est retransmis à la source (il quittera le système) sinon avec p il atteint l'espace orbite. Ce dernier fait le rappel jusqu'à ce que le paquet soit servi. Une fois le nombre de paquet dans l'espace orbite égal à zéro et tous les paquets au niveau buffer servis le système revient à l'état idle.

4.2.6 Forme générale de la CMTC correspondante

4.2.6.1 Description de la chaîne générale

En observant notre chaîne de Markov nous avons constaté qu'elle peut être décrite par trois variables (i, j, O) tel que :

$X(i)$: Représente le nombre de paquet au niveau de buffer.

$X(j)$: Représente l'état de système .Idle si $X(j)=0$ sinon l'état busy.

$X(O)$: Représente le nombre de paquet dans l'orbite.

Le générateur infinitésimal Q est construit comme suit :

$$Q[(i, j, O), (X, Y, Z')] = \begin{cases} Q[(i, j, O), (X, Y, Z')] & \text{si } (i, j, O) \neq (X, Y, Z') \\ -\sum_{\{(i, j, O) \neq (X, Y, Z')\}} Q[(i, j, O), (X, Y, Z')] & \text{si } (i, j, O) = (X, Y, Z') \end{cases}$$

Où $Q[(i, j, O), (X, Y, Z')]$ est la transition de l'état (i, j, O) vers l'état (X, Y, Z') . En analysant notre CMTC nous obtenons les transitions suivantes :

TABLE 4.3 – Représentation des taux de transitions entre les états

Description	Etat sortant	Taux de transition	Etat entrant
$0 \leq i \leq N - 1, j = 0, O = 0$	(i, j, O)	$\lambda(R-i)$	$(i+1, j, O)$
$i=N, j=1, O=0$	(i, j, O)	$\lambda(R-i)$	$(i+1, j, O)$
$i=N, j=1, O=0$	(i, j, O)	μ	$(i-1, j, O)$
$N+1 \leq i \leq K - 1, j = 1, O = 0$	(i, j, O)	$\lambda(R-i)$	$(i+1, j, O)$
$N+1 \leq i \leq K - 1, j = 1, O = 0$	(i, j, O)	μ	$(i-1, j, O)$
$i=K, j=1, O=0$	(i, j, O)	μ	$(i-1, j, O)$
$i=K, j=1, O=0$	(i, j, O)	$\lambda(R-i)P$	$(i, j, O+1)$
$2 \leq i \leq N - 1, j = 1, O = 0$	(i, j, O)	μ	$(i-1, j, O)$
$2 \leq i \leq N - 1, j = 1, O = 0$	(i, j, O)	$\lambda(R-i)$	$(i+1, j, O)$
$i=1, j=1, O=0$	(i, j, O)	μ	$(i-1, j-1, O)$
$i=1, j=1, O=0$	(i, j, O)	$\lambda(R-i)$	$(i+1, j, O)$
$i=1, j=1, O=R-k$	(i, j, O)	$\lambda(R - i - O)$	$(i+1, j, O)$
$2 \leq i \leq K - 1, j = 1, O = 1$	(i, j, O)	$\lambda(R-i-O)$	$(i+1, j, O)$
$2 \leq i \leq K - 1, j = 1, O = 1$	(i, j, O)	μ	$(i-1, j, O)$
$2 \leq i \leq K - 1, j = 1, O = 1$	(i, j, O)	$\lambda'O$	$(i+1, j, O-1)$
$i=1, j=1, O=1$	(i, j, O)	μ	$(i-1, j-1, O)$
$i=1, j=1, O=1$	(i, j, O)	$O\lambda'$	$(i+1, j, O-1)$
$0 \leq i \leq N - 2, j = 0, O = 1$	(i, j, O)	$\lambda(R-i-O)$	$(i+1, j, O)$
$0 \leq i \leq N - 2, j = 0, O = 1$	(i, j, O)	$O\lambda'$	$(i+1, j, O-1)$
$i=N-1, j=0, O=1$	(i, j, O)	$\lambda(R-i-O)$	$(i+1, j+1, O)$
$i=N-1, j=0, O=1$	(i, j, O)	$O\lambda'$	$(i+1, j+1, O-1)$
$i=K, j=1, O=1$	(i, j, O)	μ	$(i-1, j, O)$
$i=k, j=1, O=1$	(i, j, O)	$\lambda((R-i-O)P)$	$(i, j, O+1)$
$2 \leq i \leq K - 1, j = 1, O = R - K$	(i, j, O)	$\lambda(R-i-O)$	$(i+1, j, O)$
$2 \leq i \leq K - 1, j = 1, O = R - k$	(i, j, O)	μ	$(i-1, j, O)$
$2 \leq i \leq K - 1, j = 1, O = R - K$	(i, j, O)	$O\lambda'$	$(i+1, j, O-1)$
$i=1, j=1, O=R-K$	(i, j, O)	μ	$(i-1, j-1, O)$
$i=1, j=1, O=R-k$	(i, j, O)	$O\lambda'$	$(i+1, j, O-1)$
$i=1, j=1, O=R-k$	(i, j, O)	$\lambda(R - i - O)$	$(i+1, j, O)$
$i=K, j=1, O=R-K$	(i, j, O)	μ	$(i-1, j, O)$
$0 \leq i \leq N - 2, j = 0, O = R - K$	(i, j, O)	$\lambda(R-i-O)$	$(i+1, j, O)$
$0 \leq i \leq N - 2, j = 0, O = R - K$	(i, j, O)	$O\lambda'$	$(i+1, j, O-1)$
$i=N-1, j=0, O=R-k$	(i, j, O)	$O\lambda'$	$(i+1, j+1, O-1)$
$i=N-1, j=0, O=R-k$	(i, j, O)	$\lambda(R - i - O)$	$(i+1, j+1, O)$

4.2.6.2 Formule générale du nombre d'états de la CMTC

Soient N et K et R les trois paramètres de notre modèle tel que, K représente la capacité de buffer, N le seuil et le R représente la taille de la source (le nombre de paquet dans le système), a partir de ces trois paramètres nous pouvons déterminer le nombre total d'états de la CMTC par la formule suivante : $(R-K+1)(K+N)$.

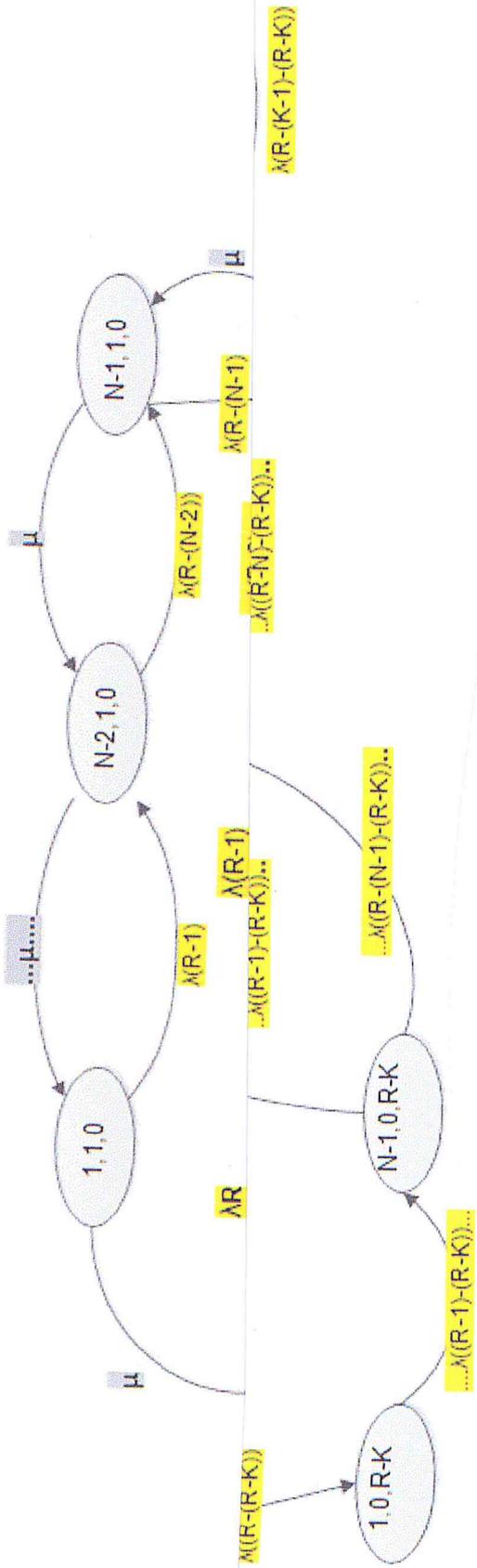
Algorithme de construction du générateur infinitésimal de la CMTC

