

MA-004-325-1

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida

N° D'ordre :



Faculté des sciences

Département d'informatique

MEMOIRE DE FIN D'ETUDE POUR L'OBTENTION DU DIPLOME DE MASTER EN
INFORMATIQUE OPTION: GENIE DES SYSTEMES INFORMATIQUE

Thème :

***APPLICATION DES MODELES AVEC VACANCES
NON EXHAUSTIVE POUR LA CONSERVATION
D'ENERGIE DES RESEAUX DE CAPTEURS SANS
FIL***

Sujet proposé par :

Mme. BOUTOUMI BACHIRA

Présenté par :

**Mr. Boualam Mahmoud
Mr. Dahmane Abdelhak**

Soutenu le : XX/XX/XXXX

**Mme. REZOUG NACHIDA Promotrice
Mme. BOUTOUMI BACHIRA Encadrant**

**Promotion
2015 / 2016**

MA-004-325-1

Remerciement

Nous tenons tout d'abord à exprimer toute notre reconnaissance à notre directrice du mémoire madame BOUTOUMI Bachira, nous la remercions de nous avoir encadré, orienté, et aidé à trouver des solutions pour avancer.

Nous voudrions également remercier madame REZOUG Nachida pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion.

Nous remercions tous les membres de notre famille pour leur contribution, leur soutien et leur patience.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Résumé

Les réseaux de capteurs sans fil (RCSF) sont composés de dispositifs électroniques conçus pour la collecte des grandeurs physiques de l'environnement dans lequel ils sont déployés. Ces derniers ont fait naître d'énorme problème de recherche, de par les applications qu'ils permettent d'entrevoir et les nombreuses contraintes qu'ils supposent, la conservation d'énergie dans ces réseaux est très importante. Les capteurs que nous considérons sont alimentés par des batteries non rechargeables. Ainsi, la conservation de cette ressource très limitée au niveau des nœuds capteurs influe directement sur la durée de vie du RCSF. Afin de préserver l'énergie au niveau des nœuds capteurs, nous proposons d'introduire la notion de vacance issue des modèles de file d'attente avec vacance dans leurs comportements. Dans un premier temps nous modélisons un seul nœud capteur, puis deux nœuds qui communiquent entre eux et par la suite on généralise pour la communication de plusieurs nœuds vers un seul nœud. Nous avons optés pour les Réseaux de Petri Stochastique Généralisé (RdPSG) comme outil de modélisation et d'évaluation des performances.

Mots clés : Les réseaux de capteurs sans fil (RCSF), les Réseaux de Petri Stochastique Généralisé (RdPSG), Service non exhaustif, conservation d'énergie, Evaluation des performances , Politique de vacance .

Abstract

Wireless sensor networks (WSN) consist of electronic devices designed to collect physical parameters of the environment in which they are deployed, These have given rise to enormous research problems, because of the applications they allow a glimpse and the many constraints they imply, energy conservation in these networks is very important, The sensors that we consider are powered by non-rechargeable batteries. Thus, the conservation of this very limited resource at the sensor nodes affects directly the lifetime of the WSN, to conserve energy at the sensor nodes, we propose to introduce the concept of vacation outcome of queuing vacation models in their behaviors. First we model a single sensor node. After, we propose models for two nodes that communicate with each other. Then, we generalize to the communication of multiple nodes to a single node, we opted for the Generalized Stochastic Petri Nets (GSPNs) as a modeling and performance evaluation tool.

Key words : Wireless Sensor Networks (WSN), Generalized stochastic Petri nets(GSPNs), non exhaustive service ,Power saving, Performance analysis, Vacation policy.

Table des matières

Introduction générale	1
-----------------------------	---

CHAPITRE 1 Réseau De Capteurs Sans Fil

1. Introduction	3
2. Définition d'un capteur	3
3. Anatomie d'un nœud capteur	4
4. Architecture en couches des réseaux de capteurs sans	5
5. Réseaux de capteurs sans fil	7
6. Topologie d'un réseau de capteurs sans fil	8
7. Problèmes étudiés en RCSF	11
8. Nature du trafic	13
9. Routage dans les RCSF	14
10. Applications des RCSF	16
11. Conclusion	18

CHAPITRE 2 Réseau De Petri

1. Introduction	19
2. Concept de base et définition	19
3. Evolution d'un réseau de Petri	21
4. Situations possibles entre transitions	22
5. Propriétés des réseaux de Petri	23
6. Les réseaux de Petri à arcs inhibiteurs	24
7. Réseaux de Petri stochastiques (SPN)	25
8. Réseaux de Petri stochastiques généralisés (RDPSG)	26
9. Analyse des performances d'un RDPSG	28
10. Conclusion	31

CHAPITRE 3 Politiques De Vacances Et Modélisation

1. Introduction	32
2. Files d'attente	33
2.1 Files d'attente standards	33
2.2 Files d'attente avec vacance.....	35
3. Vacances d'un capteur sans fil	38
4. Travaux connexes	38
5. Modélisation des RCSF à l'aide des RdPSG	40
5.1 Modèles pour un seul nœud capteur	40
5.1.1 Modèle générique (M0)	40
5.1.2 Modèle de vacances multiple aléatoire avec barrière (M1).....	42
5.1.3 Modèle de vacances unique aléatoire avec barrière (M2)	44
5.1.4 Modèle de vacances unique N-vacances avec barrière(M3).....	45
5.1.5 Modèle avec service barrière à vacances multiple adaptative aléatoire (M4)	46
5.1.6 Modèle avec service limité à vacances multiple aléatoire (M5)	47
5.1.7 Modèle avec service limité à vacances unique aléatoire (M6)	50
5.1.8 Modèle avec service limité à N-vacances unique (m7)	51
5.1.9 Modèle avec service limité à vacances multiple adaptative aléatoire (M8)	52
5.2 Modèle pour deux nœuds capteurs.....	53
5.2.1 Modèle de transmission entre deux nœuds avec service limité à vacances multiple (M9)	53
5.2.2 Modèle de transmission entre deux nœuds avec service barrière à vacances multiple aléatoire (M10)	57
5.2.3 Modèle de transmission entre plusieurs nœuds et un avec service limité à vacances multiple (M11)	59

5.2.4	Modèle de transmission entre plusieurs nœuds à un seul avec service barrière à vacance multiple aléatoire	59
6.	Les Indices de performance	60
7.	Conclusion	63

CHAPITRE 4 Test et Résultats

1.	Introduction	64
2.	Description du GreatSPN	65
3.	Mesure de gain	66
4.	Etude expérimentale	67
4.1	Etude expérimental pour les modèles pour un seul nœud de capteur	68
4.2	Etude expérimental pour les modèles à deux nœuds de capteurs (émetteur, récepteur)	78
5.	Conclusion	85
	Conclusion générale	86
	Bibliographie	

Table des figures et des tableaux

Figure 1.1 - Exemples de capteur	4
Figure 1.2 - les composants d'un nœud capteur	4
Figure 1.3 - La pile protocolaire des RCSF	6
Figure 1.4 - Architecture globale d'un réseau de capteurs sans fil [07]	7
Figure 1.5 - Topologie étoile.....	9
Figure 1.6 - Topologie maillée	10
Figure 1.7 - Topologie arbre (cluster-tree)	11
Figure 1.8 - Une taxonomie des applications RCSF [08]	17
Figure 2.1 - Exemple de réseau de petri	19
Figure 2.2 - Exemple de marquage réseau de petri	19
Figure 2.3 - Situations de conflit et de confusion	22
Figure 2.4 - Exemple d'arc inhibiteur	22
Figure 3.1- Système de file d'attente	34
Figure 3.2 - Système de file d'attente avec vacance	35
Figure 3.3 - Modèle générique(M0)	40
Figure 3.4 - Modèle vacance aléatoire multiple avec barrière(M1)	42
Figure 3.5 Modèle de vacance unique aléatoire avec barrière(M2).....	44
Figure 3.6 Modèle de vacance multiple N-vacance avec barrière(M3)	45
Figure 3.7 Modèle service avec service barrière à vacance multiple adaptative aléatoire (M4)	46
Figure 3.8 Modèle avec service limité à vacance multiple aléatoire (M5)	48
Figure 3.9Modèle avec service limité à vacance unique aléatoire (M6)	50
Figure 3.10 Modèle avec service limité à vacance multiple aléatoire (M7)	51
Figure 3.11 Modèle avec service limité à vacance multiple adaptative aléatoire (M8)	52

