

**Université Saad Dahlab Blida**

N° D'ordre :14/103



Faculté des sciences

**Département d'informatique**

Mémoire Présenté par :

Mr.KHODJA Mohamed Riad

Mr.AFIR Nabil

**En vue d'obtenir le diplôme de master2**

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Génie des Systèmes informatiques

**Sujet :**

*Application des modèles avec vacance exhaustif pour la conservation  
d'énergie des réseaux de capteurs*

**Soutenu le :**

Mme. S.Aroussi  
Mme. H.Ghribi  
Mme. B.Boutomi

Président  
Examineur  
Promotrice

**Promotion**

2015 / 2016

# Résumé

Le domaine de la télécommunication repose essentiellement sur les technologies sans fil. Celle-ci sont en développement constatant .grâce au progrès réalisé, un nouveau type de réseau est apparu. Ce sont les réseaux de capteur sans fil. Leur particularité est qu'ils n'ont pas de structure fixe. On peut les déployer dans des environnements hostiles facilement et de manière très rapide. Ces réseaux sont constituer de nœud capteur de petite taille qui peuvent communiquer entre eux. Leur objectif principal est de surveiller une zone en récoltant divers mesure physique et de les retransmettre sous forme de donnée numérique. Cependant, ces réseaux posent un certain nombre de défis scientifiques intéressant pour la communauté des chercheurs, dont certains d'entre eux sont liés directement aux contraintes physiques des nœuds capteurs. Ce dernier, à cause de la nécessité de miniaturisation de leur composant. Ils disposent de ressource très restreinte en matière de mémoire, énergie, calcul et de communication. Dans notre mémoire nous nous somme donne comme objectif de minimise la consommation en énergie du nœud capteur.

Pour cela, nous avons modélisé le comportement d'un nœud capteur avec les réseaux de Petri stochastique généralise. Et nous introduisons la notion de vacance exhaustive dans son comportement. Enfin nous donnerons les formules des différents indices de performance de nos modèles.

**Mots clés :** Réseaux de capteurs sans fil, Vacance exhaustif, Modélisation avec RdPSG, Conservation d'énergie, Evaluation des performances.

# Abstract

The field of telecommunication is mainly based on wireless technologies. This finding is in development. Thanks to that progress, a new type of network appeared. These are wireless sensor networks. Their peculiarity is that they do not have a fixed structure. We can deploy in harsh environments easily and very quickly. These networks form of small sensor node that can communicate with each other. Their main objective is to monitor an area by collecting various physical measurement and retransmit them as digital provides. However, these networks present a number of interesting scientific challenges for the research community, some of them are directly related to the physical constraints of sensor nodes. The latter, due to the need for miniaturization of their components. They have very limited resources in terms of memory, energy, computing and communication. In our brief we sum's objective is to minimize the energy consumption of the sensor node.

For this, we modelled the behaviour of a sensor node with general stochastic Petri nets. And we will introduce the concept of comprehensive vacancy in his behaviour. Finally we give the formulas of the various performance indicators of our models.

**Keywords:** Networks wireless sensors, Exhaustive vacancy, Modelling with GSPN, Energy Storage, Performance evaluation.

# Remerciement

*Au terme de ce travail, nous tenons à exprimer notre gratitude et remerciement pour toutes les personnes qui ont contribué à sa réalisation.*

*Nous tenons à remercier Mesdames et Messieurs les jurys pour le temps qu'ils nous accordent.*

*Nous remercions aussi notre promotrice Madame le professeur Boutoumi Bachira pour toute l'aide qu'elle nous a apporté. Elle sait toujours montrer disponible et à l'écoute tout au long de la réalisation de ce mémoire.*

*Nous remercions aussi tous les membres de nos familles pour leur aides et soutient moral.*

*Aussi, nous remercions nos proches et amis qui ont toujours était présent a nos coté pour nous aider.*

## LISTE DES ABRÉVIATIONS

RCSF	réseau de capteur sans fil
RdP	réseau de petri
GMA	graphe de marquage accessible
RdPSG	réseau de petri stochastique généralisé
CM	Chaine de Markov
CMR	chaîne de Markov réduite
QOS	quality of service

# Table des matières

<b>Table des figures</b>	<b>4</b>
<b>Liste des tableaux</b>	<b>5</b>
<b>Introduction générale</b>	<b>6</b>
<b>1 Les réseaux de capteurs sans fil</b>	<b>8</b>
1.1 Définition d'un capteur sans fil . . . . .	9
1.2 Anatomie d'un capteur sans fil . . . . .	9
1.3 Les réseaux de capteur sans fil . . . . .	10
1.4 Domaine d'application . . . . .	11
1.5 Architecture d'un réseau de capteur sans fil . . . . .	11
1.6 Modèle en couche . . . . .	12
1.7 Routage dans les RCSF . . . . .	13
1.7.1 Routage à plat . . . . .	13
1.7.2 Routage hiérarchique . . . . .	13
1.7.3 Routage géographique . . . . .	14
1.8 Nature du Trafic . . . . .	14
1.8.1 Le Trafic périodique . . . . .	14
1.8.2 Le Trafic apériodique . . . . .	14
1.9 Topologie d'un réseau de capteur sans fil . . . . .	14
1.9.1 Topologie en étoile . . . . .	14
1.9.2 Topologie en maille . . . . .	15
1.9.3 Topologie hybride . . . . .	16
1.10 Contraintes de conception des RCSF . . . . .	16
1.10.1 Contrainte conceptuelles . . . . .	16
1.10.2 Contraintes matérielles . . . . .	18
1.11 Consommation énergétique d'un nœud capteur . . . . .	19
1.11.1 Formes de dissipation d'énergie . . . . .	19
1.11.2 Sources de surconsommation d'énergie . . . . .	19
1.11.3 Mécanismes de conservation d'énergie . . . . .	20
1.12 Conclusion . . . . .	21

<b>2 Les réseaux de Petri</b>	<b>22</b>
2.1 Introduction . . . . .	23
2.2 Concepts de base et définition . . . . .	23
2.2.1 Définition formelle . . . . .	24
2.2.2 Représentation graphique . . . . .	24
2.2.3 Représentation matricielle . . . . .	24
2.3 Évolution des réseaux de Petri . . . . .	25
2.3.1 Règles de sensibilisation et de franchissement . . . . .	25
2.3.2 Ensemble d'accessibilité et graphe des marquages accessibles . . . . .	26
2.4 Domaine d'applications des réseaux de Petri . . . . .	26
2.5 Situation possible entre transition . . . . .	27
2.6 Propriétés des réseaux de Petri . . . . .	28
2.6.1 Bornitude . . . . .	28
2.6.2 Absence de blocage . . . . .	28
2.6.3 Vivacité . . . . .	28
2.6.4 Réinitialiabilité et état d'accueil . . . . .	28
2.6.5 Persistance . . . . .	29
2.7 Les réseaux de Petri à arcs inhibiteur . . . . .	29
2.7.1 Définition [1] . . . . .	29
2.8 Les réseaux de Petri stochastiques . . . . .	29
2.8.1 Evolution des réseaux de Petri stochastiques . . . . .	30
2.8.2 Analyse d'un réseau de Petri stochastique . . . . .	30
2.8.3 Principaux paramètres de performance . . . . .	31
2.9 Les réseaux de Petri stochastiques généralisé . . . . .	32
2.9.1 Définition informelle . . . . .	32
2.9.2 Définition formelle . . . . .	32
2.9.3 Processus stochastique associé à un RdPSG . . . . .	33
2.9.4 Les règles de franchissement dans les RdPSG . . . . .	33
2.9.5 Analyse des performances d'un RdPSG . . . . .	34
2.10 Conclusion . . . . .	36
<b>3 Les Politiques de vacances et modélisation des RCSF</b>	<b>37</b>
3.1 Introduction . . . . .	38
3.2 Modèle général de file d'attente avec vacance . . . . .	38
3.2.1 Description du modèle général de file d'attente classique . . . . .	38
3.2.2 Notion de KENDALL . . . . .	39
3.3 Description du modèle général de file d'attente avec vacance . . . . .	40
3.3.1 Les politiques de vacance . . . . .	40
3.4 Vacance d'un capteur sans fil . . . . .	41
3.5 Travaux connexes . . . . .	42
3.6 Modélisation des RCSF a l'aide des RdPSG : . . . . .	44
3.6.1 Modèle générique (M0) . . . . .	44
3.6.2 Modèle N-vacance (M1) . . . . .	45
3.6.3 Modèle avec vacance multiple(M2) . . . . .	46
3.6.4 Modèle N-vacance multiple (M3) . . . . .	47

## TABLE DES MATIÈRES

3

3.6.5	Modèle avec vacance unique (M4)	48
3.6.6	Modèle N-vacance unique (M5)	49
3.6.7	Modèle avec vacance différencié (M6)	50
3.6.8	Modèle avec N-vacance et vacance différencié (M7)	51
3.7	Les indices de performances	52
3.7.1	Modèle générique	52
3.7.2	Modèle avec vacance	53
3.7.3	Modèles à N-vacance	54
3.7.4	Modèle à N-vacance et service exhaustive	54
3.7.5	Modèle avec vacance différencie	54
3.7.6	Modèle avec vacance différencie et N-vacance	54
3.8	Conclusion	55
<b>4</b>	<b>Test et Expérimentation</b>	<b>56</b>
4.1	Introduction	57
4.2	Description de GreatSPN	57
4.3	Le calcul du gain en énergie	58
4.4	L'étude expérimentale	58
4.5	Conclusion	69
	<b>Bibliographie</b>	<b>72</b>

# Table des figures

1.1	Exemple de capteur . . . . .	9
1.2	Composants d'un noeud capteur . . . . .	10
1.3	Architecture d'un réseau de capteurs sans fil . . . . .	12
1.4	Modèle en couches d'un réseau de capteurs sans fil . . . . .	13
1.5	Topologie en étoile . . . . .	15
1.6	Topologie maillée . . . . .	15
1.7	Topologie hybride étoile - maillée . . . . .	16
2.1	Représentation graphique d'un réseau de Petri . . . . .	24
2.2	Exemple de transitions . . . . .	26
2.3	Situations de conflit et de confusion . . . . .	27
3.1	Système de files d'attente a un serveur . . . . .	38
3.2	File d'attente avec vacance . . . . .	40
3.3	Modèle générique (M0) . . . . .	45
3.4	Modèle N-vacance(M1) . . . . .	46
3.5	Modèle avec vacance multiple(M2) . . . . .	47
3.6	Modèle N-vacance multiple et service exhaustive(M3) . . . . .	48
3.7	Modèle avec vacance unique et service exhaustive(M4) . . . . .	49
3.8	Modèle avec service exhaustive à N-vacance unique (M5) . . . . .	50
3.9	Modèle avec service exhaustive et vacance différencié(M6) . . . . .	51
3.10	Modèle avec N-vacance et vacance différencié(M7) . . . . .	52
4.1	Nombre de message dans le buffer en fonction du taux de vacance . . . . .	61
4.2	Temps de séjour du message dans le buffer en fonction du taux de vacance . . . . .	62
4.3	Fréquence de transmission des messages en fonction du taux de vacance . . . . .	63
4.4	Gain d'énergie en fonction du taux de vacance . . . . .	64
4.5	Nombre de message dans le buffer en fonction de du seuil de la N-vacance . . . . .	65
4.6	Temps de séjour du message dans le buffer en fonction du seuil de la N-vacance . . . . .	66
4.7	Fréquence de transmission des messages en fonction du seuil de la N-vacance . . . . .	67
4.8	Gain d'énergie en fonction du seuil de la N-vacance . . . . .	68

# Liste des tableaux

4.1	Paramètre de réseau . . . . .	59
4.2	Paramètres de consommation du capteur . . . . .	59

# Introduction générale

Le développement de l'informatique et de l'électronique dans le domaine des micros technologies et des technologies sans fil ont permis la créations de dispositifs et d'équipements miniature appelés micros-capteurs ou nœuds capteur. Leur fonction principale est de mesurer des grandeurs physiques, telle que l'humidité, la chaleur, la pression etc., et de les transformer en informations numériques exploitables par les calculateurs et ordinateurs. Ces capteurs sont déployés sous forme d'un réseau appelé réseaux de capteurs sans fils ou WSN (wireless sensor network). Il communique entre eux via des ondes radio afin de transmettre une information captée ou simplement de la relayer.

Ces réseaux de capteur sans fils sont principalement utilisés pour la surveillance d'une zone quelconque. Des micros capteurs sont déployés dans la zone cible afin de collecter des informations. L'utilisation de réseaux de capteurs sans fil présente de nombre avantage comme le cout réduit, le déploiement simple et rapide et l'absence d'infrastructures. Tout cela augmente leur importance et leur utilité dans la vie aujourd'hui. Grace à ces avantages, les réseaux de capteurs sans fil sont utilisés dans des domaines diverses tel que la défense, la sécurité, la santé, l'agriculture, etc. Cependant ces réseaux subissent plusieurs contraintes comme la tolérance aux pannes, le passage à l'échelle, la gestion de la topologie du réseau. S'ajoute aussi à cela l'obligation d'utiliser des composant miniatures au maximum afin de minimiser leur coups de production et de faciliter leur déploiement. Mais cela implique que les capteurs sans fils sont généralement dotés de ressources très limitées en terme de capacité de calcul, d'espace de stockage de données, de débit de transmission ainsi que l'énergie embarquée qui a une importance primordial car elle influe directement sur la durée de vie du réseau.

Les contraintes citées précédemment démontrent qu'une conception détaillée et précise des réseaux de capteur sans fil est impérative, il est même indispensable de les évaluer et de les analyser avant tout déploiement. Ceci permettrait d'évaluer les performances et la consommation en énergie du réseau. Généralement, l'analyse des réseaux de capteur sans fil se fait par simulation grâce a des logiciels tels que NS-2[2], OMNet++[3], OPNET [4] etc. Cependant cette approche a pour inconvénient le temps d'exécution énorme nécessaire pour avoir des résultats proches de la réalité. D'autre part, l'utilisation de simulateur différent pourrait générer des résultats divergents. C'est pour cela qu'une alternative récente est utilisée pour l'évaluation des réseaux de capteur sans fil, celle-ci est l'utilisation de méthodes formelles basées sur des abstractions mathématiques du réseau, ce qui a l'avantage de fournir des résultats exacts.

Parmi les modèles formels utilisés, nous trouvons les réseaux de Petri. Cet outil graphique puissant permet l'expression des caractéristiques des systèmes parallèles, telles que la synchronisation, la concurrence, l'exclusion mutuelle, le conflit, etc. Par ailleurs, l'introduction de la notion de temps aléatoire dans les réseaux de Petri mène aux réseaux de Petri stochastiques généralisés qui ont de plus, l'avantage de faire une évaluation des performances des systèmes, en déterminant leurs propriétés quantitatives en plus des propriétés qualitatives.

Tout au long de ce mémoire, nous nous intéressons à la conservation d'énergie des capteurs sans fil dans le but de réduire leur consommation en énergie. Pour réaliser notre objectif, nous avons introduit la notion de vacance qui consiste à mettre un capteur en état de veille. Nous modéliserons ce comportement avec différentes politiques de vacance grâce aux réseaux de Petri stochastique généralisés.

Notre document est organisé comme suit : le premier chapitre est dédié à la présentation des réseaux de capteur sans fil. Nous présenterons leurs caractéristiques, leurs contraintes ainsi que quelques exemples de leurs domaines d'application. Nous aborderons dans le chapitre deux les réseaux de Petri et réseaux de Petri stochastique généralisés. Le troisième chapitre sera consacré aux différentes politiques de vacance, leurs définitions ainsi que notre modélisation et notre conception des différentes politiques. Finalement, nous présenterons dans le quatrième chapitre les différents résultats obtenus après avoir effectué des tests sur nos modèles. La conclusion de notre mémoire sera un résumé sur les principaux points abordés et les résultats obtenus.

# Chapitre 1

Les réseaux de capteur sans fil

## Introduction

Le domaine de l'informatique et de l'électronique ont connus d'énorme progrès ces dernières années, cela a permis l'introduction de la technologie des micros capteurs ou plus communément appelés nœuds de capteur dans différents dispositifs et équipements de surveillance et de mesure. Ces dispositif collecte des informations et les numérise .

Les réseaux de capteurs sans fil (RCSF) sont apparus après l'année 1990 grâce au grand progrès qu'ont connus les technologies sans fils.les (RCSF) sont devenues très importants en raison de leur multiples applications et avantages, ainsi que leur utilisations variées dans le domaine industriels et environnemental. Un réseau de capteurs sans fil est composé d'un très grand nombre de capteurs de petites tailles dispersés sur une zone géographique appelée "Zone de captage". Ces capteurs communiquent entre eux d'une manière sans fil dans le but d'accomplir des taches communes en détectant et communiquant les différents évènements se produisant dans leur entourage. Notre premier chapitre présente des généralités sur les réseaux de capteurs sans fil. Leur définition, ainsi que l'anatomie d'un capteur, ensuite une description des réseaux de capteur sans fil, suivi des différentes architectures des RCSF, les contraintes de conception des RCSF, et pour finir leur différents domaines d'application et la consommation énergétique d'un nœud capteur.

### 1.1 Définition d'un capteur sans fil

Un capteur est un dispositif qui sert à capter une grandeur physique observée dans son entourage et ainsi de la transformer en une grandeur numérique exploitable. Un capteur par exemple peut capter des variations de lumière ou de température et de retranscrire ses informations en données numériques. la Figure 1.1 représente un capteur sans fil modèle MPR400CB. Il est équipé d'une carte réseau qui lui permet de communiquer avec des capteurs du même type et peut ainsi échanger des informations, ceci donc créera un réseau de capteurs sans fil [5].

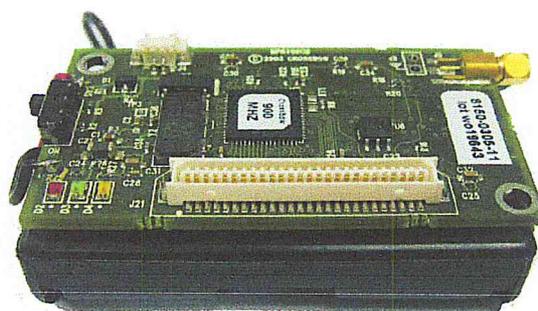


FIGURE 1.1 – Exemple de capteur

### 1.2 Anatomie d'un capteur sans fil

Un nœud capteur est composé principalement de quatre unités de base : une unité de captage (Sensing Unit), une unité de traitement (Processing Unit), une unité de transmission

(Transmission Unit) et une unité d'énergie (Power Unit) comme cela est décrit dans la Figure 1.2. D'autres unités peuvent être ajoutées en fonction de l'utilisation du capteur et du type de l'application, on peut citer un système de régénération d'énergie, un système de déplacement ou un système de localisation. Les caractéristiques des capteurs sont les contraintes en énergie, en capacité de stockage et en vitesse de calcul[6].

**Unité de captage :** cette unité permet la collecte d'informations et de mesurer des grandeurs physiques et de les convertir en une mesure numérique afin de la transmettre à l'unité de traitement [7].

**Unité de traitement :** composée d'un microprocesseur, d'une mémoire vive RAM, d'une ROM et d'une unité de stockage à capacité limitée, cette unité peut exécuter des logiciels ou système d'opération consacré spécialement pour les capteurs. Cette unité est aussi chargée de l'exécution des protocoles de communications qui permettent à un nœud capteur de communiquer et de transmettre les données qu'il a capté aux nœuds capteurs du réseau [8].

**Unité de transmission :** cette unité est responsable de toutes les émissions et les réceptions de données, via un support de communication sans fil commun entre les nœuds.

**Unité d'énergie :** elle est considérée comme l'un des composants les plus importants d'un nœud capteur. Elle alimente les trois unités précédentes. L'unité d'énergie peut être soutenue par une source d'énergie telle qu'un panneau solaire.

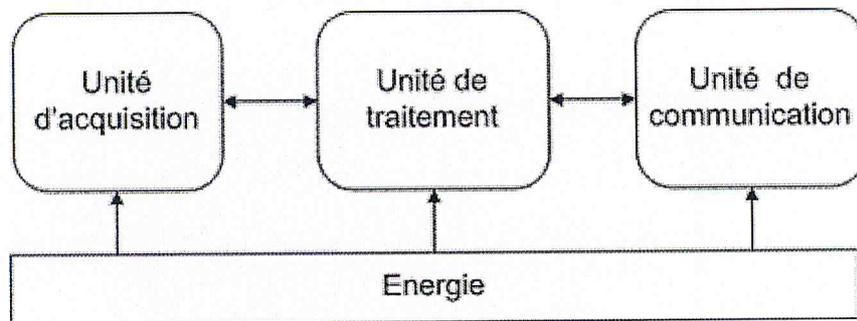


FIGURE 1.2 – Composants d'un nœud capteur

### 1.3 Les réseaux de capteur sans fil

Les réseaux de capteurs sans fil RCSF (réseaux de capteur sans fil) sont constitués d'un grand nombre de petits dispositifs (micro-capteurs), ils sont apparentés aux réseaux AD-HOC du point de vue infrastructure, architecture, autonomie d'énergie et utilisation des liens sans fil pour la communication (communiquent entre eux de façon autonome via un lien radio).

Les RCSF sont déployés aléatoirement dans une zone cible afin de recueillir régulièrement des mesures et de faire remonter des alarmes vers certains nœuds capables de relayer l'information à grande échelle.

L'alimentation électrique de chaque dispositif est assurée par une batterie individuelle elle représente la seule source d'énergie du nœud capteur qui lui permet d'effectuer ces tâches c'est pour cela que l'optimisation de la consommation de l'énergie est très importante.