Algorithm 3 Algorithme de Modèle 3

```

for  $j \leftarrow 0$  to 1 do
  if  $j = 0$  then
     $O \leftarrow 0$ ;
    for  $i \leftarrow 0$  to  $N - 1$  do
       $Q[(i, j, O)(i + 1, j, O)] \leftarrow \lambda(R - i)$ ;
    end for
     $O \leftarrow 1$ ;
    for  $i \leftarrow 0$  to  $N - 2$  do
       $Q[(i, j, O)(i + 1, j, O)] \leftarrow \lambda(R - i - O)$ ;
       $Q[(i, j, O)(i + 1, j, O - 1)] \leftarrow O\lambda'$ ;
    end for
     $i \leftarrow N - 1$ ;
     $Q[(i, j, O)(i + 1, j + 1, O)] \leftarrow \lambda(R - i - O)$ ;
     $Q[(i, j, O)(i + 1, j + 1, O - 1)] \leftarrow O\lambda'$ ;
     $O \leftarrow R - K$ ;
    for  $i \leftarrow 0$  to  $N - 2$  do
       $Q[(i, j, O)(i + 1, j, O)] \leftarrow \lambda(R - i - O)$ ;
       $Q[(i, j, O)(i + 1, j, O - 1)] \leftarrow O\lambda'$ ;
    end for
     $i \leftarrow N - 1$ ;
     $Q[(i, j, O)(i + 1, j + 1, O)] \leftarrow \lambda(R - i - O)$ ;
     $Q[(i, j, O)(i + 1, j + 1, O - 1)] \leftarrow O\lambda'$ ;
  else
     $O \leftarrow 0$ ;
     $i \leftarrow N$ ;
     $Q[(i, j, O)(i + 1, j, O)] \leftarrow \lambda(R - i)$ ;
     $Q[(i, j, O)(i - 1, j, O)] \leftarrow \mu$ ;
    for  $i \leftarrow N + 1$  to  $K - 1$  do
       $Q[(i, j, O)(i + 1, j, O)] \leftarrow \lambda(R - i)$ ;
       $Q[(i, j, O)(i - 1, j, O)] \leftarrow \mu$ ;
    end for
     $i \leftarrow K$ ;
     $Q[(i, j, O + 1)(i, j, O + 1)] \leftarrow \lambda((R - i)P)$ ;
     $Q[(i, j, O)(i - 1, j, O)] \leftarrow \mu$ ;
    for  $i \leftarrow 2$  to  $N - 1$  do
       $Q[(i, j, O)(i - 1, j, O)] \leftarrow \mu$ ;
       $Q[(i, j, O)(i + 1, j, O)] \leftarrow \lambda(R - i)$ ;
       $Q[(i, j, O)(i - 1, j, O)] \leftarrow \mu$ ;
    end for
     $i \leftarrow 1$ ;
     $Q[(i, j, O + 1)(i, j, O)] \leftarrow \lambda((R - i)$ ;
     $Q[(i, j, O)(i - 1, j - 1, O)] \leftarrow \mu$ ;
     $O \leftarrow 1$ ;
    for  $i \leftarrow 2$  to  $K - 1$  do
       $Q[(i, j, O)(i + 1, j, O)] \leftarrow \lambda(R - i - O)$ ;
       $Q[(i, j, O)(i - 1, j, O)] \leftarrow \mu$ ;
       $Q[(i, j, O)(i + 1, j, O - 1)] \leftarrow O\lambda'$ ;
    end for
     $i \leftarrow 1$ ;
     $Q[(i, j, O)(i - 1, j, O)] \leftarrow \mu$ ;
     $Q[(i, j, O)(i + 1, j, O - 1)] \leftarrow O\lambda'$ ;
     $i \leftarrow K$ ;
     $Q[(i, j, O + 1)(i - 1, j, O)] \leftarrow \mu$ ;
     $Q[(i, j, O)(i, j, O + 1)] \leftarrow \lambda(R - i - O)P$ ;
     $O \leftarrow R - K$ ;
    for  $i \leftarrow 2$  to  $K - 1$  do
       $Q[(i, j, O)(i + 1, j, O)] \leftarrow \lambda(R - i - O)$ ;
       $Q[(i, j, O)(i - 1, j, O)] \leftarrow \mu$ ;
       $Q[(i, j, O)(i + 1, j, O - 1)] \leftarrow O\lambda'$ ;
    end for
     $i \leftarrow 1$ ;
     $Q[(i, j, O + 1)(i - 1, j - 1, O)] \leftarrow \mu$ ;
     $Q[(i, j, O)(i + 1, j, O - 1)] \leftarrow O\lambda'$ ;
     $i \leftarrow K$ ;
     $Q[(i, j, O + 1)(i - 1, j, O)] \leftarrow \mu$ ;
  end if
end for

```



4.2.7 Le modèle N-policy avec rappel et avec l'état sleep

Nous considérons un capteur comme une file d'attente avec vacance et avec rappel. Les paquets arrivent dans le système avec un taux λ , et le transmettre avec le taux μ . Nous décrivons le modèle correspondant à un nœud de capteur dans sa période active et sleep. Au début le capteur est dans l'état sleep, une fois un paquet arrivé nous allons deux probabilités, soit le capteur réveil avec la probabilité P , ou bien le paquet atteindre l'espace orbite avec la probabilité q . Si le système réveil nous passe à l'état idle, et le buffer reçoit les paquets avec le taux d'arrivée λ une fois le nombre de paquet dans le capteur atteindre un certain seuil N , le système passe à l'état busy, dans ce cas le capteur peut transmettre les paquets avec un taux de transition μ . Si le buffer est plein et nous arrivons un nouveau paquet qui arrive le paquet transmettre avec le taux d'arrivée λ et la probabilité q vers l'orbite, dans ce cas l'orbite peut recevoir jusqu'à $R - k$ paquets, et il fait le rappel jusqu'à le paquet. Lorsque le nombre des paquets dans buffer égal à zéro le système passe à l'état sleep. Sinon comme le capteur réveil à l'arrivée d'un nouveau paquet il peut aussi réveil si l'orbite envoie un paquet (paquet secondaire) et le système fait les mêmes états précédents. L'orbite peut recevoir R paquet dans le cas où les paquets arrivés trouvent le système en vacances.

4.2.8 Forme générale de la CMTC correspondante

4.2.8.1 Description de la chaîne générale

En observant notre chaîne de Markov nous avons constaté qu'elle peut être décrite par quatre variables (i, j, k, O) tel que :

$X(i)$: Représente le nombre de paquet au niveau de buffer.

$X(j)$: Représente l'état idle.

$X(k)$: Représente l'état busy.

Si $(X(j)=0, X(k)=0)$ le système est dans l'état sleep.

Si $(X(j)=1, X(k)=0)$ le système est dans l'état idle.

Si $(X(j)=0, X(k)=1)$ le système est dans l'état busy.

$X(O)$: Représente le nombre de paquet au niveau de l'orbite

Le générateur infinitésimal Q est construit comme suit :

$$Q = \begin{cases} Q[(i, j, k, O), (X, Y, Z, Z')] & \text{si } (i, j, k, O) \neq (X, Y, Z, Z') \\ -\sum_{\{(i, j, k, O) \neq (X, Y, Z, Z')\}} Q[(i, j, k, O), (X, Y, Z, Z')] & \text{si } (i, j, k, O) = (X, Y, Z, Z') \end{cases}$$

Où $Q[(i, j, k, O), (X, Y, Z, Z')]$ est la transition de l'état (i, j, k, O) vers l'état (X, Y, Z, Z') .

