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

UNIVERSITE SAAD DAHLEB BLIDA

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'étude

En vue d'obtenir le diplôme de Master en informatique

Spécialité : Système Informatique et Réseaux

Thème

Un Outil de Simulation pour l'analyse des Performances d'un Nœud Capteur en Utilisant les Politiques de Vacance

Devant le jury composé de :

M^{me} Boutoumi Bachira Promotrice.

M^{me} Mancer Yasmine Présidente.

M^{me} Chikhi Imane Examinatrice.

Présenter par :

M. Kias Mohamed

M. Nekaa Hamza

Année universitaire 2019/2020

REMERCIEMENTS

*Avec un grand plaisir nous remercions **Allah** qui nous a aidé et donné la patience, le courage et la force d'achever ce travail.*

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

*Nous tenons d'exprimer toute nos reconnaissances à notre promotrice de mémoire Mme : **Bachira Boutoumi** d'avoir encadré orienté aidé et conseillé, et ainsi pour la confiance et la liberté qu'elle nous a accordé, sans oublier ses efforts fournis pendant cette période.*

*Nous adressons également nos remerciements, à tous **nos enseignants**, pour leurs aides inestimables, qui nous ont donné les bases de la science. Sans oubliant nos chers professeurs membres de jury et tous les enseignants de département d'informatique.*

*Nous remercions **nos collègues étudiants d'informatique** Nejmi Abdelouahab, Masous Ali, Mamèche Mohamed, Sahran Amin, AbdeRaouf, Djamel. Aussi, **nos collègues** de mathématique Merouani khalil, Boulares Adel, Nejri Hamza, Merabet Yasssine, pour les bons moments passés ensemble et aussi pour leur soutien et leur encouragement.*

*Enfin nous remercions **nos amis** Talaiche Imad Eddine, Idir Imran, Benzina Abdeslam, Tefiani Abdeslam « Mario », Rarbi yakoub, Bahli Khaled et Hamza « El handi », Yahia de Pav03, Ilyes Bosbia, Ilyes Bakir. Et pour finir, merci à toutes les personnes que nous avons oublié de citer et qui nous ont permis de mener à bien ce mémoire.*

« Merci à vous tous »

Nekaa Hamza et Kias Mohamed

RÉSUMÉ

Les réseaux de capteurs sans fil (RCSF) ont attiré l'attention des chercheurs du monde entier par leurs caractéristiques considérables telles que la petite taille, le faible coût et les ressources de traitement et de calcul limitées, cette limitation influence sur la durée de vie du réseau, de ce fait le facteur de la consommation d'énergie devient un problème important dans la conception des RCSF.

L'un des défis des réseaux de capteurs sans fil et qui sera l'objectif de notre travail est la conservation d'énergie dans un nœud de capteur, pour cela nous allons procéder à travailler avec des modèles de files d'attente avec vacances en incorporant deux méthodes qui sont la politique N-vacances et la politique hybride, nous allons faire une analyse du system en utilisant une approche de simulation à événement discret, de plus nous développons un outil de simulation qui permet de faire une comparaison des déférentes résultats obtenus qui sont basés sur les indices de performances entre les deux méthodes proposées.

Mots clés

Réseaux de capteurs sans fil, Simulation à événement discret, File d'attente avec vacance, politique N-vacance, politique hybride, conservation d'énergie, indices de performances.

ABSTRACT

Wireless Sensor Networks have attracted the attention of researchers around the world for their considerable features such as small size, low cost, and limited processing and computing resources, this limitation influences the life time of the network, therefore the factor of energy consumption becomes an important issue in the design of WSN.

One of the challenges of wireless sensor networks and which will be the objective of our work is the conservation of energy in a sensor node, for this we will proceed to work with models of queues with vacations in incorporating two methods which are the N-vacations policy and the hybrid policy, we are going to do an analysis of the system using a discrete event simulation approach, moreover we are developing a simulation tool which allow us to make a comparison of the various results obtained which are based on the performance indices between the two proposed methods.

Keywords

Wireless sensor networks, Discrete event simulation, Queue with vacancy, hybrid policy, N-vacancy policy, energy conservation, performance indices.

ملخص

جذبت شبكات الاستشعار اللاسلكية انتباه الباحثين في جميع أنحاء العالم لميزاتها الكبيرة مثل الحجم الصغير والتكلفة المنخفضة وموارد المعالجة والحوسبة المحدودة ، وهذه المحدودية تؤثر على العمر الافتراضي للشبكة ، وبالتالي يصبح عامل استهلاك الطاقة قضية مهمة في تصميم شبكات الاستشعار الاسلكية.

أحد التحديات التي تواجه شبكات الاستشعار اللاسلكية والذي سيكون الهدف من عملنا هو توفير الطاقة في عقدة المستشعر ، لذلك سنبدأ العمل مع نماذج قائمة الانتظار مع الإجازة من خلال دمج طريقتين ، وهما سياسة الإجازة N والسياسة المختلطة.

سنقوم بتحليل النظام باستخدام نهج محاكاة الأحداث المنفصلة ، بالإضافة إلى أننا سنقوم بتطوير أداة محاكاة تسمح بإجراء مقارنة بين النتائج المختلفة التي تم الحصول عليها والتي تستند إلى مؤشرات الأداء بين الطريقتين المقترحتين.

الكلمات الدالة

شبكات الاستشعار اللاسلكية ، محاكاة الأحداث المنفصلة ، قائمة الانتظار مع الاجازة ، سياسة الاجازة N ، السياسة المختلطة ، الحفاظ على الطاقة ، مؤشرات الأداء.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABRÉVIATIONS

WSN: Wireless sensor network.

QoS: Quality of service.

FCFS: First come first served.

DES: Discrete event simulation.

FDES: Fuzzy discrete event simulation.

FEL: Future Event List.

POO : Programmation orienté objet.

INTRODUCTION GÉNÉRALE

L’évolution de la technologie des systèmes micro électromécaniques (MEMS), de communications sans fil et de l'électronique numérique ont permis le développement de nœuds de capteurs multifonctionnels à faible coût, de faible puissance et de petite taille qui communiquent sur de courtes distances.

Ces minuscules nœuds de capteurs, constitués de composants de détection, de traitement de données et de communication et elles exploitent l'idée de réseaux de capteurs basés sur l'effort collaboratif d'un grand nombre de nœuds.

Un réseau de capteurs est composé d'un grand nombre de nœuds de capteurs, qui sont densément déployés soit à l'intérieur du phénomène, soit très près de celui-ci. L’effort coopératif des nœuds de capteurs constitue une caractéristique unique des réseaux de capteurs. Les nœuds de capteur sont équipés d'un microcontrôleur intégré. Au lieu d'envoyer les données brutes aux nœuds responsables de la fusion, les nœuds capteurs utilisent leurs capacités de traitement pour effectuer localement des calculs simples et ne transmettre que les données requises et partiellement traitées.

Au cours de la dernière décennie, les réseaux de capteurs sans fil (WSN) ont fait l'objet de nombreuses recherches en raison du grand nombre d'applications émergentes qui couvrent plusieurs domaines tel que : domaine militaire, sanitaire domestique commercial et Il est possible d'être appliqué avec plus de catégories telles que l'exploration spatiale, le traitement chimique et les secours en cas de catastrophe.

De nombreux facteurs qui influence la conception d'un réseau de capteurs dont la tolérance aux pannes, évolutivité, coûts de production, environnement d'exploitation, topologie du réseau de capteurs il est inclus les contraintes matérielles telles que le support de transmission et la consommation d'énergie. [1]

La plupart des capteurs sont équipés de batteries non rechargeables dont la durée de vie est limitée et qui dépend fortement de leur consommation d'énergie. En raison du faible coût d'un nœud individuel, il est plus rentable de remplacer le nœud entier que de localiser le nœud et de remplacer ou recharger sa batterie, pour cela la minimisation de la consommation d'énergie a fait l’objet de plusieurs recherche et travaux importants, des exemples de ces travaux sont cité dans [2], [3], [4], [5] pour but de prolonger la durée de vie des RCSF et rendre les réseaux de capteurs sans fil RCSF déployables.

De nombreux systèmes de files d'attente du monde réel peuvent être modélisés en tant que modèle de vacances avec différente Stratégies, les modèles de file d'attentes dans lesquels les serveurs ne sont pas disponibles du service pendant une période aléatoire sont appelés « modèle de file d'attente avec vacances ». Pendant les vacances, le serveur peut effectuer des tâches supplémentaires, être vérifier pour une réparation par exemple ou simplement faire une pause. [6].

Les modèle de file d'attente avec vacances du serveur ont une large applicabilité dans de nombreux domaines, en revanche l’analyse à l’aide de ces modèles dans le domaine des réseaux

de capteurs sans fil présente un défi pour de nombreux chercheurs ces dernières années, parmi les travaux existant qui se focalisent sur la conservation d'énergie, ce sont les cas des travaux cité dans [7], [8], [9],de plus les modèles de vacances de serveur se présentent comme étant l'image plus réaliste et plus flexible des files d'attente réelles.

Parmi les techniques de modélisation du monde réel, c'est la simulation qui est devenue un outil important et utile avec le besoin croissant de mieux comprendre le comportement des systèmes complexes.

La simulation est utilisée dans de nombreux domaines, notamment les industries, la médecine, la conception mécanique et électrique, les organisations d'ingénierie et de services, les systèmes de transport, les systèmes mondiaux, les sciences sociales et comportementales. [10]

Au cours de ce mémoire, nous nous intéressons à la conservation d'énergie dans les réseaux de capteurs dans fil en introduisant une approche de simulation à évènement discret, pour la conception et la modélisation des nœuds de capteurs nous avons adopté un modèle de file d'attente avec vacances en prenant en considération la limitation des buffers , ce model offre la possibilité à un capteur d'être en état de veille, en revanche nous avons développé un outil de simulation permettant de présenter notre approche ainsi que le modèle utilisé avec les déférentes stratégies de vacances adoptées, de plus nous avons développé les principaux indices de performances liée à ce modèle.

Le présent document est organisé comme suit :

- Le premier chapitre est dédié à présenter une généralité sur les réseaux de capteurs sans fil. On citera leurs caractéristiques, leurs facteurs et contraintes ainsi que les différents domaines d'application qui peuvent avoir.
- Le deuxième chapitre est consacré à la description des modèles de files classique et files d'attente avec vacance en définissant leurs caractéristiques et en citant quelques notions importantes pour l'analyse et la modélisation telles que le processus stochastique et les chaine de Markov.
- Dans le troisième chapitre, on définira les concepts de la simulation a événement discret ainsi que leurs différentes approches une section qui sera consacré à présenter quelques travaux connexes.
- Le quatrième chapitre est consacré à la conception de l'outil de simulation que nous allons réaliser son fonctionnement.
- Le dernier chapitre présente la mise en œuvre de l'outil de simulation, y compris les résultats de la simulation.

On termine ce mémoire par une conclusion générale et quelques perspectives

1. LES RÉSEAUX DE CAPTEURS SANS FIL

1.1 Introduction

Le réseau de capteurs sans fil RCSF est devenu l'une des technologies les plus prometteuses pour l'avenir, cela a été rendu possible grâce aux progrès technologiques et à la disponibilité de petits capteurs intelligents et peu coûteux, ce qui a donné lieu à des réseaux de capteurs sans fil économiques et facilement déployables. Cependant, les chercheurs doivent relever une variété de défis pour faciliter le déploiement généralisé de la technologie RCSF dans des domaines du monde réel. [11]

Dans ce premier chapitre nous allons aborder tous les concepts nécessaires d'un RCSF. Nous présenterons la définition et les principaux composants d'un capteur, les réseaux de capteurs sans fil, l'architecture des RCSF, les contraintes et facteurs de conception des RCSF la consommation énergétique d'un nœud capteur. Les différents types de RCSF et les domaines d'application de ce type de réseaux.

1.2 Définition d'un capteur

C'est un dispositif électronique qui a pour tâche principale de détecter des événements, d'effectuer un traitement local rapide des données, puis de transmettre les données. La consommation d'énergie peut donc être divisée en trois domaines : détection, communication, et traitement de l'information. [12]

Un nœud de capteur de type LSN50 433/868/915MHz est présenté dans la figure 1.1 ci-dessous

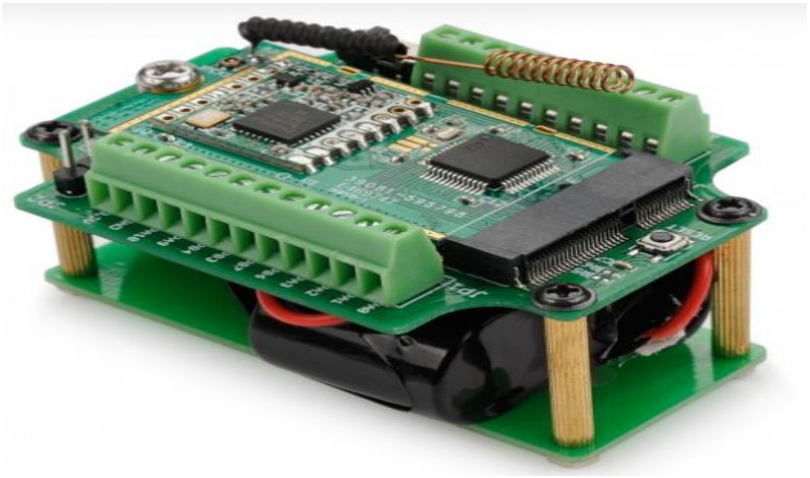

Figure 1.1 Un nœud de capteur de type LSN50 433/868/915MHz.

1.3 Composant de base d'un capteur

Un nœud de capteur est un miniature dispositif, il comprend principalement quatre unités de base :

- **Unité de de captage ou de détection**

Pour l'acquisition de données à partir de l'environnement physique. Elle est composée de deux sous unités, un dispositif de capture physique qui prélève l'information de l'environnement local et un convertisseur analogique/numérique appelé ADC (Analoge to Digital Converter).

- **Unité de calcule ou de traitement**

Elle est conçue pour le traitement et le stockage des données locales, les données captées sont exécutées par le processeur ou elles sont stockées dans la mémoire.

- **Unité de communication sans fil**

Cette unité de communication est appelée aussi transceiver, la transmission des données dans cette unité sera notre point de travail alors, elle se présente comme étant un serveur dans un nœud de capteur qui sert à transmettre des données, elle est composée d'un émetteur/récepteur (module radio) permettant la communication entre les différents nœuds du réseau. Cette unité possède quatre modes de fonctionnement : émission, réception, oisif, et sommeil. Une importante observation est que le mode oisif pour la plupart des unités de transmission consomme autant d'énergie que le mode réception. Il sera donc plus adapté d'éteindre l'unité que de la laisser en mode oisif quand on n'a aucun besoin d'émission ou de réception de donnée. [13]

- **Source d'alimentation**

Fournit l'énergie nécessaire à l'appareil pour effectuer la tâche programmée. Cette source d'alimentation est souvent constituée d'une batterie avec un budget énergétique limité. De plus, il pourrait être impossible ou gênant de recharger la batterie. [14]

Un nœud de capteur peut également avoir des composants supplémentaires dépendant de l'application, tels qu'un système de localisation, un générateur d'alimentation et unité mobile. [12]. La figure I.2 ci-dessous représente les composants de base d'un nœud de capteur.

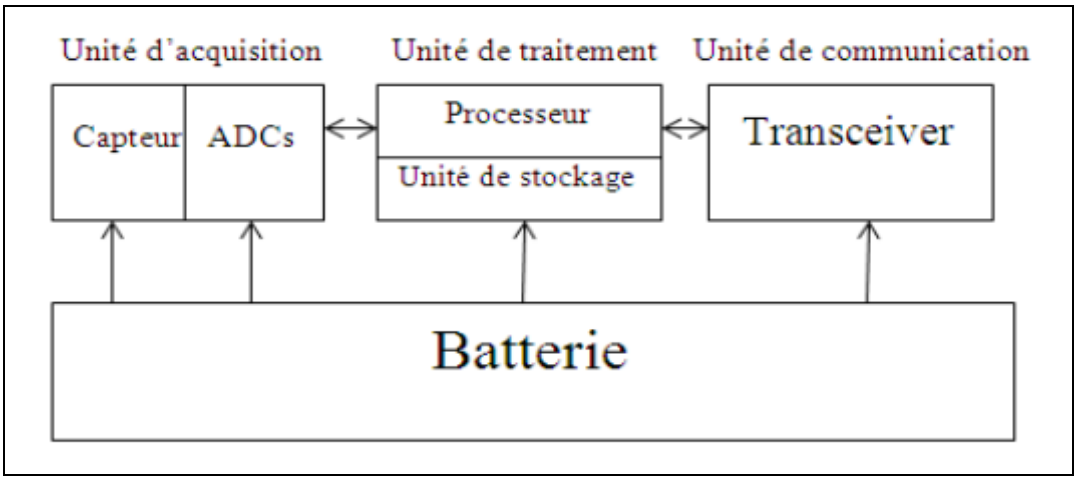

Figure 1.2 Les composants d'un nœud de capteur. [15]

1.4 Définition d'un Réseaux de capteur sans fil

Un RCSF est composé d'un nœud récepteur (ou station de base) et un grand nombre de nœuds de capteurs déployés sur une grande zone géographique (champ de détection). Les données sont transférées des nœuds de capteur vers le puits (station de base) via un paradigme de communication à sauts multiples. [14]

Une architecture d'un RCSF est illustrée dans la figure 1.3.

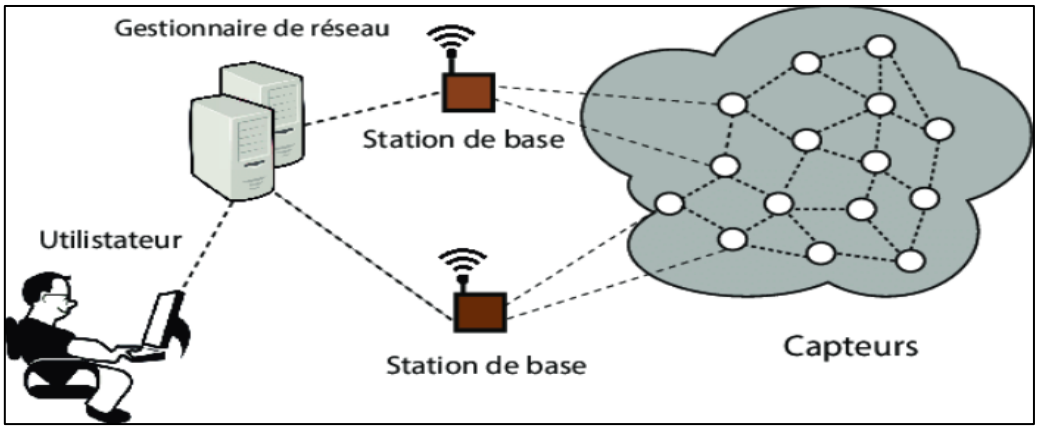

Figure 1.3 Architecture d'un réseau de capteur sans fil. [11]

1.5 Caractéristiques des RCSF

Par rapport aux réseaux de communication sans fil traditionnels tels que les réseaux mobiles ad hoc (MANET) et les réseaux cellulaires, les réseaux de capteurs sans fil présentent les caractéristiques et contraintes uniques suivantes : [16]

- **Forte densité de nœuds de capteurs** :

Les nœuds de capteur sont généralement densément déployés et peuvent être de plusieurs ordres de grandeur supérieurs à ceux d'un MANET.

- **Nœuds de capteur alimentés par batterie :**

Les nœuds de capteur sont généralement alimentés par batterie et sont déployés dans un environnement hostile où il est très difficile de remplacer ou de recharger les batteries.

- **Contraintes énergétiques de calcul et de stockage sévères :**

Les nœuds de capteurs ont des capacités d'énergie, de calcul et de stockage très limitées.

- **Auto-configurable**

Les nœuds de capteur sont généralement déployés de manière aléatoire et se configurent de manière autonome dans un réseau de communication.

❖ **Nœuds de capteur peu fiables**

Étant donné que les nœuds de capteur sont exposés à des endommages physiques ou à des pannes en raison de leur déploiement dans un environnement hostile.

❖ **Redondance des données**

Dans la plupart des applications de réseau de capteurs, les nœuds de capteurs sont déployés de manière dense dans une région d'intérêt et collaborent pour accomplir une tâche de détection commune ainsi que les données détectées par plusieurs nœuds de capteur ont généralement un certain niveau de redondance.

❖ **Objectif ciblé**

Un réseau de capteurs est généralement conçu et déployé pour une application spécifique. Les exigences de conception d'un réseau de capteurs changent avec son application.

❖ **Modèle de trafic plusieurs-à-un**

Dans la plupart des applications de réseau de capteurs, les données détectées par les nœuds de capteur s'écoulent de plusieurs nœuds de capteur source vers un récepteur particulier en présentant un modèle de trafic plusieurs à un.

❖ **Changement fréquent de topologie**

La topologie du réseau change fréquemment en raison des défaillances de nœuds, d'ajout ou déplacement, suppression de capteurs, l'épuisement d'énergie ou de la décoloration des canaux. [16]

❖ **La durée de vie d'un réseau**

C'est l'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où le réseau devient non fonctionnel par exemple l'énergie du premier nœud s'épuise. [17]

1.6 La pile protocolaire dans un RCSF

Cette pile de protocoles prend en charge le problème de consommation d'énergie, intègre le traitement des données transmises dans les protocoles de routage et favorise les efforts de coopération des nœuds capteurs.

La pile de protocoles se compose de la couche physique, couche liaison de données, couche réseau, couche transport, couche application, plan de gestion de l'énergie, plan de gestion de la mobilité, et plan de gestion des tâches.

❖ **La couche physique :**

Elle répond aux besoins de techniques de modulation, de transmission et de réception simples mais robustes. Étant donné que l'environnement est bruyant et que les nœuds de détection peuvent être mobiles.

❖ **La couche liaison :**

Elle assure la liaison point à point et multipoint dans un réseau de communication. Le protocole MAC (Media Access Control) de la couche liaison assure la gestion de l'accès au support physique.

❖ **La couche réseau :**

Prend en charge le routage des données fournies par la couche transport.

❖ **La couche transport :**

Permet de maintenir le flux de données si l'application des réseaux de capteurs l'exige. Selon les tâches de détection, différents types de logiciels d'application peuvent être créés et utilisés sur la couche d'application.

❖ **La couche application :**

Cette couche permet d'assurer une interface avec les applications. C'est la couche la plus proche des utilisateurs, géré directement par le logiciel.

Les composants de la pile protocolaire sont illustrés dans la figure 1.4.