Figure 3.12 Modèle de transmission entre deux nœuds avec service limite à vacance multiple aléatoire (M9)54
Figure 3.13 Modèle de transmission entre deux nœuds avec service barrière à vacance multiple(M10)57
Figure 3.14 Modèle de transmission entre plusieurs nœuds et un avec service limite à vacance multiple (M11)59
Figure 3.15 Modèle de transmission entre plusieurs nœuds à un seul avec service barrière à vacance multiple aléatoire(M12)60
Tableau 4.1 paramètre du réseau67
Figure 4.1 nombre moyen de paquet dans le buffer en fonction du taux de vacance68
Figure 4.2 nombre moyen de paquet dans le buffer en fonction nombre de vacance successive (J)69
Figure 4.3 nombre moyen de paquet dans le buffer en fonction du nombre de paquet dans le buffer (N)70
Figure 4.4 probabilité de vacance de l'unité de transmission en fonction du taux de vacance70
Figure 4.5 probabilité de vacance de l'unité de transmission en fonction du nombre de paquets dans le buffer71
Figure 4.6 nombre moyen de paquet dans le buffer en fonction de la taille du buffer72
Figure 4.7 nombre moyen de paquet dans le buffer en fonction du taux de d'arrivée73
Figure 4.8 probabilité de vacance de l'unité de transmission en fonction du taux d'arrivée .	.74
Figure 4.9 temps moyen passé par un paquet dans le buffer en fonction du taux d'arrivée .	.74
Figure 4.10 la consommation totale d'énergie en fonction du taux de vacance75
Figure 4.11 gain d'énergie (%) en fonction du taux de vacance76
Figure 4.12 la consommation d'énergie en fonction du nombre de paquet dans le buffer (N)77
Figure 4.13 gain d'énergie (%) en fonction du nombre de paquet dans le buffer (N)77

Figure 4.14 le nombre de paquet moyen dans le buffer en fonction du taux de vacance.....	.78
Figure 4.15 probabilité de vacance de l'unité de transmission en fonction du taux de vacance79
Figure 4.16 temps moyen passé par un paquet dans le buffer en fonction du taux de vacance80
Figure 4.17 la consommation d'énergie en fonction du taux de vacance80
Figure 4.18 gain d'énergie (%) en fonction du taux de vacance81
Figure 4.19 nombre de paquet moyen dans le buffer en fonction du taux de vacance.....	.82
Figure 4.20 la probabilité de vacance de l'unité de transmission en fonction du taux de vacance82
Figure 4.21 temps moyen passé par un paquet dans le buffer en fonction du taux de vacance83
Figure 4.22 la consommation d'énergie en fonction du taux de vacance83
Figure 4.23 gain d'énergie en fonction du taux de vacance84

INTRODUCTION GENERALE

Au cours de ces dernières années, la technologie des réseaux sans fil n'a cessé de croître grâce aux développements technologiques dans divers domaines lié à l'informatique et l'électronique. En plus, avec l'émergence des Réseaux de Capteurs sans fil (RCSF) de nouveaux besoins sont apparus et plusieurs défis ont vu le jour pour répondre aux besoins des personnes et aux exigences de plusieurs domaines d'application (industriel, culturel, militaire et environnemental). Ces réseaux font face à plusieurs contraintes comme la tolérance aux pannes, le passage à l'échelle, la gestion de la topologie du réseau. D'autres contraintes sont liées directement aux nœuds capteurs eux même. Ces petits dispositifs ont des capacités de stockage et une vitesse de calcul très limités. Ils sont équipés également d'une radio à portée réduite et sont alimentés par de petites batteries qui sont dans la plupart des cas non rechargeable. Un plan de gestion efficace de la consommation d'énergie est primordial au niveau des nœuds capteurs afin de prolonger leurs durées de vie ainsi que celle du réseau de capteur sans fil

Vues les contraintes citées précédemment, une conception détaillée et précise des réseaux de capteurs sans fil est impérative, il est même indispensable de les analyser et de les évaluer avant tout déploiement. L'analyse et l'évaluation des réseaux de capteurs se font généralement par simulation grâce aux outils logiciels, tels que NS-2, OMNet++, OPNET, etc. Cependant, l'inconvénient de la simulation est qu'elle donne des résultats différents, en dépendance de l'outil utilise. D'autre part, pour obtenir des résultats proches de la réalité, la simulation devrait prendre un temps d'exécution très long. Une autre alternative, qui reste encore jeune dans le domaine des réseaux de capteur, est l'utilisation des méthodes formelles basées sur des abstractions mathématiques des réseaux, et qui ont l'avantage de fournir des résultats exactes.

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. Ces derniers ont de plus, l'avantage de faire une évaluation des performances des systèmes, en déterminant leurs propriétés quantitatives en plus des propriétés qualitatives.

Dans Notre travail, nous nous intéressons à la consommation d'énergie des capteurs sans fil en essayant de minimiser cette dernière durant leur temps libre. Pour cela nous introduisons la notion de vacance qui consiste à mettre un capteur en état de veille. Nous modélisons donc le comportement des capteurs sans fil avec différentes politiques de vacance et services non exhaustif grâce aux réseaux de Petri stochastique généralisés. Dans un premier plan nous modélisons le comportement d'un seul nœud, dans le second plan nous modélisons le comportement de deux nœuds qui communiquent entre eux et nous clôturons par la proposition d'un modèle pour la communication de plusieurs nœuds dans le scénario plusieurs-à-un (many-to-one). Ce dernier est une généralisation du modèle de communication de deux nœuds.

Ce document est organisé en quatre chapitres. Le premier chapitre est consacré aux réseaux de capteurs sans fil. Dans lequel, nous citerons leurs caractéristiques, leurs contraintes ainsi que quelques exemples d'application. Dans le deuxième chapitre, nous parlons présentons les réseaux de Petri et réseaux de Petri Stochastiques généralisés. Ensuite, Dans le troisième chapitre, nous définirons les politiques de vacances, leurs aspects, ainsi que notre modélisation et notre conception des différentes politiques. Finalement, le quatrième chapitre sera consacré aux principaux résultats obtenus ainsi qu'une comparaison entre les différentes modélisations. Ce mémoire se terminera par une conclusion générale résumant les principaux points abordés, ainsi que quelques perspectives intéressantes à notre travail.

1. Introduction

Les réseaux de capteurs sont de plus en plus utilisés dans l'environnement et l'industrie grâce notamment aux derniers développements réalisés dans le domaine des technologies sans-fils (« Wireless »). Depuis quelques années, le besoin d'observer, d'analyser et de contrôler des phénomènes physiques sur des zones étendues est essentiel pour de nombreuses applications environnementales et scientifiques. Cette nouvelle manière d'envisager la métrologie, en détectant un phénomène à différents points disséminés sur un système ou un site, fait émerger de nouvelles problématiques technologiques, par exemple sur l'autonomie énergétique des capteurs, et de nouveaux types d'applications nous permettant de mieux connaître notre environnement et d'anticiper les problèmes de sécurité, de pollution, de risques naturels, de défaillances, de maintenances, ou plus généralement de tous phénomènes non désirés qui pourraient être anticipés.

Ce premier chapitre est consacré aux réseaux de capteurs sans fil (RCSF). Nous allons présenter les points nécessaires à la compréhension d'un RCSF, tels que : la définition d'un capteur, l'anatomie d'un nœud capteur, l'architecture en couches des réseaux de capteurs sans fils, les différentes topologies d'un réseau de capteurs sans fil, les Problèmes étudiés en RCSF, la nature du trafic, le routage et les domaines d'applications des RCSF

2. Définition d'un capteur

Un capteur est un équipement qui permet de mesurer une grandeur physique observée dans l'environnement qui l'entoure, telle que la température, le taux d'humidité, les vibrations, et qui la transforme en une grandeur numérique. Un capteur seul (qui ne peut pas communiquer avec son environnement) peut permettre des applications locales. Par exemple, un capteur de présence va permettre d'éclairer une lampe lors du passage d'une personne. Lorsque le capteur est équipé d'une carte réseau, il peut alors être mis en réseau, ce qui lui donne une autre dimension [1], [3]. La figure 1.1 donne des exemples de capteurs.