## 1.4 Domaine d'application

La taille de plus en plus réduite des micro-capteurs, leur coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, etc.) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir rapidement plusieurs domaines d'application [9]. Nous résumons ci-dessous les principaux domaines :

- **Applications militaires** : les réseaux de capteurs sans fil sont susceptibles d'être une partie intégrante du commandement militaire, contrôle, communications, renseignement, surveillance de champs de batailles, reconnaissance et systèmes de ciblage.
- **Surveillance de zones** : pour la surveillance de zones, les nœuds capteurs sont déployés dans une zone où un phénomène doit être surveillé. Lorsque les capteurs détectent l'évènement surveillé (chaleur, pression, etc.), l'information est rapportée à la station de base, qui prend par la suite les mesures nécessaires.
- **Transport** : des informations en temps réel sur le trafic sont collectées par les réseaux de capteurs pour suivre plus tard des modèles de transport et alerter les chauffeurs de la congestion et des problèmes de circulation.
- **Applications dans la santé** : certaines des applications des réseaux de capteurs dans la santé sont : la surveillance intégrée des patients, les diagnostics des maladies et l'administration de médicaments dans les hôpitaux, la télésurveillance des données physiologiques de l'homme, le suivi et la surveillance des médecins ou des patients à l'intérieur des hôpitaux.
- **Détection de l'environnement** : le terme réseaux de capteurs environnementaux [10] couvre de nombreuses applications des réseaux de capteurs à la recherche dans les sciences de la terre. Cela comprend la détection des volcans, ou d'autres évènements dans les océans, les glaciers, les forêts, etc [10]. Nous citons à titre d'exemple :
  - la surveillance de la pollution de l'air .
  - la détection des feux de forêts .
  - la détection des glissements de terrains.
- **Surveillance des structures** : les réseaux de capteurs sans fil peuvent être utilisés pour surveiller les mouvements à l'intérieur des bâtiments et des infrastructures, tels que les ponts et les tunnels.
- **agricole** : l'utilisation d'un réseau sans fil libère l'agriculteur de l'entretien du câblage dans un environnement difficile. L'automatisation de l'irrigation permet une utilisation plus efficace de l'eau et de réduire les déchets. D'autre part, les capteurs sont capables de surveiller le taux de pesticides dans l'eau potable, le degré d'érosion du sol, et le niveau de pollution de l'air en temps réel.

## 1.5 Architecture d'un réseau de capteur sans fil

Un réseau de capteurs est constitué d'un très grand nombre de capteurs déployés sur une zone de captage spécifique afin de recueillir des informations sur l'environnement qui les

entoure et de les communiquer entre eux. Dans un RCSF, la présence unique de capteurs n'est pas suffisante, la présence d'une station de base ou un nœud collecteur est nécessaire au fonctionnement du réseau. D'où l'introduction des stations de contrôle appelées nœuds puits, 'Sink' ou stations de base [11] plus communément.

Une station de base peut être un nœud fixe ou mobile qui a pour rôle de collecter les différentes informations des différents capteurs du réseau et de les stocker localement. Ayant beaucoup plus de capacité qu'un simple nœud, la station de base peut exécuter des calculs plus complexes, stocker beaucoup plus d'informations, et même être connectée à internet [12]

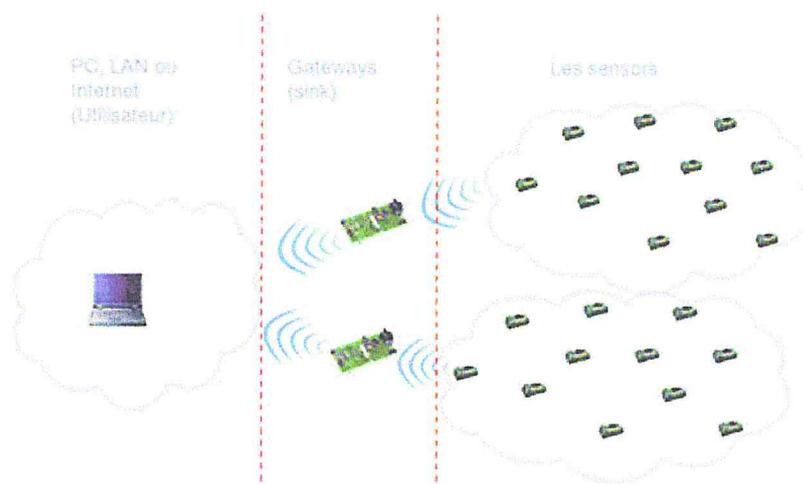


FIGURE 1.3 – Architecture d'un réseau de capteurs sans fil

## 1.6 Modèle en couche

Le modèle de communication en couches figure 1.4 utilisé par tous les nœuds du réseau favorise la coopération en combinant le routage de l'information et l'économie d'énergie des équipements [7]. Il comprend cinq couches : la couche application, la couche transport, la couche réseau, la couche liaison de données et la couche physique, ainsi que des plans de gestion d'énergie, de mobilité et de tâches.

La couche application diffère d'un capteur à un autre selon la fonctionnalité de chacun. Elle permet d'assurer la transparence des couches sous-jacentes aux utilisateurs, elle apporte aussi une interface conviviale facile d'utilisation. La couche Transport fait le lien entre la couche applicative et la couche réseau en segmentant ou réordonnant les données en transit avant d'être route par la couche réseau. La gestion des liaisons entre les stations de base est assurée par la couche liaison de données, elle assure aussi le contrôle d'erreur. Le protocole MAC (Media Access Control) de cette couche gère l'accès au support physique. La dernière couche physique assure les techniques de modulation, de cryptage, de transmission et de réception de données.

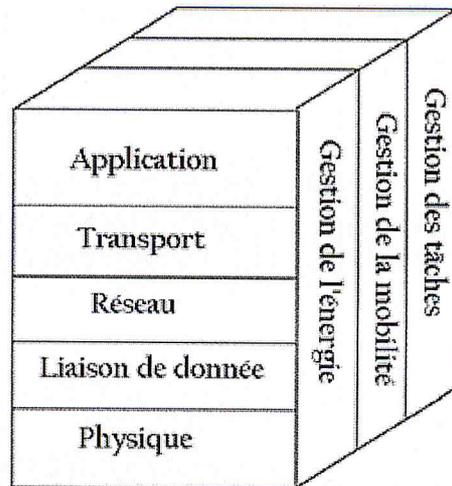


FIGURE 1.4 – Modèle en couches d'un réseau de capteurs sans fil

## 1.7 Routage dans les RCSF

Router l'information dans les réseaux de capteurs est considéré comme un problème complexe à cause des caractéristiques inhérentes des nœuds capteurs[13]. Il existe plusieurs façons de classer les protocoles de routage. En général, ces protocoles sont divisés en trois classes : routage plat, routage hiérarchique et routage géographique.

### 1.7.1 Routage à plat

Dans ce type de routage, tous les nœuds sont au même niveau hiérarchique, ont les mêmes rôles et mêmes fonctionnalités. Ils collaborent entre eux afin d'accomplir une tâche de captage[12]. La distinction de chaque nœud du réseau par l'attribution d'un identificateur unique n'est pas faisable à cause du grand nombre de nœud dans le réseau. Ceci a conduit à l'apparition de protocoles Data-Centric, où la station de base envoie des requêtes à certaines régions et attend des réponses à partir des nœuds situés dans ces régions. Un exemple de protocoles Data-Centric est : SPIN [14] et Directed Diffusion[15].

### 1.7.2 Routage hiérarchique

Dans ce type de routage, tous les nœuds n'ont pas un niveau hiérarchique similaire et jouent différents rôles suivant leur rang. Ce type de routage se base sur une organisation en arbre et sur un partitionnement du réseau en un ensemble de groupe disjoint afin de garantir le passage à l'échelle, assure aussi la connexité et prolongation de la durée de vie du réseau[16]. La communication entre les différents clusters est assurée grâce à des nœuds de niveau supérieur appelés cluster-heads. Le rôle de ces derniers est principalement le routage intra et extra cluster.

### 1.7.3 Routage géographique

Le routage géographique se base sur la localisation relative ou absolue des nœuds capteurs dans le but de trouver une route vers les destinataires. Il s'agit d'utiliser les coordonnées du nœud source et de ses voisins afin de transmettre un paquet au plus près de sa destination[17].

## 1.8 Nature du Trafic

On distingue dans les réseaux de capteurs sans fil deux méthodes d'envois de données (souvent appelées "nature du trafic") Trafic périodique et Trafic apériodique [18] :

### 1.8.1 Le Trafic périodique

Dans les réseaux à trafic périodique, la transmission des données collectées se fait à chaque fin de période suivant des intervalles réguliers. Chaque nœud génère donc des flux intermittents de message. Ceci leur permet de conserver leur énergie en restant en mode sommeil [19]. Ce type de trafic est utilisé dans des applications de supervision.

### 1.8.2 Le Trafic apériodique

Dans le trafic apériodique ,la transmission se fait de deux façons :

**Transmission par évènement** : la transmission de données se fait que lorsqu' un évènement est détecté à proximité. Un évènement peut être un changement significatif de la valeur mesurée par un capteur. Cette transmission est utilisée dans les applications de surveillance telle que la détection d'incendie.

**Transmission par requête** la transmission de l'information se fait que lorsqu'un nœud reçoit une demande explicite du besoin de la valeur captée.

## 1.9 Topologie d'un réseau de capteur sans fil

Les RCSF communiquent entre eux de manière sans fil et s'organisent de différentes façons. Ci-dessous, nous décrivons brièvement les différentes topologies pouvant être adaptées à ce genre de réseau.

### 1.9.1 Topologie en étoile

Comme son nom l'indique ce réseau s'organise en étoile voir Figure 1.5, C'est à dire que la station de base qui communique avec tous les autres nœuds capteurs se trouve au centre, quant aux nœuds capteurs sont déployés autour d'elle. Ces dernier ne sont pas autorisés à communiquer entre eux, seulement avec la station de base. L'avantage de ce type de réseau est la simplicité de déploiement, et la minimisation de la charge du réseau et de la consommation d'énergie des différent capteurs. Cependant cette topologie dépend fortement de la station

de base qui doit être dans à la portée de tous les autres nœuds, et pour que le réseau reste actif, la station de base [20] doit toujours être active.

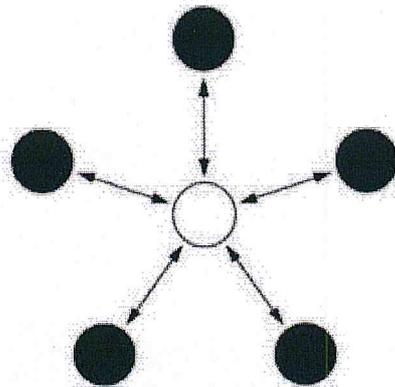


FIGURE 1.5 – Topologie en étoile

### 1.9.2 Topologie en maille

Cette topologie (figure 1.6) permet la communication de tous les nœuds entre eux, même s'ils ne sont pas à portée de communication, on dit que c'est une communication multi saut c'est à dire que si un nœud veut communiquer avec un autre en dehors de sa portée de communication, il utilisera un nœud intermédiaire[20]. L'avantage de la topologie en maille est de permettre l'extension du domaine du réseau par l'ajout de nouveau nœud. Ainsi que la redondance qui assure une communication sûre même en cas de nœuds défailants. Cependant l'acheminement des messages dans ce réseau consomme plus d'énergie que dans les réseaux en étoiles et ajoute une certaine latence avant la délivrance des messages.

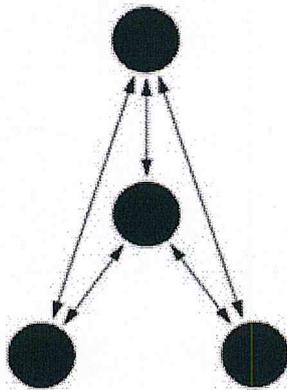


FIGURE 1.6 – Topologie maillée

### 1.9.3 Topologie hybride

Cette topologie qui est représenté dans la Figure 1.7 combine entre les deux topologies citées ci-dessus en assurant un réseau robuste et souple tout en assurant une consommation d'énergie minimale. Dans cette topologie, les nœuds capteurs ayant un taux d'énergie faible ne font pas l'acheminement d'informations en multi saut, ceci est dans le but d'économiser leur énergie. Quant aux autres nœuds, ceux qui ont un taux plus élevé d'énergie, ils doivent effectuer ce type de transmission[21].

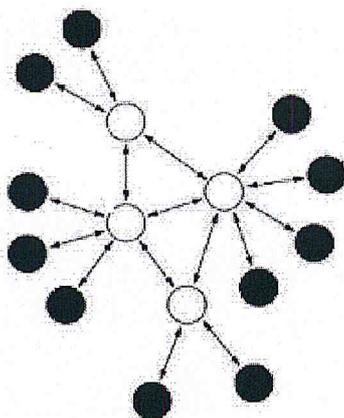


FIGURE 1.7 – Topologie hybride étoile - maillée

## 1.10 Contraintes de conception des RCSF

Les contraintes de réalisation des RCSF sont principalement des contraintes conceptuelles et matérielles. Ces facteurs servent comme directives pour le développement des algorithmes et protocoles s'utilisant dans les RCSF.

### 1.10.1 Contrainte conceptuelles

La conception des RCSF, leurs protocoles et algorithmes sont guidés par plusieurs paramètres :

- **La tolérance aux pannes :**

L'épuisement de la batterie peut engendrer des mal-fonctionnements des nœuds capteur qui peuvent générer par la suite des erreurs, il existe aussi des problèmes physique ou des interférences. Ces problèmes ne devraient Pas affecter le fonctionnement du reste du réseau. Ceci est le principe de tolérance aux pannes.

- **Le coût de production :**

Le coût de production d'un seul micro-capteur est très important pour l'évaluation du coût global du réseau. Si ce dernier est supérieure à celui nécessaire pour le déploiement des capteurs classiques. L'utilisation de cette nouvelle technologie ne serait pas rentable, Par conséquent réduire le coût de production d'un capteur jusqu'à moins de 1 dollar

par nœud est un objectif important pour la faisabilité de la solution des réseaux de capteurs sans fils[7] .

- **Le Passage à l'échelle :**

Atteindre des centaines ou des milliers de nœuds capteurs dans un RCSF. Suivant l'application. Ce nombre peut être augmenté jusqu'à des millions de capteurs. Le réseau doit fonctionner avec des densités de capteurs très grandes, ce qui engendre un grand trafic dans les réseaux. C'est pourquoi il est nécessaire que la station de base soit équipée d'une mémoire suffisante pour stocker les data reçues.

- **Média de transmission et Environnement :**

Les nœuds communicants sont reliés par liens sans fil. Ce lien peut être réalisé par radio, infrarouge ou bien par un média optique. Les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : humidité, pluie, forte chaleur, etc.

- **Consommation énergétique :**

Un des principaux défis des réseaux de capteurs sans fils tourne autour de la limite des sources d'énergie. La réduction de la taille des nœuds capteurs implique la réduction de leur batteries internes, ce qui implique par la suite de l'énergie en moins. C'est pour cela qu'il faudra envisager des politiques de gestion de cette denrée précieuse afin de maximiser la durée de vie de tout un réseau[6].

- **Agrégation de donnée :**

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

### 1.10.2 Contraintes matérielles

La principale contrainte matérielle est la taille des capteurs. Leur petite taille peut présenter de nombreux avantages, elle permet notamment un déploiement flexible et simple du réseau. Cependant, la puissance des batteries utilisées pour alimenter les nœuds capteurs est limitée par la petite taille de ces derniers. Cependant, d'autres contraintes exigeantes pour la construction des nœuds capteurs existent, ces nœuds doivent :

- Consommer le minimum d'énergie pour que le réseau survive le plus longtemps possible.
- s'adapter aux différents environnements (forte chaleur, eau...).
- Être autonomes (pouvoir opérer sans assistance) et très résistant vu qu'ils sont souvent déployés dans des environnements hostiles.

- Les processus des réseaux de capteurs sont différents de ceux d'une machine classique car ils utilisent souvent des microcontrôleurs de faibles fréquences

## 1.11 Consommation énergétique d'un nœud capteur

### 1.11.1 Formes de dissipation d'énergie

Un capteur est principalement alimenté par une batterie dont il est équipé. Etant donné que cette batterie n'est ni remplaçable ni rechargeable, un capteur doit donc consommer l'énergie dont il dispose de manière économique pour survivre le plus longtemps possible. Afin de concevoir de bonnes solutions énergétiques à ce genre des nœuds capteurs, il est important de connaître comment cette énergie est consommée et d'analyser les différents facteurs provoquant sa dissipation [22] De manière générale, la dissipation d'énergie se fait selon les modes suivant :

- **Unité de traitement** : généralement l'unité de traitement opère de différentes façons selon les situations. Elle possède trois modes de fonctionnement : actif, oisif, et sommeil. Chaque mode est caractérisé par sa propre consommation d'énergie.
- **Unité de transmission** : cette unité par contre 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 données[23].
- **Unité de captage** : les sources de consommation d'énergie de l'unité de captage sont généralement l'échantillonnage et la conversion des signaux physiques en signaux électriques, le conditionnement des signaux et la conversion analogique-numérique. Comme il existe divers types de capteurs, il n'y a pas d'informations typiques sur la consommation énergétique d'un capteur, on peut avoir par exemple des capteurs de température ou de pression qui peuvent consommer peu d'énergie ou des capteurs d'images ou des sonars qui peuvent consommer beaucoup d'énergie. 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. Cependant, des expériences ont montré que l'unité de transmission est celle qui consomme le plus d'énergie dans un capteur [22].

### 1.11.2 Sources de surconsommation d'énergie

La surconsommation d'énergie représente toute consommation superflue qui peut être évitée afin de Conserver l'énergie d'un nœud capteur. Les sources de cette surconsommation sont nombreuses. Elles peuvent être engendrées lors d'un captage d'information mal géré, ou lors d'un échantillonnage mal contrôlé [24].

La communication peut aussi conduire à une surconsommation lorsque certains phénomènes

se produisent au niveau du support physique de communication. Certains de ces phénomènes causant cette perte d'énergie ont été recensés dans [25], [26], [27].

- **Les collisions** : lors de l'émission, il est possible que deux trames de données se heurtent et deviennent ainsi inexploitable et doivent être abandonnées. Ainsi, leur re-transmission engendre une consommation accrue d'énergie. Ce phénomène est considéré comme la source majeure de la perte d'énergie.
- **L'écoute à vide** : l'état oisif des nœuds les met dans une situation où ils écoutent constamment le support de communication pour recevoir des informations alors qu'ils peuvent ne rien recevoir. Ceci est coûteux et inutile dans certains types de réseaux à faible échange de trafic. Eteindre la radio dans ce cas peut être une solution mais comme cité plus haut la transition entre les différents modes du module de communication consomme aussi de l'énergie.
- **L'écoute abusive ou overhearing** : cette situation se présente quand un nœud reçoit des données qui ne lui sont pas destinées. overhearing peut engendrer une forte consommation d'énergie lorsque la charge du trafic est élevée et la densité des nœuds est grande.
- **Overmitting** : cette situation présente le cas d'un nœud transmetteur qui envoie des données alors que le nœud destinataire n'est pas prêt à les recevoir.
- **Overhead des paquets de contrôle** : l'envoi, la réception et l'écoute des paquets de contrôles consomment de l'énergie. Comme les paquets ne transportent pas directement des données, ils réduisent également le débit utile effectif.

### 1.11.3 Mécanismes de conservation d'énergie

Comme cité précédemment, la transmission de données se révèle extrêmement consommatrice par rapport aux tâches du nœud capteur. Nous présentons dans ce qui suit certains mécanismes de base pour la conservation d'énergie.

- **Mode économie d'énergie** : cette technique consiste à éteindre le module radio dès que possible à titre d'exemple le protocole de communication basé sur la méthode TDMA nous donne la possibilité d'implémenter un tel mécanisme pendant que le nœud n'est pas en train de transmettre.
- **Traitement local** : l'idée de cette technique est une collaboration entre les nœuds voisins et très proches, et où un changement significatif dans l'environnement provoque un envoi de paquets. Ainsi, à la détection d'un événement, au sein de ce voisinage, les données seront confondues localement et agrégées au sein d'un seul message. Cette stratégie permet de réduire considérablement le trafic.

- **Organisation des échanges** : cette technique permet de limiter les problèmes de retransmission dus aux collisions. Une solution serait d'utiliser la technique d'accès au medium TDMA. Cependant, cette solution n'est pas très évidente à mettre en place puisque cela demande une synchronisation très délicate entre les capteurs. Une autre solution est l'utilisation de S-MAC [25] basé sur la méthode d'accès au canal CSMA/CA qui implémente des techniques d'évitement de collisions avec le mécanisme RTS/CTS ainsi qu'un mécanisme de mise en veille distribué. Ceci entrainera naturellement à une économie d'énergie.
- **Limitation des accusés de réception** : les accusés de réceptions sont principalement utilisés à des fins de synchronisation ou comme assurance de délivrance d'information. Cependant, en abuser, conduira à une congestion du réseau et évidemment des collisions et des interférences.  
Clustérisations : certaines architectures des RCSF ne sont pas optimisées pour l'économie d'énergie. Effectivement, lors d'une transmission directe vers la station de base tous les nœuds sont sujets au phénomène overhearding. Dans un autre cas d'une transmission par saut, les nœuds relayeurs seront soumis à une forte charge et épuiseront par conséquent leur énergie rapidement. La clustérisation dans ce cas, peut-être la mieux adaptée. Effectivement, l'élection d'un cluster Head en prenant en compte son taux d'énergie peut être favorable pour une consommation énergétique optimisée.

## 1.12 Conclusion

Dans ce chapitre nous avons pu constater l'importance des réseaux de capteurs sans fils qui surpassent les réseaux sans fils traditionnels grâce à leur flexibilité, leur capacité de captage, leur cout réduit, leur installation rapide ainsi que leur utilisation varié dans différents domaines d'application.

Nous avons aussi montré que les réseaux de capteurs sans fils sont soumis à des contraintes qui peuvent être soit matérielles ou conceptuelles et la qualité du réseau de capteurs sans fils dépend de la façon dont nous abordons ces problèmes, parmi ces contraintes il y a la consommation de l'énergie qui est très importante car elle se répercute directement sur la durée de vie du RCSF. et le plus longtemps ces réseaux sont en vie, le mieux c'est.

Nous aborderons dans la suite de ce mémoire les réseaux de Petri stochastiques généralisés. Ce modèle formel et graphique permet une description précise des systèmes complexes dans divers domaines. En effet, cet outil permet de déterminer et de calculer les paramètres de performance des systèmes modélisés.

# Chapitre 2

## Les réseaux de Petri

## 2.1 Introduction

En 1962, Carl Adam Petri définissait les réseaux de Petri qui portent depuis son nom. Il constitue un modèle mathématique et un outil graphique puissant servant à présenter et à décrire formellement des systèmes complexes dans lesquels les notions d'évènement et d'évolution sont importantes. Les réseaux de Petri sont un excellent outil de modélisation des systèmes caractérisés par la concurrence, l'exclusion mutuelle, la synchronisation et le conflit. Cependant aucune notion de temps n'a été considérée.

Par la suite, la notion de temps a été introduite par RAM-CHANDANI [28], MERLIN [29], SIFAKIS [30]. Il existe aussi une autre approche des réseaux de Petri utilisant des temps aléatoire qui sont les Réseaux de Petri Stochastique, introduite par SYMONS [31], FLORIN ET NATKIN [32] ET MALLOY [33].