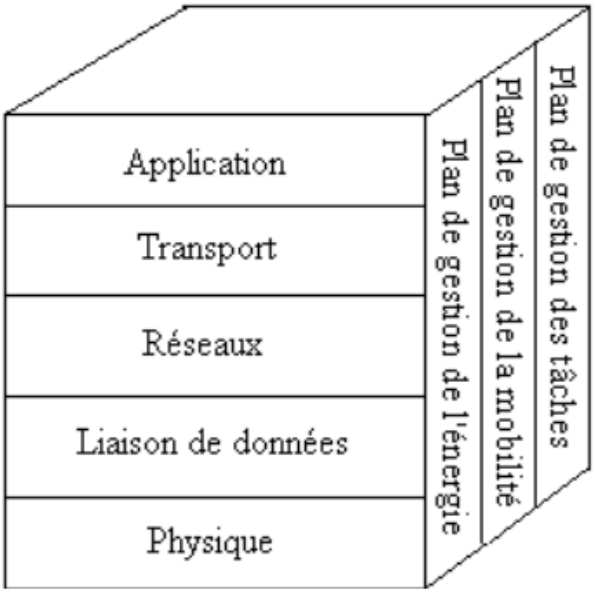

Figure 1.4 La pile protocolaire [12].

De plus, les plans de gestion d'énergie, de mobilité et de gestion des tâches surveillent la puissance, les mouvements et la répartition des tâches entre les nœuds de capteur. Ces plans aident les nœuds de capteur à coordonner la tâche de détection et à réduire la consommation électrique globale. [12]

1.7 La sous-couche MAC dans les RCSF

Les techniques de couche de liaison jouent également un rôle indirect dans la réduction de la consommation d'énergie. L'utilisation d'un bon schéma de contrôle d'erreur minimise le nombre de retransmissions d'un paquet, réduisant ainsi la puissance consommée au niveau de l'émetteur ainsi que du récepteur. [18]

La sous-couche MAC fait partie de la couche liaison de données. Elle fournit le mécanisme d'accès au canal à plusieurs dispositifs de partage de support. Sur un support sans fil, partagé par plusieurs appareils et diffusé dans la nature, lorsqu'un nœud transmet, tous les autres nœuds de la portée de transmission reçoivent sa transmission. Cela pourrait entraîner une interférence et une collision des trames lorsqu'une transmission à partir de deux nœuds ou plus arrive en un point simultanément. Les nœuds de capteur communiquent généralement via des chemins à sauts multiples sur le support sans fil dans un champ de capteur dispersé, dense et rugueux.

Un protocole MAC gère le trafic de communication sur un support partagé et crée une infrastructure réseau de base permettant aux nœuds de capteurs de communiquer entre eux. Ainsi, il fournit une capacité d'auto-organisation aux nœuds et essaie de renforcer la singularité dans le réseau en permettant à l'expéditeur et au récepteur de communiquer entre eux de manière sans collision et sans erreur. [19]

La sous couch MAC est présenté dans La figure 1.5 ci-dessous.

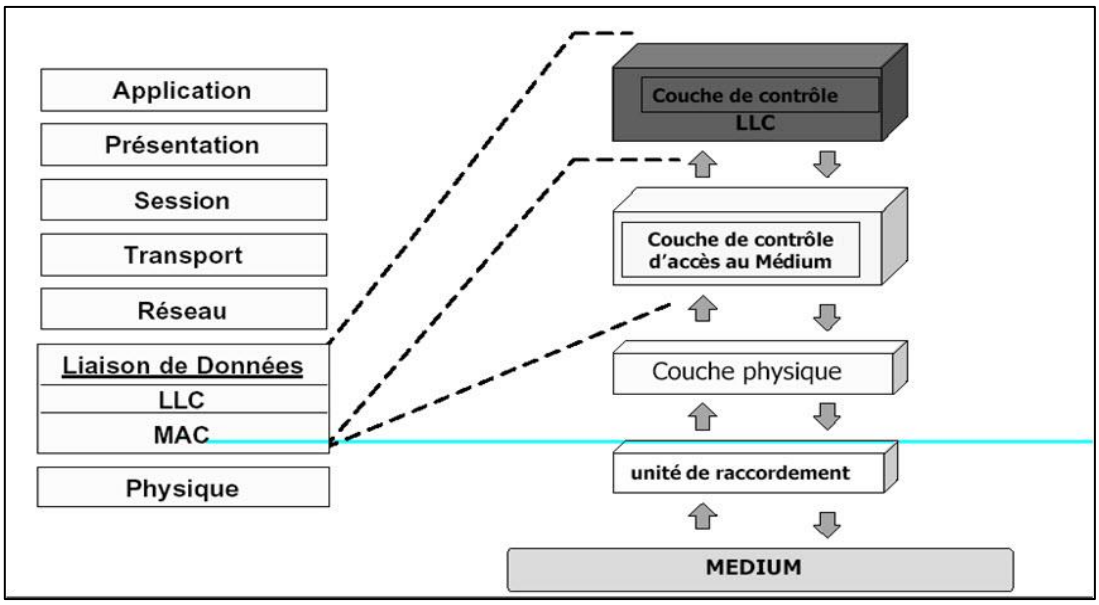

Figure 1.5 Les sous-couches MAC et LLC montré dans la pile protocolaire. [19]

1.8 Sources de gaspillages d'énergie dans les RCSF

Un gaspillage d'énergie pour un nœud de capteur dans un RCSF peut être dû à un ou plusieurs des faits suivants : [20]

- **L'écoute inactive (*idle listening*)**

C'est-à-dire écouter un canal inactif afin de recevoir un trafic éventuel.

- **La collision**

Lorsqu'un nœud reçoit plus d'un paquet en même temps, ces paquets sont appelés collision même s'ils ne coïncident que partiellement. Tous les paquets qui causent la collision doivent être rejetés et retransmis, ce qui augmente la consommation d'énergie.

- **L'écoute (*overhearing*)**

Un nœud reçoit des paquets qui sont destinés à d'autres nœuds.

❖ **La surcharge des paquets de contrôle**

Un nombre minimal de paquets de contrôle doit être utilisé pour effectuer une transmission de données.

❖ **Sur-émetteur(*over-emitting*)**

Ce qui est causé par la transmission d'un message lorsque le nœud de destination n'est pas prêt**.**

1.9 Mécanismes d'économie d'énergie

Les mécanismes d'économie d'énergie peuvent être classées en trois catégories : L'approche du *Data Driven*, l'approche du *Duty cycling* et la mobilité.

1.9.1 L'approche du *Data Driven*

Les données échantillonnées de l'environnement environnant ont une corrélation temporelle ou spatiale élevée, cette approche présente la nécessité de ne pas transmettre des données redondantes au puits.

Dans ce cas, la transmission de données inutiles et redondantes prend beaucoup de temps et d'énergie. Ainsi, les approches basées sur les données fonctionnent comme des solutions écoénergétiques pour la redondance des données. Ils sont utilisés pour réduire la quantité de données échantillonnées à transmettre en maintenant la précision de détection à un niveau acceptable imposé par les exigences de l'application.

Par conséquent, les approches pilotées par les données sont chargées d'équilibrer la consommation d'énergie lors de la transmission des données détectées. Ils sont essentiellement divisés en deux schémas : *la réduction des données* et *l'acquisition des données*. [21]

1.9.2 L'approche du Duty cycling

L'opération d'économie d'énergie la plus efficace de cette approche consiste à mettre l'émetteur-récepteur radio en mode *Low power*, c'est-à-dire mode *sleep* lorsque la communication entre les nœuds n'est pas nécessaire. La radio doit être éteinte dès qu'il n'y a pas de données à transfert ou à recevoir et doit être rendu actif dès qu'un paquet de données devient prêt.

La commutation entre le mode actif et le mode veille peut économiser de l'énergie. De cette manière, les nœuds alternent entre les périodes d'activité et de sommeil. [22]

1.9.3 Les protocoles *Sleep/*wake-*up*

Les protocoles de *Sleep* /*wake-up* sont définis via le sous-système radio du nœud de capteur, ce qui réduit le temps pendant lequel un nœud reste à l'état inactif. Les protocoles de *Sleep* /*wake-up* peuvent être classés principalement en trois catégories :

Les protocoles à la demande, *les rendez-vous planifiés* et *les schémas asynchrones*.

❖ **Les protocoles à la demande**

Ils sont considérés comme l'approche la plus intuitive pour réduire la consommation d'énergie. Un nœud de capteur ne doit être réveillé à la demande.

Pour informer le nœud qui est en veille qu'il y a des voisins possibles qui vont tenter de communiquer avec lui, les protocoles de *Sleep* /*wake-up* utilisent plusieurs radios à faible puissance et débit. En effet, un nœud relais est obligé d'attendre que le nœud se réveille pour recevoir les données.

- **Le rendez-vous planifié**

Cette approche garantit que tous les nœuds se réveillent en même temps. Selon un programme de réveil, les nœuds de capteurs se réveillent et sont toujours actifs pendant un court intervalle de temps pour communiquer avec leurs voisins, puis ils s'endorment jusqu'au prochain heure de rendez-vous.

- **Schéma asynchrone**

Il permet aux nœuds de capteurs de gérer leurs horaires de *Sleep* /*wake-up*. Les nœuds sélectionnent l'heure à laquelle se réveiller lorsqu'ils peuvent communiquer avec leurs voisins. Pour y parvenir, les voisins doivent avoir un chevauchement entre leurs périodes de réveil.

Les principaux objectifs de ce schéma sont : les nœuds ne doivent pas être synchronisés entre eux, il permet à chaque nœud du RCSF de définir son propre programme de *Sleep* /*wake-up* indépendamment, et en conséquence, la contention et la charge de travail sont réduites.

1.9.4 Mobilité

La mobilité des nœuds est désormais considérée comme une solution récente pour la collecte de données écoénergétiques dans les RCSF. En fait, le déploiement de nœuds mobiles peut se faire de deux manières : soit avec l'utilisation d'un mobilisateur installé sur le nœud lui-même, soit en plaçant le nœud sur un élément mobile comme un animal ou une voiture. Une fois que la mobilité a été appliquée dans les RCSF, plusieurs problèmes concernant la connectivité peuvent apparaître.

Puisque les éléments mobiles peuvent atteindre toutes les parties du réseau, ce qui réduit la consommation d'énergie. En fait, les paquets communiqués traversent le réseau vers le puits dans un chemin à sauts multiples.

Ainsi, lorsque le puits est statique, certains chemins sont plus chargés que d'autres. Par conséquent, les nœuds plus proches du puits sont dû à un épuisement énergétique. La mobilité doit être alternée entre les nœuds pour simplifier le processus de collecte de données en affectant certains collecteurs de données mobiles. Désormais, les nœuds ordinaires restent inactifs en attendant que le collecteur de données mobile achemine les messages vers lui. Par conséquent, les erreurs de liaison et les coûts de communication sont réduits. [21]

1.10 Protocole de routage dans les RCSF

De nombreux algorithmes de routage ont été développés pour les réseaux sans fil en général. Tous les principaux protocoles de routage proposés pour les RCSF peuvent être divisé en sept catégories [16]:

❖ **Les protocoles basés sur l'emplacement**

Les nœuds de capteurs sont adressés au moyen de leurs emplacements. Les informations de localisation des nœuds de capteurs sont requises pour les réseaux de capteurs par la plupart des protocoles de routage pour calculer la distance entre deux nœuds particuliers afin que la consommation d'énergie puisse être estimée. Des exemples de ces protocoles sont : *MECN, SMECN, GAF.*

❖ **Les protocoles centrés sur les données**

Diffèrent des protocoles centrés sur les adresses traditionnelles par la manière dont les données sont envoyées des capteurs sources au puits.

Dans les protocoles centrés sur l'adresse, chaque capteur source qui a des données appropriées répondent en envoyant ses données au puits indépendamment de tous les autres capteurs. Cependant, dans les protocoles centrés sur les données, lorsque les capteurs sources envoient leurs données au puits, les capteurs intermédiaires peuvent effectuer une certaine forme d'agrégation sur les données provenant de plusieurs capteurs sources et envoyer les données agrégées vers le puits.

Ce processus peut entraîner des économies d'énergie en raison de la réduction de la transmission requise pour envoyer les données des sources au puits.

Des Example de ces protocoles sont : *SPIN, Directed Diffusion, Rumor Routing, COUGAR, ACQUIRE, EAD, Information-Directed Routing, GradientBased Routing.*

❖ **Protocoles hiérarchiques**

De nombreux projets de recherche au cours des dernières années ont exploré le regroupement hiérarchique dans RCSF sous différents angles.

Le clustering est un protocole de communication économique en énergie qui peut être utilisé par les capteurs pour rapporter leurs données détectées au puits.

Les nœuds sont regroupés en clusters avec une tête de cluster qui est responsable du routage du cluster vers les autres têtes de cluster ou stations de base.

Les données voyagent d'une couche en cluster inférieure à une autre supérieure. Bien qu'il saute d'un nœud à un autre, mais lorsqu'il saute d'une couche à une autre, il couvre de plus grandes distances. Cela déplace les données plus rapidement vers la station de base.

Le clustering fournit des capacités d'optimisation inhérentes aux têtes de cluster. Des exemples de protocole de routage hiérarchique sont : *LEACH, PEGASIS, HEED, TEEN, APTEEN.*

❖ **Protocoles basés sur la mobilité**

La mobilité pose de nouveaux défis aux protocoles de routage dans les RCSF, la mobilité des puits nécessite des protocoles écoénergétiques pour garantir la livraison des données provenant des capteurs sources vers les puits mobiles.

Des exemples de ces protocoles sont : SEAD, *TTDD, Joint Mobility and Routing.*

❖ **Protocoles basés sur des trajets multiples**

En ce qui concerne la transmission de données entre les capteurs source et le puits, il existe deux paradigmes de routage :

Le routage à un seul trajet et le routage à trajets multiples. Dans le routage à un seul chemin, chaque capteur source envoie ses données au puits via le chemin le plus court.

Dans le routage par trajets multiples, chaque capteur source trouve les k premiers chemins les plus courts vers le puits et répartit sa charge uniformément entre ces chemins.

Des exemples de ces protocoles sont *: Sensor-Disjoint Multipath, Braided Multipath, N-to-1 Multipath Discovery.*

❖ **Protocoles basés sur l'hétérogénéité**

Dans l'architecture de réseau de capteurs hétérogène, il existe deux types de capteurs à savoir les capteurs alimentés en ligne sans contrainte d'énergie et les capteurs alimentés par batterie ayant une durée de vie limitée, et devraient donc utiliser efficacement leur énergie disponible en minimisant leur potentiel de communication de données et de calcul.

Des exemples de ces protocoles sont : *IDSQ, CADR, CHR.*

❖ **Protocoles basés sur la QoS**

En plus pour minimiser la consommation d'énergie, il est également important de prendre en compte les exigences de qualité de service (QoS) en termes de délai, de fiabilité et de tolérance aux pannes dans le routage dans les WSN.

Des exemples de ces protocoles sont : *SAR, SPEED, Energy-aware routing.*

1.11 Facteurs et Contraintes de conception des RCSF

Les principaux facteurs et contraintes influençant sur la conception et la réalisation des réseaux de capteurs sans fils, leurs protocoles et algorithmes peuvent être résumés comme suit :

❖ **Tolérance aux pannes**

Certains nœuds de capteur peuvent tomber en panne ou être bloqués en raison d'un manque d'alimentation ainsi qu'ils peuvent présenter des dommages physiques ou des interférences environnementales. La défaillance des nœuds de capteur ne devrait pas affecter la tâche globale du réseau de capteurs.

❖ **Évolutivité**

Le nombre de nœuds capteurs déployés pour étudier un phénomène peut être de l'ordre de centaines voire de milliers. Selon l'application, le nombre peut atteindre une valeur extrême de millions. Les nouveaux schémas doivent pouvoir fonctionner avec ce nombre de nœuds. Ils doivent également utiliser la haute densité des réseaux de capteurs.

❖ **Coûts de production**

Les réseaux de capteurs étant constitués d'un grand nombre de nœuds de capteurs, le coût d'un seul nœud est très important pour justifier le coût global du réseau. Si le coût du réseau est plus cher que le déploiement de capteurs traditionnels, le réseau de capteurs n'est pas justifié

par les coûts. Par conséquent, le coût de chaque nœud de capteur doit être maintenu bas. La technologie de pointe permet à un système radio Bluetooth d'être inférieur à 10 $ US.

- **Environnement**

Les nœuds capteurs sont densément déployés très près ou directement à l'intérieur du phénomène à observer. Alors, ils travaillent généralement sans surveillance dans des zones géographiques éloignées. Ils peuvent travailler à l'intérieur de grosses machines, au fond d'un océan, dans un champ contaminé biologiquement ou chimiquement, sur un champ de bataille au-delà des lignes ennemies, et dans une maison ou un grand bâtiment.

- **Supports de transmission**

Dans un réseau de capteurs à sauts multiples, les nœuds communicants sont reliés par un support sans fil. Ces liens peuvent être formés par des supports radio, infrarouges ou optiques. Pour permettre l'exploitation globale de ces réseaux, le support de transmission choisi doit être disponible dans le monde entier.

- **Consommation d'énergie**

Le nœud de capteur sans fil, étant un appareil microélectronique, ne peut être équipé que d'une source d'alimentation limitée (<0,5 Ah, 1,2 V). Dans certains scénarios d'application, la reconstitution des ressources d'alimentation peut être impossible. La durée de vie du nœud de capteur montre donc une forte dépendance à la durée de vie de la batterie. Dans un réseau de capteurs ad hoc multi saut, chaque nœud joue le double rôle de créateur de données et de routeur de données. Le dysfonctionnement de quelques nœuds peut entraîner des changements topologiques importants et peut nécessiter un réacheminement des paquets et une réorganisation du réseau.

Le nœud de capteur sans fil, étant un appareil microélectronique, ne peut être équipé que d'une source d'alimentation limitée (<0,5 Ah, 1,2 V). Dans certains scénarios d'application, la reconstitution des ressources d'alimentation peut être impossible. La durée de vie du nœud de capteur montre donc une forte dépendance à la durée de vie de la batterie. Dans un réseau de capteurs ad hoc multi saut, chaque nœud joue le double rôle de créateur de données et de routeur de données. Le dysfonctionnement de quelques nœuds peut entraîner des changements topologiques importants et peut nécessiter un réacheminement des paquets et une réorganisation du réseau.

- **La topologie du RCSF**

Des centaines à plusieurs milliers de nœuds sont déployés dans tout le champ du capteur. Ils sont déployés à des dizaines de pieds les uns des autres. Les densités de nœuds peuvent atteindre 20 nœuds / m3. Le déploiement d'un nombre élevé de nœuds nécessite une gestion minutieuse de la maintenance de la topologie. [12]

- **Les contraintes matérielles :**

La principale contrainte matérielle est la taille du capteur. La taille requise peut être plus petite que même un centimètre cube ce qui est suffisamment léger pour rester en suspension dans l'air. Outre la taille, il existe d'autres contraintes strictes pour les nœuds de capteur. Ces nœuds doivent consommer une puissance extrêmement faible, fonctionner à des

densités volumétriques élevées, avoir un faible coût de production, être dispensables et autonomes, fonctionner sans surveillance et être adaptatifs à l'environnent. [12]

Les autres contraintes sont que la consommation d'énergie doit être moindre pour que le réseau survive le plus longtemps possible, qu'il s'adapte aux différents environnements (fortes chaleurs, eau...) qu'il soit autonome et très résistant vu qu'il est souvent déployé dans des environnements hostiles.

Les autres contraintes sont que la consommation d'énergie doit être moindre pour que le réseau survive le plus longtemps possible, qu'il s'adapte aux différents environnements (fortes chaleurs, eau...) qu'il soit autonome et très résistant vu qu'il est souvent déployé dans des environnements hostiles.

1.12 Types des réseaux de capteurs sans fil

Actuellement, de nombreux RCSF sont déployés sur terre, sous terre et sous l'eau. Ils affrontent différents défis et contraintes en fonction de leur environnement. Nous présentons cinq types de RCSF :

1.12.1 Les RCSF terrestre

Consiste en un grand nombre (des centaines à des milliers) de nœuds déployés à terre dans une zone donnée, généralement de manière ponctuelle (par exemple, nœuds tombés d'un avion). Dans les RCSF terrestres les nœuds de capteurs doivent être capables de communiquer efficacement avec la station de base dans un environnement dense. Comme la puissance de la batterie est limitée et généralement non rechargeable, les nœuds de capteurs terrestres peuvent être équipé d'une source d'alimentation secondaire telle que des cellules solaires. L'énergie peut être économisée grâce à un routage optimal à sauts multiples, une courte portée de transmission, une agrégation de données en réseau et des opérations à faible cycle de service.

Les RCSF terrestre inclus la détection et la surveillance de l'environnement, la surveillance industrielle et la surface explorations comme application courante. [23]

1.12.2 Les RCSF souterrain

Se compose d'un certain nombre de nœuds de capteurs déployés dans des grottes ou des mines ou souterraine pour surveiller les conditions souterraines. Afin de relayer les informations des nœuds capteurs souterrains à la station de base, des nœuds puits supplémentaires sont situés au-dessus du sol. Ils sont plus chers que les RCSF terrestres car ils nécessitent des équipements appropriés pour assurer une communication fiable à travers le sol, les roches, et de l'eau. La communication sans fil est un défi dans un tel environnement en raison d'atténuation et perte de signal. De plus, il est difficile de recharger ou de remplacer la batterie de nœuds enfouis sous terre, il est donc important de concevoir un protocole de communication économe en énergie pour une durée de vie prolongée.

Les RCSF souterrains sont utilisés dans de nombreuses applications telles que la surveillance de l'agriculture, la gestion du paysage, la surveillance souterraine du sol, de l'eau ou des minéraux et la surveillance des frontières militaires. [23]

1.12.3 Les RCSF sous-marins

Sont constitués de capteurs déployés sous l'eau, par exemple, dans environnement océanique. Ces nœuds étant coûteux, seuls quelques nœuds sont déployés et des véhicules sous-marins autonomes sont utilisés pour explorer ou recueillir des données. La communication sans fil sous-marine utilise des ondes acoustiques qui présentent divers défis tels qu'une bande passante limitée, un long délai de propagation, une latence élevée et signaler des problèmes de décoloration. Les nœuds sont équipés d'une batterie limitée qui ne peuvent pas être remplacés ou rechargés nécessitant des techniques de communication et de mise en réseau sous-marines économes en énergie.

Les applications des RCSF sous-marins incluent surveillance de la pollution, surveillance et exploration sous-marines, prévention des catastrophes et surveillance, surveillance sismique, surveillance des équipements et robotique sous-marine.

1.12.4 Les RCSF multimédia

Se compose de nœuds de capteurs à faible coût équipés de caméras et de microphones, déployés de manière planifiée pour garantir la couverture. Les dispositifs capteurs multimédia sont capables de stocker, de traiter et de récupérer des données multimédias telles que vidéo, audio et images. Ils doivent faire face à divers défis tels que demande de bande passante, consommation d'énergie élevée, approvisionnement en qualité de service (QoS), techniques de traitement et de compression des données, et conception de couches croisées. C'est requis développer des techniques de transmission qui prennent en charge une bande passante élevée et une faible consommation d'énergie afin de fournir du contenu multimédia tel qu'un flux vidéo. Bien que le provisionnement du (QoS) est difficile dans les RCSF multimédias en raison de la capacité et du délai de liaison variables, un certain niveau de QoS doit être atteint pour une livraison fiable du contenu.