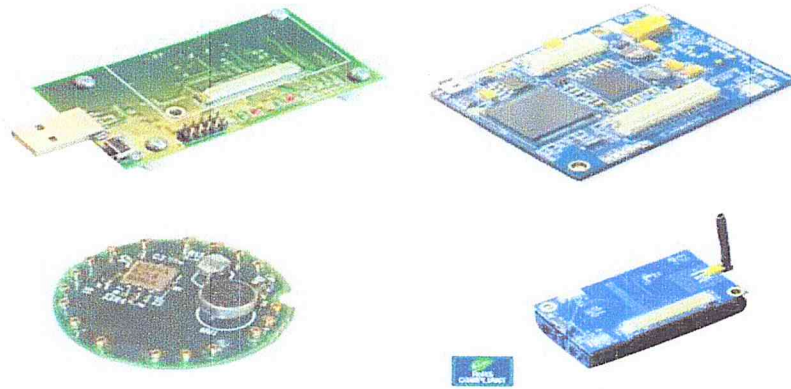


Figure 1.1 - Exemples de capteurs

3. Anatomie d'un nœud capteur

Un nœud capteur est composé de quatre unités principales, qui sont présentées dans la figure I.2 [04] [06] :

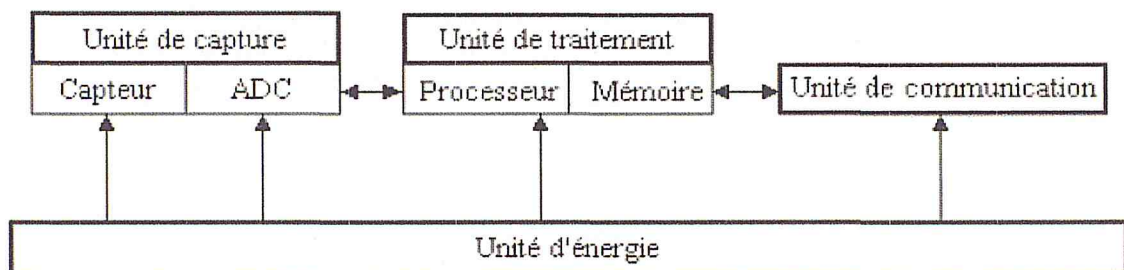


Figure 1.2 - les composants d'un nœud capteur.

- **Unité de capture (*Sensing unit*)** : elle est composée de deux sous-unités, un dispositif de capture physique qui prélève l'information de l'environnement local et un convertisseur analogique/numérique appelé ADC (Analog to Digital Converters), qui fait la conversion de l'information analogique relevée en information numérique et la transmet à l'unité de traitement [06].
- **Unité de traitement (*Processing unit*)** : cette unité est chargée de l'exécution des protocoles de communications qui permettent au nœud de collaborer avec les autres nœuds du réseau pour accomplir les tâches de captage [05]. Elle est composée d'un microprocesseur et d'une unité de stockage de capacité limitée, généralement de moins de 10 Ko de RAM (Random Access Memory) et de moins de 100 Ko de ROM (Read Only Memory) [06], nécessaire à l'implantation et à l'exécution d'un

programme logiciel qui peut être un système d'exploitation spécialement conçu pour les micro-capteurs, comme le système d'exploitation TinyOS par exemple [04],[06]. La mémoire d'un nœud capteur est donc très limitée par rapport à ce que nous pouvons trouver sur les ordinateurs personnels actuels.

- **Unité de communication (*Transceiver unit*)** : elle est composée d'un émetteur/récepteur (module radio) permettant la communication entre les différents nœuds du réseau.
- **Unité d'énergie (*Power unit*)** : 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 [06]

4. Architecture en couches des réseaux de capteurs sans fil

Vu le grand nombre de fonctionnalités implémentés dans les réseaux de capteurs, l'architecture de ces derniers est particulièrement complexe. L'architecture en couches dans les RCSF comme dans les réseaux en général, permet de réduire cette complexité en décomposant les processus qui y sont mis en œuvre. Un tel découpage permet au réseau de traiter en parallèle les fonctions attribuées aux différentes couches. Cette architecture qui est représentée sur la figure 1.3 tient compte des contraintes liées au routage et à la consommation d'énergie. Il est clair que c'est une architecture qui met en œuvre le modèle de référence des communications réseau dit OSI (Open System Interconnexion) avec une réduction de couches selon le type et les caractéristiques des réseaux de capteurs sans fil.

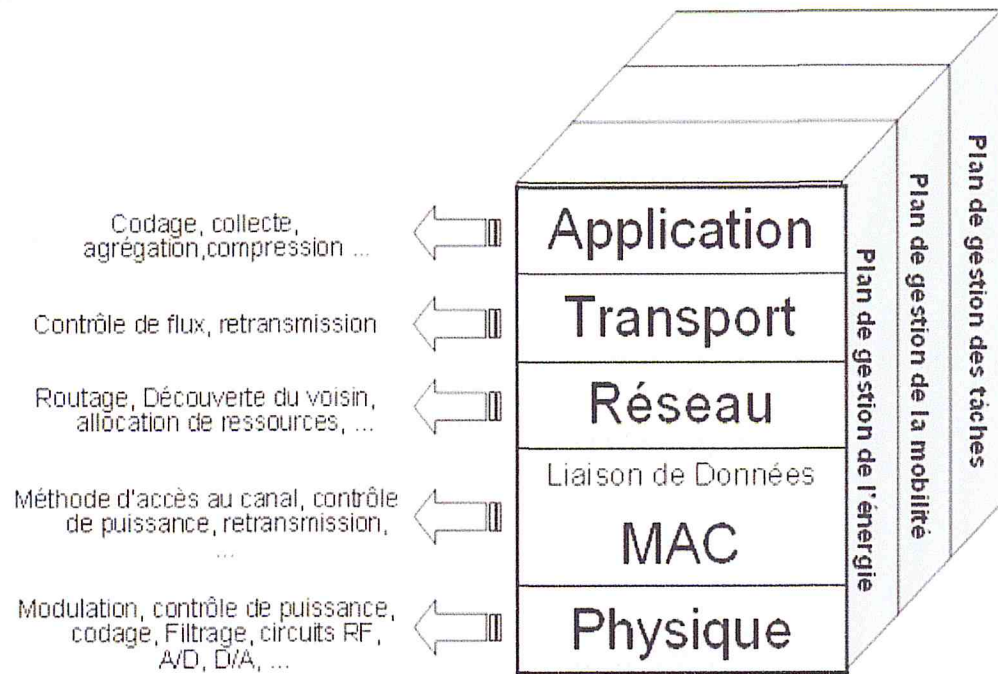


Figure 1.3 - La pile protocolaire des RCSF

Le découpage consiste en une couche physique, une couche liaison de données, une couche réseau, une couche transport, une couche application en plus d'un plan de gestion de l'énergie, un plan de gestion de mobilité et un plan de gestion des tâches. En fonction de l'usage du réseau de capteurs, différents types d'outils exploitant la couche application peuvent être développés. La couche transport permettra de maintenir le flux de données. La couche réseau pourra s'occuper du routage des données qui lui seront présentées par la couche transport. Puisque, les nœuds capteurs sont aussi déployés en grand nombre, la couche liaison de données se chargera d'éviter les collisions qui peuvent être dues aux communications simultanées. La couche physique, quant à elle, assurera les besoins non moins importants de modulation, de réception et d'émission. En plus de ces couches, les plans de gestion de l'énergie, de la mobilité et des tâches gèrent la consommation d'énergie, les déplacements et la distribution des tâches entre les nœuds capteurs. Ils aident les nœuds capteurs à coordonner les tâches de détection et de limiter la consommation d'énergie [07] .

5. Réseaux de capteurs sans fil

Un réseau de capteurs sans fil est constitué de plusieurs nœuds appelés nœuds capteurs ou tout simplement capteurs, permettant de capter et collecter des événements, d'effectuer des traitements et de transmettre les informations recueillies dans différents environnements. Ces nœuds peuvent avoir des positions fixes ou bien être déployés aléatoirement pour surveiller l'environnement. Les communications dans un réseau de capteurs se font souvent d'une manière multi-saut. L'écoulement des données se termine vers des nœuds spéciaux appelés nœuds-puits ou stations de base (Sink). Ces nœuds-puits sont des bases de contrôle qui possèdent plus de ressources matérielles et permettent de collecter et stocker les informations issues des capteurs [03]

Le nœud puits est responsable, en plus de la collecte des rapports, de la diffusion des demandes sur les types de données requises par les capteurs via des messages de requêtes. Notons qu'un réseau de capteurs peut contenir plusieurs nœuds puits diffusant des intérêts différents. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du nœud puit.