En analysant notre CMTC nous obtenons les transitions suivantes :

TABLE 4.4 – Représentation des taux de transitions entre les états

Description	Etat sortant	Taux de transition	Etat entrant
$i=0, j=0, k=0, O=0$	(i, j, k, O)	$\lambda R P$	$(i+1, j+1, k, O)$
$i=0, j=0, k=0, O=0$	(i, j, k, O)	$\lambda R q$	$(i, j, k, O+1)$
$1 \leq i \leq N-1, j=1, k=0, O=0$	(i, j, k, O)	$\lambda(R-i)$	$(i+1, j, k, O)$
$i=N, j=0, k=1, O=0$	(i, j, k, O)	$\lambda(R-i)$	$(i+1, j, k, O)$
$i=N, j=0, k=1, O=0$	(i, j, k, O)	μ	$(i-1, j, k, O)$
$N+1 \leq i \leq K-1, j=0, k=1, O=0$	(i, j, k, O)	$\lambda(R-i)$	$(i+1, j, k, O)$
$N+1 \leq i \leq K-1, j=0, k=1, O=0$	(i, j, k, O)	μ	$(i-1, j, k, O)$
$i=K, j=0, k=1, O=0$	(i, j, k, O)	$\lambda(R-K)q$	$(i, j, k, O+1)$
$i=K, j=0, k=1, O=0$	(i, j, k, O)	μ	$(i-1, j, k, O)$
$2 \leq i \leq N-1, j=0, k=1, O=0$	(i, j, k, O)	μ	$(i-1, j, k, O)$
$2 \leq i \leq N-1, j=0, k=1, O=0$	(i, j, k, O)	$\lambda(R-i)$	$(i+1, j, k, O)$
$i=1, j=0, k=1; O=0$	(i, j, k, O)	μ	$(i-1, j, k-1, O)$
$i=1, j=0, k=1; O=0$	(i, j, k, O)	$\lambda(R-i)$	$(i+1, j, k, O)$
$2 \leq i \leq K-1, j=0, k=1, O=1$	(i, j, k, O)	$\lambda(R-i-O)$	$(i+1, j, k, O)$
$2 \leq i \leq K-1, j=0, k=1, O=1$	(i, j, k, O)	μ	$(i-1, j, k, O)$
$2 \leq i \leq K-1, j=0, k=1, O=1$	(i, j, k, O)	$O\lambda'$	$(i+1, j, k, O-1)$
$i=1, j=0, k=1, O=1$	(i, j, k, O)	$O\lambda'$	$(i+1, j, k, O-1)$
$i=1, j=0, k=1, O=1$	(i, j, k, O)	μ	$(i-1, j, k-1, O)$
$i=0, j=0, k=0, O=1$	(i, j, k, O)	$\lambda(R-O)P$	$(i+1, j+1, k, O)$
$i=0, j=0, k=0, O=1$	(i, j, k, O)	$\lambda(R-O)q$	$(i, j, k, O+1)$
$i=0, j=0, k=0, O=1$	(i, j, k, O)	$O\lambda'$	$(i+1, j+1, k, O-1)$
$1 \leq i \leq N-2, j=1, k=0, O=1$	(i, j, k, O)	$\lambda(R-i-O)$	$(i+1, j, k, O)$
$1 \leq i \leq N-2, j=1, k=0, O=1$	(i, j, k, O)	$O\lambda'$	$(i+1, j, k, O-1)$
$i=N-1, j=1, k=0, O=1$	(i, j, k, O)	$O\lambda'$	$(i+1, j-1, k+1, O-1)$
$i=N-1, j=1, k=0, O=1$	(i, j, k, O)	$\lambda R-i-O$	$(i+1, j-1, k+1, O)$
$i=K, j=0, k=1, O=1$	(i, j, k, O)	$\lambda R-i-O)q$	$(i, j, k, O+1)$
$i=K, j=0, k=1, O=1$	(i, j, k, O)	μ	$(i-1, j, k, O)$
$2 \leq i \leq K-1, j=0, k=1, O=R-K$	(i, j, k, O)	$\lambda(R-i-O)$	$(i+1, j, k, O)$
$2 \leq i \leq K-1, j=0, k=1, O=R-K$	(i, j, k, O)	μ	$(i-1, j, k, O)$
$2 \leq i \leq K-1, j=0, k=1, O=R-K$	(i, j, k, O)	$O\lambda'$	$(i+1, j, k, O-1)$
$i=1, j=0, k=1, O=R-K$	(i, j, k, O)	$O\lambda'$	$(i+1, j, k, O-1)$
$i=1, j=0, k=1, O=R-K$	(i, j, k, O)	μ	$(i-1, j, k-1, O)$
$i=0, j=0, k=0, O=R-K$	(i, j, k, O)	$\lambda(R-O)p$	$(i+1, j+1, k, O)$
$i=0, j=0, k=0, O=R-K$	(i, j, k, O)	$\lambda(R-O)q$	$(i, j, k, O+1)$
$i=0, j=0, k=0, O=R-K$	(i, j, k, O)	$O\lambda'$	$(i+1, j+1, k, O-1)$
$1 \leq i \leq N-2, j=1, k=0, O=R-K$	(i, j, k, O)	$\lambda(R-i-O)$	$(i+1, j, k, O)$
$1 \leq i \leq N-2, j=1, k=0, O=R-K$	(i, j, k, O)	$O\lambda'$	$(i+1, j, k, O-1)$
$i=N-1, j=1, k=0, O=R-K$	(i, j, k, O)	$O\lambda'$	$(i+1, j-1, k+1, O-1)$
$i=N-1, j=1, k=0, O=R-K$	(i, j, k, O)	$\lambda(R-i-O)$	$(i+1, j-1, k+1, O)$
$i=0, j=0, k=0, O=R-1$	(i, j, k, O)	$\lambda(R-O)P$	$(i+1, j+1, k, O)$
$i=0, j=0, k=0, O=R-1$	(i, j, k, O)	$\lambda(R-O)q$	$(i, j, k, O+1)$
$i=0, j=0, k=0, O=R-1$	(i, j, k, O)	$O\lambda'$	$(i+1, j+1, k, O-1)$
$i=0, j=0, k=0, O=R$	(i, j, k, O)	$O\lambda'$	$(i+1, j+1, k, O-1)$

4.2.8.2 Formule générale du nombre d'états de la CMTC

Soient N et K et R les trois paramètres de notre modèle tel que k représente la capacité de buffer, N le seuil et le R représente la taille de la source a partir de ces trois paramètres nous pouvons déterminer le nombre total d'états de la CMTC par la formule suivante : $(R-K+1)(K+N) + \sum_{i=1}^K i + (N-1)(K-N)$.