Les RCSF multimédia améliorent les applications RCSF existantes telles que le suivi et la surveillance.

1.12.5 Le RCSF mobile

Se compose de nœuds de capteurs mobiles qui peuvent se déplacer et interagir avec l'environnement physique. Les nœuds mobiles peuvent se repositionner et s'organiser dans le réseau en plus de pouvoir détecter, calculer et communiquer. Un algorithme de routage dynamique doit donc être utilisé contrairement au routage fixe en statique RCSF. Les RCSF mobiles sont confrontés à divers défis tels que le déploiement, la gestion de la mobilité, la localisation avec mobilité, la navigation et le contrôle des nœuds mobiles, le maintien d'une couverture de détection adéquate, la minimisation de la consommation d'énergie en locomotion, maintenir la connectivité réseau et la distribution des données.

Les principaux exemples d'applications des RCSF mobiles sont la surveillance (environnement, habitat, sous-marin), militaire surveillance, suivi des cibles, recherche et sauvetage. Un degré de couverture plus élevé et la connectivité peut être obtenue avec des nœuds de capteurs mobiles par rapport aux nœuds statiques.

1.13 Domaine d'application des RCSF

- **Application militaire**

Les réseaux de capteurs sans fil peuvent faire partie intégrante de l'armée commander, contrôler, communications, intelligence, surveillance, reconnaissance et ciblage. Le déploiement rapide, l'auto-organisation et les caractéristiques de tolérance aux pannes des réseaux de capteurs en font une technique de détection très prometteuse pour le domaine militaire**.**

Les applications des réseaux de capteurs surveillent les forces armées, l'équipement et les munitions, surveillance sur le champ de bataille et reconnaissance des forces du terrain opposés, évaluation des dégâts de combat, et détection et reconnaissance des attaques nucléaires, biologiques et chimiques. [12]

- **Application environnementale**

Certaines applications environnementales des réseaux de capteurs comprennent le suivi des mouvements des oiseaux, des petits animaux et des insectes, surveiller les conditions environnementales qui affectent les cultures et le bétail ,détection chimique ou biologique ,surveillance biologique, terrestre et environnementale dans des contextes marins, pédologiques et atmosphériques, détection des incendies de forêt, recherche météorologique ou géophysique, détection des inondations cartographie de la bio-complexité de l'environnement, et étude de pollution. [12]

- **Application sanitaire**

Dans le domaine de la santé des nœuds de capteur peuvent être utilisés pour surveiller à distance des patients. Dans ce cas, ils permettent non seulement d'améliorer la qualité de vie des malades, qui peuvent rester chez eux, mais aussi intervenir le plus rapidement possible si les mesures effectuées par les capteurs sont anormales. [24]

- **Application domestique**

À mesure que la technologie progresse, les nœuds de capteurs intelligents et les actionneurs peuvent être enterrés dans appareils, tels que les aspirateurs, micro-ondes fours, réfrigérateurs et magnétoscopes. À l'intérieur des appareils domestiques ces nœud de capteurs peuvent interagir entre eux et avec le réseau externe via l'Internet ou satellite. Ils permettent aux utilisateurs finaux digérer les appareils domestiques localement et à distance plus facilement. [12]

- **Applications commerciales**

Certaines des applications commerciales surveillent l'état de matériel de construction, automatisation des processus en usine, détecter et surveiller les vols de voitures et le suivi et la détection des véhicules.

- **Application de sécurité**

Dans ce domaine des capteurs peuvent être exploitées et placé à différents points stratégiques pour prévenir des cambriolages.

1.14 Conclusion

Dans ce chapitre nous avons présenté brièvement les généralités sur les réseaux de capteurs sans fil. Ceci aidera le lecteur à mieux assimiler la problématique abordée dans la suite de ce mémoire.

La recherche dans le domaine de capteur subit une révolution importante, le progrès de minimisation de la consommation d'énergie et l'allongement de durée de vie des batteries annoncent un futur prometteur à la technologie des réseaux de capteurs qui reste le vrai problème de tous protocoles proposés.

Dans le prochain chapitre, nous nous intéressons à présenter les files d'attente avec vacances et leurs caractéristiques.

2. LES FILES D'ATTENTE AVEC VACANCE

2.1 Introduction

Les file d'attente sont rencontrés presque partout, y compris les comptoirs de caisse dans les épiceries et dans les banques en attente de service, les bureaux de poste, les cinémas et les cafétérias. Un système de file d'attente se compose d'un ou plusieurs serveurs qui s'occupent des clients qui arrivent selon un processus stochastique bien défini. [25] d'un autre côté, le système de file d'attente est une application classique des chaînes de Markov continues.

Les chaînes de Markov sont une classe fondamentale de processus stochastiques. Elles sont très importantes et largement utilisées pour résoudre des problèmes dans un grand nombre de domaines tels que la recherche opérationnelle, l'informatique et les systèmes distribués, les réseaux de communication, la biologie, la physique, la chimie, l'économie, la finance et les sciences sociales. Le succès des chaînes de Markov tient principalement à leur simplicité d'utilisation, au grand nombre de résultats théoriques disponibles et à la qualité des algorithmes développés pour l'évaluation numérique des différentes métriques qui leur sont associées et qui peuvent être appliqué aux modèles de files d'attente [26].

Au cours de ce chapitre nous allons présenter des modèles de files d'attente simple et les files d'attente avec vacance et leurs caractéristiques et nous allons introduire les mesures de performance approprie ainsi que les concepts de la chaine de Markov et le processus stochastique qui joue un rôle pour l'analyse mathématique de ces modèles.

2.2 Les files d'attente

La théorie des files d'attente est une technique de la Recherche opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions. [27]

2.2.1 Description du modèle des files d'attente

Dans un système de file d'attente, les clients arrivent au hasard à un taux moyen de λ, ils sont servis sans délai s'il y a des serveurs disponibles sinon, ils attendent dans la file d'attente jusqu'à ce que ce soit leur tour d'être servi. Une fois être servis, ils sont supposés quitter le système.

La description de tout système de file d'attente nécessite la spécification de trois parties : [28]

- Le processus d'arrivée.
- Le mécanisme de service tel que le nombre de serveurs et la distribution du temps de service.
- La discipline de la file d'attente (par exemple, premier arrivé, premier servi).

Une représentation de file d'attente simple est illustrée à la figure 2.1 ci-dessous.

Figure 2.1 Un simple système de file d'attente.

2.2.2 Caractéristiques des files d'attente

Dans la plupart des cas, un système de file d'attente est caractérisé par : [29]

- **Processus d'arrivée des clients**

Le processus d'arrivée signifie la manière dont les arrivées se produisent. Il est spécifié par le temps entre deux arrivées consécutives. Habituellement les temps inter arrivées sont supposés suivre une distribution commune et sont indépendants les uns des autres.

Le modèle de saisie indique le comportement des clients lorsqu'ils arrivent au système de service. Certains clients peuvent attendre longtemps en patience, d'autres clients sont moins patients et partent après un certain temps. Par exemple, les patients qui visitent l'hôpital pour avoir un rendez-vous avec leur médecin, si le médecin n'est pas disponible alors certains d'entre eux partira et peut-être changer de rendez-vous. Il est également très important de savoir si les clients arrivent par groupe (batch) ou un par un.

- **Temps de service**

Le modèle des temps de service est la manière dont le service est rendu. C'est spécifié par le temps nécessaire pour terminer un service. On suppose généralement que les temps de service suivent une distribution commune et sont indépendants les uns des autres et indépendamment des périodes inter arrivées. Les distributions les plus courantes que le service les temps peuvent avoir sont des distributions déterministes et exponentielles. Les temps de service peuvent dépendre également de la longueur de la file d'attente.

- **Discipline de service**

La discipline de service indique la manière dont les unités sont prises dans la file d'attente et autorisé en service. Les clients peuvent être servis en groupe ou un par un.

Les disciplines les plus connues sont :

FIFO :*(Premier entré, premier sorti)* : la discipline de file d'attente habituelle est le premier arrivé, premier servi (FCFS), un client qui trouve le centre de service occupé va à la fin de la file d'attente.

LIFO :*(Last in, First out)* : ou dernier arrivé, premier servi (LCFS), un client qui trouve le centre de service occupé passe immédiatement en tête de file d'attente, ce le client sera servi ensuite, étant donné qu'aucun autre client n'arrive.

Service aléatoire : également appelé (SIRO), les clients de la file d'attente sont servis ordre aléatoire.

Round Robin (RR) : chaque client obtient une tranche de temps. Si l'entretien d'un le client n'est pas terminé à la fin de ce délai, le client est préempté et retourné à la file d'attente pour être servi selon la discipline FCFS.

- **Disciplines de priorité**

Chaque client a une priorité (statique ou dynamique), le serveur sélectionne les clients les plus prioritaires en fonction de leur temps d'arrivée au système. Ce régime peut utiliser une préemption.

Dans le cas d'une préemption, le client ayant la priorité la plus élevée est autorisé à entrer en service et à service d'un client de priorité inférieure dont le service doit être repris après la un client de priorité plus élevée est servi. En cas d'absence de préemption, le plus haut Le client prioritaire se place en tête de file d'attente et attend que le service en cours soit terminé.

- **Nombre de serveurs**

Le nombre de serveurs est une caractéristique importante d'un système de file d'attente et représente un compromis fondamental.

Un système peut avoir un seul serveur ou un groupe de serveurs fournissant des services au les clients. L'augmentation du nombre de canaux de service contribue à réduire le temps d'attente. Étant donné un certain nombre de canaux de service, ils peuvent fonctionner en parallèle et servir les clients simultanément. On suppose généralement que les mécanismes de service des canaux parallèles fonctionnent indépendamment les uns des autres. Une arrivée qui trouve plusieurs serveurs gratuits peuvent choisir l'un d'entre eux pour recevoir le service. S'il trouve tous les serveurs occupés, il rejoint une file d'attente commune à tous les serveurs, le premier client dans la file d'attente commune va au serveur qui devient libre en premier.

- **Capacité du système**

Dans certains systèmes, il existe une limitation physique de la quantité d'espace d'attente des clients, de sorte que lorsque la ligne atteint une certaine longueur, aucun autre client n'est autorisé à entrer jusqu'à ce que l'espace soit disponible. Celles-ci sont appelées situations de file d'attente finies, c'est-à-dire qu'il existe une limite finie à la taille maximale du système.

- **Étapes de service**

Les clients peuvent passer par une ou plusieurs étapes pour compléter leur service avant de quitter le système. Dans le cas de systèmes de files d'attente à plusieurs étages, le client entre dans une file d'attente en attendant le service, ensuite il quitte la station-service pour entrer dans une nouvelle file d'attente pour un autre service, etc.

Un exemple d'un système de file d'attente à plusieurs étapes est une procédure d'examen physique où chaque patient doit passer par plusieurs étapes, comprenant les antécédents médicaux, examen des oreilles, du nez et de la gorge, des analyses de sang, électrocardiogramme, œil.

2.2.3 La notation de Kendall

Dans la plupart des cas, on modélise un système de files d'attente à l'aide de la notation de Kendall. Cette notation est de la forme A/B/C[/D/E] où : [30]

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

- *M* : pour *Markovian* ou *Memoryless,* correspondant à un processus d'arrivée Poisson,
- *D* pour déterministe,
- *G* : pour generallydistributed.

B : *Distribution du temps de service*

- *M* : pour *Markovian* ou *Memoryless,* correspondant à un temps de service distribué selon la loi exponentielle, D pour *Deterministe*, G (ou GI) pour *Generallydistributed* (et independent).

C : *Nombre de serveurs*

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

- Valeur par défaut : ∞.

E : *Discipline de service*

- *FIFO* pour first-in, first-out (premier arrivé, premier servi),
- *LIFO* pour last-in, first-out (qui peut être préemptif ou non),
- *PS* pour processor-sharing.
- Valeur par défaut *: FIFO*.

2.2.4 Les mesures de performance d'un système de file d'attente

Les mesures de performance du système de file d'attente sont un aspect très critique et important. L'analyse mathématique du système de file d'attente est effectuée uniquement pour obtenir les différentes mesures de performance qui peuvent être utilisées pour déterminer la mesure de l'efficacité d'un processus donné. [6]

Les mesures de performances utilisées lors de l'analyse du modèle d'attente sont les suivantes :

Etant donné que les paramètres :
$\boldsymbol{\lambda}$ **:** le taux d'arrivé.

$\boldsymbol{\mu}$ **:** le taux de service.

- **La probabilité du nombre de client dans le système** P_n

C'est la probabilité que n clients résidant dans le système soient servis ou en attente.

$$P_n = \text{P [il y a des n clients dans le système]}$$

- **L'intensité du trafic** ρ *(Traffic intensity)*

L'intensité du système est donnée par la relation du taux d'arrivée et le taux de service.

$$\rho = \frac{\lambda}{\mu}$$

- **L'utilisation du serveur u** *(utilization)*

L'utilisation est donnée par la relation entre le taux d'arrivé et le taux de service, tel que m représente le nombre de serveurs.

$$u = \frac{\lambda}{m\mu}$$

Si le nombre de serveurs m=1, l'utilisation (*Utilization*) est égale à l'intensité du trafic (*traffic intesity*).

- **Le débit d'un système σ** (*Throughput)*

Le débit est défini comme le nombre moyen de clients quittant le système. Le débit d'un système de file d'attente est égal à son taux de départ, c'est-à-dire au nombre moyen de clients qui sont traités par unité de temps.

Dans un système de file d'attente dans lequel tous les clients qui arrivent sont finalement servis et quittent le système, le débit est égal au taux d'arrivée, ce n'est pas le cas dans les systèmes de file d'attente avec un tampon fini, car les arrivées peuvent être perdues avant de recevoir le service.

- **Temps de réponse T** *(Response time)*

Le temps qu'un client passe dans le système de l'instant de son arrivée à la file d'attente jusqu'à l'instant de son départ du serveur est appelé temps de réponse ou temps de séjour.

- **Temps d'attente W** *(Waiting time)*

Le temps d'attente est le temps qu'un client passe dans une file d'attente pour être traité. Par conséquent, nous avons :

Temps de réponse = temps d'attente + temps de service.

Le temps moyen passé par un client en file d'attente pour obtenir son tour de service est appelé temps d'attente moyen dans la file d'attente W_q tandis que le temps total passé dans la file d'attente plus le temps de service est appelé temps d'attente moyen dans le système W_s.

- **Longueur de la file d'attente L** *(Queue length)*

La longueur de la file d'attente *L* représente le nombre de clients dans la file d'attente.

- **Nombre moyen de clients en attente**

Le nombre moyen de clients en attente dans la file d'attente pour obtenir le service est appelé longueur moyenne de la file d'attente et est noté L_q tandis que le nombre total de clients en attente dans la file d'attente plus le nombre de clients servies est appelé longueur moyenne de la file d'attente dans le système L_S.

2.3 Les files d'attente avec vacance

2.3.1 Description du modèle des files d'attente avec vacance

Dans un modèle de file d'attente classique, les serveurs sont toujours disponibles. Cependant, dans de nombreux systèmes de file d'attente pratiques, les serveurs peuvent devenir indisponibles pendant un certain temps pour diverses raisons. Cette période d'absence du serveur peut représenter le fait que le serveur travaille sur des tâches supplémentaires où il fait l'objet d'une vérification de maintenance ou fait simplement une pause. Permettre aux serveurs de prendre des vacances rend les modèles de files d'attente plus réalistes et flexibles dans l'étude des systèmes de file d'attente du monde réel.

Une représentation de file d'attente avec vacance est illustrée dans la figure ci-dessous. [31]

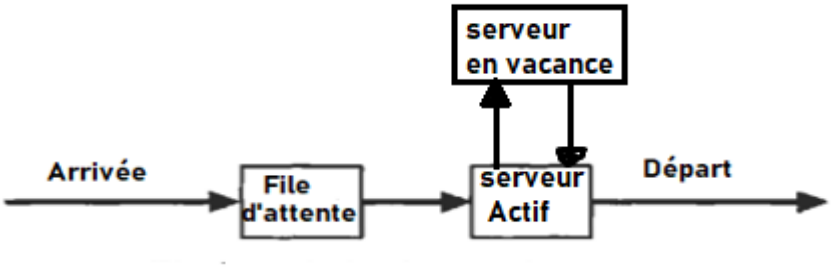

Figure 2.2 Un système de files d'attente avec vacance.

2.4 Classification des politiques de vacances de serveurs

On distingue trois parties de base de politiques de vacance c'est les types de vacance et la discipline de service et la durée de vacance.

2.4.1 Types de vacances

Les vacances dans un contexte de file d'attente sont une période pendant laquelle le serveur n'est pas disponible pour fournir le service. Les arrivées qui ont lieu pendant les vacances ne peuvent être mises en service qu'après le retour de vacances du serveur. Il existe différents types de modèles de vacances :

- **Les vacances uniques**

Dans ce modèle il y a exactement une vacance après la fin de chaque période de pointe. Si le serveur revient de ces vacances, il n'en prend pas d'autres, même si le système est encore vide à ce moment-là.

Ce type de vacances peut être lié aux cas comme la maintenance des systèmes de production, la maintenance pouvant être considérée comme des vacances.

- **Les vacances multiples**

Lorsque le système est inactif, le serveur prend des vacances et à son retour de vacances, le serveur ne démarre le service que s'il trouve K clients ou plus dans la file d'attente, si le nombre de clients en attente est inférieur à K alors il prend de nouvelles vacances.

- **Les vacances hybrides**

Le serveur commence une autre vacance après une période aléatoire qui suit une loi exponentielle s'il trouve le system vide et aucun client n'est arrivé à la fin de cette période. [32]

2.4.2 Disciplines de service

C'est la règle de début de vacance qui détermine quand le serveur commence des vacances. Il existe deux types principaux :

- **Service exhaustif**

Dans ce cas, le serveur sert les clients jusqu'à ce que le système soit vidé, puis il part en vacances.

- **Service non exhaustif**

Le serveur d'un système de service non exhaustif peut prendre des vacances même si le système n'est pas vide. Ce service peut inclure une autre politique a service limité.

- **Service limité**

Dans ce cas, une limite fixe de K est placée sur le nombre maximum de clients qui peuvent être servis avant que le serveur ne parte en vacances. Le serveur part également en vacances : lorsque le système est vide ou lorsque K clients. [32]

2.4.3 Les politiques de fin de vacance

Les différentes disciplines de service constaté précédemment définissent l'instant du début de la vacance, par contre Il existe une autre classification possible qui considère la règle de fin de vacances. Cette règle détermine quand le serveur reprend le service.

Trois politiques de vacance sont définies ci-dessous et chacune d'elle, la disciplines de service est exhaustif [31]

- **N-vacance :** Dans cette politique le serveur retourne de la vacance quand la taille de la file d'attente est supérieure ou égale à N.
- **T-vacance :** Dans ce cas, le serveur est non activé (en vacances) T unités de temps après la fin d'une période de service. Après cette vacance (de longueur T), le serveur est réactivé pour servir les clients seulement s'il y avait au moins un client présent, sinon le serveur prend une autre période de vacance de longueur T.
- **D-vacance :** Dans cette politique le serveur retourne d'une vacance quand la somme des temps de service des clients dans le buffer dépasse la valeur D.

2.4.4 Durée d'une vacance

Les vacances au serveur sont souvent supposées être des variables aléatoires indépendantes et de distribution identique avec une fonction de distribution générale notée V (x).

Cependant, certains modèles de vacances exigent différents types de vacances et suivent des distributions différentes. [31]

2.5 La chaînes de Markov

Soit $\{X_n\}_{n\geq 0}$ une suite de variables aléatoires à valeur dans l'espace dénombrable E.

Si pout tout entier $n \geq 0$et tout états$i_0, i_1, \dots \dots, i_{n-1}, i, j \in E$ on a :

$$\mathrm{P}\{X_{n+1} = j | X_n = \mathrm{i}, X_{n-1} = i_{n-1}, \dots X_0 = i_0\} = \mathrm{P}\{X_{n+1} = j |\, X_n = \mathrm{i}\}$$

Alors le processus$\{X_n\}_{n\geq 0}$ est appelé chaine de Markov.

Et elle est homogène si $\mathrm{P}\{X_{n+1} = j |\, X_n = \mathrm{i}\}$ ne dépend pas de n.

La matrice $\mathrm{P} = \{p_{ij}\}_{\mathrm{i,j}\in \mathrm{E}}$ où :

$$p_{ij} = \mathrm{P}\{X_{n+1} = j |\, X_n = \mathrm{i}\}$$

Est la probabilité de transition de i vers j, est appelée matrice de transition de la chaîne. [33]

2.5.1 Variable aléatoire

Une variable aléatoire est une fonction qui reflète le résultat d'une expérience aléatoire. Le but principal de l'utilisation d'une variable aléatoire est de pouvoir définir certaines fonctions de probabilité qui rendent à la fois pratique et facile le calcul des probabilités de divers événements.

Considérons une expérience aléatoire avec un espace d'échantillonΩ. Soit ω un point d'échantillon dansΩ.on s'intéresse à attribuer un nombre réel à chaque $\omega \in \Omega$. Une variable aléatoire $X(\omega)$ est une fonction réelle à valeur unique qui affecte un nombre réel appelé la valeur de $X(\omega)$ à chaque point d'échantillonnage $\omega \in \Omega$.

Autrement dit, il s'agit d'un mappage de l'espace échantillon sur la ligne réelle.

L'espace d'échantillonnageΩ est appelé le domaine de la variable aléatoire X et la collection de tous les nombres qui sont des valeurs de X est appelée la plage de la variable aléatoire X. [25]

2.5.2 Loi exponentielle

Pour $\lambda > 0$, Une variable aléatoire X qui suit une loi exponentielle, si et seulement si sa densité de probabilité (PDF) est donnée par :

$$g_\lambda(t) = \begin{cases} 0 & si \quad t < 0; \\ \lambda e^{-\lambda t} & si \quad t \geq 0; \end{cases}$$

Et sa fonction de distribution cumulative (CDF) avec le paramètre t :

$$G(\lambda) = 1 - e^{-\lambda t}, t \geq 0$$

La moyenne et la variance pour une variable qui suit la loi exponentielle sont : [34]

$$E[X] = \frac{1}{\lambda} \text{ Et } V[X] = \frac{1}{\lambda^2}$$

2.5.3 Processus stochastique

Soit S un espace échantillon et $X_n(\omega)$une variable aléatoire où $\omega \in S$. Un processus stochastique peut être défini comme une collection de points

$X_n(t, \omega), t \geq 0$Indexés sur le paramètre t, représentant généralement le temps mais pouvant en représenter d'autres comme l'espace, etc.