Dans un réseau de capteurs sans fil, des points d'agrégation peuvent être introduits. Cela a pour but de résoudre le problème de la consommation d'énergie [02], [03]. En effet, la communication entre les nœuds consomme beaucoup d'énergie, Ainsi, ceci a pour but de réduire cette communication entre les nœuds en privilégiant celle entre les points d'agrégation. Un type de regroupement appelé "clustering" peut aussi être appliqué dans ce cas. Un chef de cluster joue le rôle d'un point d'agrégation. La communication au sein d'un groupe doit passer à travers le chef, qui ensuite la transmet à un autre chef du cluster voisin jusqu'à ce qu'il atteigne sa destination. La figure 1.4 présente une architecture simple de réseau de capteurs sans fil.

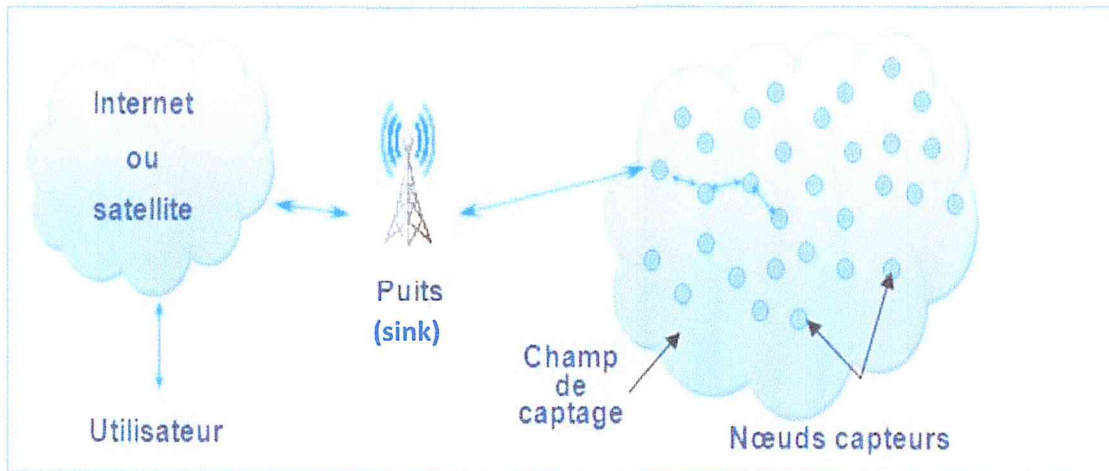


Figure 1.4 - Architecture globale d'un réseau de capteurs sans fil [07]

6. Topologie d'un réseau de capteurs sans fil

La topologie détermine l'organisation des nœuds capteurs dans le réseau. Il existe en général 3 types de topologies du réseau : la topologie étoile, la topologie maillée, la topologie hybride étoile – maillée (cluster).

- **Topologie en étoile :**

La topologie étoile est composée d'un nœud central appelé coordinateur ou puits (sink) du réseau de capteurs et d'une pluralité de nœuds capteurs sans fil (voir figure 1.5). Dans cette topologie, tous les nœuds capteurs transmettent leurs données directement vers le coordinateur. Ainsi, l'architecture d'une telle topologie est caractérisée par une transmission à un saut. Dans cette topologie, le nœud central a la responsabilité de contrôler et de coordonner les nœuds capteurs qui communiquent exclusivement avec lui. Il est en charge de relayer les données vers d'autres systèmes. L'intérêt de cette architecture est qu'elle limite au maximum le transport des données et est donc très performante, en plus d'être relativement simple.

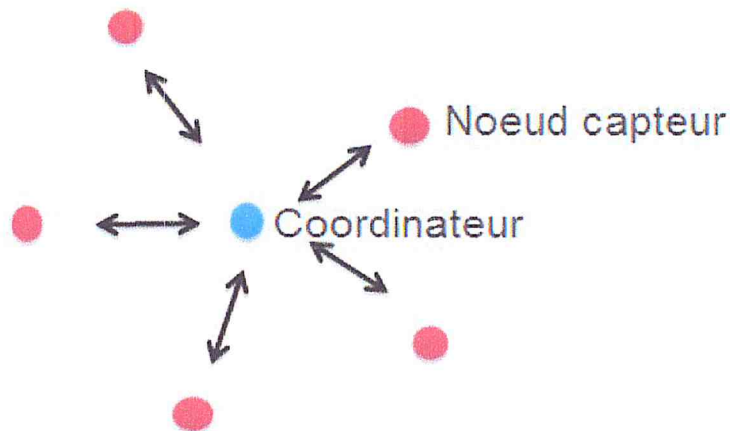


Figure 1.5 - Topologie étoile

La topologie étoile est particulièrement adaptée à deux types d'applications :

- Transmission radiofréquence en champs libre, où très peu d'obstacles sont présents dans la zone de couverture du réseau
- Transmission radiofréquence en dynamique : l'absence de routeur permet de réduire fortement le temps de latence dû aux mécanismes de ré-routage des données. Elle est surtout appliquée en médecine dans le cas de réseau à l'échelle humaine (Wireless Body Area Networks, les nœuds capteurs sont disséminés sur la peau du patient et le coordinateur récolte les données provenant de ces nœuds capteurs.
- **Topologie maillée (Mesh)**
Dans les réseaux de capteurs mobiles classiques, la topologie maillée est la topologie standard. "MANET" ou Mobile Ad hoc NETWORK est le type de topologie maillée le plus utilisé. C'est un réseau distribué sans infrastructure, les nœuds peuvent communiquer entre eux sans contrôle centralisé ni point d'accès.

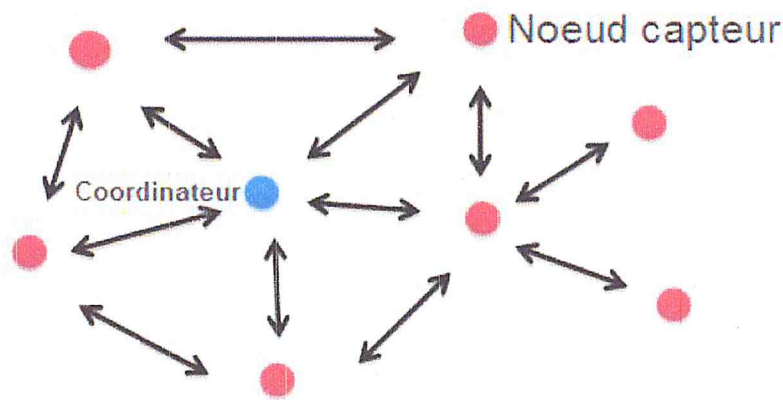


Figure 1.6 - Topologie maillée

La topologie maillée de la figure 1.6 est constituée d'un ensemble de nœuds reliés les uns aux autres et agissant comme des routeurs. D'ailleurs, tous les nœuds possèdent le même rôle. Chacun des nœuds transmet ses données à son voisin après avoir établi le meilleur chemin pour véhiculer les données à travers le réseau jusqu'au puits (coordinateur ou puit (sink)). Cette caractéristique permet de maintenir le fonctionnement du réseau en cas de panne d'un nœud. Cette topologie est alors plus robuste et flexible par rapport à la variation du nombre de nœuds. Quand il n'y a pas de transfert de données, les nœuds se mettent en mode veille pour économiser de l'énergie.