Grace à cela, les réseaux de Petri sont devenues une partie intégrante dans le domaine de l'évaluation des performances. Une autre extension des RdP a été proposée dans [34], où des temps aléatoires sont combinés avec des temps déterministes nuls. Ce formalisme est connu sous le nom des Réseaux de Petri Stochastique Généralisés. Dans ce chapitre, nous allons faire une description des réseaux de Petri, leur concept de base et définitions. Ainsi que les deux extensions existante que sont les réseaux de Petri stochastique et stochastique généralise leur concept de base et définition.

## 2.2 Concepts de base et définition

Un réseau de Petri est constitué de deux parties : une structure et un marquage. La structure est un graphe biparti orienté, qui représente la partie statique du système modélisé. Elle est constituée de deux types de nœuds : les places (appelée aussi Condition) et les transitions (appelée aussi évènement). Les places représentent les différents états possibles du système alors que les transitions représentent les actions qui peuvent modifier l'état du système. Les places et les transitions sont reliées par des arcs [35]. On dit d'un arc qu'il est en amont s'il est dirigé d'une place à une transition, dans ce cas il décrit l'état local à partir duquel un évènement peut avoir lieu.

Un arc en aval est un arc dirigé d'une transition vers une place et exprime les changements de l'état local causés par l'évènement. Les arcs sont pondérés par des entiers positifs appelés poids de l'arc. Les poids représentent le nombre de ressources nécessaires pour effectuer une action dans le cas d'un arc en amont, et le nombre de ressources produites par une action dans le cas d'un arc en aval.

Une place contient un nombre entier positif de jetons ou marques qui représente les ressources. On appelle une place condition si cette dernière contient un seul jeton au maximum. La condition est ainsi vérifiée si le jeton est disponible dans cette place. On appelle une situation une place qui contient plusieurs jetons.

Un marquage est l'ensemble des jetons présents dans les différentes places à un instant donné. L'état initial du système est un marquage initial noté  $M_0$ , qui correspond aux marques

initiales des différentes places.

### 2.2.1 Définition formelle

Un réseau de petri est défini formellement par un tuple  $(P, T, \text{Pré}, \text{Post})$ [36], où :

- $P = (p_1, p_2, \dots, p_n)$ , est un ensemble fini de places.
- $T = (t_1, t_2, \dots, t_n)$ , est un ensemble fini de transitions.
- $\text{Pré} : P \times \mathbb{N} \rightarrow \mathbb{N}$ , est l'application d'incidence avant, correspondante aux arcs
- $\text{Post} : P \times \mathbb{N} \rightarrow \mathbb{N}$ , est l'application d'incidence arrière, correspondante aux arcs allant des transitions aux places.

### 2.2.2 Représentation graphique

Les réseaux de Petri se compose d'un ensemble fini de places  $P_i$  et des transitions  $T_i$ . Les places sont reliées par des **arcs orientés**. Comme cela est présenté dans la figure 2.1. Un réseau de Petri est dit **graphe biparti alterné**, c'est à dire qu'il y a alternance des types de nœuds : tout arc, qui doit obligatoirement avoir un nœud à chacune de ses extrémités, relie soit une place à une transition soit une transition à une place.

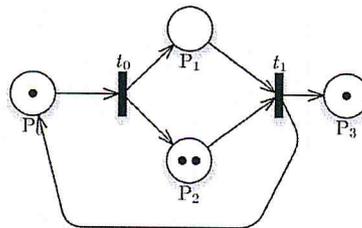


FIGURE 2.1 – Représentation graphique d'un réseau de Petri

### 2.2.3 Représentation matricielle

Un réseau de Petri est représenté de manière matricielle par une matrice appelée matrice d'incidence, elle-même est calculée à partir d'une matrice d'incidence arrière et d'une matrice d'incidence avant de la manière suivante :

$$\omega = \omega^+ - \omega^-.$$

Où :

- $\omega$  est la matrice d'incidence.
- $\omega^+$  est la matrice d'incidence arrière, qui définit les relations entre les transitions et les places. Tel que  $\omega^+(p,t) = \text{Post}(p,t)$ .
- $\omega^-$  est la matrice d'incidence avant, qui définit les relations entre les places et les transitions tel que  $\omega^-(p,t) = \text{Pre}(p,t)$ .

## 2.3 Évolution des réseaux de Petri

Les définitions et concepts de bases abordés précédemment désignent la description instantanée et statique du système. Cependant un système évolue avec le temps. Cette évolution correspond à l'évolution du marquage initiale suivant plusieurs évènements propres au système. Il est intéressant d'étudier un système évolutif, effectuant certaines actions et changeant d'état constamment. Ainsi on disposera d'une description de son fonctionnement ainsi qu'une modélisation comportementale du système. Les notions suivantes permettent d'exprimer la dynamique des réseaux de Petri. Nous utilisons les notations suivantes [37]

- $t = p \in P \mid \text{Pré}(p,t) \geq 0$ , est l'ensemble des places d'entrée de  $t$ .
- $t = p \in P \mid \text{Post}(p,t) \geq 0$ , est l'ensemble des places de sortie de  $t$ .

### 2.3.1 Règles de sensibilisation et de franchissement

Les règles de franchissement dans un réseau de Petri sont liées aux transitions. La règle de sensibilisation définit les conditions qui permettent à une transition  $t$  d'être franchie, et la règle de franchissement décrit le changement de l'état du système produit par le franchissement.

Une transition  $t$  est sensibilisée ( tirable ou franchissable ) depuis un marquage  $M$  si et seulement si chaque place d'entrée  $P$  contient un nombre de marques supérieur ou égal au poids associé à l'arc allant de  $P$  à  $t$ . D'une façon formelle on écrit :

$$\forall p \in P, M(p) \geq \text{pré}(p,t).$$

Et on note :  $M[t]$ . Le tir d'une transition sensibilisée depuis un marquage  $M$  exclut de chaque place  $p \in t$  un nombre de marques égal au poids associé à l'arc allant de  $p$  à  $t$ , et dépose dans chaque place  $p \in t$  un nombre de marques égal au poids associé à l'arc allant de  $t$  à  $p$ . Ceci, conduit à un nouveau marquage  $M'$  défini par :

$$\forall p \in P, M'(p) = M(p) - \text{Pré}(p,t) + \text{Post}(p,t). \text{ Dans ce cas, le marquage } M' \text{ est dit}$$

accessible à partir du marquage  $M$  et on note :  $M [t]M'$ . [38, 39, 40] .

### 2.3.2 Ensemble d'accessibilité et graphe des marquages accessibles

A partir d'un marquage initial  $M_0$  et en franchissant successivement des transitions dans un ordre donné on atteindra un marquage  $M_k$ . on désigne par le terme séquence de franchissement les différentes transitions successives franchies à partir de  $M_0$  jusqu'à  $M_k$ [38]. et on note :  $\sigma = t_1 t_2 \dots t_k$ .

On dit d'un marquage  $M_n$  accessible à partir de  $M_0$  s'il existe une séquence de franchissement  $\sigma$  permettant de transformer  $M_0$  à  $M_n$ .

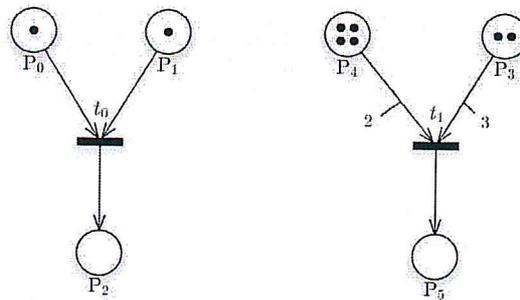


FIGURE 2.2 – Exemple de transitions

dans ce cas  $M_n$  est accessible à partir de  $M_0$  par  $\sigma$  et on écrit :  $M_0 [\sigma ]M_n$ . L'ensemble des marquages accessibles (ensemble d'accessibilité) noté  $A(R, M_0)$ , est l'ensemble des marquages atteignables à partir du marquage initial  $M_0$  par une séquence de franchissement  $\sigma$ . Formellement :

$$A(R, M_0) = \{M \in \mathbb{N}^n \mid \exists \sigma \in T^* \text{ tel que } M_0 [\sigma ]M.\}$$

Le graphe d'accessibilité (graphe de marquages accessibles GMA) d'un réseau de Petri, noté  $G(R, M_0)$  qui définit entièrement la sémantique du réseau, est défini comme un graphe dont les sommets (nœuds) sont les marquages accessibles appartenant à  $A(R, M_0)$  et les arcs sont pondérés par les noms des transitions, qui définissent la relation de tir entre les marquages. Donc, un arc étiqueté par  $t$  joint  $M'$  à  $M$  si et seulement si :  $M [t]M'$ . Aussi, le graphe des marquages accessibles n'est pas nécessairement fini.

## 2.4 Domaine d'applications des réseaux de Petri

Les réseaux de Petri forment un outil de modélisation à la fois élégant et d'une grande puissance. C'est notamment dans le contexte en pleine croissance des systèmes informatiques, que cet outil prend toute son importance. Il devient nécessaire de pouvoir concevoir des systèmes toujours plus complexes, et surtout de s'assurer de leur bon fonctionnement.

Les réseaux de Petri font preuve d'un degré d'adaptation étonnant, et c'est ce qu'on peut remarquer au travers des différents exemples que nous avons rencontrés au fil des articles. On les trouve dans des domaines allant de la biologie à l'étude des organisations.

Tout d'abord les réseaux de Petri, de par les GRAFCET, ont d'abord servi à la spécification des cahiers de charges, dans le domaine industriel, avant de devenir un langage de modélisation graphique pour les automates programmables.

Les réseaux de Pétri trouvent leur place dans le domaine informatique et sont particulièrement adaptés à la modélisation des protocoles de communication. Ils ont trait à la compilation de programmes. Enfin, dans la mesure où l'informatique s'oriente vers les systèmes répartis et où les réseaux de Pétri se montrent plus intuitifs pour modéliser les comportements parallèles, on les utilise de plus en plus pour modéliser des systèmes informatiques.

## 2.5 Situation possible entre transition

Plusieurs situations sont possibles lorsqu'un RdP transit d'un état à un état successif. Les plus importantes à citer sont : le conflit, la confusion, la concurrence et l'exclusion mutuelle.

**Conflit** : il existe deux sortes de conflit voir Figure 2.3, entre deux transitions  $t_1$  et  $t_2$ . On parle d'un conflit structurel si et seulement  $t_1$  et  $t_2$  ont au moins une place commune en entrée. Elles sont en conflit effectif pour un marquage  $M$  si elles sont en conflit structurel et :

$$M[t_1] \text{ et } M[t_2] \text{ et } \exists p \in P : M(p) < \text{Pré}(p, t_1) + \text{Pré}(p, t_2).$$

Un conflit effectif correspond donc à un choix exclusif entre deux franchissements .

**Concurrence** : la concurrence désigne le parallélisme des actions, contrairement au conflit. Donc, deux transitions  $t_1$  et  $t_2$  sont concurrentes dans un marquage  $M$  si elles sont toutes les deux franchies dans  $M$ , et elles ne sont pas en conflit.

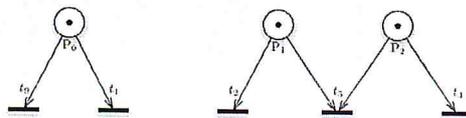


FIGURE 2.3 – Situations de conflit et de confusion

**Confusion** : une situation de confusion définit une situation où deux transitions  $t_1$  et  $t_2$  sont en même temps en conflit et en concurrence.

**Exclusion mutuelle** : l'exclusion mutuelle dans les RDP représente la situation où deux transitions  $t_1$  et  $t_2$  ne peuvent jamais être franchies en même temps dans un même marquage  $M$ . autrement dit,  $t_1$  et  $t_2$  sont mutuellement exclusif si et seulement si :

$$\forall M \in R(M_0) : t_1 \in E(M) \text{ et } t_2 \in E(M)$$

$E(M)$  désigne l'ensemble des transitions sensibilisées dans le marquage  $M$ .

## 2.6 Propriétés des réseaux de Petri

Les réseaux de Petri ont plusieurs propriétés qui peuvent être regroupées en deux classes majeurs, les propriétés concernant la structure du réseau intitulées propriétés structurelles et les propriétés qui se rapportent au comportement dénommé propriétés comportementales. Les propriétés structurelles ne dépendent que de la structure du RdP et non pas de la manière dont les jetons évoluent dans le réseau. Quant aux propriétés comportementales dépendent du marquage initial. Si ce marquage est changé, rien ne garantit que les propriétés tiennent encore. Nous citons dans ce qui suit quelques définitions relatives aux propriétés comportementales.

### 2.6.1 Bornitude

Dans un RdP  $(R, M_0)$ , une place est dite bornée pour un marquage initial  $M_0$  si pour tout marquage accessible à partir du marquage initial le nombre de marque dans la place reste toujours borné et est dite K-borné si le nombre de jetons est toujours inférieur ou égal à un entier positif K. On dit d'un RdP qu'il est K-borné si toutes ses places sont bornées. Dans le cas où k vaut 1 on désignera le RdP comme étant Sain.

### 2.6.2 Absence de blocage

Un système modélisé par un réseau de Petri est en situation de blocage, s'il aboutit à un état (marquage) où aucune transition n'est franchissable. un réseau de Petri est dit sans blocage pour un marquage initial  $M_0$  si aucun marquage accessible ne représente une situation de blocage.

### 2.6.3 Vivacité

Une transition  $t$  dans un réseau de Petri  $(R, M_0)$  est dit vivante pour le marquage initial  $M_0$  si et seulement si pour chaque marquage accessible  $M$ , il existe un marquage  $M'$  et une séquence de transition  $\sigma$  contenant  $t$  tel que :  $M [\sigma ]M'$ . autrement dit, il y'a toujours une possibilité de franchissement de  $t$  à partir de tout marquage accessible de  $M_0$ . Une transition qui n'est pas vivante est dite morte. Si toutes les transitions sont vivantes, le réseau de Petri est dit vivant. Une conséquence très importante de la propriété de vivacité, est que s'il existe au moins une transition vivante, aucune situation de blocage n'aurait lieu. [38].

### 2.6.4 Réinitiability et état d'accueil

Un état d'accueil est un marquage toujours accessible quelle que soit l'évolution du réseau. Un RdP admet un état d'accueil  $M_a$  si et seulement si :

$$\forall M \in A(R, M_0), \exists \sigma \in T_* \mid M[\sigma]M_a \quad (2.1)$$

On parle de réinitiability dans le cas où le réseau de Petri permet un état d'accueil. Ainsi, à partir de n'importe quel marquage  $M$  on pourra toujours atteindre un marquage .

### 2.6.5 Persistance

La persistance fait référence au cas où, dans un réseau de Petri, pour tout couple de transitions franchissable, l'une est tirée et l'autre reste franchissable. Une fois qu'une transition est sensibilisée dans un graphe persistant, elle reste franchissable jusqu'à ce qu'elle soit tirée.

## 2.7 Les réseaux de Petri à arcs inhibiteur

Certains phénomènes ne peuvent être représentés par les RdP simple comme le test d'absence de marque dans une place, a conduit à définir les RdP avec arcs inhibiteurs. Ce modèle permet d'interdire le franchissement d'une transition si le marquage d'une place est supérieur à une valeur.

D'autre part, les matrices d'incidence sont complétées par une matrice d'inhibition qui impose que le marquage soit strictement inférieur à une valeur donnée.

### 2.7.1 Définition [1]

Un réseau de Petri à arcs inhibiteurs représente une extension des RDP qui permet de réaliser explicitement le test à zéro par exemple. C'est-à-dire, qu'une condition de franchissement sera exprimée cette fois-ci par l'absence de jetons dans une place en entrée d'un arc inhibiteur.

Un RDP à arcs inhibiteurs est un doublet  $R_z = \langle R, Inh \rangle$  tel que :

- R est un Réseau de Petri.
- $Inh : P \times T \rightarrow (\mathbb{N} \setminus \{0\})$  est la fonction d'inhibition,  $Inh(p, t)$  représente alors le poids de l'arc inhibiteur.

Graphiquement, un arc inhibiteur est un arc dans lequel l'extrémité incidence aux transitions est représentée par un petit cercle au lieu d'une flèche[35].

### Règles de franchissement

Soit M un marquage d'un réseau de Petri à arcs inhibiteurs et t une transition, t est franchissable à partir de M si seulement si :

$$\forall p \in P, M(p) \geq \text{pré}(p, t) \text{ et } M(p) \leq \text{Inh}(p, t). \quad (2.2)$$

Ce franchissement à partir de M conduit à un nouveau marquage M' défini comme suit :

$$\forall p \in P, M'(p) = M(p) - \text{Pré}(p, t) + \text{Post}(p, t). \quad (2.3)$$

## 2.8 Les réseaux de Petri stochastiques

Un RdP stochastique est un 5-uplet  $\text{RdPS} = (P, T, E, u, M_0)$ ,  $P = P_1, P_2, \dots, P_n$  : ensemble fini et non vide de places.  $T = T_1, T_2, \dots, T_m$  : ensemble fini et non vide de transitions. A

chaque  $T_i$  on associe un taux de franchissement  $u_i$ .  $E$  = ensemble des arcs  $u = u_1, u_2, \dots, u_n$  : ensemble des taux de franchissements  $M_0$  : Vecteur marquage initial.

**Définition :**

Un réseau de Petri stochastique (RDPS) est un couple  $\langle R, W \rangle$  où :

- $R$  est un réseau de Petri.
- $W : T \rightarrow R^+$  est une fonction qui associe à chaque transition temporisée un taux de franchissement.

**Remarque :** le taux de franchissement  $\lambda_i$  peut être dépendant du marquage. Par conséquent, une transition peut avoir un taux de franchissement qui diffère d'un marquage à un autre. Dans ce cas, on note  $\lambda_i(M_j)$ , le taux de franchissement associé à la transition  $t_i$  pour le marquage  $M_j$  [41].

### 2.8.1 Evolution des réseaux de Petri stochastiques

L'évolution d'un RdPS est décrite par un ensemble de marquages obtenus à des instants de franchissement successifs. Lorsqu'un nouveau marquage est atteint, chaque transition  $t_i$  sensibilisée prend une instance de tir de délai aléatoire de la fonction de densité de probabilité associée. Le paramètre de cette fonction associé à la transition  $t_i$  est le taux de tir  $\lambda_i$ . Si le marquage sensibilise plusieurs transitions, la transition à tirer est déterminée par une des politiques de franchissements. La politique la plus appliquée dans la pratique est la politique de course (modèle concurrentiel). Cette politique choisit comme transition à tirer parmi les transitions franchissables celle qui a le plus petit délai de franchissement. Il existe une autre politique de franchissement appelée politique de pré-sélection, où la transition tirée est choisie selon une autre variable aléatoire indépendamment du délai de franchissement. Le nouveau marquage généré est obtenu selon la règle de franchissement standard des RdP classiques. [42].

**Remarque :** La probabilité que la transition sensibilisée  $t_k$  soit celle qui prend le délai de franchissement minimal est :

$$P(t_k) = \frac{\lambda_k}{\sum \lambda_i} \quad (2.4)$$

Où : la somme est faite sur tous les indices des transitions sensibilisées.

### 2.8.2 Analyse d'un réseau de Petri stochastique

Un des intérêts majeurs des RdPS est de pouvoir combiner l'analyse qualitative et l'analyse quantitative.

**Analyse qualitative**

Cette analyse consiste à vérifier les propriétés qualitatives. Les méthodes d'analyse utilisées pour les RdP ordinaires peuvent être appliquées (vivacité, Bornitude, etc...). En se basant sur le graphe des marquages accessibles, ces propriétés sont vérifiées de la même façon que pour les RdP simples. La seule différence est que les arcs du graphe temporisé sont valus par les taux correspondant aux transitions. [41]

### Analyse quantitative

Elle consiste à calculer les paramètres quantitatifs de performance. Elle est basée essentiellement sur les chaînes de Markov (CM) [41]. En effet, en utilisant la propriété d'ergodicité de la CM déduite, le régime stationnaire est atteint au bout d'un temps fini. Ainsi, on peut évaluer les performances du système modélisé.

#### Étapes de l'évaluation des performances quantitatives des RdPS :

Évaluer les performances d'un système constitue, d'abord, une modélisation du système ainsi que la construction du graphe de marquage accessible. Il faudra par la suite construire la matrice carrée des taux de transition  $Q$  dont l'ordre est égal au nombre de marquages dans le graphe d'accessibilité et les éléments désignant les taux de transition entre les marquages [43]. Ceci est défini comme suit :

$$Q(i, j) = \begin{cases} \lambda_{ij} & \text{si } i \neq j, \\ -\sum_{j=1 \text{ et } j \neq i}^n Q(i, j) & \text{sinon.} \end{cases}$$

Où :  $n$  correspond au nombre de marquages accessibles et les autres éléments de la matrice  $Q$  sont nuls. On calcule ensuite le vecteur des probabilités à l'état stationnaire  $\pi$  en résolvant le système d'équations suivant :

$$\pi \cdot Q = 0 \text{ et } \sum_i \pi_i = 1 \quad (2.5)$$

Enfin on calcule les paramètres de performances associées.

### 2.8.3 Principaux paramètres de performance

**Fréquence moyenne de franchissement d'une transition ( $\lambda(t_i)$ ) :** On appelle fréquence moyenne (ou encore débit moyen) de franchissement d'une transition  $t_i$ , le nombre moyen de tirs de  $t_i$  en une unité de temps. Elle est calculée par :

$$F^*(t_i) = \sum_{M_i \in E(t_i)} \lambda_i(M_j) \cdot \pi_j \quad (2.6)$$

Où :  $E(t_i)$  Est l'ensemble des marquages où la transition  $t_i$  est franchissable.  $\lambda_i(M_j)$  Est le taux de franchissement de  $t_i$  en  $M_j$ .

**Nombre moyen de marquage dans une place ( $n(P)$ ) :** Correspond au nombre de jeton d'une place en moyenne pendant l'état stationnaire. Le nombre moyen de marques dans une place  $p$  est calculé en appliquant la formule :

$$M^*(P) = \sum_{M_i \in E} M_i(P) \cdot \pi_j \quad (2.7)$$

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

**Le temps moyen de séjour d'une marque dans une place :** Le délai moyen qu'un jeton passe dans une place  $p_i$  d'un RDPS à l'état stationnaire, peut être calculé comme suit :

$$T^*(P_i) = \frac{M^*(p_i)}{C_i^+ \cdot F^*} \quad (2.8)$$

Où :  $C_i^+$  est la ligne de la matrice  $C^+$  correspond à la place  $P_i$ .  $F^*$  Est le vecteur des fréquences moyennes de franchissement des transitions.  $M^*$  Est le vecteur des nombres moyens de marques. Les trois premiers paramètres cités ci-dessus peuvent représenter, dans le cas d'un protocole de communication par exemple, le débit du canal de transmission, la charge du canal et le temps moyen d'attente pour la transmission.