Sous une forme plus simple, nous pouvons dire qu'un processus stochastique est une collection de variables aléatoires X (t), t ≥ 0, qui est observée au temps t.

Des exemples de processus stochastique pourraient être la quantité de pluie X (t) le jour t, l'essence restante dans le réservoir d'une automobile à tout moment, etc. La plupart des quantités étudiées dans un système de file d'attente impliquent un processus stochastique. [35]

2.5.4 Chaînes de Markov à temps discret

Un processus stochastique $X = \{X_n, n \in N\}$ dans un état d'espace S est dite une chaine de Markov a temps discret si :

- Pour tout $n \geq 0$, $X_n \in S$
- Pour tout $n \geq 1$ et pour tout $i_0, \ldots \ldots i_{n-1}, i_n \in S$,on a :

$$P\{X_n = i_n | X_{n-1} = i_{n-1}, \ldots \ldots, X_0 = i_0\} = P\{X_n = i_n | X_{n-1} = i_{n-1}\}$$

Une CMTD $X = \{X_n, n \in N\}$ dans un état d'espace S est homogène si pour tout $n, k \in N$ et pour tout $i, j \in S$ on a : [26]

$$P\{X_{n+k} = i_j | X_k = i\} = P\{X_n = i_j | X_0 = i\}$$

2.5.5 Chaînes de Markov à temps continu

Un processus stochastique $X = \{X_t, t \in \mathbb{R}^+\}$ avec des valeurs dans un ensemble dénombrable $\mathcal{S}$ est dite chaine de Markov à temps continue si pour tout $n \geq 0$ et pour tous les instants $0 \leq s_0 < \cdots < s_n < s < t$ et pour tous les états $i_0, \ldots, i_n, \mathrm{i}, \mathrm{j} \in \mathcal{S}$ on a :

$$\mathrm{P}\{X_t = \mathrm{j} \,|X_s = i, X_{sn} = i_n, \ldots\ldots X_{s0} = i_0\} = \mathrm{P}\{X_t = \mathrm{j}|\, X_s = i\}.$$

Une CMTC est homogène si $t, s \geq 0$ et $\mathrm{i}, \mathrm{j} \in \mathrm{S}$ on a : [26]

$$\mathrm{P}(X_{t+s} = \mathrm{j}|X_s = i\} = \mathrm{P}\{X_t = \mathrm{j}|X_0 = i\}.$$

2.6 Conclusion

La file d'attente est une partie inévitable de la vie moderne et la modélisation de leur comportement peut être à la fois d'intérêt théorique et d'importance pratique.

Dans ce chapitre nous avons vu déférente notions liée la description des modèles de files d'attente classique et avec vacance ainsi que les mesure de performances.

Finalement, nous avons connu le processus stochastique et la chaine de Markov qui ont une importance pour la modélisation de ces modèles.

3. LA SIMULATION DES SYSTEMES DE FILE D'ATTENTE

3.1 Introduction

La simulation est toujours utilisée pour aider les décideurs et les concepteurs à mieux comprendre les performances attendues du système réel afin de tester l'efficacité de la conception du système [10]. Un système à événements discrets est un système à états discrets et événementiel dans lequel les changements d'état dépendent entièrement de l'occurrence d'événements discrets au fil du temps. La simulation d'un système à événements discrets est appelée la simulation à événements discrets.

Des exemples de systèmes à événements discrets comprennent les systèmes de fabrication, les systèmes de transport tels que les réseaux de trafic urbain, les systèmes de services tels que les hôpitaux et les systèmes de communication tels que les réseaux sans fil etc. [36]

Dans ce chapitre, nous allons mettre des points sur les principes de la simulation et avoir une présentation sur la simulation à événement discret ainsi que leurs approches. Nous allons connaitre les travaux connexes liée à l'analyse et l'évaluation des performances des systèmes de files d'attente.

3.2 Les principes général de la simulation

La simulation est l'imitation du fonctionnement d'un processus ou d'un système du monde réel au fil du temps. Elle représente un outil très utile et précieux pour analyser et évaluer chaque processus disponible ou développé. [10]

Une étude de simulation comprend les étapes suivantes :

1. Formulation du problème.
2. Fixer les objectifs et le plan global du projet.
3. Conceptualisation du modèle.
4. La Collection de données.
5. Traduction de modèle.
6. Vérification.
7. Validation.
8. Conception expérimentale.
9. Production et analyse.
10. Documentation et rapports.
11. L'implémentation. [37]

Il existe deux types d'objectifs de simulation :

L'un est d'obtenir des informations et l'autre est de former ou de divertir le personnel. La première est souvent appelée simulation analytique et la seconde c'est la simulation d'environnement virtuel.

❖ **La simulation analytique**

Son but principal est l'analyse quantitative du système source basée sur des données exactes. Ainsi, la simulation doit être exécutée de la manière la plus rapide possible et être capable de reproduire précisément la séquence d'événements du système source.

❖ **Une simulation d'environnement virtuel**

Elle est exécutée en temps réel mis à l'échelle lors de la création d'environnements virtuels et elle est souvent appelée la simulation de retard.

Il existe trois types de simulation informatique sont :

- **La simulation par événements discrets.**
- **La simulation continue**

Elle représente la simulation des systèmes continue, c'est une évaluation numérique d'un modèle informatique d'un système dynamique physique qui suit en continu les réponses du système au fil du temps selon un ensemble d'équations impliquant généralement des équations différentielles. [36]

- **La simulation de Monte Carlo**

La simulation de Monte Carlo est un type de simulation qui repose sur un échantillonnage aléatoire répété et une analyse statistique pour calculer les résultats. Cette méthode de simulation est très étroitement liée aux expériences aléatoires, expériences dont le résultat spécifique n'est pas connu à l'avance. [38]

3.3 La simulation a évènement discret

La simulation d'événements discrets est une évaluation par ordinateur d'un modèle de système dynamique à événements discrets où le fonctionnement du système est représenté comme une séquence chronologique d'événements.

Dans un système à événements discrets, un ou plusieurs phénomènes d'intérêt changent de valeur ou d'état à des points discrets dans le temps. [36]

Les principaux termes du vocabulaire de La simulation par événements discret sont :

Système : Une collection d'objets qui interagissent dans le temps selon des règles spécifiées.

Il existe deux types de systèmes :

- *Un système discret* est un système où la variable représente les valeurs et change instantanément à des moments séparés dans le temps.
- *Un système continu* est un système pour lequel les variables d'état changent continuellement en fonction du temps. [39]

Horloge : elle donne la valeur actuelle du temps simulation.

Modèle : Une représentation logique et mathématique abstraite d'un système qui décrit la relation entre les objets d'un système.

Entité : Un objet dans un système qui nécessite une représentation explicite dans un modèle du système. Par exemple, les serveurs, les clients et les machines etc.

Attribut : Un descripteur d'une propriété d'une entité. Par exemple, un serveur peut avoir un attribut de compétence qui identifie son niveau de compétence et un attribut de statut qui identifie s'il est inactif ou occupé.

Liste chainée : Une collection d'enregistrements chacun identifié avec une entité et ils sont enchaîné ensemble dans un ordre selon les valeurs assumées par un attribut particulier de chaque entité.

Un événement : Un changement d'état d'un système.

Avis d'événement : Un enregistrement décrivant quand un événement doit être exécuté.

Activité : c'est une paire d'événements, l'un initiant et l'autre complétant une opération qui transforme l'état d'une entité. Le temps s'écoule dans une activité.

Processus : Une collection d'événements classés dans le temps. Le temps s'écoule dans un processus. [40]

3.4 Les approches de la simulation à évènement discret

3.4.1 The Event Scheduling

La grande majorité des programmes de simulation reposent sur une approche de planification d'événements. En fait, l'ordonnancement des événements constitue la pierre angulaire de la simulation informatique par événements discrets. [41]

Cette approche se concentre sur les événements et leur impact sur l'état du système. Le processus d'avancement dans le temps est effectué sur la base d'une liste ordonnée appelée liste d'événements futurs (FEL). Lorsqu'un nouvel événement est généré dans le système, il est inséré dans l'ordre chronologique dans cette liste. L'horloge passe à l'instant suivant correspondant au premier élément du (FEL), traite l'événement et modifie l'état du système et introduit tous les changements nécessaires dans le (FEL). Il convient de noter que l'opération d'insertion peut devenir coûteuse s'il y a trop d'événements futurs qui sont générés et doivent être insérés dans la liste.

Cependant, cette approche est définitivement plus rapide que celle d'analyse des activités, car elle évite le contrôle inutile des activités à chaque avance d'horloge. [37]

3.4.2 The activity-scanning

Dans cette approche, le temps est décomposé en petits incréments à chaque instant ou avancement d'horloge le modèle vérifie les conditions pour chaque activité et celles qui satisfont aux conditions sont alors démarrées.

Cette approche peut devenir extrêmement lente si l'incrément de temps est trop petit. Des incréments de temps importants ne fourniront pas une représentation correcte de la façon dont les activités doivent être exécutées, car certaines activités qui satisfont aux conditions devront attendre la prochaine avance d'horloge pour démarrer. Il faut également noter que si les incréments de temps sont trop petits, une grande partie des contrôles d'activité sera inutile car il n'y aura aucun changement dans le système.

Par exemple, si le temps est subdivisé en incréments de 0,01 s et qu'une tâche peut démarrer à t = 10, cela implique 1000 vérifications des conditions de début pour toutes les activités à prendre en compte.

Cela peut facilement devenir incontrôlable s'il y a trop d'activités ou si les incréments de temps étaient beaucoup plus petits. Ce type d'approche peut conduire à des simulations qui prouvent durer plusieurs heures, voire plusieurs jours. [37]

3.4.3 The ABC approach (Three-phase approach)

Dans cette approche, il existe deux types distincts d'événements, les événements liés et les événements conditionnels, qui sont programmés comme des procédures distinctes.

Un événement lié (ou *événement B*) est un événement dont l'occurrence est prévisible et peut donc être planifiée.

D'un autre côté, un événement conditionnel (ou *événement C*) est un événement dont l'occurrence dépend du respect de certaines conditions, par exemple, la disponibilité de certaines ressources.

Les activités terminées sont les événements appelés événements liés à la fin d'une durée prescrite. Le démarrage d'une activité dépend des conditions spécifiques remplies, appelées événement conditionnel.

Les procédures d'exécution de la simulation en trois phases sont résumées comme suit :

- **La phase A (time scan):**

Consiste à Vérifier la future liste des événements (temps de fin de toutes les activités actuellement en cours, c'est-à-dire événements liés) et Trouvez le premier de ces derniers et avancez l'horloge au temps du prochain événement programmé.

- **La phase B (appels B) :**

Consiste à exécuter tous les événements liés identifiés dans la phase A comme devant se produire à l'instant courant.

- **La phase C (appels C) :**

Tester tous les événements conditionnels et exécuter ceux dont les conditions sont satisfaites. [42]

3.5 Travaux connexes

L'application du modèle de file d'attente avec vacance pour la conservation d'énergie des réseaux de capteur sans fil a fait l'objet de plusieurs travaux, l'économie d'énergie est un problème important dans les réseaux de capteurs sans fil pour la majorité des nœuds de capteurs équipés de batteries non rechargeables.

Dans les sections suivantes, nous allons présenter des travaux dans ce domaine pour l'évaluation et l'analyse des performance des systèmes de file d'attente pour objectif de réduire la consommation d'énergie dans le nœud de capteurs.

3.5.1 Analyse et évaluation des performances des systèmes de files d'attentes en utilisant la théorie des files d'attentes

Parmi les modèles formels, nous trouvons le model des réseaux de pétri**, B. Boutoumi et N. Gharbi** [7] ont opté pour la modélisation d'un nœud capteur par une file d'attentes avec buffer et source finie et différentes politiques de vacances, et ce en se basant sur le modèle des réseaux de Pétri Stochastiques Généralisés.

Ils ont proposé pour conserver l'énergie et d'avoir un délai d'attente efficace est de modéliser le modèle de sommeil/actif des nœuds en réseau avec deux politiques en tenant compte à la fois de la limitation de la capacité du buffer et de la source de trafic. Lorsqu'un nœud est en mode sommeil, le capteur ne peut pas interagir avec son environnement par conséquent, l'énergie consommé est très basse. Durant le mode actif, le capteur écoute le canal, génère des paquets, reçois et transmet les paquets de données de ces voisins donc l'énergie consommé est considérable. Ainsi, pendant la période actif le capteur transite entre l'état oisif ou il peut générer et recevoir des paquets et l'état occupé où il peut générer, recevoir et émettre. Il est à noter également que l'énergie dissipée pour la transition entre l'état oisif et l'état occupé, génère une surconsommation d'énergie importante nommé l'énergie de transition.

Dans ce travail, un capteur en sommeil peut devenir actif et reprendre son service de deux manières différentes. La première par l'écoulement d'un certain temps aléatoire tandis que la deuxième sera un réveil immédiat si N paquets sont en attente dans son buffer.

Ils appliquent d'abord le modèle basé sur la politique N-vacance avec service exhaustif comme schéma de réveil en file d'attente. Ensuite, ils proposent un deuxième model basé sur une nouvelle politique de vacances appelé la politique hybride pour minimiser le délai d'attente.

Dans le deuxième modèle ou ils ont considéré une règle de fin de vacance hybride. Ils ont optés pour une fusion entre la politique N-vacance et une vacance d'une durée aléatoire dont le but est de réduire le délai d'attente des paquets dans le buffer avant d'être envoyé.

Le modèle hybride est particulièrement intéressant dans les réseaux de capteur sans fil ou le taux d'arriver est variable. Ceci permet d'éviter la saturation rapide du buffer et la dissipation de l'énergie due à la retransmission des paquets perdus dans le cas où le taux d'arriver est élevé. De plus dans le cas où le taux des arrives est petit, la vacance aléatoire permet de réduire le délai d'attentes des paquets.

Sur la base des modèles obtenus ils donnent les formules des principales mesures de performance du réseau de capteurs et une analyse détaillée pour montrer l'impact des deux politiques de vacances sur les performances du réseau.

Pour prolonger la durée de vie des nœuds de capteurs et réduire la consommation d'énergie, **Fuu-Cheng Jiang et Der-Chen Huang et Chao-Tung Yang et Chu-Hsing Lin et Kuo-Hsiung Wang** [8] proposent une stratégie de conception pour réduire la consommation

d'énergie moyenne du nœud de capteur en utilisant le modèle de file d'attente M / G / 1 et la politique Min (N, T) au lieu d'avoir la politique N-vacance.

Le point fondamental de leur approche est que La politique Min (N, T) peut réduire les temps moyens totaux de contention de support en ayant à la fois un compteur (N) et une minuterie (T) pour le contrôle du déclenchement sur un serveur radio pour transmettre des paquets en file d'attente. L'approche N-vacance entraîne un délai d'attente pour les paquets restant dans le buffer de la file d'attente. Pour éviter que les paquets en file d'attente n'attendent trop longtemps pour être envoyés au puits de données en temps voulu, une T -vacance est nécessaire chaque fois que le nombre total de paquets en file d'attente semble ne jamais atteindre N en cas de délai d'attente trop long.

Une analyse mathématique complète des paramètres de contrôle optimaux avait été effectuée. De nombreuses analyses de données et simulations avaient été effectuées pour valider le modèle proposé.

Pour optimiser la consommation d'énergie et le délai d'attente, **FC Jiang et DC Huang et KH Wang** [2] ont proposé un cadre d'optimisation pour la consommation d'énergie du nœud de capteur avec l'approche de N-vacance de la file d'attente M / G / 1 en adoptant une architecture des réseaux de capteurs « *many-to-one* ».

Un serveur radio d'un nœud de capteur se présente par trois états de fonctionnement : un état *idle, occupé* et d'un état de *démarrage*. Le système peut être dans l'un des trois états suivants

État *idle* : la fonction de transmission de paquets dans le serveur radio est désactivée et le nombre de paquets en attente dans la file d'attente est inférieur ou égal à $N - 1$ ($N \geqq 1$).

État de *démarrage* : le serveur radio est engagé dans le processus de contention de canal avant le début du service. Cela implique que le nombre de paquets en attente dans la file d'attente est supérieur ou égal à N.

État *occupé* : le serveur radio est occupé à offrir un service de transmission de paquets jusqu'à ce que la file d'attente soit vide.

Il y a une surcharge importante pour les collisions de paquets et les conflits de canaux résultant du redémarrage du processus de contention de support. Leur cadre peut être intégré à la plupart des protocoles MAC existants pour améliorer consommation d'énergie en réduisant le nombre total de conflits de support tout au long de la durée de vie d'un nœud de capteur générique.

Bachira Boutoumi et Nawel Gharbi [9] ont proposé deux modèles pour optimiser la consommation d'énergie, le premier modèle est basé sur l'utilisation de la politique N-vacance avec service exhaustif.

Une étude détaillée et argumenté par les résultats numériques a été établie pour montrer l'influence de cette politique sur la conservation de l'énergie dans les réseaux de capteurs sans fil.

Quant au deuxième modèle, afin de minimiser le délai d'attente des paquets, ils ont introduit la notion de vacances de travail dans le comportement des capteurs. Les auteurs ont modélisé un seul capteur en full-duplex durant sa période actif. Le capteur a été modélisé par une file d'attente M/M/1/K avec la politique "two thresholds working vacation Policy" qu'ils ont proposés. Dans ce modèle, un capteur en sommeil peut devenir semi-busy si le nombre de paquets en attente atteint le seuil N2 et commence le service avec le taux de vacance de travail μ2. Dès que le nombre de paquets en attente atteint le seuil N1 (N2<N1) le nœud passe à l'état occupé et commence le service avec le taux normal μ1 (μ2 < μ1).

Pour cela, ils ont développé les formules des principaux indices de performance stationnaire et de consommation d'énergie. Enfin, ils donnent une analyse détaillée qui prouve l'efficacité de l'approche proposée.

Le tableau ci-dessous montre un résumé de ces travaux.

Les références	**Formalisme mathématique**	**Paramètres analysés**	**Test et validation**
[7]	Les files d'attente avec vacances et le modèle des réseaux de pétri stochastiques généralisés.	La consommation d'énergie et le délai d'attente.	La méthode analytique.
[8]	Une file d'attente M/G/1 avec la politique Min (N, T).	La consommation d'énergie et le délai d'attente.	La méthode analytique et de simulation
[2]	Une file d'attente M/G/1 avec la politique N-vacance.	La consommation d'énergie et le délai d'attente.	La méthode analytique.
[9]	Une file d'attente M/M/1/K avec la politique "two thresholds working vacation Policy"et le modèle des réseaux de pétri stochastiques généralisés.	La consommation d'énergie et le délai d'attente.	La méthode analytique.

Tableau 3-1 Résumé de travaux connexes-1-

3.5.2 Analyse et évaluation des performances des systèmes de files d'attentes en utilisant la simulation à événement discret

Muhammad Dermawan MULYODIPUTRO et SUBANAR [43] ont appliquer la technique de simulation en modélisant la file d'attente avec un service cyclique dans le système d'intersection à signalisation avec deux politiques de service fermé et exhaustif.

Un modèle de file d'attente avec service cyclique est un système de file d'attente avec plusieurs files d'attente de clients, qui sont servis par un serveur en une seule séquence cyclique. La file d'attente avec service cyclique et le système d'intersection signalé ont les mêmes caractéristiques qu'un serveur et plus d'une file d'attente et le serveur sert la file d'attente par ordre cyclique.

Dans ce cas, le véhicule représente le client et les feux de signalisation représente le serveur. Les véhicules seront servis lorsque le feu de signalisation affichera un signal vert (heure de visite), la temps requis pour que le signal vert se déplace d'une intersection à l'autre est appelé un temps entièrement rouge ou un temps de commutation de plus, La période du signal rouge se présente comme étant le temps entre les visites.

Le modèle adopté est une file d'attente M / M / 1 avec un taux d'arrivée qui suit la distribution de Poisson et le taux de services qui suit la distribution exponentielle.

Dans la politique de *service fermé*, le serveur ne sert que les véhicules qui sont arrivés avant que le signal vert n'apparaisse à une intersection., le serveur ne sert les clients existants juste à son arrivé. Quant au *service exhaustif,* le serveur continue de travailler sur la file d'attente jusqu'à ce que la file d'attente soit vide (l'ajout de durée de signal vert à l'intersection, lorsque la durée du signal vert à une intersection terminée). Dans le service limité à 1, le serveur ne sert qu'un seul client avant de passer à la file d'attente suivante. Dans le service *exhaustif normal* les véhicules seront servis uniquement pendant le signal vert soit toujours actifs.

Les résultats de ce programme de simulation de file d'attente sont d'obtenir les caractéristiques et les performances du système, c'est-à-dire le nombre moyen de véhicules et le temps d'attente des véhicules à l'intersection et dans le système. Ensuite, à partir de ces valeurs, ils peuvent déduire laquelle des politiques de service cyclique normale exhaustive ou bien exhaustive et fermée qui est la plus appropriée lorsqu'elle est appliquée à un système d'intersection avec signalisation.

N Sadeghi et AR Fayek et NG Seresht [44] ont adopté la simulation à événements discrets flous (FDES) qui a été proposée pour simuler des projets de construction**.** De plus, ils fournissent une nouvelle méthodologie pour considérer l'incertitude subjective dans l'analyse des files d'attente floues dans les modèles (FDES) de construction.

Ils ont intégré la théorie de la file d'attente floue avec la méthodologie (FDES) proposé pour améliorer l'applicabilité du (FDES) dans les projets de construction.

La simulation à événement discret floue est une intégration de la théorie des ensembles flous avec (DES) qui fournit un cadre pour considérer l'incertitude subjective dans les modèles de simulation de construction. Les cadres (FDES) actuels calculent uniquement le temps de simulation comme résultat de la simulation. Cependant, les mesures de performance de la file

d'attente (par exemple, la longueur moyenne de la file d'attente et le temps d'attente) à travers d'importantes sorties de modèle de simulation pour la prise de décision, la recherche de goulots d'étranglement et l'optimisation des ressources de construction ne sont pas analysées dans les méthodologies (FDES) actuelles.

La méthodologie proposée est validée par des exemples de file d'attente qui sont présenté par un seul serveur avec des temps d'inter-arrivées et des temps de service flous et résolus mathématiquement, et ses aspects pratiques sont illustrés à l'aide d'un exemple d'opération de revêtement d'asphalte.

Mohammad Ehsanifar et Nima Hamtas et Mahshid Hemesy [45] ont proposé un modèle de simulation conçu pour prédire les indices de performance tels que le temps d'attente en analysant les composants de la file d'attente dans le monde réel dans des situations incertaines et subjectives.

Ils ont tenté de développer une distribution plus pratique des modèles de file d'attente au moyen de la théorie des ensembles flous par présenter les temps d'arrivée et les temps de service comme étant des variables floues.