- **Topologie arbre (cluster-tree)**

La topologie cluster-tree (voir 1.7) est partitionnée en groupes appelés "clusters". Un cluster est constitué d'un nœud particulier appelé "cluster-head" ou "tête de cluster" et d'autres nœuds. Ces derniers ne communiquent qu'avec leur "tête de cluster". Ce dernier est ensuite en charge de faire suivre les messages reçus vers le puits du réseau. Cette topologie est alors hiérarchisée selon le rôle des éléments du réseau (coordinateur, têtes de cluster, nœuds). Il peut y avoir un changement de hiérarchie entre les nœuds et les têtes de cluster en fonction de l'énergie disponible dans chaque nœud. Cela permet d'équilibrer la différence des niveaux d'énergie de tous les nœuds afin d'éviter la disparition ou l'isolement d'un nœud et de mieux prolonger la durée de vie du réseau présentée par la suite. Dans les réseaux où l'on tient surtout compte de

l'énergie, plusieurs travaux et recherches comme dans [09] [10] [11] ont affirmé que cette topologie cluster-tree est la mieux adaptée pour équilibrer le niveau d'énergie des nœuds dans le réseau.

En adoptant la topologie cluster-tree, chaque nœud dans le réseau est souvent muni d'un dispositif pour estimer le niveau de batterie du nœud. Ensuite, un protocole spécial doit être installé pour élire la prochaine tête de cluster suivant les niveaux de batterie de tous les nœuds dans le réseau.

La topologie arbre est parfois similaire à la topologie cluster-tree, seulement il n'y a pas de tête de cluster mais ce sont les nœuds simples qui jouent le rôle de relai dans le réseau.

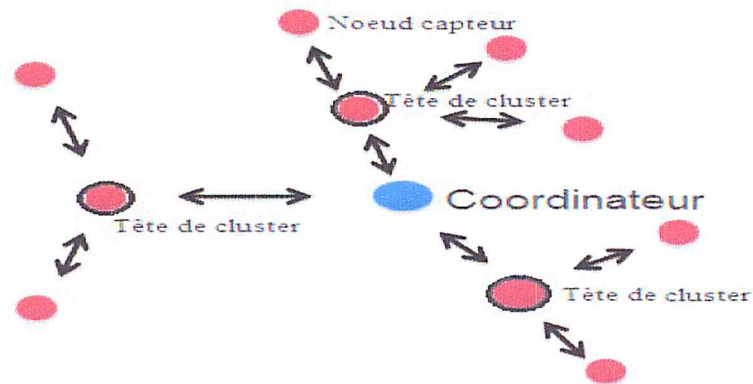


Figure 1.7 - Topologie arbre (cluster-tree)

7. Problèmes étudiés dans RCSF

Les problèmes étudiés dans ce domaine sont multiples, on rappelle ci dessous les principaux axes de préoccupation de la communauté scientifique.

- **Changement d'échelle :** L'augmentation du nombre de nœuds ou la taille du réseau peut entraîner des diminutions de performances au niveau des protocoles. Il peut être intéressant de voir le comportement des protocoles dans de telles conditions. Le défi des réseaux de capteurs soulève une question intéressante quant à leur fiabilité. Comme un réseau de capteurs est conçu pour s'adapter à un environnement en évolution, il émettra plus de paquets de contrôle face à un changement environnemental. À un certain point, le réseau ne pourra plus maintenir la surcharge, et la transmission de données deviendra de moins en moins fiable. En effet, le

passage à l'échelle et la fiabilité sont étroitement couplés et typiquement agissant l'un contre l'autre.

- **Problèmes de routage** : Plusieurs algorithmes et protocoles ont été proposés dans les dernières années, dont le but est d'optimiser la dissémination de données dans les réseaux de capteurs sans fils. Le plus évident est de faire une diffusion de paquets pour arriver d'un nœud au nœud puits(Sink), mais sans aucune maintenance ou découverte de routes, ainsi que la redondance de messages dans le réseau ce qui le charge et réduit sa durée de vie en terme d'énergie. Les stratégies de routage qui fonctionnent principalement en se basant sur des informations locales sont plus appropriées, car elles peuvent être réactives aux changements locaux. De plus, les tables de routage sont distribuées et ne nécessitent pas beaucoup de transfert de données de contrôle.
- **Problème d'énergie** : Chaque entité du réseau consomme de l'énergie, surtout l'émission et la réception des messages. Des nombreux travaux s'intéressent à la réduction de cette consommation. Une des idées est d'offrir une politique d'alternance des communications entre les nœuds. Une autre idée est de réduire la portée radio des nœuds, tout en maintenant une connexité complète du réseau.
- **Sécurité** : Comme dans le cas de réseau Internet, le problème de la sécurité se pose pour les RCSF vu le modèle totalement décentralisé de ce type de réseaux, la sécurité devient un facteur très important. Elle peut se situer au niveau des données échangées, où il est nécessaire d'établir un système qui assure l'arrivé des données sans être espionnées ou modifiées. Ceci consiste à concevoir des protocoles pour assurer la sécurité des communications. La difficulté du problème vient de nombreuses limitations dont le déploiement aléatoire.
- **Extensibilité** : Une fois mise en place les protocoles de bas niveau (routage, découverte de voisinage, ...), il existe aussi l'installation de protocoles au niveau application, dans le but d'exploiter les capacités du réseau. Le but est de pouvoir offrir certaines possibilités comme la qualité de service, l'utilisation d'applications distribuées et la distribution de services à l'intérieur du réseau.

- **Codage et stockage de données** : avec la multiplicité des applications utilisant les RCSF comme par exemple les applications multimédias, les données traitées et/ou transmises par les capteurs sont de taille importante et donc elles sont particulièrement gourmandes en énergie puisque les nœuds engagent des volumes de données très largement supérieurs aux mesures scalaires classiques. Dans ce contexte, et afin de prolonger la durée de vie du capteur multimédia et du réseau entier, le développement de méthodes de compression tenant compte de l'efficacité énergétique est nécessaire pour que le déploiement de réseaux de capteurs puisse être envisagé en pratique.

8. Nature du trafic

- **Trafic périodique**

Dans les réseaux à trafic périodique, les nœuds chargés de la collecte des données transmettent des messages à des intervalles réguliers vers leur(s) destinataire(s). Chaque nœud génère donc des flux intermittents de messages qui lui permettent de passer périodiquement en mode sommeil afin de conserver son énergie. Utilisés dans des applications de supervision, les réseaux périodiques permettent de rassembler suffisamment d'informations sur une zone donnée pour en déduire les tendances et les comportements [02]

- **Trafic apériodique**

Dans un réseau à trafic apériodique, la transmission de l'information à partir des nœuds sources peut être déclenchée de deux façons :

- **Transmission par événement (événement-driven)** : une source ne transmet des données que lorsqu'un événement est détecté dans ses environs. Un événement est décrit comme un changement significatif de la valeur mesurée par le capteur dans le temps. La transmission par événement est généralement utilisée dans des applications de surveillance, comme la détection des incendies et d'intrusion.
- **Transmission par requêtes (on demand)** : dans ce cas, les nœuds capteurs ne transmettent de l'information que lorsqu'un autre nœud du

réseau (potentiellement le puit (sink)) émet explicitement la requête via un message d'intérêt [02].

9. Routage dans les RCSF

Dans les réseaux de capteurs, l'infrastructure peut ne pas être fixe. Dans certaines topologies (maillée, arbre, etc), la participation au routage concerne chaque nœud pour transmettre les informations de la source à un nœud de destination bien précis, par des sauts multiples. Les protocoles de routages sont des applications implémentées dans les nœuds capteurs pour définir le comportement de ces derniers lors des calculs du chemin optimal, du routage de données.

Il existe plusieurs techniques de routage, selon la topologie du réseau et le modèle de trafic.

- **Les protocoles de routages proactifs**

Ils sont globalement basés sur le même principe de routage que dans les réseaux filaires. Ces types de protocoles sont indépendants du modèle de trafic. Dans ce type de routage, les routes sont définies avant la transmission de données, suivant l'état des liens (distances, obstacles, nombre de sauts, énergie restante des nœuds destinataires, etc) entre le nœud émetteur et les autres nœuds. En cas de problème de pertes de données ou erreurs de transmission de bout en bout, on a recourt à la technique de retransmission. Chaque nœud met à jour sa table de routage en échangeant des paquets de contrôle avec les nœuds voisins. En effet, si un nœud veut transmettre des données, il a la possibilité de consulter localement la table de routage et de créer le chemin dont il a besoin. La nécessité de conserver et de contrôler la validité des tables de routages en permanence (comprenant en outre des informations qui ne seront sans doute pas utilisées) est l'un des principaux inconvénients des protocoles proactifs. De plus, si la topologie change, il faut mettre à jour la table de routage toute entière. Par contre, ils présentent l'important avantage de n'introduire aucun délai avant de transmettre un paquet puisque la route est déjà définie.