Algorithme de construction du générateur infinitésimal de la CMTC

Algorithm 4 Algorithme modèle 4

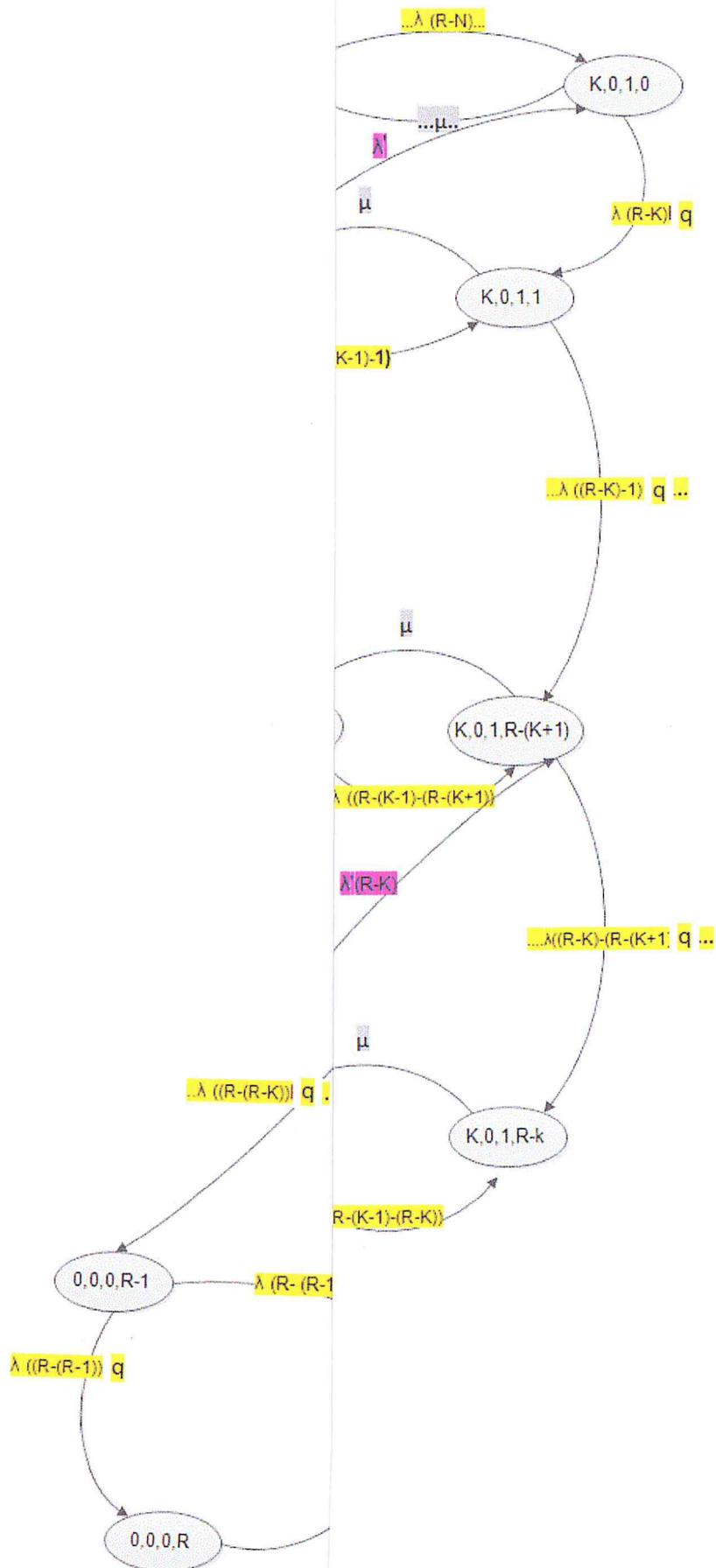
```

for  $k \leftarrow 0$  to 1 do
  if  $k = 0$  then
    for  $j \leftarrow 0$  to 1 do
      if  $j = 0$  then
         $i \leftarrow 0; O \leftarrow 0;$ 
         $Q[(i, j, k, O)(i + 1, j + 1, k, O)] \leftarrow \lambda R P;$ 
         $Q[(i, j, k, O)(i, j, k, O + 1)] \leftarrow \lambda R q;$ 
         $O \leftarrow 1;$ 
         $Q[(i, j, k, O)(i + 1, j + 1, k, O)] \leftarrow \lambda(R - O)P;$ 
         $Q[(i, j, k, O)(i, j, k, O + 1)] \leftarrow \lambda(R - O)q;$ 
         $Q[(i, j, k, O)(i + 1, j + 1, k, O - 1)] \leftarrow O\lambda';$ 
         $O \leftarrow R - K;$ 
         $Q[(i, j, k, O)(i + 1, j + 1, k, O)] \leftarrow \lambda(R - O)P;$ 
         $Q[(i, j, k, O)(i, j, k, O + 1)] \leftarrow \lambda(R - O)q;$ 
         $Q[(i, j, k, O)(i + 1, j + 1, k, O - 1)] \leftarrow O\lambda';$ 
         $O \leftarrow R;$ 
         $Q[(i, j, k, O)(i + 1, j + 1, k, O - 1)] \leftarrow O\lambda';$ 
      else
         $O \leftarrow 0;$ 
        for  $i \leftarrow 1$  to  $N - 1$  do
           $Q[(i, j, k, O)(i + 1, j, k, O)] \leftarrow \lambda(R - i);$ 
        end for
         $O \leftarrow 1;$ 
        for  $i \leftarrow 1$  to  $N - 2$  do
           $Q[(i, j, k, O)(i + 1, j, k, O)] \leftarrow \lambda(R - i - O);$ 
           $Q[(i, j, k, O)(i + 1, j, k, O - 1)] \leftarrow O\lambda';$ 
        end for
         $i \leftarrow N - 1;$ 
         $Q[(i, j, k, O)(i + 1, j - 1, k + 1, O)] \leftarrow \lambda(R - i - O);$ 
         $Q[(i, j, k, O)(i + 1, j - 1, k + 1, O - 1)] \leftarrow O\lambda';$ 
         $O \leftarrow R - K;$ 
        for  $i \leftarrow 1$  to  $N - 2$  do
           $Q[(i, j, k, O)(i + 1, j, k, O)] \leftarrow \lambda(R - i - O);$ 
           $Q[(i, j, k, O)(i + 1, j, k, O - 1)] \leftarrow O\lambda';$ 
        end for
         $i \leftarrow N - 1;$ 
         $Q[(i, j, k, O)(i + 1, j - 1, k + 1, O)] \leftarrow O\lambda';$ 
      end if
    end for
  else
     $1) \leftarrow O\lambda';$ 
  end if
end for
else
  for  $j \leftarrow 0$  to 1 do
    if  $j = 0$  then
       $O \leftarrow 0;$ 
      for  $i \leftarrow N + 1$  to  $K - 1$  do
         $Q[(i, j, k, O)(i + 1, j, k, O)] \leftarrow \lambda(R - i);$ 
         $Q[(i, j, k, O)(i - 1, j, k, O)] \leftarrow \mu;$ 
      end for
       $i \leftarrow N;$ 
       $Q[(i, j, k, O)(i + 1, j, k, O)] \leftarrow \lambda(R - i);$ 
       $Q[(i, j, k, O)(i - 1, j, k, O)] \leftarrow \mu;$ 
       $i \leftarrow K;$ 
       $Q[(i, j, k, O)(i, j, k, O + 1)] \leftarrow \lambda(R - i)q;$ 
       $Q[(i, j, k, O)(i - 1, j, k, O)] \leftarrow \mu;$ 
       $O \leftarrow 1;$ 
      for  $i \leftarrow 2$  to  $K - 1$  do
         $Q[(i, j, k, O)(i + 1, j, k, O)] \leftarrow \lambda(R - i - O);$ 
         $Q[(i, j, k, O)(i - 1, j, k, O)] \leftarrow \mu;$ 
      end for
       $i \leftarrow 1;$ 
       $Q[(i, j, k, O)(i + 1, j, k, O - 1)] \leftarrow O\lambda';$ 
       $Q[(i, j, k, O)(i - 1, j - 1, k, O)] \leftarrow \mu;$ 
       $i \leftarrow K;$ 
       $Q[(i, j, k, O)(i, j, k, O + 1)] \leftarrow O\lambda(R - i)q;$ 
       $Q[(i, j, k, O)(i - 1, j, k, O)] \leftarrow \mu;$ 
       $O \leftarrow R - K;$ 
      for  $i \leftarrow 2$  to  $K - 1$  do
         $Q[(i, j, k, O)(i + 1, j, k, O)] \leftarrow \lambda(R - i - O);$ 
         $Q[(i, j, k, O)(i - 1, j, k, O)] \leftarrow \mu;$ 
         $Q[(i, j, k, O)(i + 1, j, k, O - 1)] \leftarrow O\lambda';$ 
      end for
       $i \leftarrow 1;$ 
       $Q[(i, j, k, O)(i + 1, j, k, O - 1)] \leftarrow O\lambda';$ 
       $Q[(i, j, k, O)(i - 1, j, k - 1, O)] \leftarrow \mu;$ 
       $Q[(i, j, k, O)(i + 1, j, k, O)] \leftarrow \lambda(R - i - O);$ 
    end if
  end for
end if end for

```

Explication : Notre système avec un source fini R , donc a chaque fois que nous avons passé d'un état i a un état $i+1$, la taille de la source diminuer par 1. En générale Le nombre de paquet dans la source après chaque transition = $R-i$, une fois il exist des paquets au niveau de l'orbite donc le taux d'arrivé d'une état a une autre (i ver $i+1$) = λ $((R-i)-O)$. (O le nombre de paquet au niveau de l'orbite).