L'objectif de leur étude est de prédire le temps d'attente de chaque client dans un modèle de file d'attente M / M / C dont les arrivées suit un processus de poisson et les temps de service sont distribuée de manière exponentielle. Ce modèle qui a **c** serveurs, ses temps inter-arrivées et temps de service sont des variables floues aléatoires. Ils ont opté pour utiliser la méthode de simulation pour modéliser des systèmes complexes et comprendre le comportement des files d'attente. Pour illustrer comment le modèle est exprimé pour analyser la file d'attente floue, ils ont considéré une situation réelle dans une banques locale dans laquelle il y a cinq guichets bancaires avec un type de service complètement similaire.

Enfin, un exemple numérique comme étude de ce système bancaire est résolu pour montrer la validité du modèle développé dans la situation réelle.

Le tableau ci-dessous montre un résumé de ces travaux.

Les références	Formalisme mathématiques	Paramètres analysé	Test et validation
[43]	Une file d'attente M/M/1 avec service cyclique avec deux politiques de vacance fermé et exhaustif.	Le délai d'attente.	La simulation à événement discret.
[44]	Files d'attente floue.	Le délai d'attente.	La simulation à évènement discret.
[45]	File d'attente M/M/C	Le délai d'attente.	La méthode de simulation.

Tableau 3-2 Résumé de travaux connexes-2-

3.6 Conclusion

La simulation se présente comme étant une méthodologie de résolution de problèmes indispensable pour la solution de nombreux problèmes du monde réel.

Dans ce chapitre nous nous sommes intéressés à la simulation et ses principes et les types de simulation. Nous avons mis des points sur la simulation à événement discret et son vocabulaire principal. Nous avons connu les déférente approche de simulation à événement discret ainsi que les travails liés à l'analyse et l'évaluation des performances des systèmes de files d'attente en utilisant des modèles analytiques et de simulation.

Dans le prochain chapitre, nous nous intéressons à la conception de l'outil de simulation que nous avons développé.

4. CONCEPTION DE L'OUTIL DE SIMULATION DES NŒUDS DE CAPTEURS SANS FIL

4.1 Introduction

La modélisation est l'une des tâches les plus importantes dans le processus du développement d'un système, consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels nous nous intéressons. De plus, modéliser un system avant sa réalisation permet de mieux comprendre son fonctionnement, sa complexité et d'assurer sa cohérence.

Au cours du présent chapitre, nous procédons à la modélisation d'un nœud de capteur, de plus, nous présentons la notion de vacance ainsi que les deux politiques de vacance (N-vacance et Vacance-Hybride) liée au modèle de files d'attente adopté ainsi que son architecture, nous présentons tous les aspects importants concernant la conception du simulateur. En fin, nous montrons les différents indices de performances et leurs définitions, Nous terminerons ce chapitre par une conclusion.

4.2 Modélisation des réseaux de capteurs sans fil

Nous avons présenté dans le premier chapitre que les réseaux de capteurs sont très contraints et limité en termes de ressources. Dans la majorité des RCSF, les nœuds capteurs opèrent avec des batteries irremplaçables et leurs sources d'énergie sont très limitées, et la durée de vie devient très courte à cause de l'épuisement d'énergie des nœuds capteurs.

Pour bien décrire un système nous devons faire une modélisation, un modèle de file d'attente avec vacance est parmi les modèles que nous allons utiliser pour représenter le comportement d'un nœud de capteur.de plus d'exploiter la notion de vacances par l'utilisation des politiques de vacances dans le comportement des capteurs sans fil, ce qui permet d'économiser l'énergie précieuse durant leur temps libre.

Pour mieux décrire et comprendre la modélisation et les solutions proposée, nous allons présenter quelques notions importantes dans les sections suivants.

4.2.1 Vacance d'un nœud de capteur

D'après ce qui a été abordé dans le premier chapitre et d'après ce que nous avons comme informations sur les différentes sources de consommation d'énergie d'un capteur, Il est à noter que le passage d'un mode à un autre mode ainsi que les différentes lectures et écritures mémoire de l'unité de traitement ont un taux de consommation d'énergie [18]. Cependant, des expériences ont montré que l'unité de transmission est le plus gourmand en matière d'énergie. De plus, limiter son utilisation en le mettant dans un état inactif (se présente comme étant un état de vacance) conduira à un gain considérable sur la durée de vie de la batterie du capteur et ainsi de tout le réseau. Ceci étant, on peut confondre la notion de vacance de l'unité de transmission avec la notion de vacance d'un capteur.

Nous considérons un nœud de capteur qui alterne entre deux principaux modes : *inactif* et *actif*. Lorsqu'il est en mode *inactif* (le serveur a pris des vacances), le capteur peut uniquement recevoir des paquets de l'environnement, ce qui conduit à une consommation d'énergie très

basse. Durant le mode *actif*, le capteur écoute le canal, génère des paquets, reçois et transmet les paquets de données de ces voisins donc l'énergie consommé est considérable. Ainsi, pendant la période actif le capteur transite entre l'état oisif ou il peut générer et recevoir des paquet et l'état occupé où il peut générer, recevoir et émettre. Il est à noter également que l'énergie dissipée pour la transition entre l'état oisif et l'état occupé, génère une surconsommation d'énergie importante nommé l'énergie de transition.

4.2.1.1 La politique N-vacance

La politique N-vacances est utilisée comme un plan efficace pour réveiller le serveur de files d'attente. Cependant, cette politique augmente inévitablement le délai d'attente pour les paquets en file d'attente. Mais, elle est connue comme une technique efficace pour optimiser la consommation d'énergie dans les nœuds de capteurs sans fil pendant la période active en minimisant l'énergie de commutation causée par la fréquence de transition entre les états oisif et occupé, en réduisant le nombre de fois où l'unité de transmission commute entre les deux états.

A cet effet, un capteur en état inactif passe à l'état actif si et seulement si le nombre de paquets en file d'attente atteint le seuil « N ». [7]

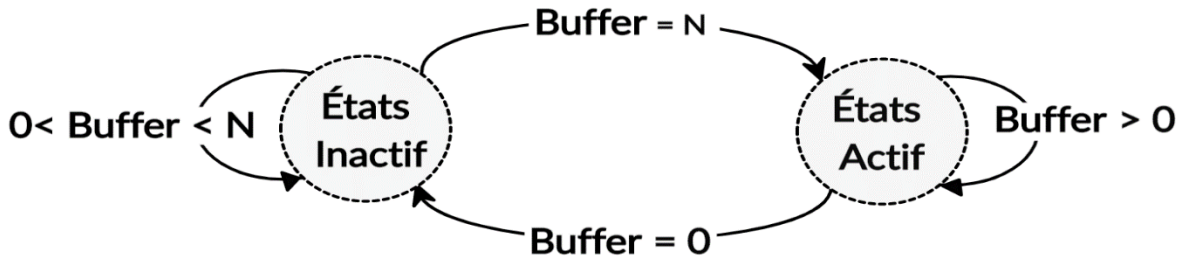

Figure 4.1 Diagramme d'état de transition d'un nœud de capteur avec la politique N-vacance

4.2.1.2 La politique vacance-Hybride

Elle est considérée comme une combinaison de la politique N-vacance avec les vacances aléatoires. Alors, un nœud peut passer de l'état inactif à l'état actif et commence la transmission des paquets en file d'attente, si sa taille de buffer a atteint le seuil N ou après avoir pris des vacances d'une durée aléatoire même si le nombre de paquets dans le buffer est strictement inférieur à N. Elle est considérée pour éviter le délai d'attente pour les paquets en attente dans le cas où le taux d'arriver est élevé.

À cette fin, le nœud du capteur peut passer de l'état inactif à l'état occupé soit :

- Au moment de l'arrivée du Nième paquet.
- A la fin d'une période de vacances qui est une durée aléatoire distribuée de façon exponentielle avec le paramètre « ζ », même si le nombre de paquets dans le tampon est inférieur à N. [7]

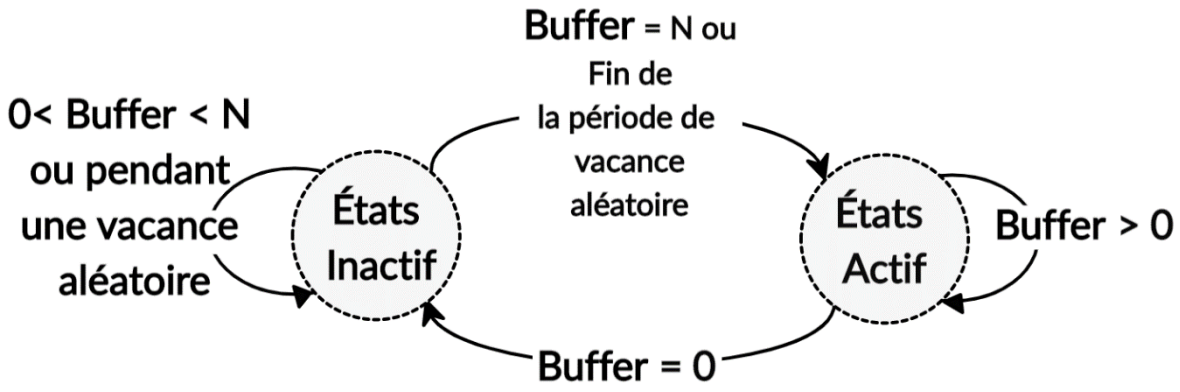

Figure 4.2 Diagramme d'état de transition d'un nœud de capteur avec Politique hybride.

4.2.2 Architecture du modèle proposé

Nous décrivons le système correspondant à un nœud de capteur par un modèle de file d'attente à un seul serveur avec vacance et un buffer de capacité finie « K », selon la notation (M/M/1/K). Dans lequel les paquets arrivent selon un processus de poisson avec le paramètre « λ » et le temps entre les arrivées est distribué exponentiellement avec la moyenne « 1/λ », les temps de service sont distribués de manière exponentielle avec une moyenne « 1/μ » . [46]

Le serveur possède deux états de fonctionnement : *actif* (décrite en vert) et *inactif* (décrite en rouge), ce qui correspondent aux états de service et de vacances du serveur respectivement. Le passage de l'état *inactif* à l'état *actif* est contrôlé par la politique N-vacance ou la politique Vacance-Hybride (une seul à la fois).

La Figure 4.3 ci-dessous montre l'architecture du modèle proposé pour le capteur.

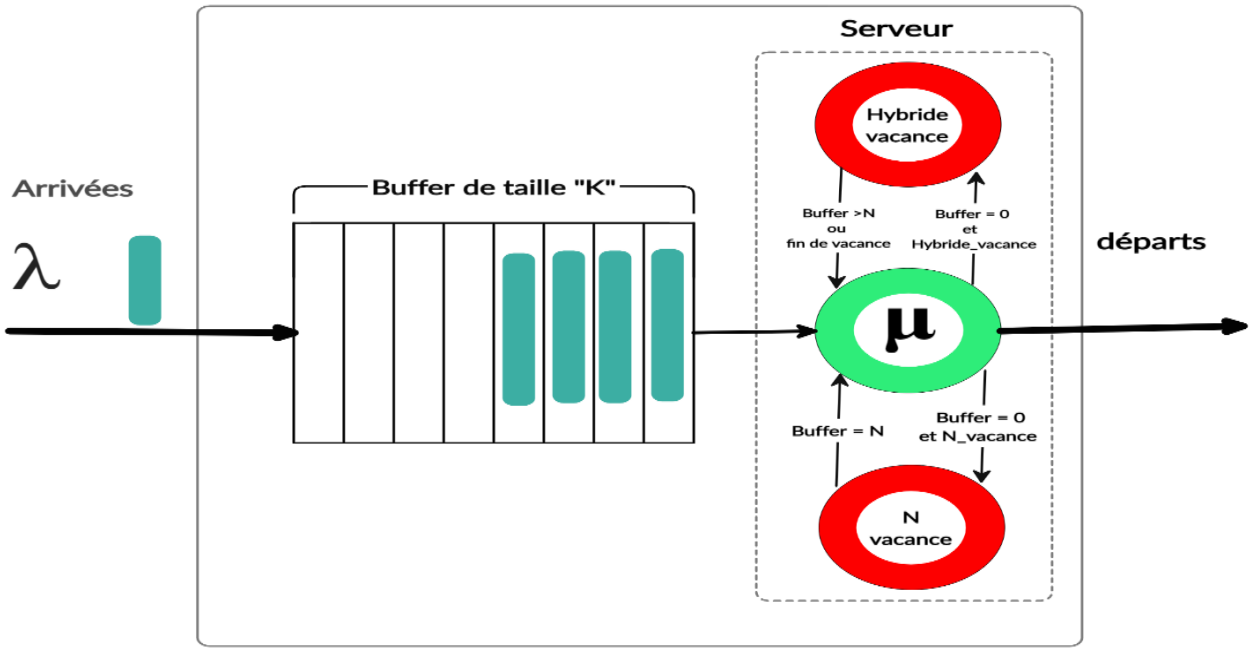

Figure 4.3 la structure de base du nœud de capteur avec le modèle de file d'attente

Initialement, le capteur est dans l'état inactif, il peut uniquement recevoir des paquets et les stocker dans le buffer. Nous considérons les deux politiques N-vacance et Vacance-Hybride comme un schéma de réveil en file d'attente pour le passage du serveur de l'état inactif à l'état actif.

Dès que le nombre de paquet en attente dans le buffer atteint le seuil « N » (pour les deux politiques) ou la période de vacance aléatoire de l'autre politique a expiré. Le capteur passe à l'état actif, le nœud peut recevoir et transmettre les paquets en attente jusqu'à ce que le buffer se vide (service exhaustif). En ce moment, il passe à l'état inactif (pour la politique hybride si le temps expire et le buffer est vide donc le serveur prend une autre période aléatoire de vacance, les périodes sont générés aléatoirement de manière distribuée exponentiellement avec la moyenne « 1/ ζ ».

La figure 4.4 ci-dessous présente le flux d'exécution pour le comportement d'un nœud capteur lors d'arrivée d'un paquet.

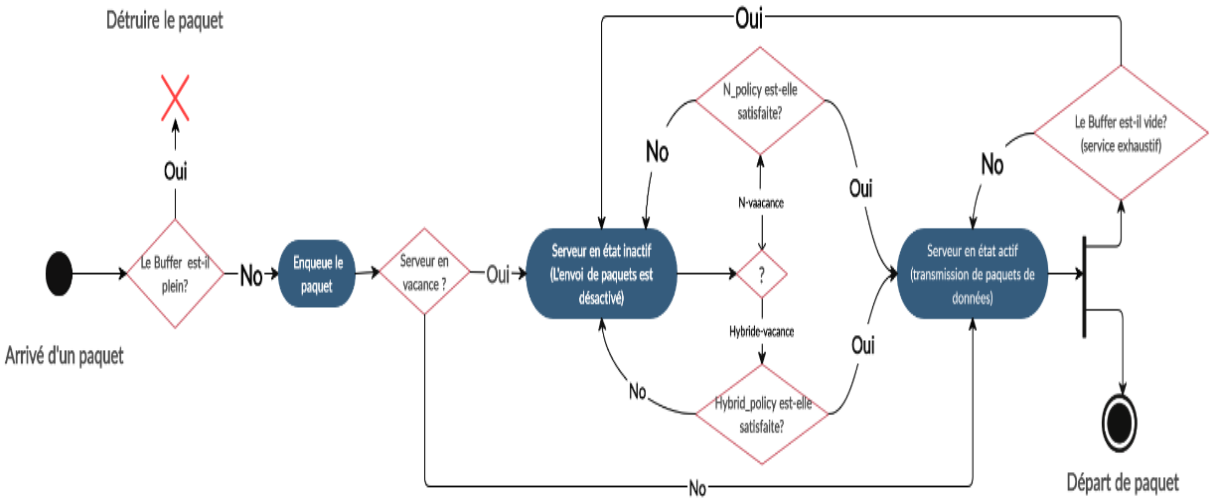

Figure 4.4 Diagramme d'activité représente le comportement de capteur sans fil

4.3 Conception du simulateur

Au sein du simulateur, nous avons choisi d'appliquer les techniques de la simulation à événements discrets pour bien imiter le comportement du système que nous avons étudié, ce système se change d'états à des points discrets, en revanche, ce simulateur permet de calculer les mesures de performances de la file d'attente qui est l'un des objectifs de ce travail, telles que la longueur moyenne et le temps d'attente...etc.

Les principaux concepts de la simulation à événement discret que nous avons utilisé sont :

- ***Le system étudié* :** le comportement d'un nœud de capteur.
- ***Le model proposé* :** une file d'attente M/M/1/K avec vacance.
- ***Entités* :** serveur, paquets.
- ***Attributs :*** actif, inactif.

Les autres concepts seront expliqués dans les prochaines sections.

Parmi les différentes approches de simulation à événements discrets que nous avons introduites (chapitre 3, section 4), c'est l'approche de planification d'événements « *Event Scheduling Approach* » pour des raisons d'efficacité et de rapidité de calcul.

Par la suite, nous allons présenter les variables et les événements ainsi que les routines nécessaires à la réalisation de ce travail.de plus, nous terminons par présenter l'architecture conceptuel de ce simulateur par un diagramme de classe.

4.3.1 Variables

Les variables(paramètres) nécessaires utilisées et leurs descriptions sont décrite dans le tableau 4-1 ci-dessous.

Nom de variable	Description
Horloge	C'est la variable globale représentant le temps simulé, l'ordonnanceur est responsable de l'avancement de ce variable
FEL	La liste triée d'événements (Futur Event List)
Buffer	La file d'attente de paquets
Etat-du-Serveur	Représente l'état instantané du serveur : *libre* ; *occupé* ; *en vacances*
X	Compteur de paquets dans le Buffer
K	La capacité maximale du buffer
N	Le seuil pour quitter l'état de vacance
λ	Le taux d'arrivée
μ	Le taux de service
ζ	Le taux de vacance pour la politique hybride

Tableau 4-1 Tableau de variables

Le tableau 4-2 ci-dessous présent les variables concernant la consommation d'énergie :

Nom de variable	Description
EC_i	La consommation d'énergie de l'unité de transmission à l'état inactif
EC_b	La consommation d'énergie de l'unité de transmission à l'état actif
EC_s	La consommation d'énergie du changement de l'état de capteur entre réception et transmission
EC_h	La consommation d'énergie de transmission pour chaque paquet

Tableau 4-2 Tableau de variables

4.3.2 Evénements

Dans la simulation à événements discrets, il est nécessaire de s'assurer que les événements se produisent dans le bon ordre et au moment approprié [47], pour cela, nous avons garanti la planification des événements par une liste triée selon l'heure nommée « FEL ». Chaque élément de la liste contient l'heure à laquelle l'événement doit se produire et le type d'événement qui doit être exécuté à ce moment.

Pour mieux comprendre le fonctionnement du simulateur, dans ce qui suit. Nous allons présenter les déférente types événements que nous avons utilisé dans la simulation.

- **L'événement arrivée**

Représente l'arrivé d'un paquet au système, il est généré pour la première fois lors de l'initialisation comme une condition de départ pour planifier le premier paquet. Les autres événements d'arrivées sont générés aléatoirement avec des intervalles distribués exponentiellement de moyenne « 1/λ » entre eux.

- **L'événement début du service**

Représente l'entrée d"un paquet au serveur et le début de son traitement. Début du service est générer lorsqu'une arrivée du paquet si le serveur vide ou bien lors de départ d'un paquet si le buffer n'est pas vide.

- **L'événement départ**

Représente la sortie d'un paquet du système, l'événement départ est généré lorsqu'un paquet finit sa période de traitement au serveur et quitte le système.

- **L'événement fin de vacance**

Nous avons ajouté un type d'événement spécial pour notre modèle proposé de la politique hybride vacance, c'est l'événement fin de vacance. Cet événement représente fin de période vacance, il est généré lorsqu'un paquet quitte le system et le buffer est vide (le serveur passe au vacance), ce type est conçu pour réveiller le serveur après avoir pris des vacances d'une durée aléatoire même si le nombre de paquets dans le buffer est strictement inférieur à N.

4.3.2 Les Routines

Au sein du simulateur, Le programme de simulation contient plusieurs routines telles que, *initialisation*, *boucle principale* et *statistiques*. etc., Ainsi, chaque événement est simulé par sa routine. Les routines se présente comme étant des fonctions pour mettre à jour les variables d'état du système et planifient l'occurrence d'événements. Nous allons les présenter dans les sections suivantes et les expliquer avec des pseudo algorithmes.

- **Routine Arrivée**

C'est la routine exécutée par l'événement « arrivée ». En générale, Cette routine présente l'arrivé d'un nouveau paquet pour l'insérer dans le buffer et vérifier s'il y a possibilité de rendre le serveur actif s'il ne l'est pas.

La figures 4.5 ci-dessous montre un pseudo algorithme pour la routine arrivé.

Routine Arrivée

```
0 : Début ;
1 :  Si  ( X < K ) Alors   // le Buffer n'est pas plein
2 :        Enfiler le paquet arrivant ;
3 :        X++ ; // incrémentez le nombre de paquets
4 :        Si ( serveur en vacance ) Alors
5 :             Si (X == N ) Alors // Le seuil a été atteint
6 :                  Rendre le serveur actif ;
7 :                  Planifiez l'événement « Début de service » : heure de l'événement = Horloge   ;
8 :                  Si (le type de vacance == Hybride)  Alors
9 :                       Supprime l'événement  « fin de vacances » de la liste FEL;
10 :                 Fin.
11:            Fin.
12 :       Fin.
13 :  Sinon
14 :        Planifiez le prochain événement « Arrivée »: heure de l'événement = Horloge + période inter-arrivée ;
          //les périodes sont générer aléatoirement de manière  distribué exponentiellement avec la moyenne « 1/λ »
15 :  Fin.
15 : Fin.
```

Figure 4.5 Pseudo algorithme pour la routine « Arrivée »

➢ Routine début de service

C'est la routine exécutée par l'événement « début de service ». Elle sert à mettre le serveur occupé pour servir un paquet depuis le buffer et le faire planifier un événement de départ. La figures 4.6 ci-dessous montre un pseudo algorithme d'une routine début de service.

Routine Début de service

```
0 :  Début ;
1 :  Définir le serveur occupé ;
2 :  Défiler le paquet du buffer  ;
3 :  Planifiez un événement  « Départ » pour ce paquet : heure de l'événement = Horloge + période de service ;
     //les temps sont générer aléatoirement de manière  distribué exponentiellement avec la moyenne « 1/μ »
4 :  Fin.
```

Figure 4.6 Pseudo algorithme pour la routine « début de service »

➢ Routine départ d'un paquet

C'est la routine exécutée par l'événement « départ ». En générale, cette routine met le serveur en état libre, si le buffer est vide, elle le met en vacances, de plus, elle fait planifier les événements nécessaires. Un pseudo algorithme est illustré dans la figure 4.7 ci-dessous.