**La probabilité d'un évènement :** La probabilité d'un évènement particulier E peut être calculée en sommant les probabilités de tous les marquages dans lesquels la condition correspondante à la définition de l'évènement est vérifiée. Ainsi, cette probabilité est calculée comme suit :

$$P(E) = \sum_{m_i \in X} \pi_i. \quad (2.9)$$

Où : X est l'ensemble des marquages accessibles ou la condition E est vérifiée.

## 2.9 Les réseaux de Petri stochastiques généralisé

Les RdPS permettent d'analyser quantitativement et qualitativement les systèmes informatiques, téléphoniques ou de production. Mais l'existence de certaines opérations qui correspondent à des aspects purement logiques et donc qui ont une durée négligeable telle qu'aucune temporisation ne peut leur être associée fait que leur modélisation par une transition temporisée ne soit pas adaptée. Pour résoudre ce problème, une nouvelle classe de RdPS est apparue dans [33] sous le nom de réseau de Petri stochastiques généralisés (RdPSG) dans lesquels les actions logiques sont représentées par des transitions de durée de franchissement nulle.

### 2.9.1 Définition informelle

On appelle RdPSG un RdPS contenant les deux types de transitions immédiates et temporisées. Les transitions immédiates possèdent une durée de franchissement nulle. Elles sont utilisées généralement pour modéliser des événements d'urgence ou des activités prioritaires, elles sont représentées par des traits. Les transitions temporisées à qui correspondent des variables aléatoires déterminant la durée de franchissement sont représenté par des rectangles

### 2.9.2 Définition formelle

Un réseau de Petri stochastique généralisé RdPSG est un trois-up let  $\langle R_0, \pi, W \rangle$   
Où :  $R_0$  est RdP classique marqué (non temporisé sous-jacent)  $\pi : T \rightarrow \mathbb{N}$ , est une fonction de priorité qui associe à chaque transition  $t$  de  $T$ , une valeur entière, appelée priorité de la transition  $t$ . Généralement, elle associe à chaque transition temporisée la valeur 0 et à chaque

transition immédiate la valeur 1 (la valeur 1 est plus prioritaire que la valeur 0). Donnant ainsi plus de priorité aux transitions immédiates.  $W : T \rightarrow \mathbb{R}_+$  est la fonction qui associe à chaque transition temporisée un taux de franchissement et à chaque transition immédiate un poids. Les poids sont utilisés dans le calcul des probabilités de franchissement des transitions immédiates et pour la résolution des conflits entre plusieurs transitions immédiates sensibilisées en même temps [41].

### 2.9.3 Processus stochastique associé à un RdPSG

L'existence des deux types de transitions conduit à la division de l'ensemble des marquages accessibles en deux sous-ensembles [38]. Les marquages tangibles : sont des marquages qui sensibilisent uniquement les transitions temporisées (aucune transition immédiate n'est franchissable). Ces marquages représentent les états où le système modélisé passe un certain temps.

Les marquages évanescents : ce sont des marquages dans lesquels il y'a au moins une transition immédiate franchissable. Ces marquages modélisent les états dans lesquels passé est nul. Le temps de séjour moyen dans un marquage évanescent est nul, tandis que, le temps de séjour dans un marquage tangible  $M$  est une variable aléatoire correspondant au minimum des temps de franchissement des transitions sensibilisées par ce marquage. Autrement dit, c'est une loi exponentielle avec un paramètre  $\lambda_M$ , qui est la somme de tous les taux de franchissement de ces transitions [38].

$$\lambda_M = \sum_{t_k \in S(M)} W(t_k) \quad (2.10)$$

Où :  $S(M)$  est l'ensemble des transitions franchissables à partir de  $M$ .

### 2.9.4 Les règles de franchissement dans les RdPSG

Tout comme un RdP ordinaire, l'évolution d'un RdPSG se fait par une successive de marquages. Ainsi le franchissement des transitions dans un RdPSG dépend du type marquage atteint  $M$ , que ce soit tangible ou évanescent. Soit  $S(M)$  l'ensemble de transitions sensibilisées de ce marquage : Si  $S(M)$  ne contient que des transitions tangibles, alors la règle de franchissement de transition est identique à la règle de franchissement de transition dans le cas d'un RdPS, i.e. que toute transition sensibilisée peut être tirée et le tir d'une transition donnée dépend des taux de franchissement des transitions sensibilisées. Si  $S(M)$  comprend au moins une transition immédiate ( $M$  est évanescent), alors seules les transitions immédiates sont franchissables car elles sont plus prioritaires que les transitions temporisées vu leur temps de franchissement nul. Les transitions immédiates peuvent être tirées simultanément dans le cas d'un conflit effectif. Si  $S(M)$  contient plusieurs transitions immédiates ayant la même priorité alors celle qui a le poids le plus élevé sera franchi. Soit  $C(M) \subset S(M)$  l'ensemble des transitions immédiates en conflit entre elles, la probabilité qu'une transition  $t_j \in C(M)$  soit tirée est donnée par :

$$P(t_j) = \frac{W(t_j)}{\sum_{t_k \in C(M)} W(t_k)}. \quad (2.11)$$

Où :  $W(t_j)$  est le poids de la transition  $t_j$ .

Formellement, une transition  $t$  est franchissable dans le marquage  $M$  si et seulement si :

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

Il n'existe pas de transition plus prioritaire que  $t$  qui soit franchissables :

$$\forall t' \in T \mid \pi(t') \geq \pi(t); \text{on}M(t').$$

Le nouveau marquage  $M'$  résultat après le franchissement de la transition  $t$  (temporisée ou immédiate) à partir d'un marquage  $M$  est défini alors comme suit :

$$M' = M - \text{Pré}(\cdot, T) + \text{Post}(\cdot, T).$$

### 2.9.5 Analyse des performances d'un RdPSG

#### Analyse qualitative :

La méthode d'analyse qualitative utilisée dans les RdPS peut être appliquée dans les RdPSG. Cependant comme les transitions immédiates ont une priorité sur les transitions temporisées, le modèle non-temporisé sous-jacent au RdPSG est un RdP avec priorité. Autrement dit on traite tous les transitions de la même façon, mais avec une certaine priorité.

#### Analyse quantitative :

L'analyse quantitative consiste à calculer les probabilités stationnaires et les indices de performance. Elle est basée essentiellement sur les techniques d'analyse des chaînes de Markov grâce à l'isomorphisme entre les RdPSG et CM (chaîne de Markov). Néanmoins ceci nécessite la vérification de certaines propriétés résumées dans la condition d'ergodicité du RdPSG. La priorité d'ergodicité garantit l'existence du régime stationnaire. Et une fois l'ergodicité du modèle est prouvée, on procède au calcul des probabilités d'états stationnaires des marquages tangibles seulement, car la probabilité de s'y trouver dans les marquages évanescents est nulle. De plus, les marquages évanescents augmentent la taille génératrice, ce qui rend le calcul de la solution plus coûteux en temps et en espace mémoire. Pour cela, la CM doit être réduite en éliminant les marquages évanescents avant la résolution. La chaîne obtenue est dite chaîne de Markov réduite (CMR).

### Evaluation des indices de performances d'un RdPSG

Une fois l'ergodicité du modèle est démontrée, on commence l'évolution des performances en suivant les étapes suivantes :

- Construire le GMA comme pour les RdP ordinaires.
- Valoriser ce graphe par des taux de franchissement lorsqu'il s'agit de transitions temporisées et par des taux infinis quand il s'agit des transitions immédiates.
- Fusionner les marquages tangibles et les marquages évanescents qui les succèdent en supprimant les marquages évanescents pour obtenir un nouveau marquage réduit. Cette fusion n'influe pas sur le comportement dynamique du système, car le processus passe un temps nul dans le marquage évanescents.
- A partir de ce nouveau marquage réduit, on construit la matrice entre états. Cette matrice est utilisée pour calculer les probabilités stationnaires. Celles-ci, à leur tour, nous permettent de réduire les indices de performances désirés[44].

### Calcul des indices de performances à l'état stationnaire :

L'analyse qualitative des RdPSG permet de vérifier la justesse du modèle et les propriétés qualitatives du système modélisé, telles que la vivacité et la bornitude, tandis que l'analyse quantitative consiste à calculer les probabilités stationnaires et les indices de performances. Dans ce qui suit, nous allons présenter les principaux indices de performance des RdPSG, qui sont : la fréquence moyenne de franchissement des transitions, le nombre moyen de marques dans une place, le temps moyen de séjour d'une marque dans un sous-réseaux, et la probabilité d'un événement particulier se produit [45].

Fréquence moyenne de franchissement des transitions : On appelle fréquence moyenne (ou encore débit moyen) de franchissement d'une transition  $t_i$ , le nombre moyen de tirs de  $t_i$  en une unité de temps. Elle est calculé par :

$$F^*(t_i) = \sum_{M_j \in E(t_i)} \lambda_i(M_j) \cdot \pi_j \quad (2.12)$$

Où :

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

Nombre moyen de marques dans une place : le nombre moyen de marques dans une place  $p_i$  est calculé à l'aide de la formule :

$$M^*(p_i) = \sum_{M_j \in A} M_j(p_i) \cdot \pi_j \quad (2.13)$$

Où :  $M_j(p_i)$  est le nombre de jetons dans la place  $p_i$  pour le marquage  $M_j$ , et A est l'ensemble des marquages accessibles.

Temps moyen de séjour d'une marque dans un sous-réseau correspond au délai moyen qu'un jeton passe dans une partie S d'un RdPSG. Il peut être calculé en appliquant la formule suivante :

$$E[T] = \frac{E[N]}{E[\gamma]} \quad (2.14)$$

Où :  $E[N]$  est le nombre de jetons dans le sous-réseau  $S$ , et  $E[\gamma]$  est le taux d'arrivé effectif des jetons dans  $S$ .

Probabilité d'un évènement : la probabilité d'un évènement particulier  $E$  est égale à la somme des probabilités de tous les marquages dans lesquels la condition correspondante à la définition de l'évènement est vérifiée. Ainsi, elle est calculé par :

$$P\{E\} = \sum_{m_i \in X} \pi_i \quad (2.15)$$

Où :  $X = \{m_i \in A(R, M_0); \gamma(m_i) = true\}$  est l'ensemble des marquages accessibles où la condition  $E$  est vérifiée.

## 2.10 Conclusion

Au cours de ce chapitre, nous avons présenté les réseaux de Petri, ainsi que leur notions de base nécessaire à leur compréhension et leur utilité dans la description et l'analyse de problème complexe.

Ce modèle formelle permet l'expression de déroulement d'action parallèles et l'analyse des systèmes concurrents ainsi que la détermination de leurs propriétés qualitatives. Les RdP permet aussi l'évaluation des performances quantitatives de ces systèmes.

Nous avons constatés aussi un manque dans les réseaux de Petri ordinaire qu'est l'absence d'informations temporelle dans ces derniers. Ce manque est la raison principal de l'élaboration des réseaux de Petri stochastiques généralisés. Ceci à permis d'étendre leur puissance descriptive grâce à l'introduction des transitions immédiates en plus des transitions temporisées dans un même modèle.

Dans le chapitre suivant nous nous intéresserons à la réduction de la consommation énergétique des capteurs en introduisant la notion de vacance issue de la théorie des modèles de files d'attentes avec vacance dans le comportement des nœuds capteurs. Grâce au formalisme des réseaux de Petri, nous proposerons une modélisation du comportement des capteurs avec différentes politiques de vacance.

# **Chapitre 3**

**Les politiques de vacance et  
modélisation des RCSF**

### 3.1 Introduction

Les réseaux de capteur sans fils connaissent une évolution rapide et constante et ne cessent de gagner de l'importance dans plusieurs domaines différents. Dans notre premier chapitre nous avons abordés les différentes contraintes des RCSF et avons conclu que parmi les plus importantes c'est la contrainte d'énergie car elle influe directement sur la durée de vie du réseau.

Au niveau d'un nœud capteur, l'unité de communication est la plus gourmande en énergie par rapport aux autres unités. Cette unité possède quatre modes de fonctionnement : émission, réception, écoute et sommeil. Une importante observation est que la consommation d'énergie est importante lorsque l'unité de communication est dans l'état émission, réception ou écoute. Cependant, l'énergie nécessaire pour l'émission est la plus importante. Il est à noter également que l'énergie dissipée pour la transition entre le mode écoute et un des modes émission réception, génère une surconsommation d'énergie importante nommé l'énergie de transition (the switching energy). Nous avons été motivées par la question suivante : Comment peut-on réduire cette consommation au niveau du nœud afin de prolonger la durée de vie de tout le réseau.

Une des solutions est l'introduction des politiques de vacance dans le comportement des capteurs sans fil, cela permettra de minimiser la consommation en énergie. Dans le présent chapitre, nous allons élaborer nos modèles pour un seul nœud capteur, en se basant sur les disciplines de vacances et en utilisant les RdPSG comme outil de modélisation et d'évaluation des performances. Pour ce faire, nous allons commencer par définir la notion de vacance, les différents aspects des vacances et la signification d'une vacance d'un capteur sans fil. Ensuite, nous citerons quelques travaux réalisés sur l'économie d'énergie des réseaux de capteurs. Et enfin, nous donnons les indices de performances relatifs aux différents modèles.

### 3.2 Modèle général de file d'attente avec vacance

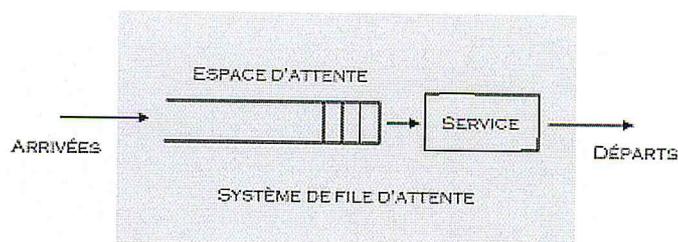


FIGURE 3.1 – Système de files d'attente a un serveur

#### 3.2.1 Description du modèle général de file d'attente classique

Le model de file d'attente classique 3.2 est constitué de trois parts qui sont les processus d'arriver, le processus de service, et la discipline de service . Le processus d'arrive détermine comment les clients entrent dans le système pour se faire servir il peut être poissonnier ou

exponentiel ou autre.

Le processus de service détermine comment les clients seront servis .

La discipline de service est la manière par laquelle l'ordre de service des clients est organisé.

Il existe plusieurs disciplines qui sont :

- **FIFO (first in first out)** : le premier arrive est le premier servi.
- **LIFO (last in first out)** : le dernier arrivé sera le premier servi.
- **Random (aléatoire)** : les clients accèdent au serveur de manière aléatoire, indépendamment de l'ordre des arrivées.
- **Priorité relative** : un client accède au service selon sa priorité. La file est gérée par ordre de priorité de la plus forte a la plus faible.
- **Priorité absolue** : le service d'un client est interrompu lorsqu'un client de priorité supérieure se présente devant la file d'attente. Le client dont ce service est interrompu est remis en tête de la file

### 3.2.2 Notion de KENDALL

Un modèle de file d'attente est totalement décrit selon la notation de Kendall[46]. Dans sa version étendue, un modèle est spécifié par une suite de six symboles :

A/B/s/N/M/D

La signification de chacun de ces symboles est :

- A : nature du processus des arrivées.
- B : nature du processus de service.
- s : nombre de serveurs.
- N : capacité d'accueil de la file d'attente.
- M : taille de la population.
- D : discipline de la file. Dans la description des processus d'arrivée et de service, les symboles les plus courants sont :
- M : loi Exponentielle (memoryless).
- E : loi d'Erlang.
- $\gamma$  : loi Gamma.
- D : loi Déterministe (temps d'inter-arrivées ou de service constant).
- G : loi Générale (quelconque). La forme abrégé : A/B/s signifie que N et M sont infinis.

### 3.3 Description du modèle général de file d'attente avec vacance

Dans les modèles de files d'attente classiques, les serveurs sont considérés toujours disponibles. Toutefois, dans beaucoup de systèmes d'attente réels, les serveurs pourraient devenir non disponibles pour une période de temps à cause d'une variété de raisons. Cette période d'absence du serveur peut représenter le fait que le serveur travaille sur des tâches supplémentaires, une opération de maintenance, ou le serveur prend simplement une pause [47]. Pour analyser ces systèmes, la notion de vacance du serveur a été introduite dans les modèles d'attente pour représenter les périodes d'absence temporaire du serveur. Ces modèles sont appelés modèles de files d'attente avec vacance.

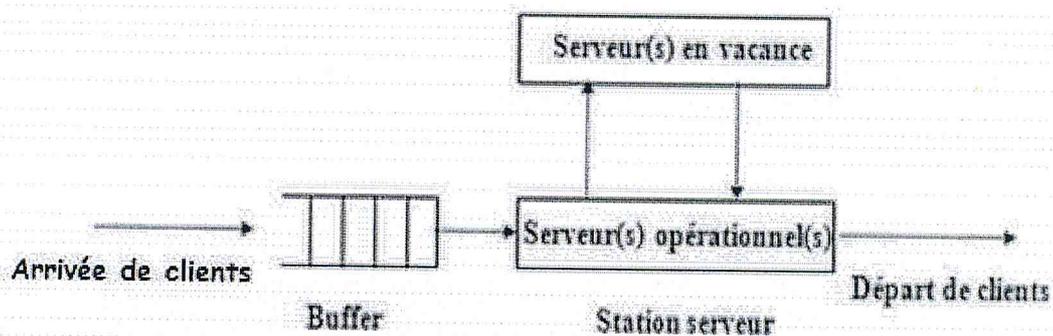


FIGURE 3.2 – File d'attente avec vacance

#### 3.3.1 Les politiques de vacance

Une politique de vacances peut être caractérisée par trois aspects, qui sont : la règle du début de vacance, la règle de fin de vacance et la distribution de la durée de vacance.

##### Règle du début de vacance :

Cette règle indique quand le serveur prend une vacance. Elle est connue aussi sous le nom de discipline de service. Il peut être **un service exhaustif** ou **non-exhaustif**.

- **service exhaustif** : Dans cette discipline le serveur ne peut prendre de vacance que lorsque le système ne contient aucun client c'est à dire qu'il est vide.
- **service non-exhaustif** : Dans ce cas le serveur peut prendre une vacance même si le système n'est pas vide. Parmi les disciplines de service non-exhaustif, on trouve par exemple .
- **le service limite** : Dans un système d'attente avec vacance a service limite, quand le serveur retourne d'une vacance, il sert un nombre  $K$  de client au maximum, puis

débutera une autre vacance.

- **le service avec barrière** : quand le serveur retourne d'une vacance, il sert seulement les clients qui étaient dans le système à son arrivée.

**Règle de fin de vacance :**

C'est la règle qui indique quand le serveur revient de vacance pour servir les clients. Il y a deux règles qui sont la politique de vacance unique et la politique de vacance multiple :

- **la politique de vacance multiple** : le principe est que le serveur prend des vacances successives jusqu'à ce qu'il trouve au moins un client attendant d'être servis dans le système à la fin d'une période de vacance.
- **politique de vacance unique** : Dans ce cas, le serveur prend seulement une seule vacance à la fin de chaque période de service. Après cette vacance, le serveur commencera à servir les clients présents dans le système, ou il reste libre s'il n'y a aucun client.

Pour la durée de vacance. Ils existent des règles générales telles que la politique de seuil (N-vacance) et la politique de T-vacance ainsi que la politique de vacance différencie :

- **La politique de N-vacance** : Elle consiste à ce que le serveur reprend du service quand le nombre de client dans le système atteint la valeur N.
- **La politique T-vacance** : le serveur reste en vacance pendant une période de longueur T. à la fin de cette période il reprend le service.
- **politique de vacance différencie** : Le principe de cette politique est que le serveur peut prendre deux type de vacance différentes (type 1 et type 2). Ces dernières sont multiples distribué suivant la même loi de probabilité avec des taux différents comme décrits dans [48, 49]. Le serveur prend une vacance de type 1 après une période actif ou au moins un client a été servis. D'autre parts il prendra une vacance de type 2 si au retour d'une vacance il trouve le serveur vide.

**Distribution de la durée de vacance :**

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

### 3.4 Vacance d'un capteur sans fil

On sait que dans un capteur sans fil, le module de transmission est le plus gourmand en énergie. Afin de minimiser la consommation en énergie d'un nœud capteur son état actif a été divisé en deux états : état occupé et état de vacance ou veille[50]. Le principe de fonctionnement est que chaque nœud capteur prend des vacances. C'est-à-dire des périodes d'activité réduite pour minimiser sa consommation énergétique. Pour la mise en œuvre de cela. Il est nécessaire d'utiliser deux canaux différents. Le premier sert à la transmission de données, le deuxième est un canal d'alarme pour réveiller le nœud capteur si nécessaire.

L'expéditeur dans ce système fonctionne en deux modes :

**Le mode actif :**

est également appelé état de fonctionnement, dans lequel les trames de données sont transmises normalement.

**le mode vacance :**

Il comprend deux phases, la phase du sommeil et de la phase d'écoute. Dans la phase de sommeil, l'expéditeur éteint le module de communication sans fil, ainsi les trames de données ne seront pas reçues ni envoyées, et l'énergie sera sauvée. Dans la phase d'écoute si certaines trames de données sont détectées à transmettre, l'émetteur va entrer dans le mode actif. Sinon, l'expéditeur sera de retour à l'étape du sommeil.

### 3.5 Travaux connexes

La recherche dans le domaine des réseaux de capteur sans fils en utilisant des files d'attente avec vacance est une activité récente. et vue son importance croissant, de plus en plus de chercheur s'y intéresse. Spécialement dans le but d'optimiser l'énergie au niveau du noeud capteur.

Dans le modèle proposé par Lau et al Dans [51]. Le noeud capteur a été modéliser à l'aide d'une file d'attente M/G/1 avec vacances du serveur.

Le serveur est le support de communication commun entre tous les noeuds capteurs, tandis que l'agrégation de tous les buffers des noeuds est considérée comme un buffer des noeuds est considéré comme un buffer centralise de capacités infinie.

Tous les noeuds sont synchronisés à prendre des vacances en même temps, en éteignant leur unité de transmission. Les auteurs ont donnés des résultats analytiques de la latence des paquets et le pourcentage du temps passe à l'état occupe par les noeuds dans le cas d'une politique de service exhaustif et d'une politique de service avec barrière.

Jiang et al [52] ont utilisés la théorie des files d'attente standards M/G/1 avec une politique de vacance de type N-vacance pour optimiser la consommation d'énergie. Dans leur modèle, ils ont considérés des noeuds capteur avec des buffers a capacité illimitée, et un schéma de communication plusieurs-a-un (many to one), dans lequel un groupe de noeuds communique seulement avec un noeud spécifique, avec un seul saut.