- **Les protocoles de routage réactifs**

Contrairement aux protocoles proactifs, les protocoles réactifs ne calculent la route que sur demande. Ils tiennent compte de l'évolution des trafics et

beaucoup d'autres paramètres dans le réseau afin de déduire le chemin optimal pour transmettre l'information d'un nœud à un autre.

Ces protocoles sont surtout utilisés dans le cas des nœuds mobiles et quand les liens entre les nœuds varient progressivement. Si un nœud source a besoin d'envoyer un message à un nœud destination, alors il envoie une requête à tous les membres de réseau. Après la réception de la requête, le nœud destination envoie un message réponse qui remonte vers la source cependant, le routage à la demande génère une lenteur à cause de la recherche des routes. Cela peut entraîner une dégradation des performances des applications. Ce type de protocole présente l'inconvénient d'être très coûteux en transmission de paquets lors de la détermination des routes. Par conséquent, il n'y a pas une route prédéfinie, on ne maintient la route que si nécessaire. Ces protocoles ont alors l'avantage de ne pas avoir à maintenir des informations inutilisées dans les tables de routage.

- **Les protocoles de routages hiérarchiques**

La structuration d'un réseau est une technique pour sauvegarder l'énergie dans chaque nœud du réseau. Et cette sauvegarde de l'énergie prolonge la durée de vie du système. Une des structures les plus connues est la structure hiérarchique. Cette technique de hiérarchisation sert à partitionner le réseau en sous ensembles afin de faciliter la gestion du réseau surtout le routage, qui se réalise à plusieurs niveaux. Dans ce type de protocoles, la vue du réseau devient locale, des nœuds spéciaux peuvent avoir des rôles supplémentaires

Après avoir défini la route ou le prochain nœud de destination vers lequel on doit véhiculer l'information, le rôle de la couche MAC est de définir la technique d'accès au canal de communication afin de connecter le nœud émetteur et le nœud récepteur.

- **Protocoles Data centric**

Dans Ces protocoles les nœuds communiquent sans avoir d'adresses ou d'identifiant. La communication s'effectue a travers certaines régions, de proche en proche a travers le réseau. Où la station de base envoie des requêtes a certaines régions et attend les données a partir des nœuds situés dans cette région.

- **Protocoles bases sur la localisation**

Ce type de protocole considère que les nœuds connaissent leur position respective et sont capables de connaître la position des autres nœuds. Il s'agit d'utiliser les coordonnées du nœud source et celles de ses voisins afin de transmettre ou de retransmettre un paquet au plus près de sa destination.

- **Protocoles basés sur la qualité de service**

Dans ce type de protocoles, des métriques de performance du réseau, telles que le délai de bout en bout, la consommation d'énergie, la bande passante, etc., sont prises en compte lors de la transmission de données vers la station de base C'est surtout le cas des applications industrielles et militaires.

10. Applications des RCSF

La taille de plus en plus réduite des micro-capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, vibrations, multimédia, ...) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs sans fil d'envahir plusieurs domaines d'application. Ils permettent aussi d'étendre les applications existantes et de faciliter la conception d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage. Les réseaux de capteurs ont le potentiel de révolutionner la manière même de comprendre et de construire les systèmes physiques complexes. Ils peuvent se révéler très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi les domaines où ces réseaux peuvent offrir les meilleures contributions, nous citons les domaines militaire, environnemental, domestique, santé, sécurité, etc.

Certaines applications militaires utilisent les RCSF afin de contrôler les différentes troupes, de surveiller toutes les activités des forces ennemies ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou des radiations). Les applications pour l'environnement incluent la poursuite du mouvement des animaux, et des insectes ainsi que le contrôle des conditions de l'environnement qui affectent les produits agricoles, l'irrigation, l'exploration planétaire, etc.

L'utilisation des réseaux de capteurs dans le domaine de la médecine pourrait apporter une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité, facilitant ainsi le diagnostic de quelques maladies.

Il est clair que ces domaines d'applications des RCSF ont des spécificités différentes en ce qui concerne la nature des données captées où les nœuds utilisés qui doivent être dotés par de tels capteurs (capteur de pression, température, mouvement, image et vidéo, ...). D'un autre côté, il y a des exigences en ce qui concerne le fonctionnement comme le débit nécessaire, les délais de transmission, la qualité de service, la sécurité, etc. On trouve dans [08] une classification des applications RCSF sous forme de taxonomie avec les performances demandées par chaque type ou domaine d'applications comme le montre la figure 1.9 ci-après.

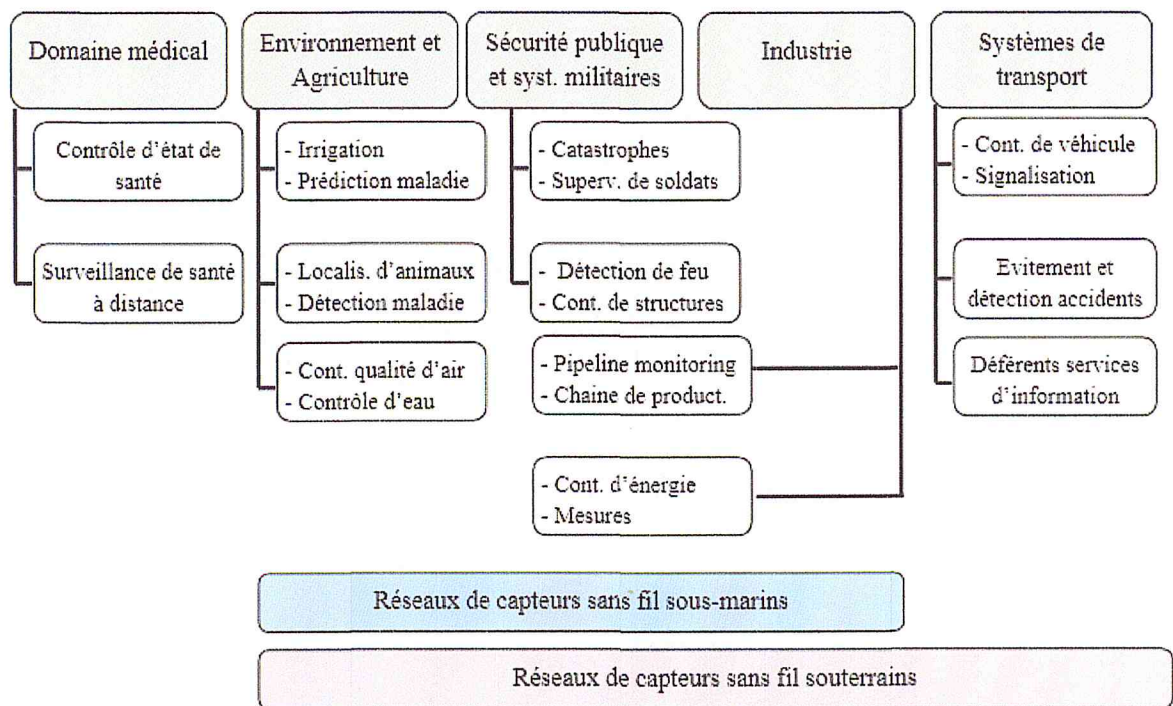


Figure 1.8 - Une taxonomie des applications RCSF [08]

11. Conclusion

Les réseaux de capteurs sans fil présentent un intérêt considérable et une nouvelle étape dans l'évolution des technologies de l'information et de la communication. Cette nouvelle technologie suscite un intérêt croissant vu la diversité de ces applications : santé, environnement, industrie et même dans le domaine sportif. Dans ce premier chapitre, nous avons présenté les réseaux de capteurs sans fil, leurs architectures de communication, la pile protocolaire des capteurs et leurs diverses applications. Cependant, nous avons remarqué que plusieurs facteurs et contraintes compliquent la gestion de ce type de réseaux. En effet, les réseaux de capteurs se caractérisent par une capacité énergétique limitée rendant l'optimisation de la consommation d'énergie dans des réseaux pareils une tâche critique pour prolonger la durée de vie du RSCF

Nous nous intéressons particulièrement à l'optimisation de la consommation d'énergie des nœuds capteur dans les RCSF en utilisant les files d'attente avec vacance et les réseaux de Petri stochastiques généralisés comme formalisme de modélisation. De ce fait, nous allons présenter dans le chapitre suivant, les ingrédients nécessaires à la compréhension de ces formalismes.