Routine Départ

```
0 : Début ;
1 :  Mettre le serveur libre ;
2 :  Si ( X == 0 ) Alors   // le Buffer est vide
3 :        Mettre   le serveur en  vacance ;
4 :        Si ( le type de vacance == Hybride )  Alors
5 :                Planifier l'événement « Fin de vacance » : heure de l'événement = Horloge + période de vacance ;
          //les périodes sont générer aléatoirement de manière  distribué exponentiellement avec la moyenne « 1/ ζ »
6 :         Fin.
7 :   Sinon
8 :     Planifier l'événement «Début de service » : heure de l'événement = Horloge;
9 :   Fin.
10 : X-- ; // décrémente le nombre de paquets
11 : Fin.
```

Figure 4.7 Pseudo algorithme pour la routine « départ ».

➢ Routine de fin de vacance

C'est la routine exécutée par l'événement « fin de vacance ». Elle est conçue pour indiquer que le serveur a fini ses vacances et vérifier s'il y a possibilité d'avoir de nouvelles vacances. La figure 4.8 ci-dessous représente un pseudo algorithme d'une routine fin de vacance.

Routine Fin de vacance

```
0 : Début ;
1 :  Si ( X==0 ) Alors    // le Buffer est vide
2 :       planifier un nouvel évènement « Fin de vacance » : heure de l'événement = Horloge + période de vacance ;
         //les périodes sont générer aléatoirement de manière  distribué exponentiellement avec la moyenne « 1/ ζ »
3 :  Sinon
4 :     Planifiez un événement « Début de service » : heure de l'événement = Horloge ;
9 :   Fin.
10 : Fin.
```

Figure 4.8 Pseudo algorithme pour la routine « fin de vacance »

- **Routine d'initialisation**

Cette routine contient tous les paramètres nécessaires du system saisis par l'utilisateur et leurs initialisations qui reflètent l'état du système au temps zéro tell que l'initialisation de l'horloge et mettre le buffer vide, les paramètres liés à la consommation d'énergie et préciser le taux d'arriver et le taux de service, le seuil, le taux de vacance. etc.de plus, un événement de type Arrivé doit être planifié comme une condition de départ.

La routine d'initialisation permet généralement de faire varier un paramètre d'une manière spécifiée.

Routine d'initialisation

```
0 :  Début ;
1 :  Définir Horloge = 0 ;
2 :  Définir X = 0 ;
3 :  Définir le Buffer sur vide ;
4 :  Définir le serveur en vacance ;
5 :  Définir la liste d'événements « FEL » à vide ;
6 :  Récupérer et initialiser les valeurs de : K,N , λ , μ,  ζ ;
7 :  Récupérer et initialiser les valeurs de : ECi, ECb, ECs, Ech ;
8 :  Planifiez un événement  « Arrivée »  à l'instant 0 ;
4 :  Fin.
```

Figure 4.9 Pseudo algorithme pour la routine « Initialisation »

- **Routine de la boucle principale (*l'ordonnanceur*)**

La routine de la boucle principale rassemble toutes les autres routines.au début, Elle fait un appel à la routine d'initialisation et elle termine par appeler la routine de sorite statistique.

A l'intérieur de la boucle elle fait appeler une routine en fonction de type d'événement précisé dans la liste d'événement « FEL » durant la simulation avec un temps prédéfinie.

Routine de la boucle principale

```
0 :  Début ;
1 :  Appeler la fonction d'éexécution de la routine d'initialisation ;
2 :  TanQue(Horloge <= un temps prédéfinie) // condition de résiliation par défaut est 10000 unité de temps
3 :      Défiler le prochain événement de la liste d'événements « FEL » ;
4 :      Horloge = l'heure de l'événement ;
5 :      Appeler une fonction d'exécution de routine  en fonction du type d'événement ;
6 :  FinTanQue ;
7 :  Appeler la fonction d'exécution de la routine statistiques;
8 :  Fin.
```

Figure 4.10 Pseudo algorithme pour la routine « Boucle principale »

- **Routine statistique**

C'est la routine de sortie qui est exécutée à la fin de la simulation. Elle est conçue pour donner des résultats des mesures et statistiques finales de la simulation. Les résultats se sont des mesures qui seront détaillés dans la section 6 de ce chapitre.

4.3.3 Diagramme de classe

Les diagrammes UML structurels sont utilisés pour modéliser la structure statique du système en utilisant des concepts orientés objets. Parmi ces diagrammes, nous allons nous focaliser sur le diagramme de classes qui décrit le modèle conceptuel du système. Il est considéré comme le plus important de la modélisation orientée objet.

Le diagramme de classes et sa description sont illustrés respectivement dans la figure 4.11 et le tableau 3 suivants :

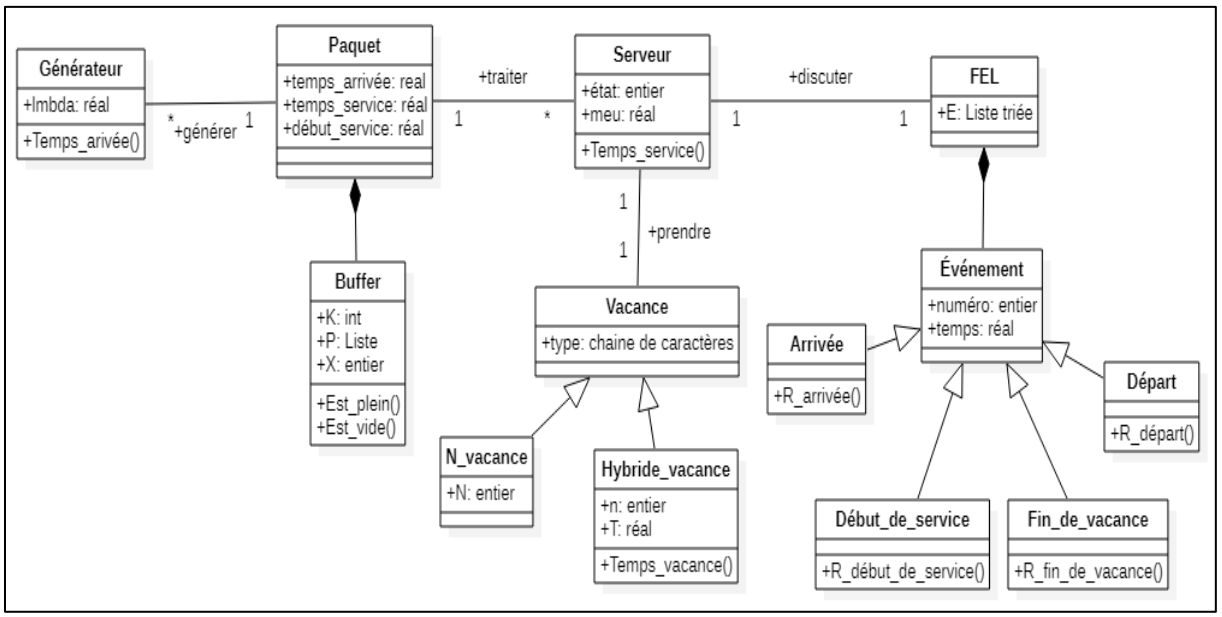

Figure 4.11 Diagramme de classes

Pour bien expliquer le diagramme précédant, le tableau 3 suivant comporte une description de données des différentes classes utilisé :

Classe	Attribut	Type	Désignation	Méthode
Générateur	lambda	réal	Le taux d'arrivée « λ ».	Temps_arrivée()
Paquet	Temps_arrivée	réal	Le temps d'arrivée de paquet.	
	Temps_service	réal	Le temps du service de paquet.	
	Début_service	réal	Le temps du début de service de paquet.	
Buffer	K	entier	La taille maximale de Buffer.	Est_plein () Est_vide()
	P	Liste	La file d'attente de paquets (Buffer)	
	X	entier	Nombre de paquet dans le buffer.	
Serveur	état	entier	L'état de serveur 0 : libre/ 1 : occupé/ 2 : vacance.	Temps_service()
	meu	réal	Le taux de service.	
Vacance	type	Chaine de caractères	Le type de vacance	
N_vacance	N	entier	La seuil	
Hybride_vacance	N	entier	La seuil	Temps_vacance()
	T	réal	Le taux de vacance	
Événement	numéro	entier	Type de événements: 0:Arrivée/1:Début_de_service 2: Départ/3: Fin_de_vacance	
	temps	réal	Le temps d'arrivée d'événement.	
Arrivée			L'événement Arrivée	R_arrivée()
Début_de_service			L'événement Début de service	R_début_de_service ()
Départ			L'événement Départ	R_départ()
Fin_de_vacance			L'événement fin de vacance	R_fin_de_vacance()
FEL	E	Liste triée	List d'événement triée selon le temps d'arrivée	

Tableau 4-3 Tableau descriptif des classes

4.4 Les indices de performances

Après avoir vue les différents aspects nécessaires liée à la conception du simulateur, nous devons également connaitre les principaux indices de performance du modèle de file d'attente utilisé, ces indices seront par la suite utiles pour étudier l'impact des deux politiques de vacances et leurs effets sur la performance du RCSF, ces indices sont présentés dans les sections suivantes.

4.4.1 Taux de perte

Il correspond au cas où un nouveau paquet ne peut être pas stocké à cause de la saturation du buffer. Autrement, c'est la fraction des nombres de paquet ayant été rejeté sur le nombre total des paquets. D'un autres coté, il peut représenter le pourcentage des paquets rejeté.

Pour calculer sa valeur la figure 4.12 montre une suite d'instruction à suivre.

Le taux de perte

Lors de L'événement d'arrivée

```
0 : cpt paquet++ ; //compteur de paquets arrivant
1 :  Si ( X==k ) Alors  //paquet est rejeté car le buffer est saturé
2 :          cpt paquets perdu++ ;
3 :  Fin.
```

À la fin de la simulation

```
4 :  taux de perte = cpt paquet perdu / cpt paquet ;
5 : Fin.
```

Figure 4.12 Pseudo algorithme « calcule du taux de perte »

4.4.2 Débit

Il est défini comme le taux de paquets qui peuvent être traitées par le serveur. Autrement, il correspond à la fraction du nombre de paquets ayant été servies sur le temps total de simulation, Le débit est mesuré en paquets par unité de temps, qui est généralement « seconde ». La manière de calculer sa valeur est illustré dans la figure 4.13.

La Débit

Lors de L'événement de départ

```
0 :  cpt paquet servi++ ;
```

À la fin de la simulation

```
1 :  Débit =  cpt paquet servi  / temps total de simulation ;
2 : Fin.
```

Figure 4.13 Pseudo algorithme « calcule du débit »

4.4.3 Probabilité de blocage (*saturation du buffer*)

Correspond à la probabilité que la capacité du buffer du capteur atteigne sa limite K (taille maximale du buffer) durant la simulation. En d'autres termes, Elle représente la fraction des temps auxquels le système a connu une saturation (le buffer est plein et il ne peut pas recevoir des paquets) sur le temps total de la simulation. Une suite d'instruction dans la figure 4.14 illustrant la manière pour calculer sa valeur.

La probabilité de blocage

Lors de L'événement d'arrivée

```
0 :  Si (X==K) Alors  //le Buffer est saturé
1 :      Début saturation =Horloge ;
2 :  Fin.
```

Lors de L'événement de départ

```
3 :  temps saturation += (Horloge - Début saturation) ;
```

À la fin de la simulation

```
4 :  Blocage=temps saturation / temps total de simulation ;
5 : Fin.
```

Figure 4.14 Pseudo algorithme « calcule de probabilité du blocage »

4.4.4 Probabilité que l'unité de transmission soit en vacances « P_v»

Celle-ci correspond à la probabilité que le serveur soit à l'état de vacance. Si aucun paquet se trouve dans la file d'attente des paquets(buffer) le serveur prend des vacances après un départ de paquet. Autrement, elle représente la fraction des temps de vacances sur le temps total de simulation.

Pour plus de détaille la figure 4.15 ci-dessous montre bien la manière pour calculer cette probabilité.

La probabilité de vacance

Lors de L'événement de départ

0 : Si (X==0) Alors // le Buffer est vide

1 : Début vacance = Horloge ;

2 : Fin.

Lors de L'événement d'arrivée

3 : Si (X == N) Alors

4 : Temps de vacance+=(Horloge – Début vacance) ;

5 : Fin.

Lors de L'événement de Fin de vacance // pour la politique Hybride

6 : Si (X>0) Alors

7 : Temps de vacance+=(Horloge – Début vacance) ;

8 : Fin.

À la fin de la simulation

9 : Si (la simulation est terminée par serveur en vacance) Alors

10 : Temps de vacance+=(Horloge – Début vacance) ;

11 : Fin.

12 : probabilité de vacance=temps de vacance / temps total de simulation ;

13 :Fin.

Figure 4.15 Pseudo algorithme « calcule de probabilité de vacance »

4.4.5 Temps de réponse moyenne (*Temps de séjour*)

Représente le temps moyen passé par un paquet depuis l'instant de son arrivé jusqu'à l'instant de son départ du serveur. Ceci est illustré dans la figure 4.16 ci-dessous.

La temps de réponse moyenne

Lors de L'événement de départ

0 : temps de réponse = T_now – le temps d'arrivée de paquet ;

1 : cpt paquet servi++ ;

À la fin de la simulation

2 : moyenne de réponse= temps de réponse / cpt paquet servi ;

3 : Fin.

Figure 4.16 Pseudo algorithme « calcule temps de réponse moyenne »

4.4.6 Temps d'attente moyenne

Cette quantité est estimée comme la moyenne arithmétique de tous les temps d'attente des paquets observés. Cependant, pour calculer cette estimation, nous devons mesurer pour chaque paquet le temps écoulé depuis son arrivée jusqu'au temps de son début du service. Autrement, Elle représente la fraction des temps d'attente pour chaque paquet sur le nombre total des paquets servies.

La temps d'attente moyenne

Lors de L'événement début de service

0 : temps d'attente=T_now – le temps d'arrivée de paquet ;

Lors de L'événement de départ

1 : cpt paquet servi++ ;

À la fin de la simulation

2 : moyenne d'attente= temps d'attente / cpt paquet servi ;

3 : Fin.

Figure 4.17 Pseudo algorithme « calcule temps d'attente moyenne ».

4.4.7 Energie consommée

Elle est calculée à la fin de la simulation par cette formule :

$$EC=P_v * EC_i +(1-P_v)* EC_b+EC_s*N_c+Q*EC_h$$

Pour calculer l'énergie consommé, nous aurons besoin de calculer la langueur moyenne du buffer et le nombre de cycles suivants :

- **La longueur moyenne du buffer (Q)**

Elle représente le nombre moyen de paquets en attente dans la file d'attente (le buffer)

La longueur moyenne du Buffer (Q)

Lors de la boucle principale

0 : Défiler l'événement ;

1 : T[X] += le temps d'événement – Horloge ; // X : le Compteur de paquets dans le Buffer

À la fin de la simulation

1 : Pour tout pair<clé, valeur> S ∈ T

2 : Q= S.clé * S.valeur ;

3 : FinPour

4 : Fin.

Figure 4.18 Pseudo algorithme « calcule la langueur moyenne du tampon ».

❖ **Le nombre de cycles « N_c»**

Il correspond au nombre de transitions de l'état inactif à l'état occupé par unité de temps. Pour plus de détaille la figure 4.19 présente la manière de calculer cette variable.

Le nombre de cycles (Nc)

Lors de L'événement d'arrivée

```
1 :  Si (( X == N  )&&(server en vacance)) Alors //le serveur reprend le service
2 :      Nbr_cycles ++ ;// nombre de cycles
3 :  Fin.
```

Lors de L'événement de Fin de vacance // pour la politique Hybride

```
4 :      Nbr_cycles ++ ;
5 :   Fin.
```

À la fin de la simulation

```
6 :   nombre de cycles= Nbr_cycles / Horloge ;
7 :  Fin.
```

Figure 4.19 Pseudo algorithme « calcul du nombre de cycles »

4.5 Conclusion

L'objectif de ce chapitre était consacré pour présenter une modélisation du system étudiée, de plus avoir une conception sur les déférentes aspects important liés au simulateur et à l'approche de simulation à événement discret que nous avons introduit.

Nous avons présenté dans un premier temps les politiques de vacance proposées, puis une modélisation avec les files d'attente avec vacance. Ensuite, nous avons présenté les indices de performances importante pour l'évaluation des performances d'un noud de capteur.

Dans le prochain chapitre, nous nous intéressons à la mise en œuvre des différents résultats expérimentaux qui permettent d'étudier l'impact des deux politiques de vacances sur la performance du RCSF.

5. RÉSULTATS ET ÉTUDES EXPÉRIMENTALES

5.1 Introduction

L'objectif principal de ce chapitre est l'implémentation d'une application pour l'évaluation des performances d'un nœud capteur. Cette application implémente les algorithmes proposés dans le chapitre précédant en fonction des paramètres du système introduit par l'utilisateur. De plus, elle affiche les différents indices de performances du nœud capteur pour le modèle proposé précédemment.

Nous commençons ce chapitre par la présentation des différentes fonctionnalités de notre application. Suivie d'une explication du fonctionnement du générateur de variables aléatoires. Finalement, une étude expérimentale comparative entre les politiques N-vacance et Hybride-vacance pour montrer l'influence de certains paramètres de système comme le taux d'arrivée des paquets, le taux de service et le seuil N sur la consommation d'énergie pour chaque politique. Nous terminons par une conclusion.

5.2 Présentation de l'application

5.2.1 Choix du langage C#

L'implémentation de notre application est réalisée sous le système Windows, en utilisant C# sur l'outil Microsoft Visual Studio 2019. C# est un langage de programmation polyvalent à plusieurs paradigmes, simple, performant, orienté objet, at à typage fort. Il a été développé vers 2000 par Microsoft. Ses principaux avantages sont les suivants :

- Il est facile de modéliser un système réel car les objets réels sont représentés par des objets de programmation en POO.
- Le temps de compilation et d'exécution du langage C # est rapide.
- Les objets sont traités par leurs données et fonctions membres, Il répond aux besoins des utilisateurs.
- La facilité de la syntaxe et le fonctionnement.
- Un moyen plus simple pour suivre le déroulement d'un programme.

Notre application contient plusieurs fenêtres, qui sont décrites dans les sections suivantes :

5.2.2 Fenêtre d'accueil

La fenêtre d'accueil présenté dans la figure 5.1 est la première fenêtre qui apparait lors du lancement de l'application et qui contient un abstract sur l'application. Cette fenêtre contient un bouton **Start**, son but est de lancer la fenêtre de choix.

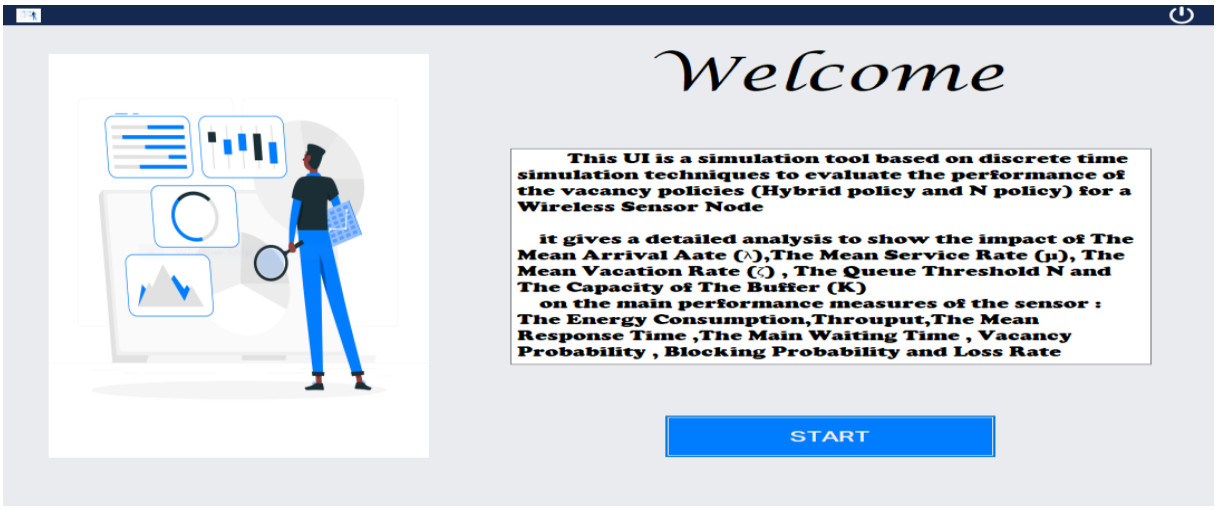

Figure 5.1 La fenêtre d'accueil

5.2.3 Fenêtre de choix

Cette fenêtre qui est illustrée dans la figure 5.2 permet à l'utilisateur de choisir l'un des paramètres afin d'étudier l'influence de la variation de ce dernier sur la performance des deux politiques. Ensuite, l'utilisateur doit remplir les champs concernant les paramètres de system telles que le taux d'arrivée, le taux de service, le seuil N, la capacité du buffer K, les d'énergies EC_i, EC_b, EC_s et EC_h afin d'obtenir les valeurs des différents indices de performances en fonction de paramètre choisi. Cette fenêtre contient un bouton **Run** pour but de lancer la simulation.

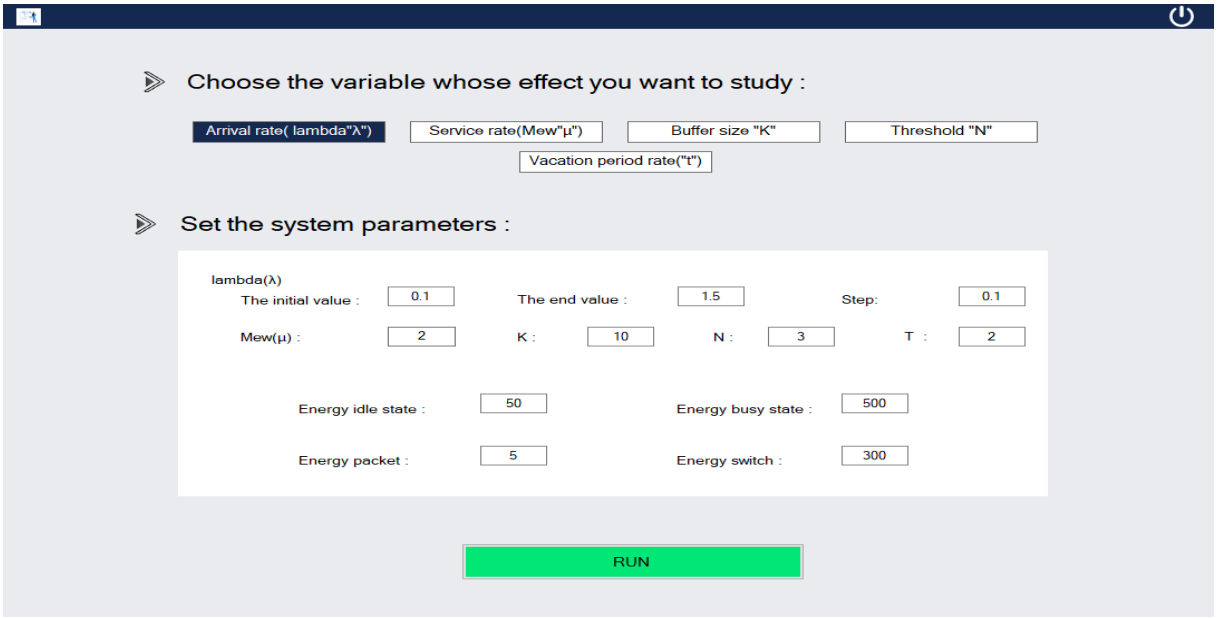

Figure 5.2 La fenêtre de choix

5.2.4 Fenêtre des résultats

Cette fenêtre qui est illustrée dans la figure 5.3 permet d'afficher les résultats obtenus de simulation, elle présente les paramètres de system entrée précédemment et les graphes pour chaque indice de performance.