Ce noeud peut être soit libre, si la fonction de transmission est désactivée. Occupe, quand la transmission est activée, et démarrage, qui est un état transitoire entre l'état occupe et l'état libre. Initialement, le noeud est en état libre démarrage, qui représente le temps nécessaire pour octroyer le canal radio. Après cette période, le noeud passe à l'état occuper et commence la transmission des paquets stockés dans le buffer. Dès que le buffer est vide, le noeud repasse à l'état libre. Les auteurs ont donné des fonctions mesurant la consommation d'énergie et la latence des messages.

Par la suite, les mêmes auteurs ont repris le même modèle avec quelque changement et on utilisant la politique de vacance  $\text{Min}(N,T)$  plutôt que la politique de vacance  $N$ -vacance[53]. Ils ont considéré seulement les états libres et occupés. C'est à dire que le nœud balance entre c'est deux états suivants les règles suivantes :

- Quand le buffer du capteur est vide, le nœud prend des vacances successives de longueur fixe  $T$ .
- Si le nombre de paquets dans le buffer atteint la valeur  $N$ , le nœud se met dans l'état occupé.
- Après la même vacance de longueur  $T$ , si le nombre de paquets dans le buffer n'atteint pas la valeur  $N$ , le nœud se met dans l'état occupé.

En se basant sur les résultats de la théorie des files d'attente  $M/G/1$  avec la politique  $\text{Min}(N, T)$  [54], et les résultats de Gakis [55], les auteurs ont donné des formules des longueurs moyennes des états libres et occupés, ainsi qu'une fonction mesurant la consommation d'énergie au niveau du nœud.

Dans[56] les auteurs ont appliqué à leur modèle la politique de  $N$ -vacance en définissant un seuil d'activation de la file d'attente. Ils ont développé un modèle analytique pour l'analyse de la performance du système en termes de consommation moyenne d'énergie et de temps de latence des paquets. Ils ont appliqué cela sur l'état actif du capteur en définissant deux états. Comme cela a été le cas dans[56]. Les hypothèses suivantes ont été faites pour l'analyse :

- Tous les nœuds de capteurs dans le réseau sont identiques
- Le taux d'arrivée correspond aux données générées
- L'arrivée des paquets de données vers des nœuds de capteur est Supposé suivre un processus de Poisson (taux d'arrivée par nœud ( $\lambda$ )).
- Les paquets sont livrés à partir du nœud de capteur (temps de service ( $\frac{1}{\mu}$ )).
- Il n'y a pas de conflit de canal
- Le tampon est supposé être vide chaque fois que le nœud Change de l'état sommeil à l'état actif.

L'auteur a par la suite donne les formules permettant de calculer la probabilité que le capteur soit dans l'état occupé, le nombre moyen de paquet dans le buffer et le temps moyen d'attente des paquets dans la file d'attente. Aussi l'auteur a calculé la consommation moyenne en énergie du capteur selon le principe que l'énergie est consommée durant sa période d'activité lorsqu'il y a :

- arrivée des paquets de données
- La transition de l'état IDLE à l'état BUSY
- synchronisation
- Transmission des paquets
- La transition de l'état BUSY à l'état de repos

Après avoir testé son modèle selon les paramètres décrits dans [19], l'auteur a conclu que en utilisant le seuil optimal  $N$ . la consommation en énergie est réduite de 68% .

Les auteurs de cet article[57] ont voulu mesurer l'impact de la politique de vacance  $N$ -vac sur le gain en énergie et les performances du réseau. Effectivement, cette politique peut s'avérer être très efficace pour économiser l'énergie mais les auteurs ont aussi voulu étudier son impact sur le délai d'attente des paquets dans la file d'attente et la qualité de service ( $Qos$ )

du réseau. Ils cherchent à trouver une valeur du seuil  $N$  qui donne un bon gain d'énergie en même temps, le réseau reste performant. Les auteurs se sont basés sur l'approche d'optimisation d'énergie de la politique  $N$ -vacance présente dans [58, 56].

Celle-ci est une approche pour économiser l'énergie durant l'état actif en définissant deux états, à savoir l'état de veille et l'état occupé. Pendant l'état de veille, des paquets sont tamponnés dans une file d'attente jusqu'à ce que le nombre de paquets atteigne un seuil connu sous le  $N^*$ . Une fois la taille de la file d'attente atteint  $N^*$ , le nœud entre dans l'état occupé où les paquets sont transmis. Les paquets dans la file d'attente sont traités selon l'ordre FIFO. Le capteur envoie tous les paquets présents en file d'attente et même les nouveaux paquets entrants jusqu'à ce que la file d'attente soit vide, après le capteur revient à l'état de veille.

Les auteurs ont donné des fonctions mesurant la consommation d'énergie et la latence des messages et ont calculé le seuil  $N^*$  optimal en ajoutant une contrainte de délai à respecter. Par la suite, ils ont effectué des simulations en variant le débit d'arrivée moyen et le seuil  $N^*$ . Et ont observé l'impact de ces variations sur le nombre de cycles par seconde, la puissance moyenne, la consommation par unité de temps et le retard moyen subi par les paquets. Ils ont conclu que la politique  $N$ -vacance permet de réduire la consommation en énergie mais les mises en attente répétées augmentent la latence du système, cela affecte la qualité du service (QoS).

## 3.6 Modélisation des RCSF à l'aide des RdPSG :

Nous avons utilisé les RdPSG afin de modéliser le comportement d'un capteur générique d'un RCSF. Et nous avons appliqué sur ces modèles les différentes politiques de vacance exhaustives abordées précédemment. Et nous introduisons aussi les différents paramètres de nos modèles. Dans ce qui suit, nous présenterons ces modèles, et nous expliquerons leur comportement.

### 3.6.1 Modèle générique (M0)

Le modèle  $M_0$  est une description du capteur générique du réseau en utilisant les RdPSG. la figure 3.3 représente le modèle dans lequel les différentes places sont :

- Place Source : représente la source des messages.
- Place Dispo : représente la capacité de stockage disponible.
- Place Buffer : représente le nombre de message dans le système en attente de transmission.
- Place Trans : représente la transmission d'un message.
- Place libre : représente l'état de l'unité de transmission.

Ce modèle a pour marquage initial  $M_0$  qui signifie qu'aucun message n'est présent dans le système :

$$M_0 = \{M(Source), M(Dispo), M(Buffer), M(Libre)\} = \{R, K, 0, 1\}$$

Lorsqu'un message arrive est qu'il y a au moins un jeton dans la place Dispo, le message sera stocké dans le Buffer en franchissant la transition Arrivée, et un jeton sera produits dans Buffer. Par la suite si un jeton se trouve dans la place libre et un autre de la place Buffer la transition Trans sera tire. C'est à dire que le message a été transmis et produiras trois jetons : un dans la place Libre décrivant le passage de l'état occupé à l'état libre du module radio, un jeton dans la place Source et un autre jeton dans la place Dispo qui représente, ainsi, la libération d'une place dans le Buffer.

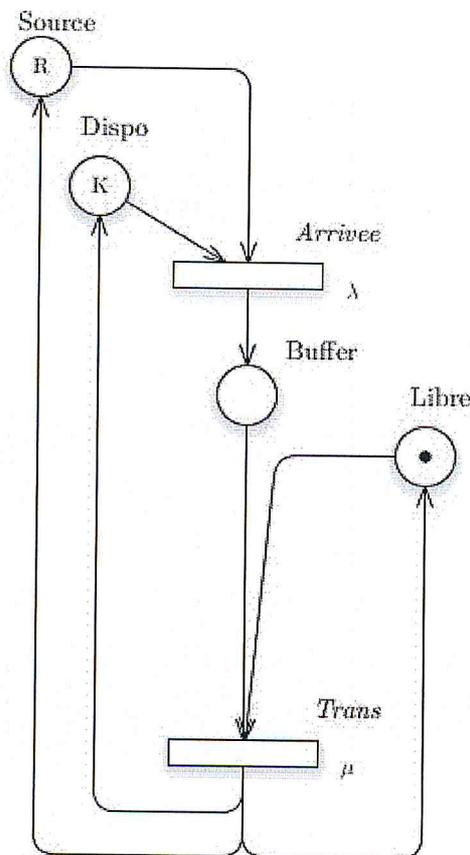


FIGURE 3.3 – Modèle générique (M0)

### 3.6.2 Modèle N-vacance (M1)

Dans ce modèle M1, nous avons repris le modèle  $M_0$  et y'avons ajouté la place Vac, ainsi que deux transitions immédiates d-vac et f-vac et deux arcs pondère a N reliant la place Buffer à f-vac. Dans M1, le service est exhaustif et la politique de vacance est N-vacance. Celle-ci consiste à ce que le serveur retourne de vacance lorsque le seuil N de message en attente dans le Buffer est atteint.

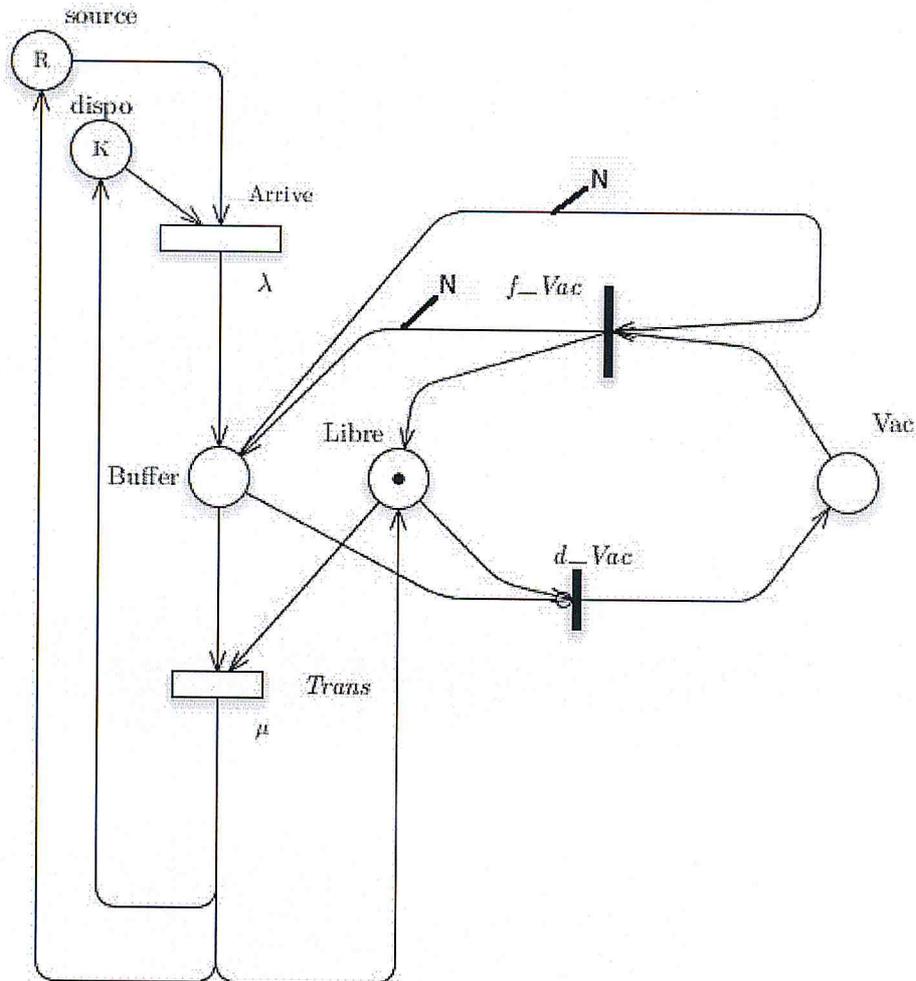


FIGURE 3.4 – Modèle N-vacance(M1)

Dans ce modèle le capteur prend une vacance lorsque la Place Buffer ne contient aucun jeton, c'est à dire qu'il n'y a aucun message à transmettre et que le capteur est libre. La transition d-Vac sera alors tiré le capteur prendra une vacance. Après cela lorsque la place Buffer contiendra N jeton la transition f-Vac sera tiré et le capteur transmettra les N message présent dans le Buffer.

### 3.6.3 Modèle avec vacance multiple(M2)

Le modèle M2 figure 3.5 représente un capteur sans fils qui applique une politique de vacance exhaustive. C'est-à-dire que le capteur prend une vacance de durée aléatoire lorsqu'il n'y a pas de message à transmettre. Et au retour de vacance, s'il trouve un message en attente dans le buffer il le transmettra, sinon il prendra une nouvelle vacance.

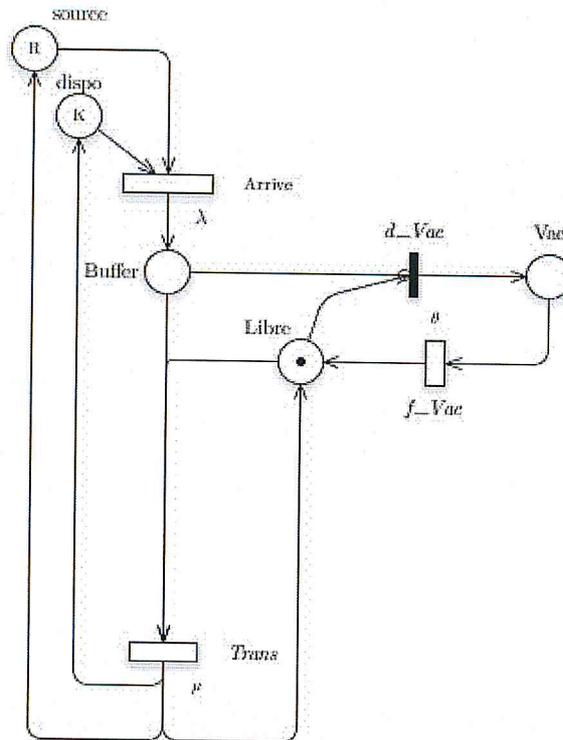


FIGURE 3.5 – Modèle avec vacance multiple(M2)

Pour modéliser cette politique nous avons repris le modèle M0, et nous avons ajouté la place vac ainsi qu'une transition direct d-vac et une autre temporisée f-vac. Dans ce modèle le capteur prend une vacance lorsque la place Buffer ne contient aucun jeton. C'est-à-dire qu'il n'y a aucun message à transmettre et que le capteur est libre, la transition d-Vac sera alors tirée et le capteur prendra une vacance de durée moyenne  $1/\theta$ . Au retour de vacance, si il n'y a aucun message à transmettre dans le Buffer le capteur prendra alors une nouvelle vacance, c'est-à-dire qu'il peut prendre des vacances multiples.

### 3.6.4 Modèle N-vacance multiple (M3)

Dans le modèle M3 nous avons appliqué la discipline de N-vacance, c'est une règle de fin de vacance. C'est à dire que le capteur reprendra son service de deux manières différentes. La première c'est après avoir fini sa vacance d'une durée aléatoire. La deuxième manière c'est un réveil immédiat lorsqu'il y a N messages en attente dans le buffer. Cette discipline permet d'éviter une saturation du Buffer

Pour modéliser cette discipline. Nous avons repris le modèle M1 et nous y avons ajouté une nouvelle transition temporisée f-Vac2 reliant la place Buffer à la place Vac.

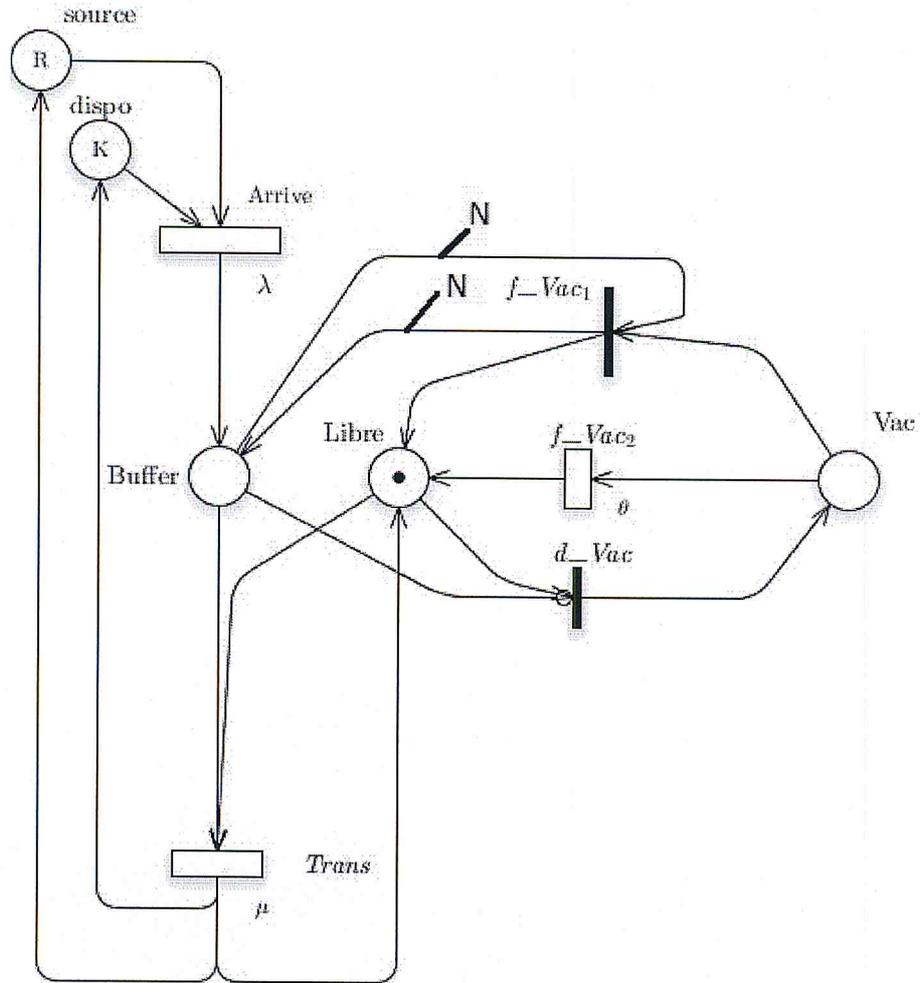


FIGURE 3.6 – Modèle N-vacance multiple et service exhaustive(M3)

### 3.6.5 Modèle avec vacance unique (M4)

Une discipline de vacance unique consiste à attendre que le capteur sert au moins un client avant de repartir en vacance. C'est-à-dire que l'unité de transmission restera en attente active jusqu'à la transmission d'au moins un message avant de pouvoir prendre une vacance.

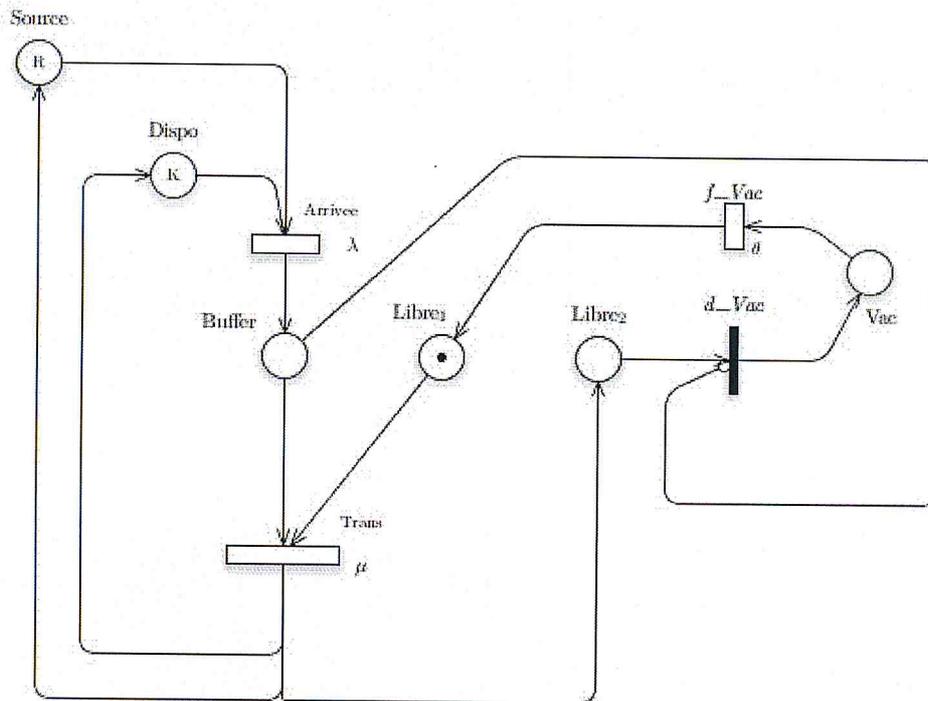


FIGURE 3.7 – Modèle avec vacance unique et service exhaustive(M4)

La modélisation de cette discipline se base sur le modèle N-vacance M3. Pour cela, la place Libre représentant l'état de module de transmission libre est divisée en deux états. La première place désigne une transmission libre qui ne peut que servir. La deuxième place par contre lui donne la possibilité de partir en vacance suivant ce processus : Initialement le jeton disponible dans la place Libre1 représente une transmission libre en attente de service. Lors du service d'une première requête un jeton sera consommé de Libre1 et de Buffer et la transition Trans sera alors tirée et produira un jeton dans la place Libre2. La présence d'un jeton dans cette dernière place représente une transmission libre et qui peut prendre une vacance suivant le même processus décrit dans les modèles précédents.

Le modèle M4 a pour marquage initial :

$$M_0 = M(\text{Source}), M(\text{Dispo}), M(\text{Buffer}), M(\text{Libre1}), M(\text{Libre2}) = \{R, K, 0, 1, 0, 0\}$$

### 3.6.6 Modèle N-vacance unique (M5)

Pour introduire la politique de N-vacance à service exhaustif. Et afin de modéliser M5 figure 3.8, nous nous sommes basés sur le modèle M4. La caractéristique de ce modèle réside dans le retour de vacance de l'unité de transmission (ou la règle de fin de vacance). Ce modèle utilise un retour de vacance hybride, le même utilisé dans le modèle M3, utilisant à la fois une variable de temps de vacance aléatoire et un mécanisme de contrôle sur le nombre de message dans le buffer. Le modèle M5 utilise aussi le même principe que le modèle M4 pour éviter des vacances multiples.