4.2.8.3 CMTc général de modèle N-Policy avec l'état orbite et l'état sleep.



4.3 Les indices de performance

En appliquant les algorithmes donnés ci-dessus, le générateur infinitésimal Q est obtenu, ce qui permet le calcul du vecteur des probabilités stationnaires par une résolution du système d'équation :

$$\begin{cases} \pi Q = 0 \\ \sum_{(i,j,k)} \pi_{ijk} = 1 \end{cases}$$

Une fois ces probabilités obtenues, divers indices de performance peuvent être calculés. En appliquant Les formules que nous avons développées et qui sont données ci-après :

N-Policy sans l'état sleep :

1. La probabilité que le capteur est dans l'état idle :

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

2. La probabilité que le capteur est dans l'état busy :

$$P_{busy} = 1 - P_{idle}$$

3. La longueur moyenne de buffer \bar{Q} : (il représente le nombre moyen de paquets en attente dans le capteur .)

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

4. Le débit de réception de paquets $\bar{\lambda}$: Cela correspond au taux effectif de réception des paquets par le nœud de capteur :

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

5. Le temps d'attente moyen des paquets dans le buffer \bar{W} :

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

6. Les formules little :

- (a) La durée moyenne d'une période occupée (busy) \bar{B} :

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

(b) La durée moyenne d'une période libre (idle) \bar{T} :

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

(c) La durée moyenne d'un cycle :

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

(d) Le nombre de cycles (N_C) : Il correspond à le nombre de transitions de l'état idle à l'état busy par unité de temps :

$$N_C = \frac{1}{C}$$

7. Le taux moyenne de service $\bar{\mu}$:

$$\bar{\mu} = \mu \cdot P_{busy}$$

8. La consommation d'énergie moyenne d'un nœud de capteur (EC) :

$$EC = EC_{(idle)} \cdot P_{(idle)} + EC_{(busy)} \cdot P_{(busy)} + EC_S \cdot N_C + EC_{TX} \cdot \bar{Q}$$

$EC_{(idle)}$: Consommation d'énergie tandis que l'unité de transmission est en état de idle.

$EC_{(busy)}$: Consommation d'énergie pendant que l'unité de transmission est en état busy.

EC_{TX} : Consommation d'énergie pour la transmission de chaque paquet présent dans le nœud du capteur.

EC_S : Représente la consommation d'énergie du changement de l'état de capteur entre réception et transmission.

N-policy avec l'état sleep :

1. La probabilité que le capteur est dans l'état sleep :

$$P_{Sleep} = \pi_{0,0,0}.$$

2. La probabilité que le capteur est dans l'état idle :

$$P_{idle} = \sum_{i=1}^{N-1} \pi_{i,1,0}.$$

3. La probabilité que le capteur est dans l'état busy :

$$P_{busy} = 1 - (P_{sleep} + P_{idle}).$$

4. La longueur moyenne buffer (\bar{Q}) : (Il représente le nombre moyen de paquets en attente dans le capteur)

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

5. Le débit de réception de paquets $\bar{\lambda}$: Cela correspond au taux effectif de réception des paquets par le nœud de capteur :

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

6. Les formules little :

- (a) La durée moyenne d'une période vacance (sleep) \bar{S} :

$$\bar{S} = \frac{1}{\lambda P}$$

- (b) La durée moyenne d'un cycle :

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

- (c) Le nombre de cycles (N_C) : Il correspond à le nombre de transitions de l'état idle à l'état busy par unité de temps :

$$N_C = \frac{1}{\bar{C}}$$

7. La consommation d'énergie moyenne d'un nœud de capteur (EC) :

$$EC = EC_{(sleep)} \cdot P_{(sleep)} + EC_{(idle)} \cdot P_{(idle)} + EC_{(busy)} \cdot P_{(busy)} + EC_S \cdot N_C + EC_{TX} \cdot \bar{Q}.$$

$EC_{(sleep)}$: Consommation d'énergie pendant que l'unité de transmission est en état sleep.

N-policy avec rappel :

1. La probabilité que le capteur est dans l'état idle :

$$P_{idle} = \sum_{i=0}^{N-1} \sum_{O=0}^{R-K} \pi_{i,0,O}.$$

2. La probabilité que le capteur est dans l'état busy :

$$P_{busy} = 1 - (P_{idel}).$$

3. Le nombre moyenne de paquet dans l'orbite :

$$Q_{orbite} = \sum_{i=0}^K \sum_{j=0}^1 \sum_{O=1}^{R-K} O \pi_{i,j,O} + \sum_{i=1}^{N-1} \sum_{O=0}^{R-K} O \pi_{i,0,O}$$

4. La longueur moyenne buffer (\bar{Q}) : (Il représente le nombre moyen de paquets en attente dans le capteur)

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

5. La consommation d'énergie moyenne d'un nœud de capteur (EC) :

$$EC = EC_{(idle)} \cdot P_{(idle)} + EC_{(busy)} \cdot P_{(busy)} + EC_S \cdot N_C + EC_{TX} \cdot \bar{Q} + EC_{RX} \cdot Q_{orbite}$$

EC_{RX} : Consommation d'énergie pour la retransmission de chaque paquet présent dans l'orbite.

N-policy avec rappel et l'état sleep

1. La probabilité que le capteur est dans l'état sleep :

$$P_{Sleep} = \pi_{0,0,0,O}.$$

2. La probabilité que le capteur est dans l'état idle :

$$P_{idle} = \sum_{i=0}^{N-1} \sum_{O=0}^R \pi_{i,1,0,O}.$$

3. La probabilité que le capteur est dans l'état busy :

$$P_{busy} = 1 - (P_{Sleep} + P_{idle}).$$

4. Le nombre moyenne de paquet dans l'orbite :

$$Q_{orbite} = \sum_{i=0}^K \sum_{j=0}^1 \sum_{K=0}^1 \sum_{O=0}^{R-K} O \pi_{i,j,K,O} + \sum_{i=0}^{N-1} \sum_{O=0}^{R-K} O \pi_{i,1,0,O} + \sum_{i=0}^K \sum_{j=0}^1 \sum_{K=0}^1 \sum_{O=R-K}^{(R-K)+(N-1)} O \pi_{i,j,K,O} + \sum_{i=0}^{N-1} \sum_{O=R-K}^{(R-K)+(N-1)} O \pi_{i,1,0,O} + \sum_{i=0}^{(N-1)} \sum_{(R-K)+(N-1)}^R O \pi_{i,1,0,O}.$$

5. La longueur moyenne buffer (\bar{Q}) : (Il représente le nombre moyen de paquets en attente dans le capteur) :

$$\bar{Q} = \sum_{i=0}^K \sum_{j=0}^1 \sum_{K=0}^1 \sum_{O=0}^{R-K} i \pi_{i,j,K,O} + \sum_{i=0}^{N-1} i \pi_{i,0,1,O} + \sum_{i=0}^{N-1} \sum_{O=0}^{R-K} i \pi_{i,1,0,O} + \sum_{i=0}^K \sum_{j=0}^1 \sum_{K=0}^1 \sum_{O=R-K}^{(R-K)+(N-1)} i \pi_{i,j,K,O} + \sum_{i=0}^{N-1} \sum_{O=R-K}^{(R-K)+(N-1)} i \pi_{i,1,0,O} + \sum_{i=0}^{(N-1)} \sum_{(R-K)+(N-1)}^R i \pi_{i,1,0,O}.$$

6. La consommation d'énergie moyenne d'un nœud de capteur (EC) :

$$EC = EC_{(sleep)} \cdot P_{(sleep)} + EC_{(idle)} \cdot P_{(idle)} + EC_{(busy)} \cdot P_{(busy)} + EC_S \cdot N_C + EC_{TX} \cdot \bar{Q} + EC_{RX} \cdot Q_{(orbite)}$$

4.4 Conclusion

Dans ce chapitre nous présentés les modèles proposés pour étudier l'impact de la retransmission des paquets sur la politique de conservation d'énergie N-Policy. Les CMTC ont été utilisés comme méthode de modélisation et d'évaluation des performances. Nous avons proposé pour chaque modèle un algorithme qui permet de calculer le générateur infinitésimal Q , et utilisant uniquement les paramètres de capteur tels que le taux d'arrivée, le taux de service, la capacité de buffer K , le seuil N , la consommation d'énergie EC . Les formules des indices de performances ont été élaborées en se basant sur le CMTC.