Cette fenêtre contient les buttons :

- **Retour « flèche » :** l'utilisateur a la possibilité de revenir à la fenêtre de choix et lancer une nouvelle simulation.
- **Energy consomption :** pour afficher le graphe de la consommation d'énergie du nœud capteur.
- **Throughput :** permet d'afficher le graphe de débit.
- **Mean Responce time :** permet d'afficher le graphe du temps de réponse moyen.
- **Mean waiting time :** permet d'afficher le graphe de temps d'attente moyen.
- **Vacancy pronbility :** permet d'afficher le graphe de temps de vacance moyen.
- **Loss rate :** permet d'afficher le graphe du taux de perte.
- **Blocking probability :** permet d'afficher le graphe du taux du blocage du system.

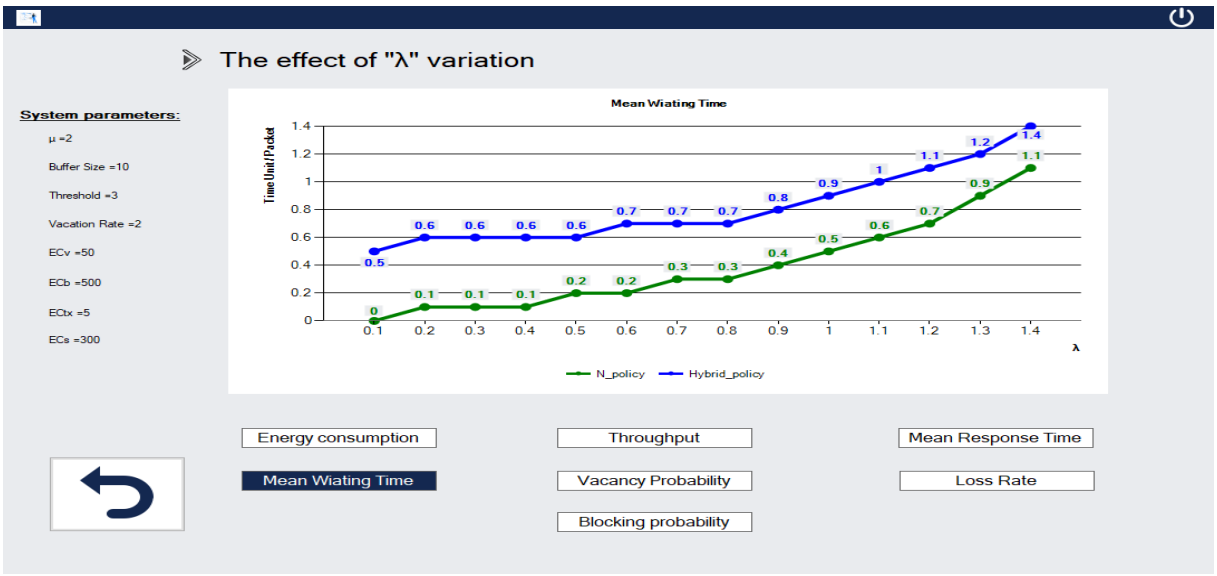

Figure 5.3 La fenêtre de résultats

5.3 Le générateur des variables aléatoires

L'une des étapes clés du développement d'une simulation est d'avoir une routine pour générer des valeurs aléatoires pour les variables, avec une distribution aléatoire spécifiée. Le procédé le plus courant de simulation d'une variable aléatoire de loi donnée s'appuie sur la notion de quantile. Cette fonction s'applique de manière tout-à-fait générale aux variables aléatoires à valeurs réelles, elle permet de définir une fonction de répartition inverse (ou fonction quantile).

La transformation inverse est une méthode basée sur l'observation qu'étant donné toute variable aléatoire X avec un FDC F(x), la variable U = F (x) est uniformément distribuée

entre 0 et 1. Par conséquent, X peut être obtenu en générant des nombres aléatoires uniformes U (0,1) et en calculant la fonction inverse, $X = F^{-1}(U)$. [47]

Notre modèle basé sur une file d'attente M/M/1, les temps inter-arrivées et les temps de service sont distribués de manière exponentielle. Les principales caractéristiques de la distribution sont Implémentés avec langage C # dans la figure 5.4. C'est fait en deux étapes :

- Tout d'abord, un nombre aléatoire réparti uniformément entre 0 et 1 est généré.
- Ensuite, le nombre est transformé pour produire une valeur aléatoire qui satisfait la distribution exponentielle par la transformation inverse de la fonction de répartition.

Pour générer une variable exponentielle X en utilisant une variable uniforme U pour un Moyenne lambda $\boldsymbol{\lambda}$,est procédez comme l'équation suivantes :

- $FDP : f(x) = \lambda e^{-\lambda x}$
- $FDC : F(x) = 1 - e^{-\lambda x} = U \quad ou \quad x = -\frac{1}{\lambda}\ln(1-U)$

Puisque U est uniformément réparti entre 0 et 1, (1 – U) est également uniformément distribué entre 0 et 1, X suit la loi exponentielle. Par conséquent, l'algorithme de génération peut être simplifié pour :

- $x = -\frac{1}{\lambda}\ln(U)$

```
1   using ...
7
8   public float Random_mean_generator(float λ)
9   {   //1er étape
10      RNGCryptoServiceProvider Rand = new RNGCryptoServiceProvider();
11
12      uint scale = uint.MaxValue;
13
14      byte[] four_bytes = new byte[4];
15      Rand.GetBytes(four_bytes);
16      scale = BitConverter.ToUInt32(four_bytes, 0);
17
18      var U = (scale / (double)uint.MaxValue);
19      //2eme étape
20      var X= (float)Math.Log(U) / (-1 * λ);
21      return X;
22  }
```

Figure 5.4 Implémentation d'un générateur de variables aléatoires

5.4 Résultats numériques

Nous allons présenter les résultats numériques pour montrer l'efficacité du modèle proposé. Les mesures de performance que nous avons considérées sont la consommation d'énergie moyenne et le temps d'attente moyen dans le Buffer et la saturation du buffer. La validité des modèles proposés peut être démontrée en comparant les résultats obtenus pour différents scénarios en faisant varier le taux d'arrivée moyen λ, le taux de service **μ** et le seuil de file d'attente **N**. Les paramètres du système utilisés dans l'étude expérimental sont répertoriés dans le tableau 5.1.

Paramétre	**Valeur**
Capacité du Buffer (*K*)	10
Seuil de file d'attente (*N*)	plage de 1 à 9
Taux moyen d'arrivée (λ)	plage de 0,5 à 2,5
Taux de service moyen (μ)	plage de 2 à 3
Taux de vacances moyen (ζ)	0,5
EC_i	50
EC_b	500
EC_h	5
EC_s	300

Tableau 5-1 Paramètres du système

5.4.1 Effet de la variation de taux d'arrivée des paquets λ

Dans ce qui suit, nous avons montré l'impact de la variation de taux d'arrivé λ sur la consommation moyenne d'énergie et le délai d'attente et la probabilité de blocage.

- **La consommation moyenne d'énergie**

La figure 5.5 apparaissant ci-dessous représente la consommation d'énergie moyenne pour les deux modèles ayant la politique N et la politique hybride pour différents taux d'arrivée moyens.

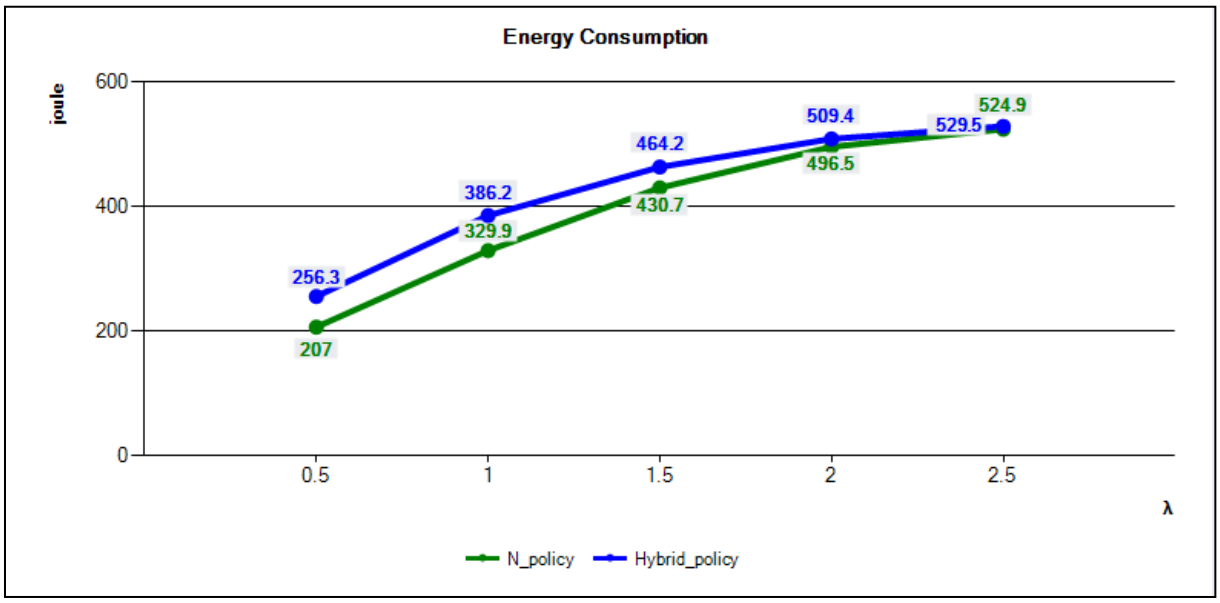

Figure 5.5 La consommation moyenne d'énergie en fonction de taux d'arrivée

Analyse : Il est clair que la consommation d'énergie moyenne pour le modèle avec N-vacance est inférieure à celle du modèle avec vacance-Hybride lorsque le taux d'arrivée est faible, mais elle est la même lorsque qu'il est plus élevé. De plus, son augmentation implique une augmentation de l'énergie consommé car plus le taux d'arrivée augmente plus le nœud consomme ses ressources en énergie pour traiter ses paquets arrivants.

- **Le délai d'attente dans le buffer**

La figure 5.6 ci-dessous représente l'influence de la variation du taux d'arrivée **λ** sur la durée moyenne d'attente dans le buffer pour les deux politiques.

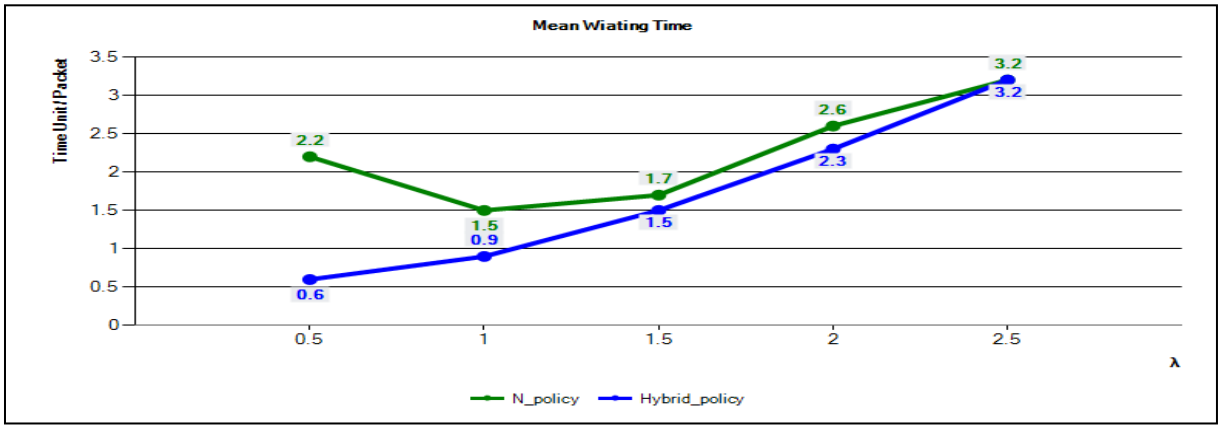

Figure 5.6 Le délai d'attente dans le buffer en fonction de taux d'arrivé

Analyse : la figure 5.6 montre que l'augmentation du taux d'arrivée **λ** augment le délai d'attente des paquets dans le buffer pour les deux politiques, d'autre part, nous notons que la politique hybride a une influence significative sur la diminution du temps d'attente moyen dans le Buffer pour des valeurs de taux d'arrivée moyen faibles. Cependant, lorsque le taux d'arrivée moyen est plus élevé, la diminution n'est pas très significative.

• La probabilité de blocage :

La figure 5.7 ci-dessous représente l'influence de la variation du taux d'arrivée **λ** sur la probabilité de blocage.

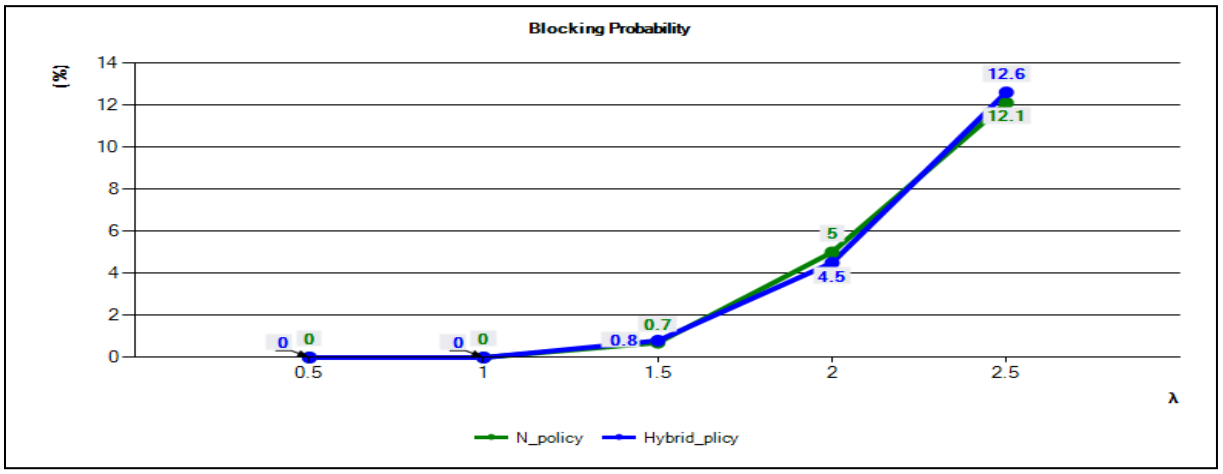

Figure 5.7 La probabilité de blocage en fonction de taux d'arrivée

Analyse : on déduit que l'augmentation de taux d'arrivée **λ** augmente la probabilité de blocage des paquets, car la capacité K est atteint rapidement.

5.4.2 Effet de la variation de taux de service μ

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

• La consommation d'énergie

Le courbe représenté par la figure 5.8 ci-dessous donnent les résultats de la consommation moyenne d'énergie en fonction du taux de service avec les deux politiques.

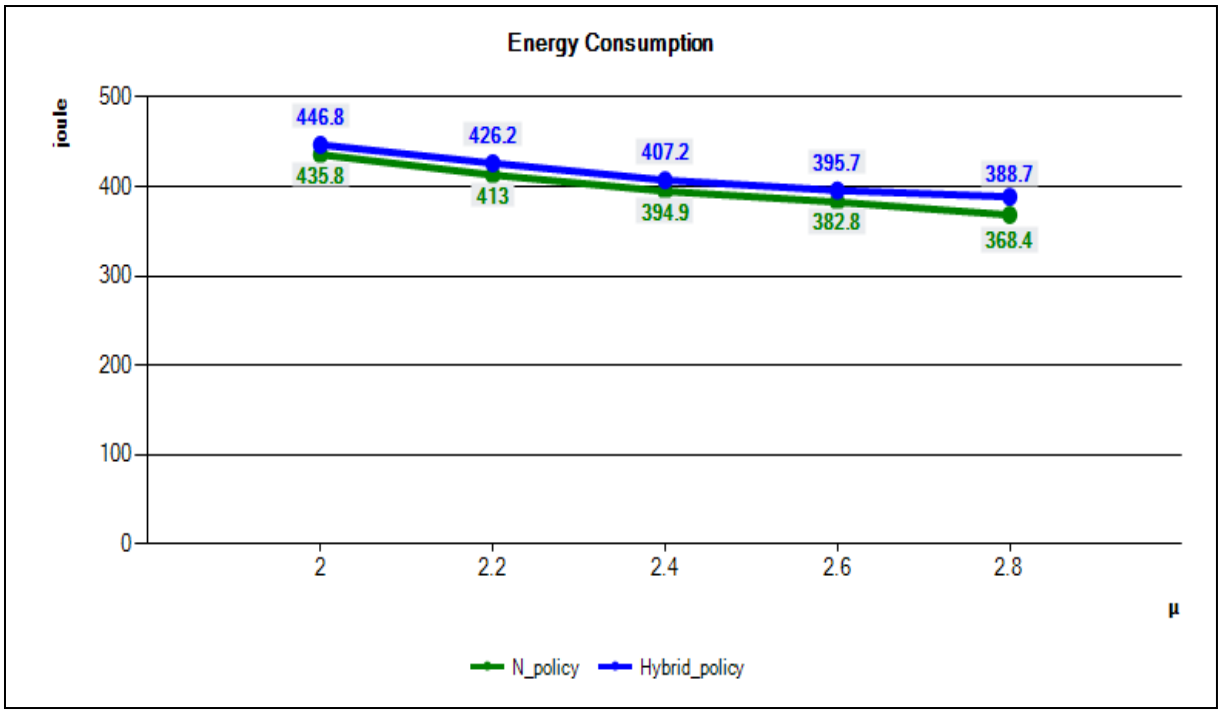

Figure 5.8 La consommation d'énergie en fonction de taux de service

Analyse : nous remarquons que la consommation d'énergie est similaire pour les deux politiques. L'augmentation du taux de service par le nœud capteur influe directement sur la consommation d'énergie, car plus le capteur fournit des services plus l'énergie est consommée se baisse.

- **Le délai d'attente dans le buffer**

Le courbe représenté par les figures 5.9 ci-dessous donnent les résultats de la durée moyenne d'attente en fonction du taux de service pour les deux politiques.

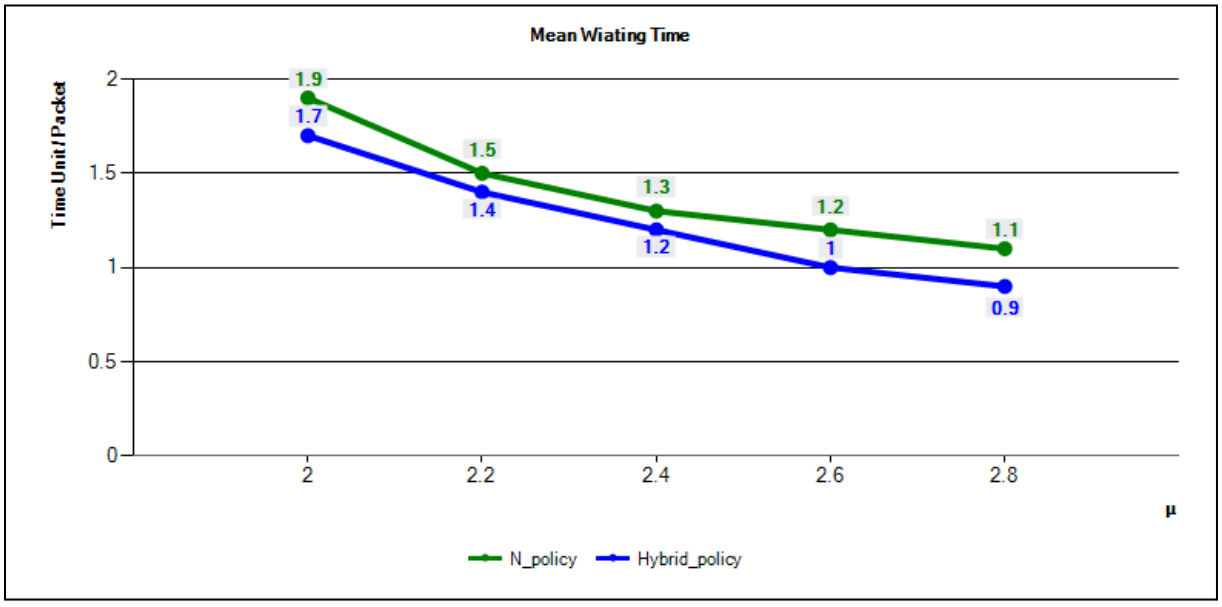

Figure 5.9 Le délai d'attente en fonction de taux de service

Analyse : la figure 5.9 illustre que l'augmentation de taux de service **µ** diminue le délai d'attente des paquets dans le buffer dans les deux politiques, ainsi qu'on remarque que la politique hybride a une influence sur la diminution du temps d'attente moyen dans le buffer mieux que la politique N-vacance.

• La probabilité de blocage

Le courbe représenté par les figures 5.10 ci-dessous donnent les résultats concernant la probabilité de blocage en fonction du taux de service pour les deux politiques.