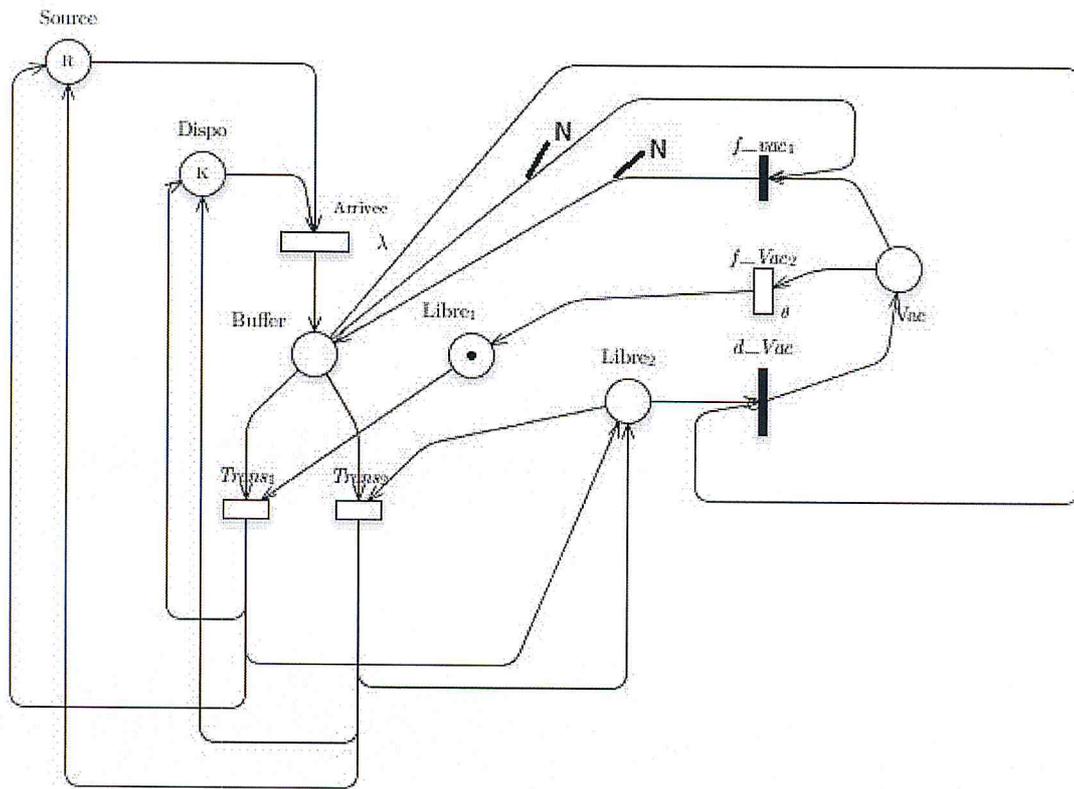


FIGURE 3.8 – Modèle avec service exhaustive à N-vacance unique (M5)

### 3.6.7 Modèle avec vacance différencié (M6)

Le modèle M6 de la figure 3.9 politique de service exhaustive et de vacance différencié c'est-à-dire que le capteur peut prendre deux types de vacances différentes selon le nombre de messages en attente.

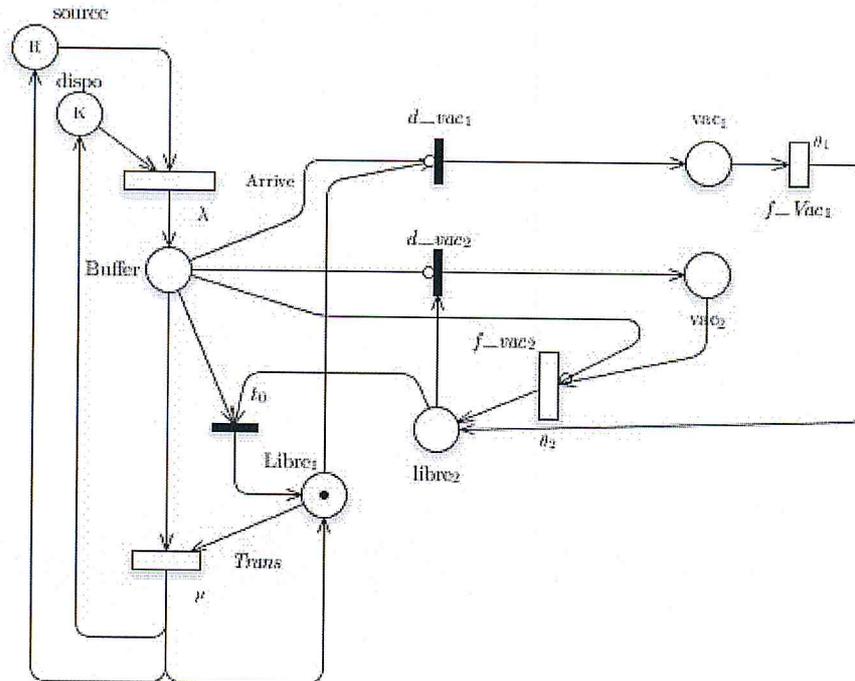


FIGURE 3.9 – Modèle avec service exhaustive et vacance différencié(M6)

Le modèle M6 a pour marquage initial :

$$M_0 = M(Source), M(Dispo), M(Buffer), M(Libre1), M(Libre2), M(Vacance1), M(Vacance2) = \{R, K, 0, 1, 0, 0, 0\}$$

Lorsque la place Buffer est vide et qu'un jeton se trouve dans la place Libre1 la transition d-vac1 sera tirer, ainsi le capteur prendra une vacance de type 1. Après que la transition f-vac1 sera franchie un jeton sera créé dans la place Libre2. Si la place Buffer contient un jeton c'est-à-dire qu'un message est en attente alors la transition t0 sera tirer et un jeton sera créé dans la place Libre1. Sinon si aucun jeton n'était présent dans la place Buffer alors la transition d-vac2 sera tirer et le capteur prendra une vacance de type 2 c'est-à-dire une petite vacance parce qu'il n'a traité aucun message.

### 3.6.8 Modèle avec N-vacance et vacance différencié (M7)

Le modèle de la figure 3.10 M7 permet d'appliquer la politique de N-vacance au modèle précédent(M6). C'est-à-dire que le capteur peut prendre deux types de vacance différente selon la quantité de message transmis et peut interrompre sa vacance lorsque le seuil N1 ou N2 de message en attente dans le Buffer est atteint.

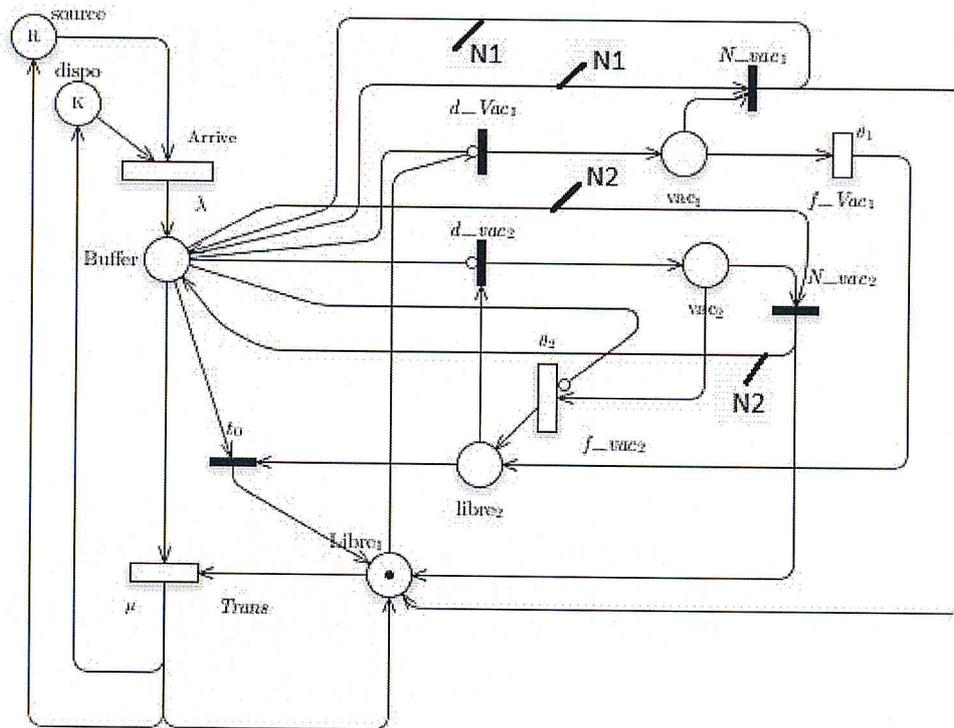


FIGURE 3.10 – Modèle avec N-vacance et vacance différencié(M7)

### 3.7 Les indices de performances

Après la modélisation des différentes politiques de vacance, nous avons développé les formules des principaux indices de performance des différents modèles. Pour cela nous avons utilisé les formules de calcul d'indice de performance d'un RdPSG et les avons adapté à nos modèles.

#### 3.7.1 Modèle générique

- **Nombre moyen de messages dans le Buffer,  $M^*(Buffer)$**  : il correspond au nombre moyen de jetons dans la place Buffer et d'après la formule 2.7 on a :

$$M^*(Buffer) = \sum_{i: M_i \in A} M_i(Buffer) \cdot \pi_i \quad (3.1)$$

- **Nombre moyen de message dans le Capteur** : il correspond au nombre moyen de jetons dans la place Dispo et il est calculé comme suit :

$$M^*(Dispo) = \sum_{i: M_i \in A} M_i(Dispo) \cdot \pi_i \quad (3.2)$$

- **Débit de réception des messages  $F^*(Arrivée)$**  : il correspond à la fréquence moyenne de franchissement de la transition Arrivée et d'après la formules 2.6 on obtient :

$$F^*(Arrivée) = \sum_{i: M_i \in A(Arrivée)} \lambda M_i(source) . \pi_i \quad (3.3)$$

- **Débit de transmission des messages  $F^*(Trans)$**  : il correspond à la fréquence moyenne de franchissement de la transition Trans. Cet indice est calculé comme suit :

$$F^*(Trans) = \beta \sum_{i: M_i \in A(Trans)} . \pi_i \quad (3.4)$$

- **Probabilité de saturation du buffer** : correspond à la probabilité que la capacité du buffer du capteur atteigne la limite N de donnée. On d'autres termes ceci représente la surcharge du capteur. On se basant sur la formule 2.9 on obtient :

$$P_{BP} = \sum_{i: M_i(Dispo=0)} . \pi_i \quad (3.5)$$

### 3.7.2 Modèle avec vacance

- **Temps moyen de la vacance de l'unité de transmission  $Tmoy_v$**  : représente le temps moyen qu'un jeton passe dans la place Vacance.

$$Tmoy_v = \frac{1}{\theta} \quad (3.6)$$

- **Temps moyen passé par un message dans le buffer  $Tmoy_b$**  : représente le temps moyen qu'un jeton passe dans la place Buffer.

$$Tmoy_b = \frac{(M^*(Buffer))}{(F^*(Arrivée))} \quad (3.7)$$

- **Temps de séjour moyen d'un message dans le capteur  $Tmoy_s$**  : représente le temps moyen passé par un message entre l'instant de son arrivé et celui de la fin de sa transmission. Ceci correspond formellement au temps moyen de réponse d'un jeton. Ceci est donné par la formule suivante :

$$Tmoy_s = Tmoy_b + \frac{1}{\mu} \quad (3.8)$$

- **Probabilité que l'unité de transmission soit en vacance  $P_v$**  : correspond à la probabilité que le nombre de jetons dans la place Vac soit égale à un. :

$$P_v = \sum_{i: M_i(vac=1)} .\pi_i \quad (3.9)$$

### 3.7.3 Modèles à N-vacance

- **Probabilité que le seuil N de donné soit atteint P** : Correspond à la probabilité que la capacité du Buffer du nœud capteur atteigne le seuil N :

$$P_v = \sum_{i: M_i(Buffer)=N} .\pi_i \quad (3.10)$$

- **Temps moyen de la vacance de l'unité de transmission Tmoyv**

$$Tmoy_v = \frac{N}{\lambda} \quad (3.11)$$

### 3.7.4 Modèle à N-vacance et service exhaustif

- **Temps moyen de la vacance de l'unité de transmission Tmoyv**

$$Tmoy_v = \min\left\{\frac{N}{\lambda}, \frac{1}{\theta}\right\} \quad (3.12)$$

### 3.7.5 Modèle avec vacance différencie

- **Temps moyen de la vacance de l'unité de transmission Tmoyv** : Cela correspond a la somme des deux vacances

$$Tmoy_v = \min\left\{\frac{1}{\theta_1}, \frac{1}{\theta_2}\right\} \quad (3.13)$$

- **Probabilité que l'unité de transmission soit en vacance Pv** :

$$P_v = \sum_{i: M_i(vac1=1) M_i(vac2=1)} .\pi_i \quad (3.14)$$

### 3.7.6 Modèle avec vacance différencie et N-vacance

- **Temps moyen de la vacance de l'unité de transmission Tmoyv** :

$$Tmoy_v = \min\left\{\left\{\frac{N1}{\lambda}, \frac{1}{\theta_1}\right\} \text{ ou } \left\{\frac{N2}{\lambda}, \frac{1}{\theta_2}\right\}\right\} \quad (3.15)$$

### 3.8 Conclusion

Tout au long de ce chapitre, Nous avons pu voir la notion de vacance d'un capteur sans fil. Cela consiste à mettre en veille l'unité de transmission. Nous avons aussi grâce aux formidables outils de modélisation formelle les réseaux de Petri stochastique généralisé modélise différentes politiques de vacances exhaustives . Politique de N-vacance et celle de vacance différencie. Nous avons aussi développé les formules des principaux indices de performance de nos différents modèles. Ces formules seront implémentées en utilisant l'outil d'analyse GREATSPN afin de calculer les indices de performance. Enfin grâce au modèles réalise et aux indice de performance calculé et à l'outil d'évaluation des performances des RdPSG GreatSPN. Nous allons montrer que l'économie d'énergie est possible grâce au différentes politiques de vacance .



# **Chapitre 4**

## **Test et Expérimentation**

## 4.1 Introduction

Dans ce chapitre, nous ferons l'analyse numérique des modèles de vacance que nous avons réalisés. Par la suite nous étudierons les résultats obtenus. Notre étude sera focalisée sur l'influence du taux de vacance, le taux de réception et des différentes politiques de vacances sur les performances d'un nœud capteur. Les indices de performances étudiées sont : le temps de séjour d'un message dans le Buffer, la fréquence de transmission et la probabilité de saturation du Buffer, et pour finir le gain en énergie. Afin de mener à bien nos expérimentations, nous avons utilisé l'outil de validation et d'analyse des réseaux de Petri stochastiques généralisés GreatSPN 2.0.2 [59]. Cet outil permet d'obtenir des résultats numériques fiables. Cela après avoir saisi les modèles des différentes politiques de vacance étudié et implémenté les formules des indices de performance correspondants.

## 4.2 Description de GreatSPN

GreatSPN (GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets), est un outil software qui permet la spécification, la validation et l'évaluation des performances des systèmes parallèles. Il utilise le formalisme des RdP stochastiques généralisés et leurs extensions colorées. GreatSPN a été développé à l'université de Turin en Italie par un groupe de chercheurs qui travaillent sur les RdPSG et l'évaluation des performances. La première version de GreatSPN a été développée en 1984 depuis il n'a cessé d'être amélioré, sa version la plus récente est 2.0.2. Il est disponible gratuitement et est compatible avec les systèmes Solaris, SunOS et Linux. Le package GreatSPN est composé de plusieurs programmes séparés qui coopèrent pour la construction et l'analyse des modèles de RdP en partageant des fichiers. D'autre part, en utilisant les systèmes de réseaux, différents modules d'analyse peuvent s'exécuter sur différentes machines dans un environnement de calcul distribué. Tous les modules sont écrits en langage de programmation C, pour garantir la portabilité et l'efficacité sur différentes machines. Les principales fonctions et techniques d'analyse implémentées dans l'outil GreatSPN sont :

- Une interface graphique simple à manipuler
- Vérification des propriétés structurelles.
- Génération et analyse du graphe d'accessibilité.
- Résolution markovienne qui permet l'évaluation des performances stationnaires aussi bien que transitoire, en exploitant des techniques numériques matricielles efficaces.
- La simulation : dans ce cas, plusieurs distributions de probabilité sont admises pour les transitions temporisées, telles que : Erlang, Cox, loi uniforme, discrète, etc.

En effet, on peut dire que GreatSPN offre un environnement de modélisation et d'analyse complet. Il permet d'utiliser plusieurs techniques et d'algorithmes d'analyse. Tout cela va nous permettre de faire notre étude expérimentale sur nos modèles.

### 4.3 Le calcul du gain en énergie

La quantité de gain en matière d'énergie dans les modèles M1-M7, est en fonction du temps passer dans la place Vac, et le nombre de transition de l'état actif à inactif, plus ces valeurs augmentent plus le gain est important. La consommation totale en énergie [57] d'un nœud capteur EC est :

$$EC = EC_v.P_{vac} + EC_a.P_{occ} + EC_s.N_c + M^*.EC_t \quad (4.1)$$

Où :

- $EC_v$  : consommation d'énergie alors que la radio est en état de repos.
- $EC_t$  : consommation d'énergie pour transmettre chaque paquet présent dans le système.
- $EC_s$  : représente l'énergie dissipée quand un nœud capteur change de l'état repos à l'état occupé et inversement.
- $P_{vac}$  : est la probabilité que le nœud capteur est en vacances.
- $P_{occ}$  : est la probabilité que le nœud capteur est en état occupé.
- $M^*$  : Représente le nombre de message dans le Buffer en attente d'être transmis.
- $N_c$  : correspond au nombre de changement de l'état on a off du capture qui est calculé comme suit :

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

ou  $C$  : est la durée moyenne d'un cycle qui est donnée par :

$$C = B + T_{moyv}$$

avec  $B$  : le temps moyen d'une période de travail

$$B = \frac{M^*}{\mu}$$

Aussi, la consommation totale d'un nœud capteur  $EC_0$  n'applique aucune politique de vacances est donnée par :

$$EC_0 = EC_a.P_{occ} + EC_s.N_c + M^*.EC_t \quad (4.2)$$

Ainsi, nous pouvons déterminer que le gain d'énergie est égal à la différence entre  $CE_0$  et  $CE_v$ , peut être exprimé en pourcentage de la manière suivante :

$$Gain = \frac{CE_0 - CE_v}{CE_0} \times 100 \quad (4.3)$$

### 4.4 L'étude expérimentale

Notre étude consiste à effectuer des tests sur les différents modèles que nous proposons et discutons ensuite des différents résultats obtenus grâce à l'utilisation de greatSPN.

	R	K	$\theta$	$\mu$	$\lambda$	n
figure4.1	100	10	0.01-0.5	10	3	3
figure4.2	100	10	0.01-0.5	10	3	3
figure4.3	100	10	0.01-0.5	10	3	3
figure4.4	100	10	0.01-0.5	10	3	3
figure4.5	100	10	0.2	10	3	1-5
figure4.6	100	10	0.2	10	3	1-5
figure4.7	100	10	0.2	10	3	1-5
figure4.8	100	10	0.2	10	3	1-5

TABLE 4.1 – Paramètre de réseau

Etat	Consommation (mA)
Vacance	0.6
Actif	6
Transmission	20
Changement d'état(switch)	1

TABLE 4.2 – Paramètres de consommation du capteur

Cette étude expérimental vas servir principalement a observé l'impact de la variation du taux de vacance et de la multiplicité n dans la politique de N-vacance sur la valeur de différents indice de performance qui sont : le temps de séjour d'un message dans le Buffer et le nombre moyen de message dans le Buffer .Ainsi que la variation de la fréquence de transmission des paquets. Nous ferons aussi une comparaison entre les différentes politiques de vacance en observant l'impact qu'elles ont sur le gain d'énergie. Apres avoir testes nos modèles avec GreatSPN. Nous allons dessiner des graphes qui représenteront les résultats obtenu.

Les tableaux 4.1 et 4.2 représentent, pour le premier les valeurs que nous avons utilisés afin de teste nos modèles. Les différents paramètres sont :

- R : Le nombre de message qui arrivent.
- K : Le nombre de places disponible dans le buffer.
- $\lambda$  : Le taux d'arrivé des messages.
- $\theta$  : Le taux de vacance.
- $\mu$  : Le taux de transmission des message.
- n : le seuil de la politique N-vacance.

Le deuxième, montre les différents niveaux de consommation d'énergie relatif aux capteur Mica Mote Sensor.

Nous avons représenté les résultats de nos tests sur des graphes. Les courbes M1, M2, M3, M4, M5, M6, M7 désigne les modèles des différentes disciplines de vacance tels que :

- M1 : modèle avec N-vacance.
- M2 : modèle avec vacance multiple et service non exhaustive.
- M3 : modèle avec N-vacance multiple et service exhaustive.

- M4 : modèle avec vacance unique et service exhaustive.
- M5 : modèle avec N-vacance unique et service unique.
- M6 : modèle avec vacance différencié et service exhaustive.
- M7 : modèle avec N-vacance et vacance différencié et service exhaustive.

Les graphes dans Les figure 4.1, 4.2, 4.3, 4.4, représente l'influence de la variation du taux de vacance  $\theta$  sur le nombre de message en attente dans le Buffer et sur le temps de séjour d'un message dans le Buffer, la fréquence de transmission des paquets, et le gain en énergie.

Le graphe dans la figure 4.1 nous montre l'influence de la variation du taux de vacance  $\theta$  sur le nombre de message présent dans le buffer. On voit que pour le modèle M4 et M1 le nombre de messages dans le Buffer diminue avec l'augmentation de  $\theta$ . Et pour les modèles M3 et M6 que le nombre de messages reste constant et inferieur aux autres modèles. Quand a M7, le nombre de messages est faible et diminue légèrement.

**remarque 1** : le changement du taux  $\theta$  n'influe pas sur le nombre de message dans le Buffer pour les modèles M3, M6, M7. Et la politique M7 nous permettons de minimiser le nombre de messages en attente dans le buffer.

Dans la figure 4.2 le graphe représente l'influence de  $\theta$  sur le temps de séjour du message dans le Buffer. Nous remarquons qu'avec l'augmentation de  $\theta$  le temps de séjour diminue pour M2, M4, M7. Et que pour les autre modèles M3, M5, M6 ce nombre reste relativement constant.

**remarque 2** : le modèle M5 et celui ou le temps de séjour est le plus petit. Mais pour un  $\theta$  supérieur à 0.2 les écarts entre les différents modèles se réduit.