1. Introduction

Historiquement, le concept de réseau de Petri a été développé pour la première fois par Carl Adam Petri, un mathématicien Allemand. Ce concept date de 1960-1962. L'auteur a défini un outil graphique et mathématique permettant de décrire les relations existant entre des conditions et des événements. Ils permettent de modéliser le comportement de systèmes à événements discrets et de capturer divers phénomènes qui les caractérisent à savoir le parallélisme, la synchronisation, le partage de ressources, la concurrence, etc. Il est à l'origine du Grafcet (ce dernier étant spécialisé dans la description de la commande de systèmes automatisés)

La notion de temps dans les réseaux de Petri a été introduite par la suite par RAM-CHANDANI[18], MERLIN[16], SIFAKIS[17], ainsi que plusieurs autres propositions avec des approches différentes et basés principalement sur des temps déterministes. Une approche des réseaux de Petri utilisant des temps aléatoires, appelée *Réseaux de Petri Stochastique*, a été introduite par SYMONS[29], FLORIN ET NATKINET MALLOY[30]. Ceci a permis l'introduction des Réseaux de Petri au domaine de l'évaluation des performances. Dans le but d'étendre la puissance de modélisation des réseaux de Petri stochastiques, une extension a été proposée dans (Marsan, Conte, & Balbo, A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems, 1984), où des temps aléatoires sont combinés avec des temps déterministes nuls. Ce formalisme est connu sous le nom de Réseaux de Petri Stochastique Généralisés.

2. Concept de base et définition

- **Définition formelle [22,23]**

Un réseau de Petri est un graphe biparti dont on particularise les deux familles de sommets : les places et les transitions. Comme dans tout graphe biparti, un arc ne relie jamais deux sommets de la même famille. Les places sont représentées par des cercles, tandis que les transitions sont représentées par des traits ou des rectangles.

Un réseau de Petri généralisé est un quadruplet R tel que $R = (P, T, Pre, Post)$

Avec :

- P : ensemble des places du réseau ;
- T : ensemble des transitions du réseau ;

- Pré : application d'incidence avant, de $P \times T$ dans N . $\forall p \in P, \forall t \in T$,
 $\text{Pré}(p, t) = v(p, t)$ si $(p, t) \in G$, 0 sinon.
- Post : application d'incidence arrière, de $P \times T$ dans N . $\forall p \in P, \forall t \in T$,
 $\text{Post}(p, t) = v(t, p)$ si $(t, p) \in G$, 0 sinon.

A chaque application d'incidence est associée une matrice $|P| \times |T|$.

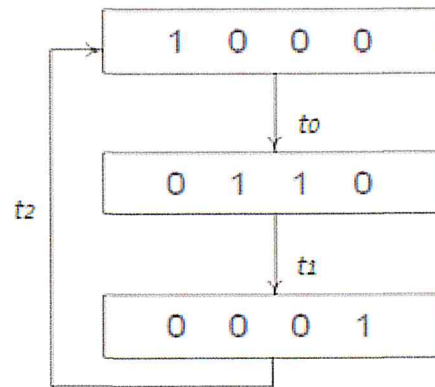
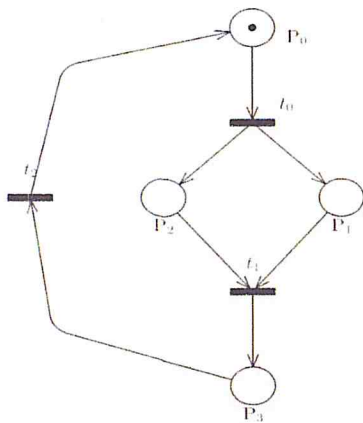


Figure - 2.1 Exemple de réseau de Petri Figure 2.2 - Exemple de marquage réseau de Petri

- **Représentation graphique**

Graphiquement les places sont représentées par des cercles, les transitions par des barres ou de petits rectangles, les arcs par des flèches et les jetons par des points. Une représentation d'un réseau de Petri est donnée dans la figure 2.1.

- **Représentation matricielle**

Le réseau de Petri est représenté par :

- un ensemble de place : $\{p_0; p_1; p_2; p_3\}$
- un ensemble de transitions : $\{t_1; t_2; t_3; t_4\}$
- une matrice Pre qui représente les arcs allant d'une place à une transition. L'élément $(i; j)$ de la matrice Pre est différent de 0 s'il y a un arc de la place i vers la transition j .

- une matrice Post qui représente les arcs allant d'une transition à une place. L'élément $(i; j)$ de la matrice Post est différent de 0 s'il y a un arc de la transition i vers la place j .

A partir de ces deux matrices, on définit la matrice d'incidence du réseau de

$$\text{Petri} : c = \text{post}^t - \text{pre}$$

L'élément $(i; j)$ de la matrice C donne le bilan pour la place i du tirage de la transition j .

3. Evolution d'un réseau de Petri

Les définitions et les notions des RdP vus précédemment représentent leur aspect statique, c'est à dire, la description instantanée du système. Les notions suivantes permettent d'exprimer la dynamique des réseaux de Petri.

Nous utilisons les notations suivantes [24] :

- $t = \{p \in P \mid \text{Pré}(p, t) \geq 0\}$, 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 .

Les règles générales d'évolution des réseaux de Petri marqué simple sont les suivantes :

- une transition est **franchissable** ou **sensibilisée** ou encore **validée** lorsque chacune des places en amont possède au moins un jeton,
- le réseau ne peut évoluer que par franchissement d'**une seule transition à la fois**, transition choisie parmi toutes celles qui sont validées à cet instant,
- le franchissement d'une transition est indivisible et de durée nulle.

4. Situations possibles entre transitions

Lors de l'évolution d'un réseau de Petri, plusieurs situations sont possibles entre les transitions, dont les plus importantes sont : le conflit, la confusion, la concurrence et l'exclusion mutuelle.

- **Conflit** : deux transitions t_1 et t_2 sont en conflit structurel si et seulement si elles 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 [25]. Les transitions t_1 et t_2 de la figure 2.3 sont en conflit effectif.

- **Concurrence** : contrairement au conflit, la concurrence est caractérisée par le parallélisme des activités. En conséquence, deux transitions t_1 et t_2 sont concurrentes dans un marquage M si elles sont toutes les deux sensibilisées dans M , et elles ne sont pas en conflit.
- **Confusion** : une situation de confusion correspond à une situation où le conflit et la concurrence sont présents en même temps. Une situation de confusion est représentée dans la figure 2.3, où t_3 et t_5 sont concurrentes et chacune d'elles est en conflit avec t_4 .
- **Exclusion mutuelle** : deux transitions t_1 et t_2 sont mutuellement exclusives ou en situation d'exclusion mutuelle si elles ne peuvent jamais être sensibilisées en même temps dans un même marquage M . Autrement dit, t_1 et t_2 sont mutuellement exclusives si et seulement si :

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

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

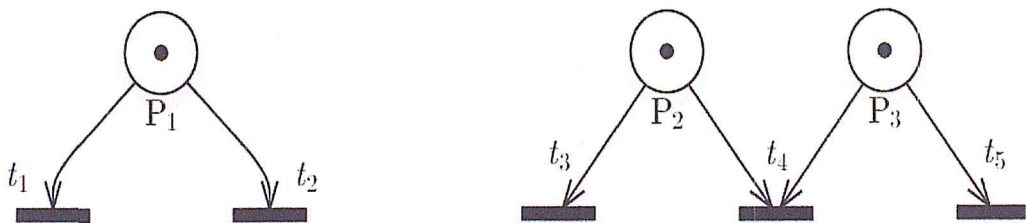


Figure 2.3 - Situations de conflit et de confusion

5. Propriétés des réseaux de Petri

Après la modélisation d'un système par un modèle formel, il est nécessaire de dériver et d'étudier des propriétés concernant ce système. Les réseaux de Petri ont une puissance majeure dans l'analyse des propriétés et problèmes associés aux systèmes parallèles. Dans ce qui suit, nous décrivons les propriétés les plus importantes des réseaux de Petri [26].