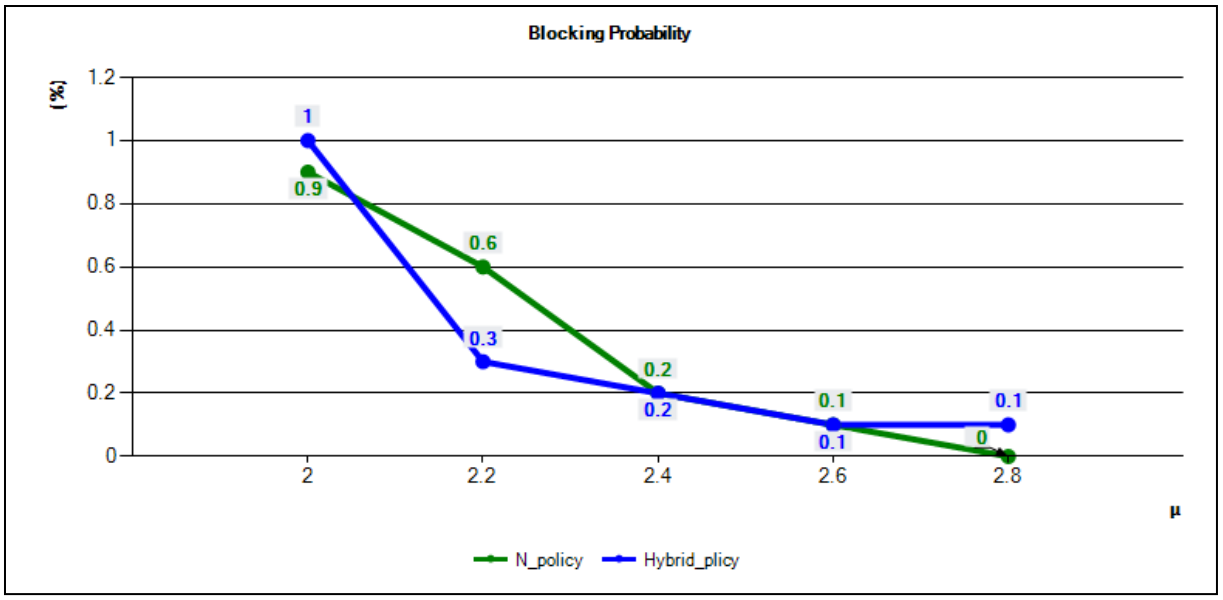

Figure 5.10 La probabilité de blocage en fonction de taux de service

Analyse : la figure 5.10 illustre que l'augmentation de taux service **µ** influence sur la probabilité de blocage dans les deux politiques. L'augmentation de taux de service implique la décroissance de la probabilité de blocage, pour raison de rapidité de service, alors le serveur prend plus de temps pour atteindre la capacité K du buffer.

5.4.3 Effet de la variation de seuil N

Nous avons évalué l'impact de variation de seuil N sur la consommation moyenne de l'énergie, la durée d'attente et la probabilité de blocage moyenne.

• La consommation d'énergie

Le courbe représentés par la figure 5.11 ci-dessous représente le résultat de l'énergie consommée en fonction de seuil N pour les deux politiques.

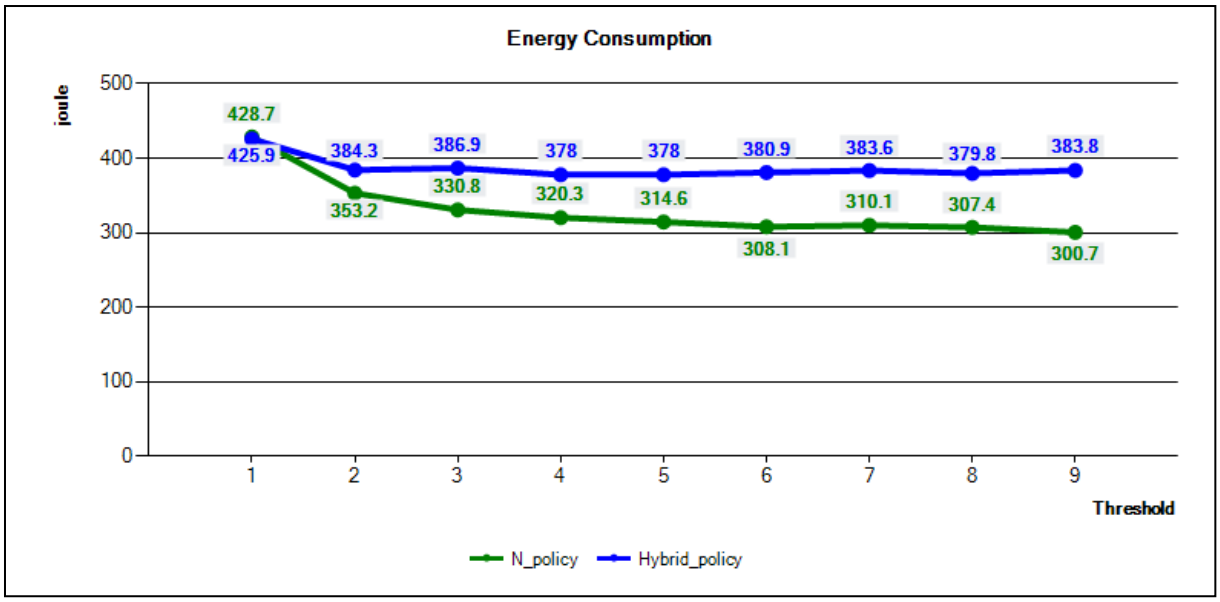

Figure 5.11 La consommation d'énergie en fonction de seuil N

Analyse : la figure 5.11 présente que la consommation d'énergie moyenne diminue avec l'augmentation du seuil de file d'attente N pour les deux politiques, jusqu'à atteindre une certaine valeur de seuil de file d'attente, car l'augmentation de seuil implique la prolongation de la durée de vacance du capteur, donc la consommation moyenne de l'énergie sera diminuée.

- **Le délai d'attente dans le buffer**

Le courbe représenté par la figure 5.12 ci-dessous représente le résultat de la durée moyenne d'attente dans le buffer en fonction de seuil N pour les deux politiques.

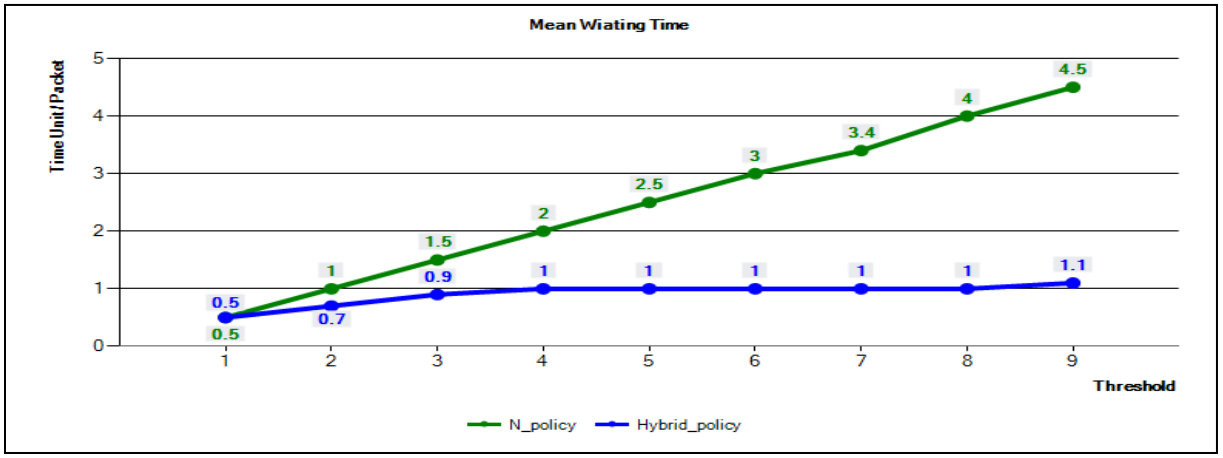

Figure 5.12 Le délai d'attente en fonction de seuil N.

Analyse : Il est clair que le temps moyen d'attente dans le buffer augmente avec le seuil N pour la politiques N-vacance, d'un autre coté le délai d'attente reste stable pour la politique hybride en augmentant le seuil N après une certaine montée. Cette stabilité est causée par les vacances aléatoires, ce qui réduit les temps d'attente des paquets.

• La probabilité de blocage :

La figure 5.13 ci-dessous représente l'influence de la variation de seuil N sur la probabilité de blocage pour les deux politiques.

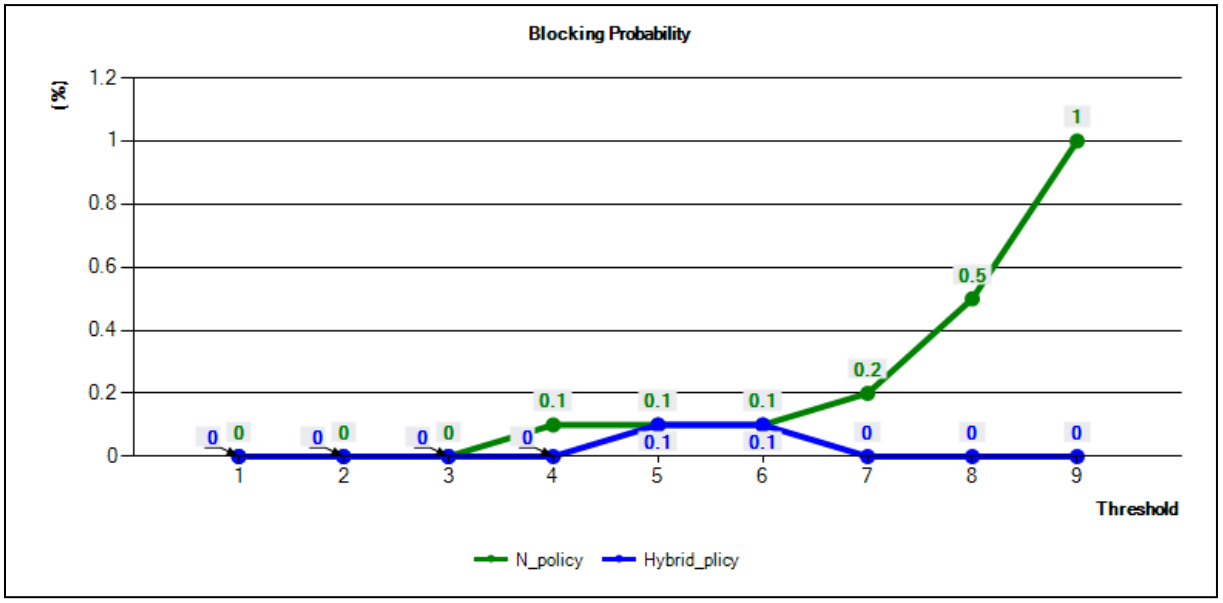

Figure 5.13 La probabilité de blocage en fonction de seuil N.

Analyse : nous remarquons que l'augmentation de seuil N augmente la probabilité de blocage des paquets dans la politique N-vacance, car la capacité du paramètre K est atteinte rapidement et les paquets servira lentement, puisque l'augmentation de seuil implique la prolongation de la durée de vacance du capteur. Au contraire, le courbe de la politique hybride est partiellement stable entre 0 et 0,1 car les vacances aléatoires de cette politique éliminent ce problème, donc le blocage est négligeable,

5.4.4 Résultat final

D'après ce nous avons vue comme études et comparaison des différentes résultats obtenus concernant les deux politique N-vacance et vacance hybride, nous pouvons déduire que la politique hybride peut mieux réduire le délai d'attente des paquets dans le buffer et d'un autres coté, elle a un impact très faible sur la consommation d'énergie par rapport à la politique N-vacance au niveau d'un nœud de capteur.

En revanche, nous pouvons également voir que le modèle avec la politique Hybride agit de la même manière que la politique N-vacance lorsque le taux d'arrivée moyen est élevé et maintient un temps d'attente acceptable dans la mémoire Buffer pour les paquets, pour un taux d'arrivée faible.

5.5 Conclusion

Dans ce chapitre, nous avons implémenté notre approche qui est basée sur deux politiques, afin de minimiser la consommation d'énergie et avoir un délai de latence efficace pour un nœud de capteur sans fil non rechargeable, et évaluer les performances de ces réseaux.

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

A la fin, nous avons procédé à présenter un résultat final qui montre l'influence de chaque politique de notre approche sur la consommation d'énergie et le délai d'attente.

CONCLUSION GÉNÉRALE

Les réseaux de capteurs sans fil ont attiré l'attention de la communauté des chercheurs ces dernières années, poussés par plusieurs défis théoriques et pratiques. Parmi ces défis, c'est la faiblesse qu'as subi les RCSF liés à leurs capacités de calcule et de stockage limité ainsi que leurs batteries irremplaçables et leurs déploiements dans un environnement hostile.

Dans les réseaux de capteurs sans fil, la consommation d'énergie est dans de nombreux cas la contrainte la plus importante puisqu'elle correspond directement à la durée de vie et au bon fonctionnement du réseau, de ce fait l'objectif de notre travail est de minimiser la consommation d'énergie dans les RCSF.

Après avoir introduire la notion de vacances au modèle que nous avons adopté, nous avons exploiter le comportement d'un nœud de capteur en utilisant deux méthodes de vacances chacune ayant ses caractéristiques de plus, l'alternance du capteur nous a permis d'avoir une économie d'énergie pour l'unité de transmission qui se présenté comme étant une source de consommation d'énergie.

La simulation est l'exécution d'un modèle, représenté par un programme informatique qui donne des informations sur le système étudié. Pour cela nous avons inclus une approche de simulation à événement discret qui se présente comme étant une méthode de résolution du problème d'énergie, grâce à ses éléments cette approche nous a permet de bien gérer les événements du système ainsi que l'étudier à des points de discrets.

Grâce à l'outil de simulation que nous avons développé nous avons eu la possibilité de comparer et tester les déférentes résultats ainsi que l'impact des modèles de vacances que nous avons proposé sur la performance du capteur, nous avons opté à faire une étude comparative entre eux pour déduire celle qu'a été efficace.de plus nous avons constaté que dans certains cas l'une de ces méthodes présenté meilleur résultats que l'autres et presque les mêmes résultats dans d'autres cas. Cet outil basé sur la simulation assure une description pour la modélisation et l'évaluation des performances des réseaux de capteur sans fils et permet d'avoir une étude comparative sur les méthodes de vacances proposées.

Enfin à titre de ce travail, l'approche par simulation pour l'analyse des modèles se diffère à l'approche analytique qui se base sur une méthode d'analyse du système purement théorique en utilisant des formules mathématiques en revanche, l'approche de simulation est adoptée pour valider les résultats obtenus de l'approche analytique et rendre le système plus efficace en terme consommation d'énergie.

BIBLIOGRAPHIE

[1] SADOUN, Balqies, "Applied system simulation: a review study," *Information Sciences,* vol. 124, pp. 173-192, 2000.

[2] JIANG, Fuu-Cheng, HUANG, Der-Chen, et WANG, Kuo-Hsiung, "Design approaches for optimizing power consumption of sensor node with N-policy M/G/1 queuing model," *In : Proceedings of the 4th International Conference on Queueing Theory and Network Applications,* pp. 1-8, 2009.

[3] SHIH, Eugene, CHO, Seong-Hwan, ICKES, Nathan, et al, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *In : Proceedings of the 7th annual international conference on Mobile computing and networking,* pp. 272-287, 2001.

[4] KIM, Kyung Tae, LYU, Chang Hoon, MOON, Sung Soo, et al, "Tree-based clustering (TBC) for energy efficient wireless sensor networks," *In : 2010 IEEE 24th international conference on advanced information networking and applications workshops.IEEE,* pp. 680-685, 2010.

[5] EBRAHIMI, Dariush et ASSI, Chadi, "Compressive data gathering using random projection for energy efficient wireless sensor networks," *Ad Hoc Networks,* vol. 16, pp. 105-119, 2014.

[6] KUMAR, Binay, VIJ, Ashu, et KUMAR, Pankaj, "Some Basic Concepts in Queuing Theory," *International Advanced Research Journal in Science, Engineering and Technology,* vol. 2, pp. 56-60, 2015.

[7] BOUTOUMI, Bachira et GHARBI, Nawel, " An energy saving and latency delay efficiency scheme for wireless sensor networks based on GSPNs. In : 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)," *IEEE, 2017. p. 0645-06.*

[8] JIANG, Fuu-Cheng, HUANG, Der-Chen, YANG, Chao-Tung, et al, " Design strategy for optimizing power consumption of sensor node with Min (N, T) policy M/G/1 queuing models," *International Journal of Communication Systems, 2012, vol. 25, no 5, p. 652-671..*

[9] BOUTOUMI, Bachira et GHARBI, Nawel, " Two Thresholds Working Vacation Policy for Improving Energy Consumption and Latency in WSNs," *In : International Conference on Queueing Theory and Network Applications.Springer, Cham,* pp. 168-181, 2018.

[10] HASSON, Saad Talib, Simulation approach to model queuing Problems.

[11] RAWAT, Priyanka, SINGH, Kamal Deep, CHAOUCHI, Hakima, et al, " Wireless sensor networks: a survey on recent developments and potential synergies.," *The Journal of supercomputing, 2014, vol. 68, no 1, p. 1-48..*

[12] AKYILDIZ, Ian F., SU, Weilian, SANKARASUBRAMANIAM, Yogesh, et al, "A survey on sensor networks.," *IEEE Communications magazine, 2002, vol. 40, no 8, p. 102-114..*

[13] Xu, Y., Heidemann, J., & Estrin, D, "Geography-informed energy conservation for ad hoc routing," *In Proceedings of the 7th aniual international conference on Mobile Computing and networking (MobiCom'01),* pp. 70-84, 2001.

[14] ANASTASI, Giuseppe, CONTI, Marco, DI FRANCESCO, Mario, et al, "Energy conservation in wireless sensor networks: A survey.," *Ad hoc networks, 2009, vol. 7, no 3, p. 537-568..*

[15] M. Lehsaini, "Diffusion et couverture basées sur le clustering dans les réseaux de," PhD, University of Franche-Comté, 2009.

[16] SINGH, Shio Kumar, SINGH, M. P., SINGH, Dharmendra K.,et al , " Routing protocols in wireless sensor networks–A survey.," *International Journal of Computer Science & Engineering Survey (IJCSES), 2010, vol. 1, no 2, p. 63-83..*

[17] CHEN, Yunxia et ZHAO, Qing., " On the lifetime of wireless sensor networks.," *IEEE Communications letters, 2005, vol. 9, no 11, p. 976-978..*

[18] Raghunathan, V., Shrugers, C., Park, S., & Srivastava, M, "Energy-awah wireless microsensor networks," *In IEEE Signal Processing Magazine,* vol. 19, pp. 40-50, 2002.

[19] ALTHOBAITI, Ahlam Saud et ABDULLAH, Manal, "Medium access control protocols for wireless sensor networks classifications and cross-layering," *. Procedia Computer Science, 2015, vol. 65, p. 4-16.*

[20] REZAEI, Zahra et MOBININEJAD, Shima, " Energy saving in wireless sensor networks," *International Journal of Computer Science and Engineering Survey,* vol. 3, p. 23, 2012.

[21] KHRIJI, Sabrine, EL HOUSSAINI, Dhouha, KAMMOUN, Ines, et al, "Energy-efficient techniques in wireless sensor networks," *In : Energy Harvesting for Wireless Sensor Networks: Technologies, Components and System Design,* 2018.

[22] AMBEKAR, Dhanashri V., BHOI, Amol D., et KHARADKAR, R. D, "A survey on sensors lifetime enhancement techniques in wireless sensor networks," *International Journal of Computer Applications,* vol. 107, 2014.

[23] Rawat, P., Singh, K. D., Chaouchi, H., & Bonnin, J. M. , "(2013). Wireless sensor networks: a survey on recent developments and potential synergies.," *The Journal of Supercomputing, 68(1), 1–48. doi:10.1007/s11227-013-1021-9..*

[24] CASTELLUCCIA, Claude et FRANCILLON, Aurélien. , "Protéger les réseaux de capteurs sans fil.," *SSTIC08, 2008..*

[25] IBE, Oliver, "Markov processes for stochastic modeling," Newnes, 2013.

[26] SERICOLA, Bruno, " Markov chains: theory and applications," *John Wiley & Sons, 2013.*

[27] FOTSO, Donatien CHEDOM et FOTSO, Laure Pauline, " Étude et simulation du phénomène d'attente dans un système bancaire".

[28] HSU, Hwei P, " Theory and problems of probability, random variables, and random processes," *New York : McGraw-hill, 1996.*

[29]] SHORTLE, John F., THOMPSON, James M., GROSS, Donald, et al, " Fundamentals of queueing theory," *John Wiley & Sons, 2018.*

[30] VERGNE, Anaıs et COMTE, Céline. , "Files d'attente.," *2018..*

[31] TIAN, Naishuo et ZHANG, Zhe George, "Vacation queueing models: theory and applications," *Springer Science & Business Media,* 2006.

[32] KHALAF, Rehab, "On some queueing systems with server vacations, extended vacations, breakdowns, delayed repairs and stand-bys.," *2012. Thèse de doctorat. Brunel University, School of Information Systems, Computing and Mathematics..*

[33] BRÉMAUD, Pierre, " Initiation aux Probabilités: et aux chaînes de Markov.," *Springer Science & Business Media, 2009..*

[34] MODICA, Giuseppe et POGGIOLINI, Laura., " A first course in probability and Markov Chains," *Wiley, 2013.*

[35] ALFA, Attahiru Sule, "Queueing theory for telecommunications: discrete time modelling of a single node system," *Springer Science & Business Media,* 2010.

[36] CHOI, Byoung Kyu et KANG, Donghun, "Modeling and simulation of discrete event systems," *John Wiley & Sons,* 2013.

[37] CHANCHAICHUJIT, Janya et SAAVEDRA-ROSAS, José F, "Discrete Event Simulation Concepts," *In : Using Simulation Tools to Model Renewable Resources. Palgrave Macmillan, Cham,* pp. 41-63, 2018.

[38] RAYCHAUDHURI, Samik, "Introduction to monte carlo simulation," *In : 2008 Winter simulation conference. IEEE,* pp. 91-100, 2008.

[39] ĎUTKOVÁ, Silvia, ACHIMSKÝ, Karol, et HOŠTÁKOVÁ, Dominika, "Simulation of queuing system of post office," *Transportation Research Procedia,* pp. 1037-1044, 2019.

[40] FISHMAN, George S, "Discrete-event simulation: modeling, programming, and analysis," *Springer Science & Business Media,* 2013.

[41] TYSZER, Jerzy, "Object-oriented computer simulation of discrete-event systems," Springer Science & Business Media, 2012.

[42] LIN, James T. et LEE, Chia-Chu, " A three-phase discrete event simulation with EPNSim graphs," *Simulation,* vol. 60, pp. 382-392, 1993.

[43] MULYODIPUTRO, Muhammad Dermawan et SUBANAR, Subanar, "Simulation of queue with cyclic service in signalized intersection system," *International Journal of Advances in Intelligent Informatics,* vol. 1, pp. 30-40, 2015.

[44] SADEGHI, N., FAYEK, A. Robinson, et SERESHT, N. Gerami, "Queue performance measures in construction simulation models containing subjective uncertainty," *Automation in Construction,* vol. 60, pp. 1-11, 2015.

[45] EHSANIFAR, Mohammad, HAMTA, Nima, et HEMESY, Mahshid, "A Simulation Approach to Evaluate Performance Indices of Fuzzy Exponential Queuing System (An M/M/C Model in a

Banking Case Study)," *Journal of Industrial Engineering and Management Studies,* vol. 4, pp. 35-51, 2017.

[46] Stewart, William J, "Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling," *Princeton university press,* 2009.

[47] Jain, Raj, The Art Of Computer Systems Performance Analysis: Techniques For Experimental Measurement, Simulation, And Modeling, john wiley & sons, 2008.