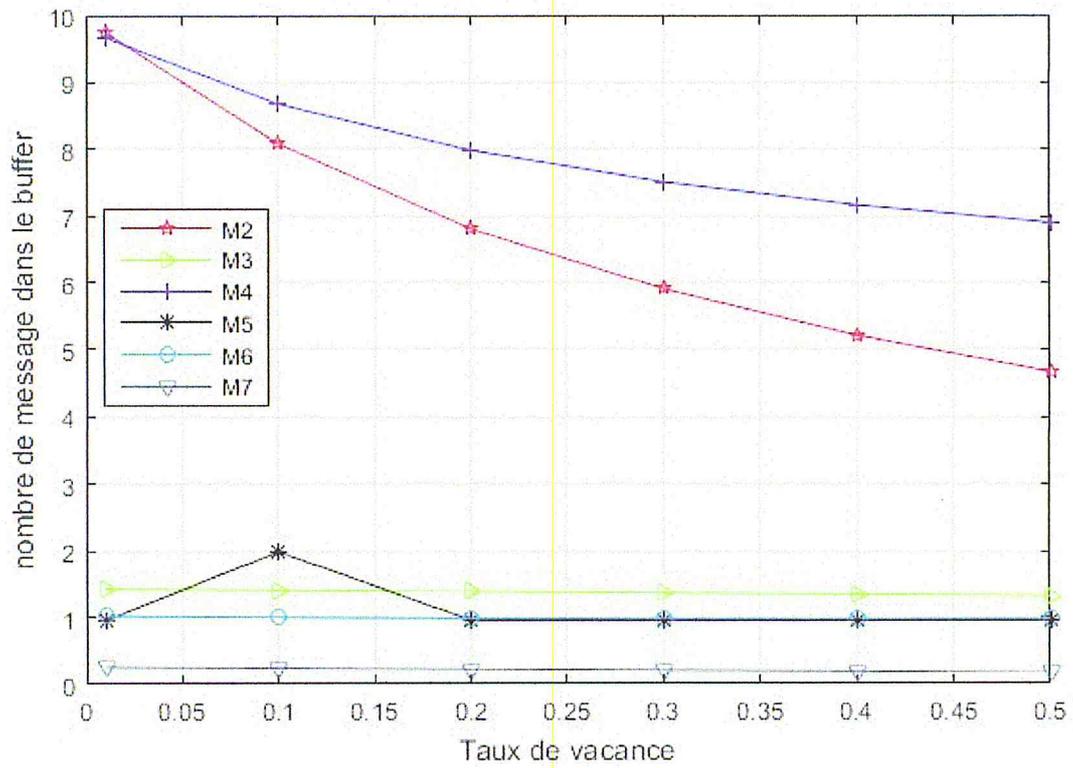


FIGURE 4.1 – Nombre de message dans le buffer en fonction du taux de vacance

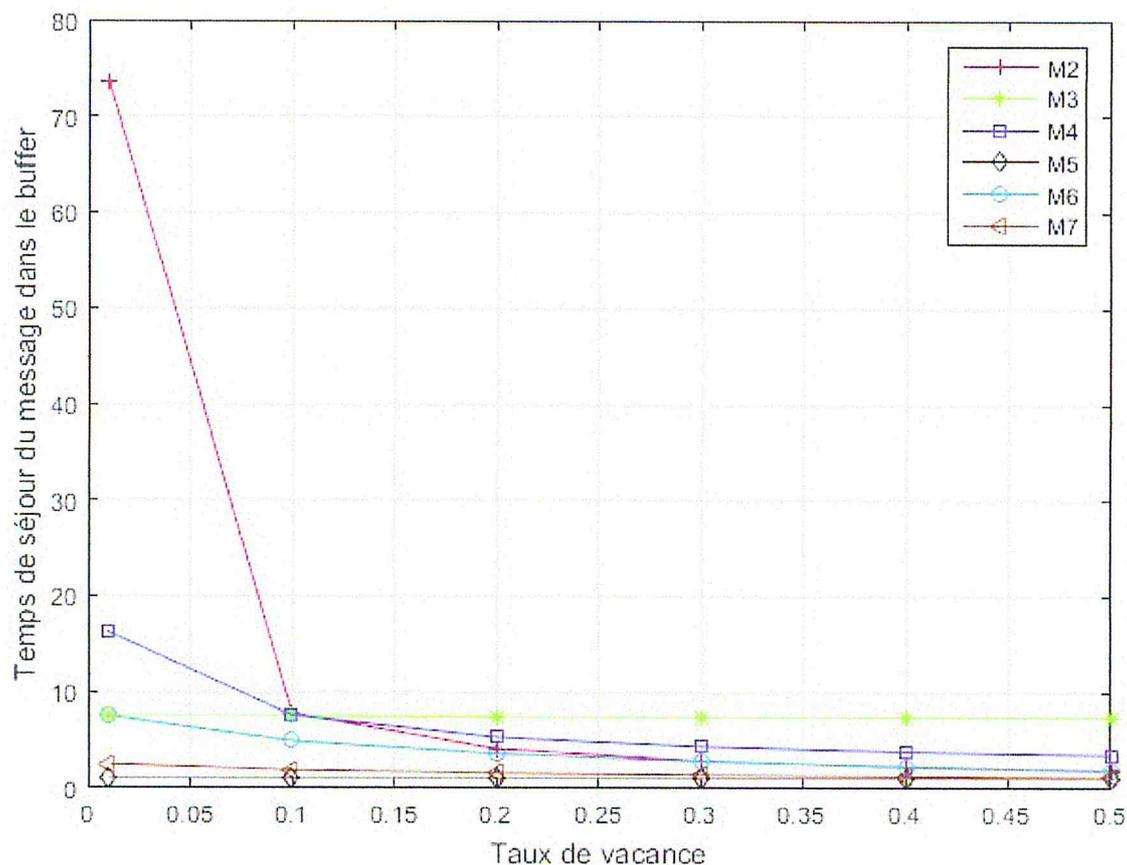


FIGURE 4.2 – Temps de séjour du message dans le buffer en fonction du taux de vacance

dans la figure 4.3, le graphe montre que pour les modèles M2 et M7 la fréquence de transmission augmente avec l'augmentation du taux de vacance  $\theta$ . Alors que pour les autres modèles la fréquence de transmission reste stable (M3) ou augmente légèrement (M4, M5, M6).

Pour le gain en énergie qui est représenté dans la figure 4.4, on remarque que ce dernier varie d'un modèle à un autre. Et qu'en fonction du taux de vacance  $\theta$  le gain augmente (M3, M1), ou diminue (M6).

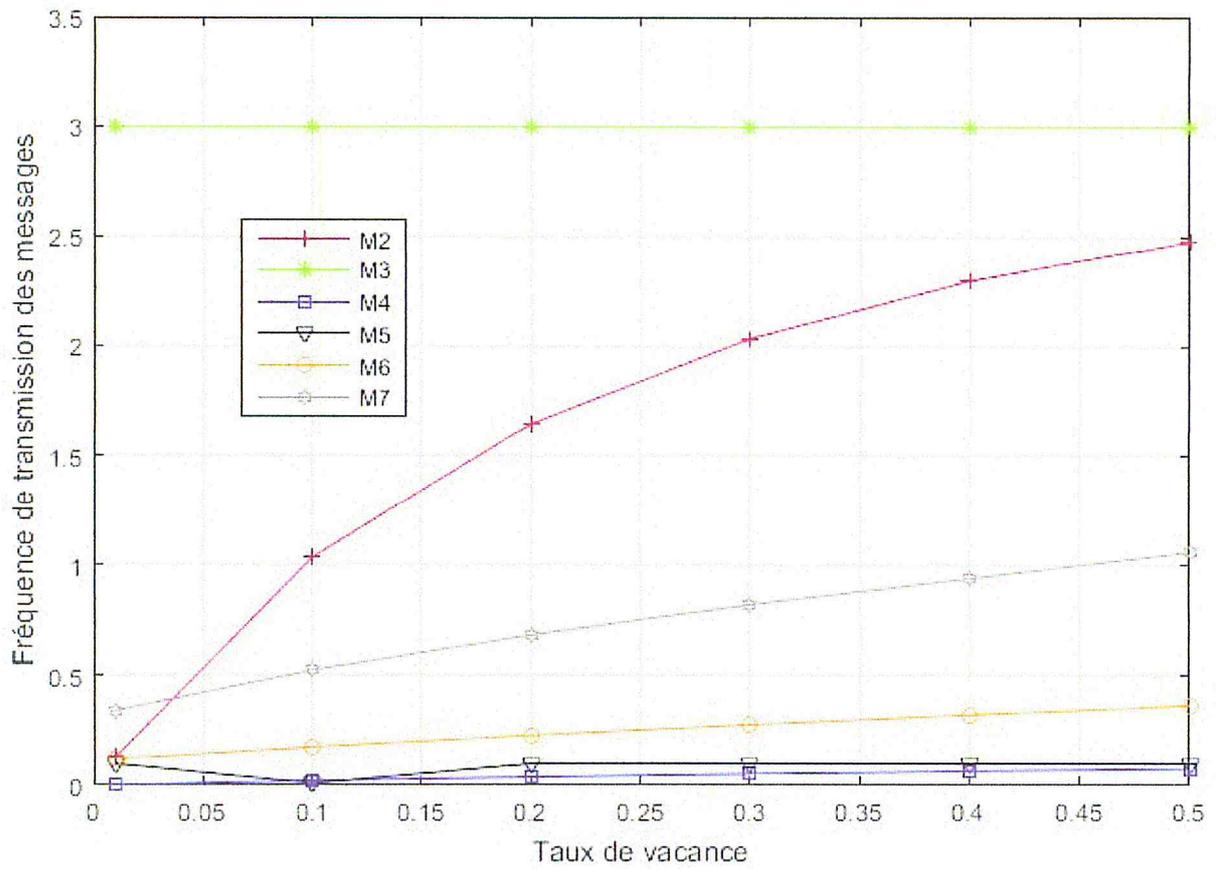


FIGURE 4.3 – Fréquence de transmission des messages en fonction du taux de vacance

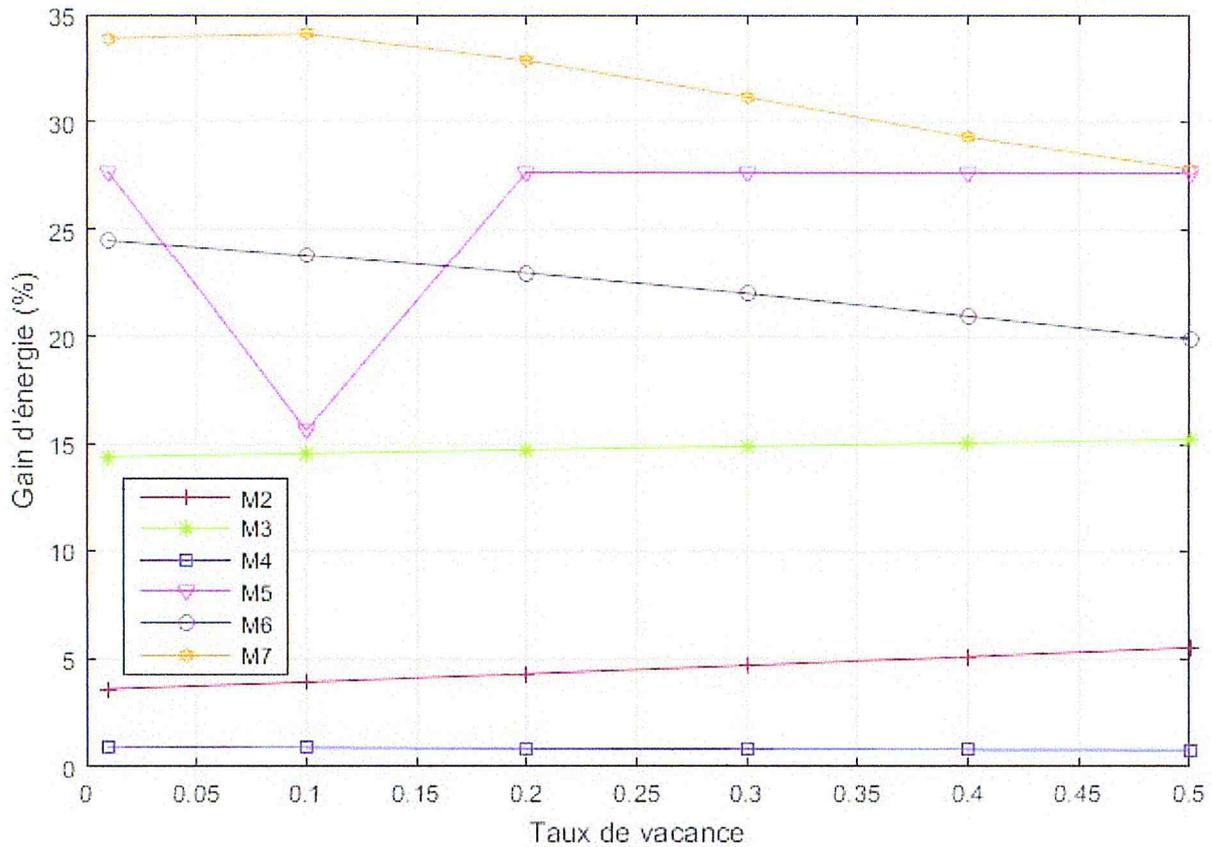


FIGURE 4.4 – Gain d'énergie en fonction du taux de vacance

dans la figure 4.3, le graphe montre que pour le modèle M2 et M7 la fréquence de transmission augmente avec l'augmentation du taux de vacance  $\theta$ . Alors que pour les autres modèles la fréquence de transmission reste stable (M3) ou augmente légèrement (M4, M5, M6).

**remarque 3 :** une fréquence de transmission trop élevée entraîne une surconsommation d'énergie et le contraire engendrera une latence élevée du réseau. Donc le meilleur modèle à choisir est le modèle M5 car il offre un bon compromis.

Le gain en énergie est représenté dans la figure 4.4, on remarque que ce dernier varie d'une politique à une autre. Et qu'en fonction du taux de vacance  $\theta$  le gain augmente (M3, M1), ou diminue (M6, M7). Pour le modèle M5 le pourcentage du gain se stabilise pour  $\theta$  supérieur à 0.2 et le gain de la politique M4 est faible et reste stable.

A partir des remarques 1, 2, et 3. On a constaté que la variation du taux de vacance  $\theta$  influence sur les indices de performance du capteur pour certains modèles. Alors que dans d'autres cas, les indices de performance restent stables malgré la variation du taux de vacance  $\theta$ . Donc selon le modèle utilisé, le choix de  $\theta$  affecte les performances du capteur et par conséquent les performances du réseau.

Nous avons aussi remarqué que le modèle M7 offre un bon compromis entre performance du réseau et gain en énergie .contrairement au modèle M4 qui ne présente aucun avantage a l'utiliser. Quant aux autre modèles M2, M3, M5, M6 chacune présente des avantages a son utilisation ainsi que des inconvénients car cela depend de la valeur de  $\theta$ .

Maintenant, Les graphes dans Les figure 4.5, 4.6 , 4.7, 4.8 représentent l'influence de la variation du seuil  $n$  dans la politique N-vacance sur le nombre de message en attente dans le Buffer et sur le temps de séjour d'un message dans le Buffer, la fréquence de transmission des paquets, et le gain en énergie.

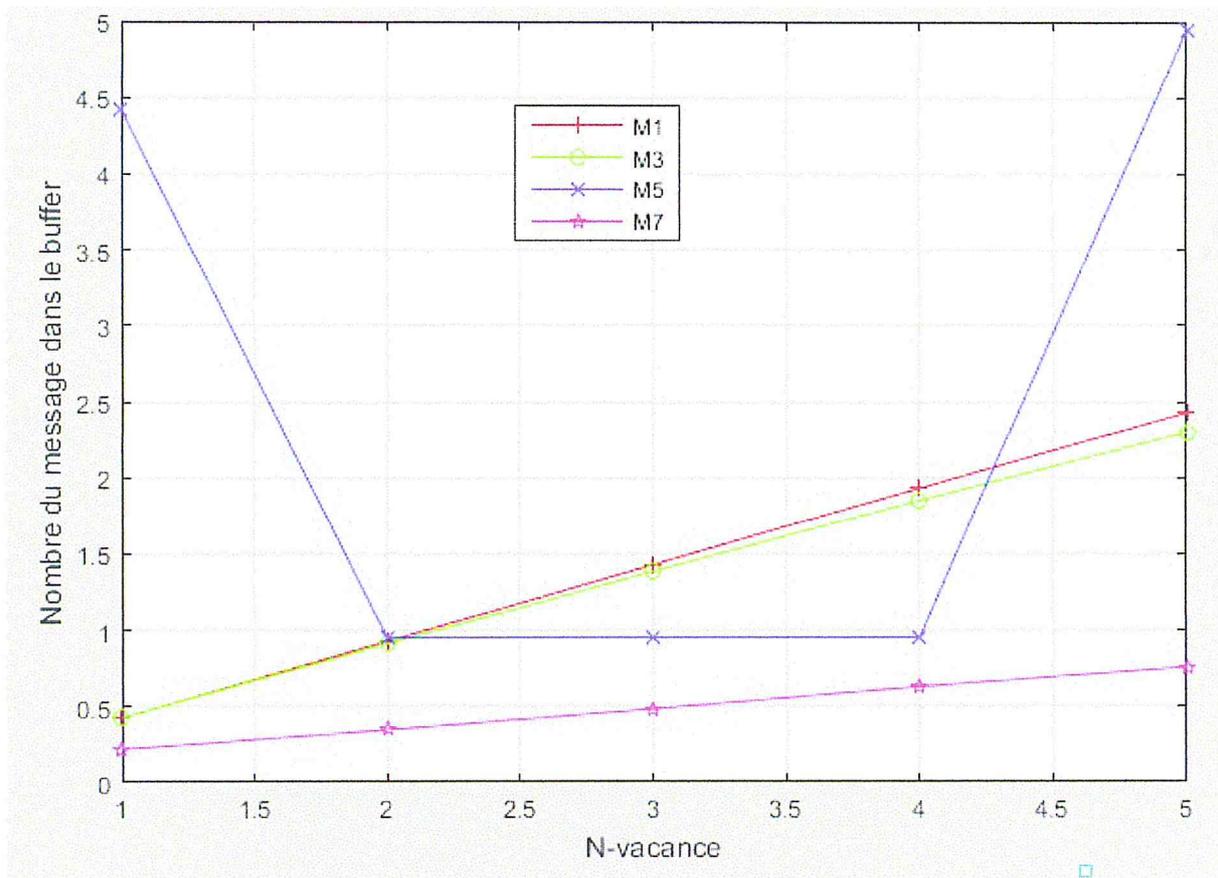


FIGURE 4.5 – Nombre de message dans le buffer en fonction de du seuil de la N-vacance

le graphe dans la figure 4.5 représente l'influence du seuil  $n$  sur le nombre de message présent dans le Buffer. Les courbe des modèles M1, M3, M7 suivent une augmentation constante. Tandis que la courbe du modèle M5 varie selon la valeur de  $n$ .

**remarque 4** : d'après le graphe la meilleure politique c'est M7, parce qu'avec cette politique les messages ne s'accumulent pas dans le Buffer.

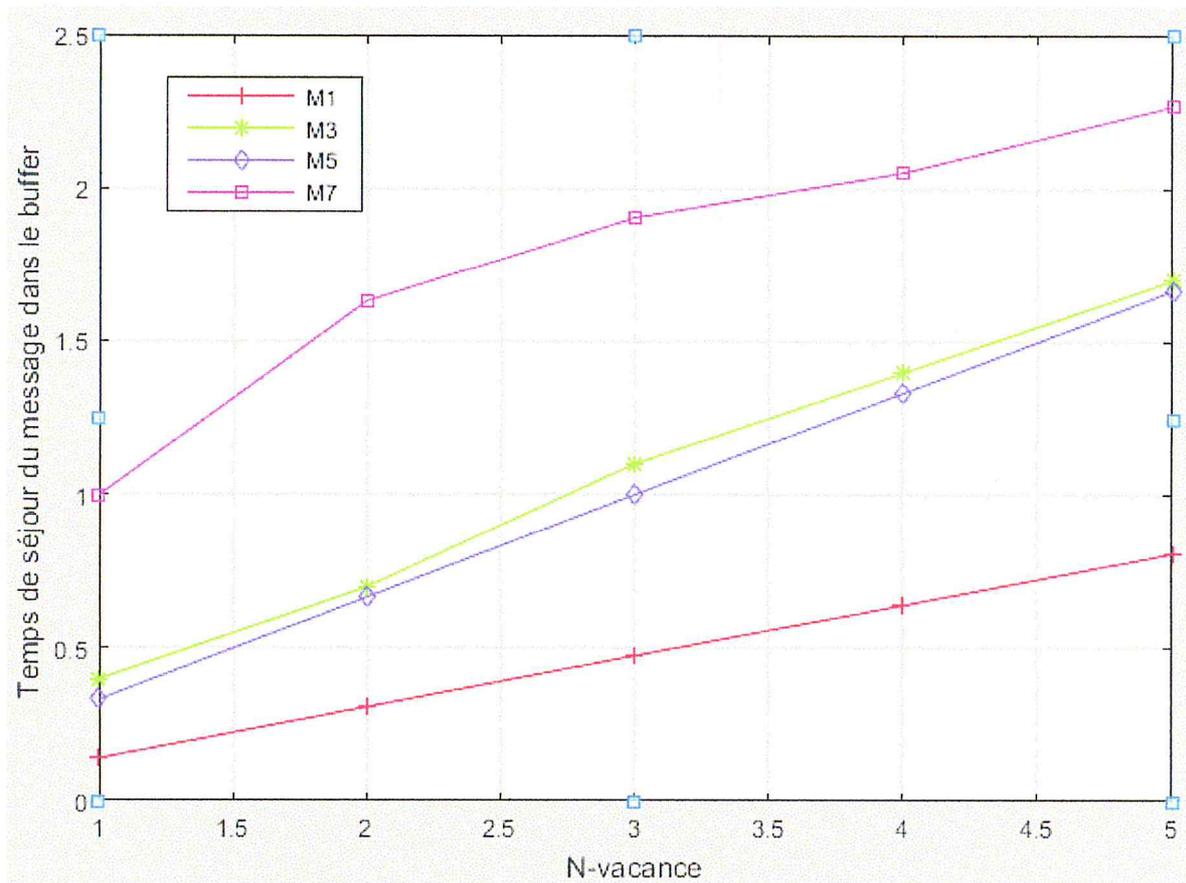


FIGURE 4.6 – Temps de séjour du message dans le buffer en fonction du seuil de la N-vacance

La figure 4.6 représente le graphe du temps de séjour du message dans le Buffer en fonction du seuil  $n$ . on remarque que pour tout les modèles, lorsque le seuil  $n$  augmente le temps de séjour du message dans le Buffer augmente.

**remarque 5 :** le modèle M1 garantie un temps de séjour des messages inferieur par rapport aux autre politiques. C'est-à-dire que le modèle M1 traite les messages stocké dans le Buffer rapidement.