Implémentation et étude expérimentale

5.1 Introduction

Après avoir présenté et bien détaillé les modèles proposés dans le chapitre de modélisation, notre objectif à cette étape est l'implémentation d'une application qui se charge de calculer les indices de performances, ainsi que la consommation d'énergie d'un nœud capteur pour chaque modèle proposé précédemment. Nous commençons ce chapitre par une présentation sur l'environnement logiciel utilisé, puis nous présentons notre application et finalement nous allons traiter une étude expérimentale.

5.1.1 L'environnement logiciel

La technologie java définit à la fois un langage de programmation et une plateforme informatique. Créée par l'entreprise "Microsystems", et reprise par la société "Oracle", la technologie java est indissociable du domaine de l'informatique et du web. On la retrouve donc sur les ordinateurs, mais aussi sur les téléphones mobiles, les consoles de jeux, ...ect[22]

5.2 Présentation de l'application

5.2.1 Fenêtre d'accueil

Lors du lancement de l'application, s'ouvre la fenêtre d'accueil permettant de commencer la manipulation du logiciel, La fenêtre contient un menu composé d'un seul élément "Aide", et deux boutons : "Démarrer" et "Quitter".

Une fois on démarre l'application il s'affiche un fenêtre qui compose d'un bouton quitter et un menu qui contiens un seul élément modèle. Lorsque on clique sur le menu

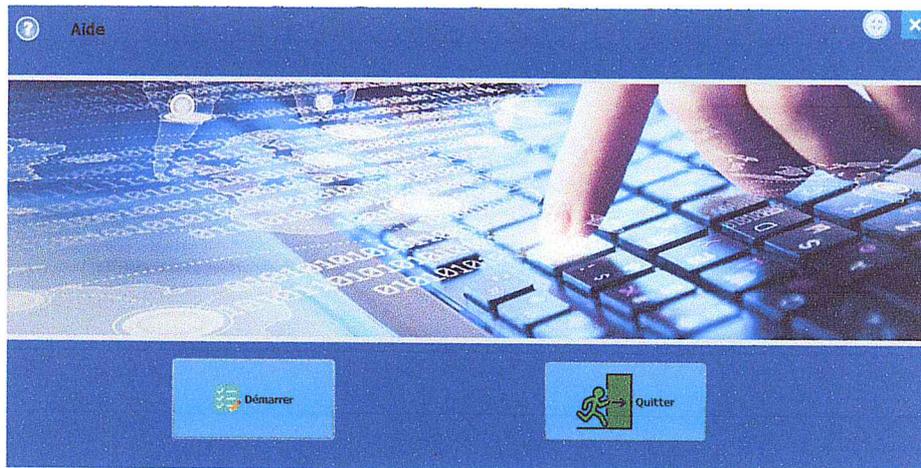


FIGURE 5.1 – Fenêtre d'accueil de l'application .

modèle on obtiens quatre modèles a choisir comme nous allons voir dans la figure ci-dessous :

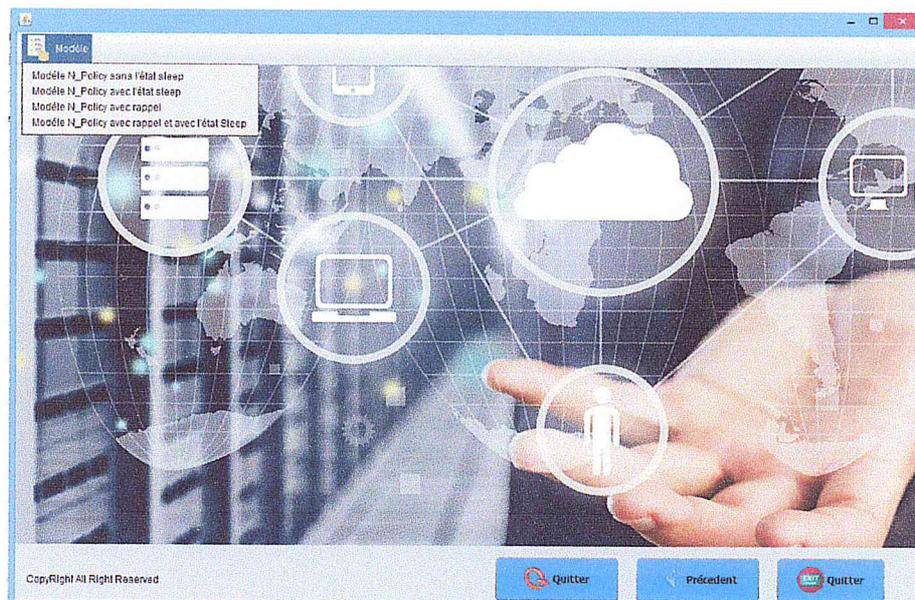


FIGURE 5.2 – Menu principale des modèles

5.2.2 Fenêtre de résultats de calcul.

Cette fenêtre pour saisir les paramètres de système afin d'afficher les indices de performance calculés à l'aide des formules données dans la section précédente.

Paramètre	Valeur
K	15
H	7
R	20
Lamda	1.5
lamda'	0.5
P	0.5
mu	2
La probabilité que le capteur est dans l'état idle	0.04279881778876271
La probabilité que le capteur est dans l'état busy	0.9572011622112173
Le nombre moyen de paquets en attente dans l'orbite	0.49445503599540003
Le nombre moyen de paquets en attente dans le capteur	12.79635268115463
Le débit de réception de paquets	6.398176240577315
La durée moyenne d'une période occupée (busy)	6.398176240577315
La durée moyenne d'une période libre (idle)	14.0
La durée moyenne d'un cycle	20.398176340577315
Le nombre de cycles	0.049023990348173159
Le taux moyen de service	1.9144023644224348
La consommation d'énergie d'un nœud de capteur	559.4178621644759

FIGURE 5.3 – Interface d'analyse du modèle N-policy avec l'état sleep .

5.3 Etude expérimentale

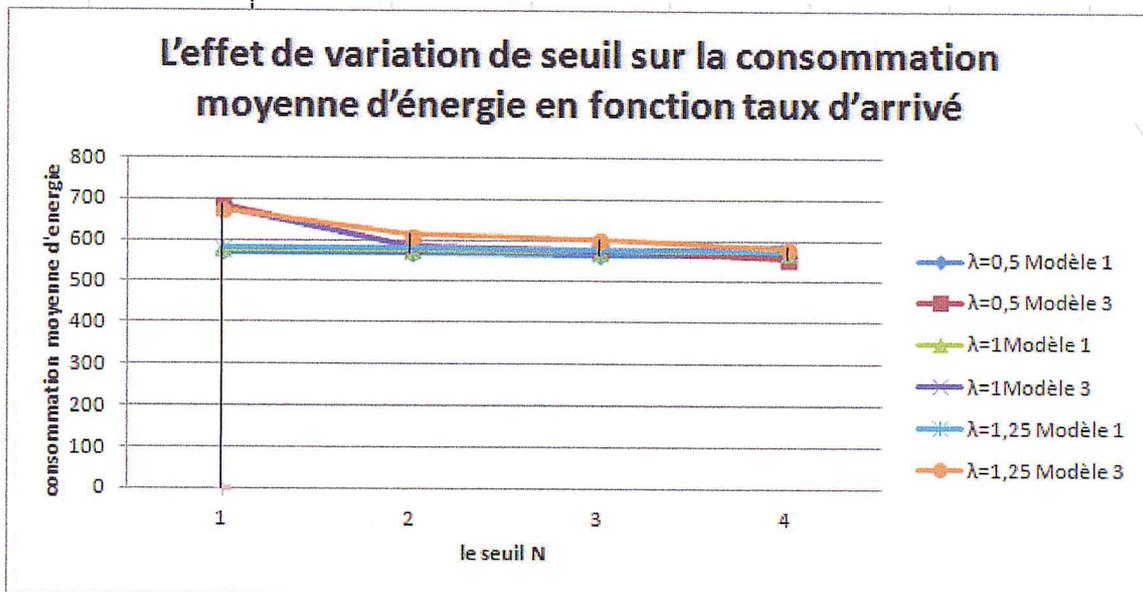
Pour étudier l'impact de la retransmission des paquets, une étude expérimentale a été mesurée. Les paramètres de configuration du réseau de capteur sans fil sur les performances de ce dernier. Les paramètres considérés sont le taux d'arrivée, le taux de service, le seuil N et la capacité du buffer de réseau de capteur. Le tableau suivant résume les valeurs :

TABLE 5.1 – Paramètres du système.