Dans la figure 4.7 le graphe représenté est celui de fréquence de transmission en fonction du temps. On remarque que la fréquence de transmission pour les modèles M1 et M3 reste stable et est plus élevé par rapport aux autres modèles. pour M7 elle augment d'une façon régulière. En revanche, la courbe de de M5 montre que la fréquence de transmission baisse jusqu' a atteindre la valeur 0.098 par la suite elle ne varie plus.

**remarque 6 :** une fréquence de transmission élever implique un réseau performant mais très gourmand en énergie. C'est pour cela que le modèle M7 est le plus intéressante en comparaison avec les autre car, la fréquence de M3 et M1 et très élevé. Quant à M5, pour  $n$  supérieur à 2 elle devient trop basse, donc le temps de latence augmente.

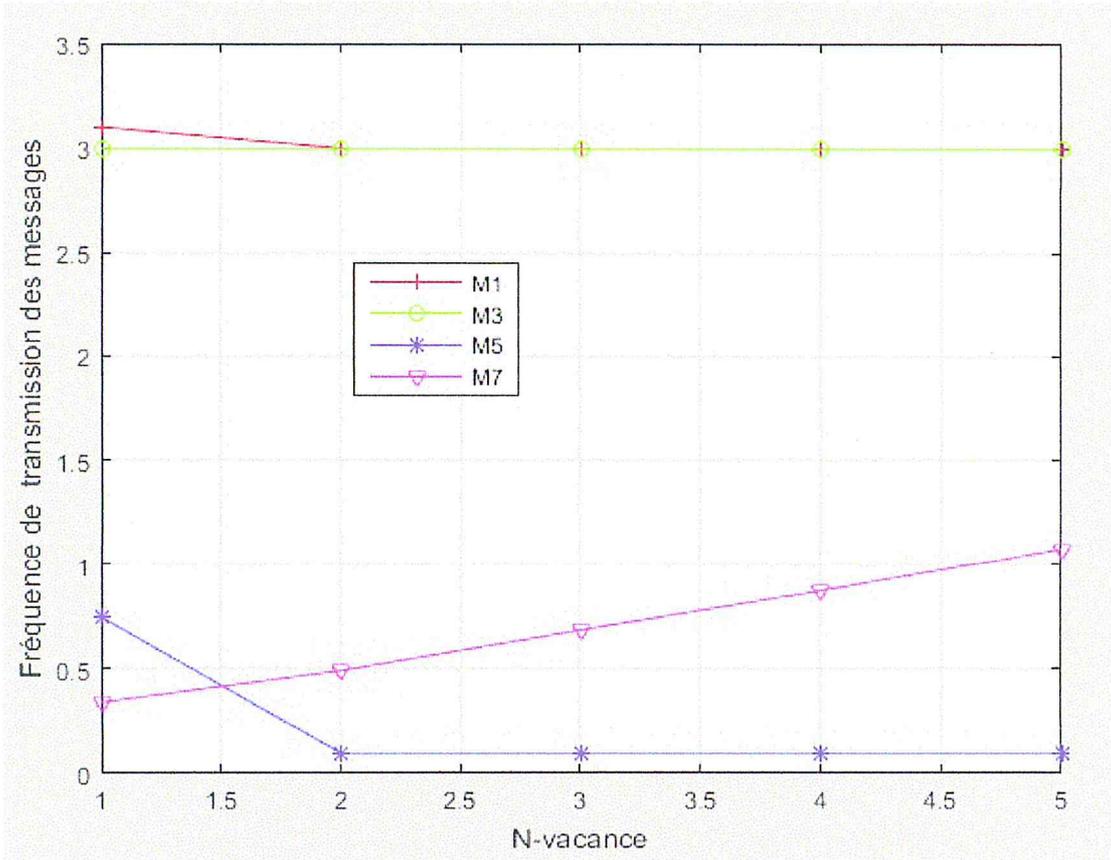


FIGURE 4.7 – Fréquence de transmission des messages en fonction du seuil de la N-vacance

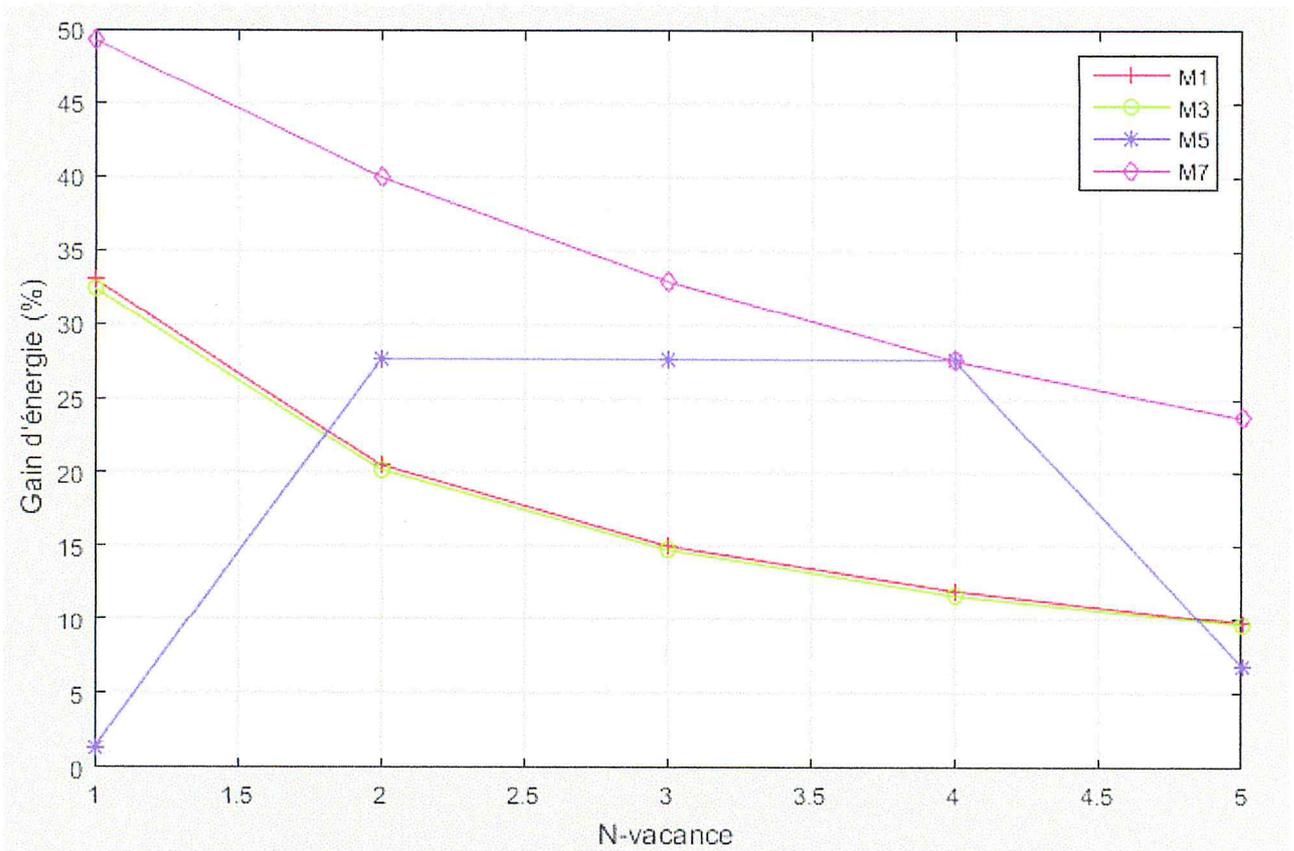


FIGURE 4.8 – Gain d'énergie en fonction du seuil de la N-vacance

Le graphe dans les figure 4.8 représente le pourcentage du gain d'énergie en fonction de la multiplicité  $n$ . on remarque que les politiques M1 et M3 ont le même pourcentage de gain d'énergie et que celui-ci diminue avec l'augmentation de  $n$ . cela est de même pour la politique M7 mais pour des pourcentages plus élevé. Quand a M5 le gain atteint une valeur optimale pour un  $n$  compris entre 2 et 4.

A partir des remarques 4,5 et 6. le choix de la valeur du seuil  $n$  dans les modèles qui utilise la politique N-vacance est très important. On a remarqué que pour certains modèles, leur indice de performance atteint une valeur optimal pour un certain seuil  $n$ . cela vas nous permettre de maximiser les performances du réseau en choisissant les bons paramètres. Aussi, la politique M7 c'est avéré être la plus adaptatif par rapport au changement du seuil  $n$ . cela a étai aussi le cas pour  $\theta$  précédemment. On peut expliquer cela par le fait que M7 combine entre plusieurs politiques.cela la rendu plus polyvalent et moins influençable par le changement de paramètre

Enfin, les modèles M1, M2, M3, M5 présentent chacun des avantages ainsi que des inconvénients a leur utilisation. Ces politique nécessitent d'être bien paramétrés pour maximiser leur rendement.

## 4.5 Conclusion

Tout au long de ce chapitre, nous avons paramétré et calculé les indices de performance de nos modèles en utilisant l'outil GreatSPN pour démontrer que les paramètres du réseau et la discipline suivie influent sur les performances du capteur et par la suite sur les performances du réseau.

Par la suite, nous avons comparé entre les résultats des différentes politiques de vacance, et avons constaté que dans certains cas, certaines politiques présentent de meilleurs résultats que d'autres. Et aussi que la politique de N-vacance différenciée (M7) et celle qui s'adapte le mieux aux différents changements des paramètres réseaux.

Et elle garantit à chaque fois un bon gain d'énergie ce qui implique une bonne durée de vie du réseau. Nous avons aussi observé l'importance du choix des paramètres du réseau. Ces derniers influent sur les indices de performance du noeud capteur.

Au final, tous ces tests nous ont permis de savoir comment améliorer le gain en énergie pour maximiser la durée de vie du capteur. Cela peut être possible en combinant plusieurs politiques. Comme c'est le cas dans M7 qui pour des taux de vacance  $\theta$  et un seuil  $n$  différent. On obtient avec cette politique le meilleur gain d'énergie.

## Conclusion Générale

L'importance des réseaux de capteur sans fil ne cesse de croître et avec cela ces réseaux commencent à être utilisés dans de nombreux domaines variés. Mais ces réseaux subissent différentes contraintes telles que des ressources matérielles très restreintes, comme la capacité de communication, l'espace de stockage, la vitesse de calcul et la ressource énergétique.

Ces contraintes ont rendu nécessaire l'analyse et l'évaluation des RCSF avant tout déploiement. C'est pour cela que mis à part les outils de simulation qui sont la méthode traditionnelle utilisée, l'utilisation des méthodes formelles ayant l'avantage de fournir des résultats exacts. Ceci dit l'obtention de ces résultats n'est pas toujours facile, cela dépend de la complexité du modèle.

Les réseaux de Petri sont un modèle mathématique. C'est aussi un outil graphique et mathématique qui offre une forte capacité de description. Il permet la modélisation et d'évaluation des performances de systèmes complexes caractérisés par la concurrence, l'exclusion mutuelle, la synchronisation et le conflit. Il existe aussi les réseaux de Petri stochastique généralisés qui sont une extension au RdP. Ils ont permis l'utilisation de taux de transition temporisés et immédiats dans un même modèle.

Par la suite, nous avons introduit la notion de vacance dans nos modèles grâce aux réseaux de Petri stochastique généralisés. Cela dans le but d'atteindre notre objectif qui est la minimisation de la consommation d'énergie d'un nœud capteur sans trop affecter les performances du réseau. Et afin d'introduire ces notions, nous avons utilisé diverses politiques de vacance pour paramétrer les départ et retour de vacance du capteur.

L'étape suivante a été les tests et l'évaluation des paramètres de performances des modèles que nous avons implémentés. Cela a été possible grâce à l'outil d'évaluation des paramètres de performances des réseaux de Petri qu'est GreatSPN. Avec cet outil nous avons étudié l'impact du taux de vacance  $\theta$  et du seuil  $n$  dans la politique de N-vacance sur les performances du capteur. Nous avons conclu que le choix de bons paramètres réseau est primordial car cela influe sur les performances du capteur et par la suite sur les performances du réseau. Nous avons aussi calculé le gain en énergie et avons conclu que l'utilisation de politiques de vacance permet d'économiser l'énergie. Mais ces politiques ont besoin d'être bien paramétrées pour maximiser le gain en énergie.

Enfin comme perspective à ce travail, il serait intéressant de réaliser des modèles qui

combinent plusieurs politiques de vacance. Et il serait aussi très utile d'implémenté une application de saisie automatique pour faciliter les tests sur GreatSPN. Et aussi, il serait important de faire des calcules en parallèle en utilisant NS2 et greatSPN. Enfin, une étude plus approfondie sur la consommation exacte des capteurs et leur comportements dans différentes situations, pour que par la suite nous puissions la modéliser.

# Bibliographie

- [1] Malika Ioualalen et al. *Evaluation des performances et de la fiabilité des systèmes multi-classes avec rappel à l'aide des réseaux de petri stochastiques colorés*. PhD thesis, 2007.
- [2] Teerawat Issariyakul and Ekram Hossain. Introduction to network simulator ns2 manual. 2009.
- [3] Andras Varga. Omnet++. In *Modeling and Tools for Network Simulation*, pages 35–59. Springer, 2010.
- [4] Xinjie Chang. Network simulations with opnet. In *Proceedings of the 31st conference on Winter simulation : Simulation—a bridge to the future-Volume 1*, pages 307–314. ACM, 1999.
- [5] Radak Jovan, Pavkovic Bogdan, Nathalie Mitton, Franck Rousseau, and Ivan Stojmenovic. Emulation of large scale wsn : from real neighbors to imaginary destination. In *14èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*, pages 1–4, 2012.
- [6] Martin Maier, Martin Lévesque, and Liviu Ivanescu. Ng-pons 1&2 and beyond : the dawn of the uber-fiwi network. *Network, IEEE*, 26(2) :15–21, 2012.
- [7] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *Communications magazine, IEEE*, 40(8) :102–114, 2002.
- [8] David Culler, Deborah Estrin, and Mani Srivastava. Guest editors' introduction : Overview of sensor networks. *Computer*, (8) :41–49, 2004.
- [9] Kay Römer, Holger Karl, and Friedemann Mattern. Wireless sensor network. In *conference*. Citeseer, 2006.
- [10] Mari Carmen Domingo. An overview of the internet of things for people with disabilities. *Journal of Network and Computer Applications*, 35(2) :584–596, 2012.
- [11] Yann Busnel. Système d'information pair-à-pair pour les réseaux de capteurs larges échelles. *Lettre de la fondation Metivier*, 1(2) :3–5, 2006.
- [12] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks : a survey. *Wireless communications, IEEE*, 11(6) :6–28, 2004.
- [13] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3) :325–349, 2005.
- [14] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.

- [15] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion : a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 56–67. ACM, 2000.
- [16] Arati Manjeshwar and Dharma P Agrawal. Apteem : A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Ipdps*, volume 2, page 48, 2002.
- [17] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing : A recursive data dissemination protocol for wireless sensor networks, 2001.
- [18] Abderrezak Rachedi, Stéphane Lohier, Sylvain Cherrier, and Ismail Salhi. Wireless network simulators relevance compared to a real testbed in outdoor and indoor environments. *International Journal of Autonomous and Adaptive Communications Systems*, 5(1) :88–101, 2012.
- [19] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.
- [20] Jon S Wilson. *Sensor technology handbook*. Elsevier, 2004.
- [21] MA Matin and MM Islam. *Overview of wireless sensor network*. INTECH Open Access Publisher, 2012.
- [22] Vijay Raghunathan, Curt Schurgers, Sung Park, and Mani B Srivastava. Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2) :40–50, 2002.
- [23] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2001.
- [24] Cesare Alippi, Giuseppe Anastasi, Cristian Galperti, Francesca Mancini, and Manuel Rove. Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pages 1–6. IEEE, 2007.
- [25] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3) :493–506, 2004.
- [26] Mahmood Ali, Andreas Bohm, and Magnus Jonsson. Wireless sensor networks for surveillance applications—a comparative survey of mac protocols. In *Wireless and Mobile Communications, 2008. ICWMC'08. The Fourth International Conference on*, pages 399–403. IEEE, 2008.
- [27] Uwe Keller, Rubén Lara, Holger Lausen, Axel Polleres, and Dieter Fensel. Automatic location of services. In *The Semantic Web : Research and Applications*, pages 1–16. Springer, 2005.
- [28] Chander Ramchandani. Analysis of asynchronous concurrent systems by timed petri nets. 1974.
- [29] Ruth Robbins, Sara Gould, and Merlin Bergdoll. Detecting the enterotoxigenicity of staphylococcus aureus strains. *Applied microbiology*, 28(6) :946–950, 1974.

- [30] J Sifakis. Modeling and performance evaluation of computer systems. In *Proc. Third International Symposium on Measuring, Modeling and Evaluating Computer Systems*, (H. Beilner and E. Gelenbe (Eds.)), North-Holland, Amsterdam, pages 75–93, 1977.
- [31] Frederick John Walter Symons. *Modelling and analysis of communication protocols using numerical Petri nets*. PhD thesis, University of Essex, 1978.
- [32] Gerard Florin and Stéphane Natkin. On open synchronized queuing networks. In *International Workshop on Timed Petri Nets*, pages 226–233. IEEE Computer Society, 1985.
- [33] Michael K Molloy. Performance analysis using stochastic petri nets. *Computers, IEEE Transactions on*, 100(9) :913–917, 1982.
- [34] Marco Ajmone Marsan, Gianni Conte, and Gianfranco Balbo. A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems (TOCS)*, 2(2) :93–122, 1984.
- [35] Marc Boyer and Michel Diaz. Multiple enabledness of transitions in petri nets with time. In *pnpm*, page 0219. IEEE, 2001.
- [36] GW Brams. *Reseaux de petri : theorie et pratique*. tome 2. 1983.
- [37] Wael Khansa. *Reseaux de Petri p-temporels contribution a l'etude des systemes A evenements discrets*. PhD thesis, 1997.
- [38] Marco Giuseppe Ajmone Marsan, Gianfranco Balbo, Gianni Conte, Susanna Donatelli, and Giuliana Franceschinis. *Modelling with generalized stochastic petri nets*. 1995.
- [39] Annie Choquet-Geniet. *Les réseaux de Petri : un outil de modélisation : cours et exercices corrigés*. Dunod, 2006.
- [40] Tadao Murata. Petri nets : Properties, analysis and applications. *Proceedings of the IEEE*, 77(4) :541–580, 1989.
- [41] Nawel Gharbi. A numerical approach for performance evaluation of cellular mobile networks with channels breakdowns. *PROCEEDINGS OF INTERNATIONAL CONFERENCE ON NETWORKS (ICN)*, 2013.
- [42] M Ajmone Marsan. Stochastic petri nets : an elementary introduction. In *Advances in Petri Nets 1989*, pages 1–29. Springer, 1988.
- [43] Bruno Lopes, Mario Benevides, and Edward Hermann Haeusler. Extending propositional dynamic logic for petri nets. *Electronic Notes in Theoretical Computer Science*, 305 :67–83, 2014.
- [44] Andrea Bobbio, Antonio Puliafito, Miklós Telek, and Kishor S Trivedi. Recent developments in non-markovian stochastic petri nets. *Journal of Circuits, Systems, and Computers*, 8(01) :119–158, 1998.
- [45] Nawel Gharbi and Leila Charabi. Wireless networks with retrials and heterogeneous servers : Comparing random server and fastest free server disciplines. *International Journal on Advances in Networks and Services Volume 5, Number 1 & 2, 2012*, 2012.
- [46] David G Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics*, pages 338–354, 1953.

- [47] Naishuo Tian and Zhe George Zhang. *Vacation queueing models : Theory and Applications*, volume 93. Springer Science & Business Media, 2006.
- [48] Oliver C Ibe and Olubukola A Isijola. M/m/1 multiple vacation queueing systems with differentiated vacations. *Modelling and Simulation in Engineering*, 2014 :18, 2014.
- [49] Olubukola A Isijola-Adakeja and Oliver C Ibe. M/m/1 multiple vacation queueing systems with differentiated vacations and vacation interruptions. *IEEE Access*, 2 :1384–1395, 2014.
- [50] Shunfu Jin, Wuyi Yue, and Qingtian Sun. Performance analysis of the sleep/wakeup protocol in a wireless sensor network. *International Journal of Innovative Computing Information and Control*, 8(5) :3833–3844, 2012.
- [51] Ka Ming Lau and OnChing Yue. Modeling synchronous wakeup patterns in wireless sensor networks : server vacation model analysis. In *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, volume 2, pages 1312–1316. IEEE, 2005.
- [52] Fuu-Cheng Jiang, Der-Chen Huang, and Kuo-Hsiung Wang. Design approaches for optimizing power consumption of sensor node with n-policy m/g/1 queueing model. In *Proceedings of the 4th International Conference on Queueing Theory and Network Applications*, page 3. ACM, 2009.
- [53] Fuu-Cheng Jiang, Chao-Tung Yang, Kuo-Hsiung Wang, and Der-Chen Huang. design framework to optimize power consumption and latency delay for sensor nodes using min (n, t) policy m/g/1 queueing models. In *Future Information Technology (FutureTech), 2010 5th International Conference on*, pages 1–8. IEEE, 2010.
- [54] Sun Hur, Jongsoo Kim, and Changwook Kang. An analysis of the m/g/1 system with n and t policy. *Applied Mathematical Modelling*, 27(8) :665–675, 2003.
- [55] KG Gakis, Hyun-Ku Rhee, and BD Sivazlian. Distributions and first moments of the busy and idle periods in controllable m/g/1 queueing models with simple and dyadic policies. *Stochastic Analysis and Applications*, 13(1) :47–81, 1995.
- [56] R Maheswar and R Jayaparvathy. Optimal power control technique for awireless sensor node : A new approach. *International Journal of Computer and Electrical Engineering*, 3(1) :37, 2011.
- [57] Aladdin Sleem, Ming Bao, and Wei Wayne Li. Analyzing the impact of the n-policy on quality of service in wireless sensor networks. In *Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference on*, pages 150–154. IEEE, 2014.
- [58] Fuu-Cheng Jiang, Der-Chen Huang, Chao-Tung Yang, and Fang-Yi Leu. Lifetime elongation for wireless sensor network using queue-based approaches. *The Journal of Supercomputing*, 59(3) :1312–1335, 2012.
- [59] Giovanni Chiola, Giuliana Franceschinis, Rossano Gaeta, and Marina Ribaud. Greatspn 1.7 : graphical editor and analyzer for timed and stochastic petri nets. *Performance evaluation*, 24(1) :47–68, 1995.