	R	K	λ	λ'	μ	N'	p	q	P_{idle}	P_{sleep}	P_{busy}	$\bar{\lambda}$	\bar{Q}
1	20	10	0.5	-	2	3	0.5		0.00000166	-	0.999998	1.491959	9.491939
2	20	10	0.5	-	2	3	0.5		0.000001619	0.00000085	0.9999975	1.491958	4.91631
3	20	10	0.5	0.5	2	3	0.5		0.01269	-	0.9873	4.064	8.1297
4	20	10	0.5	0.5	2	3	0.5	0.6	0.0121	0.0000003	0.987	3.52	7.044

5.3.1 Le taux d'arrivée

1. Les courbes graphiques de la figure 5.4 représentent les différentes valeurs obtenues en mesurant la consommation moyenne d'énergie, par rapport a le seuil N en fonction de taux d'arrivé ,avec les deux modèle N-policy sans l'état sleep et le modèle N-policy avec rappel.

FIGURE 5.4 – La consommation moyenne d'énergie par rapport à seuil N, λ : 0.5 - 1.25.

Analyse : Nous constatons que l'augmentation du taux d'arrivée influe sur la consommation moyenne d'énergie. La consommation moyenne d'énergie diminue, lorsque le seuil de la file d'attente augmente dans les deux modèles. Après la comparaison de modèle N-Policy sans l'état sleep, et N-Policy avec rappel, nous avons trouvé pour chaque valeur de λ proposé, la consommation de l'énergie moyenne de modèle avec rappel est supérieur a la consommation d'énergie moyenne par rapport au modèle N-Policy sans l'état sleep.

2. Les courbes graphiques de la figure 5.5 représentent les différentes valeurs obtenues en mesurant la consommation moyenne d'énergie par rapport a le seuil N en fonction de taux d'arrivé ,avec les deux modèle N-policy avec l'état sleep et

le modèle N-policy avec l'état sleep et avec rappel.

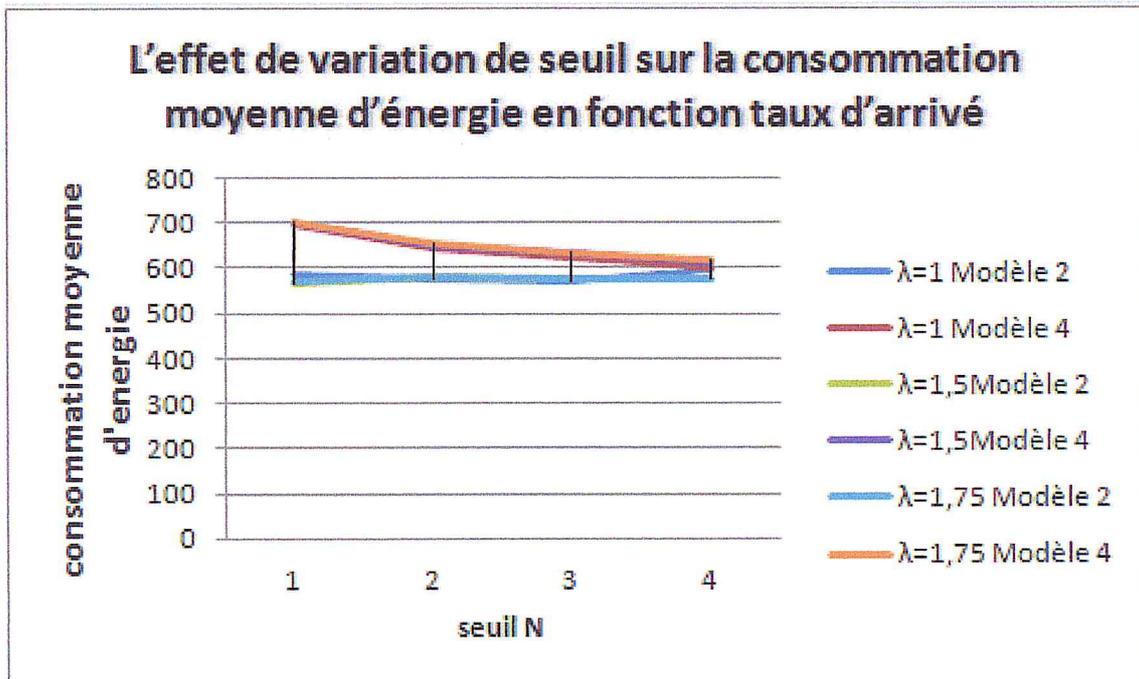


FIGURE 5.5 – La consommation moyenne d'énergie par rapport à le seuil N, λ : 1-1.75.

Analyse : La figure 5.5, montre que l'énergie moyenne de la consommation d'énergie augmente lorsque la capacité le seuil augmente. Et pour tout les valeurs λ proposé nous avons déduit que la consommation d'énergie moyenne dans le modèle N-Policy avec l'état sleep inférieur au modèle N-Policy avec l'état sleep et avec rappel.

5.3.2 Conclusion

Dans ce chapitre nous avons présenté notre outil utilisé pour faire l'évaluation de performance implémenté. La première partie du chapitre décrit les différentes interfaces ainsi que les fonctionnalités fournissent par chaque interface. Ensuite nous avons entamé une étude expérimentale qui nous a permis de voir et comparer les résultats de la consommation moyenne de l'énergie entre le modèle N-policy sans l'état sleep avec N-policy avec rappel, et entre le modèle N-policy avec l'état sleep avec N-policy avec rappel et l'état sleep. Après les résultats obtenus nous avons déduit que la retransmission des paquets consomme l'énergie dans les réseaux de capteur sans fils.

Conclusion Générale

Un réseau de capteur est un ensemble de nœud capteurs déployées dans une zone de captage, pour prélever des mesures physiques, qui sont transmis par la suite a une station de base. Ces capteurs sont déployées généralement dans des zones inaccessibles, ce qui rend difficile le rechargement ou le remplacement des batteries. C'est pour cela que la conservation de l'énergie de ces capteurs constitue un problème de recherche ouvert, malgré le nombre importants de solutions proposées dans la littérature.

Dans le cadre de notre travail, nous sommes focalisés sur la problématique d'impact de la retransmission des paquets a la politique de conservation d'énergie dans les RCSF, nous avons travaillé avec la notion de file d'attente avec vacance et avec rappel pour réaliser l'objectif de notre projet, donc nous avons proposé quatre modèle N-Policy deux modèle avec rappel et deux modèle sans rappel après nous avons analyser chaque modèle et déduire les indices de performance qui nous aide a calculer l'énergie consommer pour chaque modèle et faire la comparaison pour voir est-ce-que la retransmission des paquets consomme l'énergie. Donc après les résultats obtenus de chaque modèle et après la comparaison. Nous avons déduit que la retransmission des paquets consomme l'énergie dans les RCSFs.



Bibliographie

- [Abd08] Fares ABDELFAH. « Développement d'une bibliothèque de capteurs ». In : *projet de* (2008) (cf. p. 4, 6, 10).
- [Aby16] Affoua Therese ABY. « Réseaux de capteurs sans fil étendus dédiés aux collectes de données environnementales ». Thèse de doct. Université Blaise Pascal-Clermont-Ferrand II, 2016 (cf. p. 4, 5).
- [BB] Amar BOURAI et Abdelhamid BENTABET. « la localisation dans les réseaux de capteurs Cas étudié : feu de forêt ». Thèse de doct. (cf. p. 4, 5, 7–9, 15, 16, 19–21).
- [BBC11] Walid BECHKIT, Abdelmadjid BOUABDALLAH et Yacine CHALLAL. « Un Prototype de Réseaux de Capteurs sans Fil pour l'Agriculture et le Contrôle de l'Environnement ». In : *CFIP 2011-Colloque Francophone sur l'Ingénierie des Protocoles*. 2011 (cf. p. 3).
- [BG17] Bachira BOUTOUMI et Nawel GHARBI. « An energy saving and latency delay efficiency scheme for wireless sensor networks based on GSPNs ». In : *Control, Decision and Information Technologies (CoDIT), 2017 4th International Conference on*. IEEE. 2017, p. 0645–0650 (cf. p. 46, 47, 50).
- [Car06] Mickaël CARTRON. « DEVANT L'UNIVERSITÉ DE RENNES ». Thèse de doct. Université de Rennes 1, 2006 (cf. p. 18).
- [Dia14] Ibrahima DIANE. « Optimisation de la consommation d'énergie par la prise en compte de la redondance de mesure dans les réseaux de capteurs ». Thèse de doct. Université de Toulouse, Université Toulouse III-Paul Sabatier, 2014 (cf. p. 21).
- [HL13] Der-Chen HUANG et Jong-Hyouk LEE. « A dynamic N threshold prolong lifetime method for wireless sensor nodes ». In : *Mathematical and Computer Modelling* 57.11-12 (2013), p. 2731–2741 (cf. p. 47).
- [II14] Oliver C IBE et Olubukola A ISIJOLA. « M/M/1 multiple vacation queueing systems with differentiated vacations ». In : *Modelling and Simulation in Engineering* 2014 (2014), p. 18 (cf. p. 43).
- [Jia+12a] Fuu-Cheng JIANG et al. « Design strategy for optimizing power consumption of sensor node with Min (N, T) policy M/G/1 queueing models ». In : *International Journal of Communication Systems* 25.5 (2012), p. 652–671 (cf. p. 46, 47).
- [Jia+12b] Fuu-Cheng JIANG et al. « Lifetime elongation for wireless sensor network using queue-based approaches ». In : *The Journal of Supercomputing* 59.3 (2012), p. 1312–1335 (cf. p. 1, 45, 47).
- [Kac09] Rahim KACIMI. « Techniques de conservation d'énergie pour les réseaux de capteurs sans fil ». Thèse de doct. 2009 (cf. p. 7, 21, 22).

- [Kha12] ARRAR Nawel KHADIDJA. *Problèmes de convergence, optimisation d’algorithmes et analyse stochastique de systèmes de files d’attente avec rappels*. 2012 (cf. p. 37, 39, 40, 42, 44, 45).
- [LY13] Doo Ho LEE et Won Seok YANG. « The N-policy of a discrete time Geo/G/1 queue with disasters and its application to wireless sensor networks ». In : *Applied Mathematical Modelling* 37.23 (2013), p. 9722–9731 (cf. p. 46, 47).
- [MAMB+17] Naima MEGUELATI, Aicha ABOU M’HEMMED, Mohamed BOUALEM et al. « Etude numérique des bornes stochastiques du système M/G/1 avec rappels et clients prioritaires ». Thèse de doct. Université Abderrahmane Mira, 2017 (cf. p. 38).
- [Mih12] Adriana MIHAITA. « Approche probabiliste pour la commande orientée événement des systèmes stochastiques à commutation ». Thèse de doct. Grenoble, 2012 (cf. p. 27).
- [MJ11] R. MAHESWAR et R. JAYAPARVATHY. « Optimal Power Control Technique for a Wireless Sensor Node : A New Approach ». In : *International Journal of Computer and Electrical Engineering* 3.1 (2011), p. 37 (cf. p. 45, 47).
- [Ram12] Cédric RAMASSAMY. « Analyse des protocoles des réseaux de capteurs sans fil. » Thèse de doct. Antilles-Guyane, 2012 (cf. p. 9, 10, 18).
- [Sah] Belkheyr SAHRAOUI. « La Géo-localisation dans les Réseaux de Capteurs sans Fil. » Thèse de doct. (cf. p. 3, 12, 13, 17).
- [Sof07] MOAD SOFIANE. « Optimisation de la consommation d’énergie dans les réseaux de capteurs sans fil ». In : *Master en informatique, Université : IFSIC-Rennes1* 2008 (2007) (cf. p. 12, 14, 15).
- [You10] Yaser YOUSEF. « Routage pour la gestion de l’énergie dans les réseaux de capteurs sans fil ». Thèse de doct. Mulhouse, 2010 (cf. p. 6, 16–18, 21).

Webographie

- [1] : <http://www-igm.univ-mlv.fr/dr/XPOSE2006/Bunel/Presentation.html>
- [2] <http://igm.univ-mlv.fr/dr/XPOSE2006/Bunel/DomainesContraintes.html>
- [3] http://www.tafats.fr/Techniques/Reseaux_de_captteurs/Reseaux_captteur_ext.html
- [4] <https://cermics.enpc.fr/delmas/Enseig/mod-stoch.pdf>
- [5] <https://www.techno-science.net/?onglet=glossairedefinition=6395>
- [6] <https://www.iro.umontreal.ca/lecuyer/ift6561/model-p3.pdf>
- [7] <https://www.emse.fr/xie/master/MajeurGPL/PolyEvaPartielFrancais.pdf>
- [8] <http://mathematique.coursgratuits.net/probabilite/chaine-markovetch.php>
- [9] https://www.math.u-psud.fr/pansu/webifips/Processus_stochastiques_ch8.pdf
- [10] www.univ-bejaia.dz/dspace/bitstream/handle/123456789/8181/Mod
- [11] <https://pagesperso.g-scop.grenoble-inp.fr/gayonj/Documents/Enseignement/ModelesStochastiques.pdf>
- [12] <http://www.bibmath.net/dico/index.php?action=affichequoi=/p/poissonprocess.html>
- [13] <https://www.springer.com/cda/content/document/cda.../9783662494547-c5.pdf?>
- [14] http://www.unit.eu/cours/EnsROtice/05FileAttente_cours/co/naissancemort.html
- [15] https://pmb.univ-saida.dz/buscopac/doc_num.php?explnum_id=254
- [16] <http://www.univ-bejaia.dz/dspace/handle/123456789/5697>
- [17] <http://lesdefinitions.fr/reseau-san>
- [18] https://www.java.com/fr/download/faq/whatis_java.xml
- [19] www.prism.uvsq.fr/mogue/cours.../Gestion_des_operations_C_HAPITRE_19_QX.pdf
- [20] [y_c_hallal.esi.dz/wp-content/uploads/2016/05/PDF-RCSF.pdf\(support de cours\).](http://y_c_hallal.esi.dz/wp-content/uploads/2016/05/PDF-RCSF.pdf(support de cours).)
- [21] <http://www.math.u-psud.fr/meliot/agreg/markov.pdf>
- [22] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203555-java-definition/>
- [23] www.lirmm.fr/leclere/enseignements/TER/2008/Rapport/19.pdf
- [24] http://www.researchgate.net/publication/261299586_Approche_regenerative_de_la_file_dattente_MG1_a_vec_appels_classiques_et_acances_exhaustives_d_erveur
- [25] <http://www.openautomation.net/page/productos/id/22/title/MICAz-2.4-GHz>
- [26] https://www.researchgate.net/figure/Pile-protocolaire-SNB08_fig7_75027616
- [27] https://www.researchgate.net/figure/Domaines-dapplication-des-reseaux-de-captteurs-sans-fil_fig2_78827122
- [28] <http://irt.enseiht.fr/beylot/enseignement/ChainesMarkov.pdf>