- **Bornitude**

Un réseau de Petri (R, M_0) est dit k -borné ou simplement borné si et seulement si, pour tout marquage accessible à partir de M_0 , le nombre de marques dans chaque place du réseau ne dépasse pas un nombre entier k . Dans ce cas, chaque place p est bornée.

Un réseau marqué est borné si toutes ses places sont bornées. Les réseaux **1-bornés** sont appelés des réseaux **saufs**.

- **Vivacité**

- Le réseau de Petri est **pseudo-vivant** si : $\forall M, \exists t$ telle que $M[t >$

C'est-à-dire que pour tout marquage il existe au moins une transition tirable à partir de ce marquage.

En d'autres termes on peut dire que dans le GMA il y a au moins un arc (transition) sortant de chaque état (marquage), ou alors on peut aussi dire que si tout marquage du GMA a un successeur alors ce réseau est pseudo-vivant. Donc un réseau **pseudo-vivant** n'as pas de marquage puits (sachant qu'un marquage puits ou mort est un marquage sans transition étirable, un marquage à partir duquel on ne peut pas tirer une transition).

- Une transition t d'un RdP marqué $(R;M_0)$ est **quasi-vivante** s'il existe un état M accessible à partir de M_0 pour lequel la transition t soit franchissable :

$$M_0 \xrightarrow{s} M \text{ et } M \xrightarrow{t}$$

- Une transition t d'un RdP marqué $(R;M_0)$ est **vivante** si de tout état accessible à partir de M_0 , il existe une séquence de franchissement qui utilise t :

$$\forall M' \in A(R; M_0), \exists s \text{ telle que } M' \xrightarrow{s} M'' \text{ et } M'' \xrightarrow{t}$$

Un RdP est dit vivant si toutes ses transitions le sont franchissables.

- **Réinitialisabilité**

Un RdP est réinitialisable (propre) si de tout état accessible, il est possible de revenir à l'état initial $\forall M \in A(R; M_0) \exists s$ telle que $M \xrightarrow{s} M_0$

- **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.

6. Les réseaux de Petri à arcs inhibiteurs

L'impossibilité pour les RDP simples de représenter certains phénomènes 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.

- **Définition [25]**

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 $R_z = \langle R, \text{Inh} \rangle$ tel que :

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

Exemple :

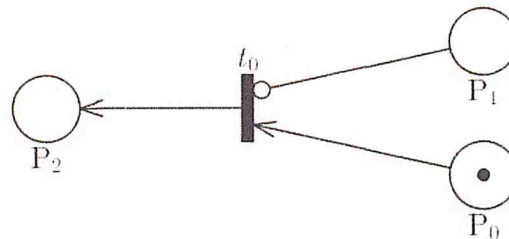


Figure 2.4 - Exemple d'arc inhibiteur

- **Règle de franchissement [28]**

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 :

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

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

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

7. Réseaux de Petri stochastiques

Les réseaux de Petri stochastiques ont été introduits par Florin dès 1978 pour répondre à certains problèmes d'évaluation liés à la sûreté de fonctionnement de systèmes informatiques. Ces problèmes faisant intervenir des phénomènes aléatoires, les transitions du réseau de Petri ont comporté des temps de franchissement aléatoires, distribués par une loi exponentielle. Cette distribution exponentielle permet d'exploiter les propriétés mathématiques d'un processus de Markov. Bien entendu, ce concept a largement été développé dès le début des années 80 pour répondre aux exigences de la modélisation de plus en plus complexes comme par exemple la modélisation des systèmes de production. De nombreuses classes de réseaux de Petri stochastiques sont proposées pour l'analyse des performances des systèmes de production. Les caractéristiques des différentes classes de réseaux de Petri stochastiques se situent essentiellement dans la nature des transitions utilisées. Initialement un réseau de Petri stochastique a toutes ses transitions temporisées avec un temps aléatoire qui est

distribué avec une loi exponentielle, mais nous retrouvons d'autres types de transitions [26,27].

8. Réseaux de Petri stochastiques généralisés (RdPSG)

Le réseau se compose de transitions avec une temporisation nulle dites transitions immédiates et de transitions avec une temporisation aléatoire distribuée exponentiellement dites transitions stochastiques. Le processus stochastique sous-jacent au graphe des marquages est un processus de Markov à temps continu.

- **Définition formelle [25]**

Un réseau de Petri stochastique généralisé (RdPSG) est un huit-uplet

$\langle P, T, Pre, Post, Inh, pri, W, M_0 \rangle$ Ou :

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

- **Conflit dans les RdPSG**

Lorsque deux ou plusieurs transitions sont en conflit dans un réseau de Petri stochastique généralisé, deux politiques de sélection de la transition qui sera tirée sont envisageables:

- Politique de présélection : la transition qui va franchir est présélectionnée selon une certaine métrique, comme la priorité. C'est la politique la mieux adaptée pour les transitions immédiates.
- Politique de course : la transition qui a le délai de franchissement le plus court sera franchi. Cette politique est généralement adaptée pour résoudre les situations de conflit entre les transitions temporisées.

- **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:

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

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

Et il n'existe pas de transition plus prioritaire que t qui soit franchissables :

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

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 :

$$M' = M - Pre(., T) + Post(., T).$$

9. Analyse des performances d'un RDPSG [27]

- **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 toutes 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. 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 évanescent.
- 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

- **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-réseau, et la probabilité d'un événement particulier se produit.

- 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 :

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

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 ;
- o 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 :

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

Où :

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

$$E[T] = \frac{E[N]}{E[\gamma]}$$

Où :

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

$$P\{E\} = \sum_{m_i \in X} \pi_i$$

Où :

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

10. Conclusion

Au cours de ce chapitre, nous avons présenté les notions de base nécessaires à la compréhension des réseaux de Petri et leur utilité pour la description et l'analyse de problème complexe.

Les modèles de réseaux de Petri sont très efficaces pour l'analyse des systèmes concurrents et la détermination de leurs propriétés qualitatives, telles que le blocage, la bornitude et la vivacité. Leur point faible était leur incapacité de calcul des propriétés quantitatives, et cela est dû à l'absence de l'information temporelle. L'introduction de la notion de temps aléatoire dans les modèles de réseaux de Petri a permis d'étendre leur puissance de modélisation, et de les lier au domaine de l'évaluation des performances. Leur puissance descriptive a été encore plus mise en valeur avec l'introduction des transitions immédiates en plus des transitions temporisées dans un même modèle, ce qui a donné naissance aux réseaux de Petri stochastiques généralisés.

Dans le chapitre suivant nous allons commencer par présenter les files d'attente avec vacance, et les différentes règles de début et de fin de vacance, nous allons parler de quelques travaux connexes à notre travail, nous présenterons aussi nos différents modèles de vacance pour un seul nœud capteur, et pour deux nœuds capteurs qui communiquent entre eux, et nous proposons une généralisation pour plusieurs nœuds qui communiquent avec un seul nœud (many-to-one).

1. Introduction

Les réseaux de capteurs sans fil ne cessent d'évoluer et on les trouve dans plusieurs domaines aussi bien militaires que civils. Nous avons discuté dans le premier chapitre des différentes contraintes de ces réseaux et parmi elles la plus importante se trouve la contrainte d'énergie. Une des solutions pour cette contrainte est l'utilisation de politiques de vacances dans le comportement des capteurs sans fil, ce qui permet d'économiser l'énergie précieuse durant leur temps libre.

Vue l'importance des réseaux de capteurs sans fil et leur utilisation plus fréquente, il devient impératif d'appliquer des approches de modélisation et d'évaluation de leurs performances avant tout déploiement. Pour cela, nous allons utiliser une approche d'analyse basée sur l'utilisation de modèles formels tels que les réseaux de Petri. Cette approche présente l'avantage de fournir des mesures de performances exactes.

Au cours de ce chapitre, nous allons introduire et définir la notion de vacance, les différents aspects des vacances et la signification d'une vacance d'un capteur sans fil. Par la suite, nous citerons quelques travaux réalisés sur l'économie d'énergie des réseaux de capteurs en se basant sur les files d'attente avec vacance, puis nous présenterons nos propositions des différentes disciplines de vacance pour un capteur sans fil en utilisant les réseaux de Petri stochastiques généralisés, toujours dans le but d'économiser l'énergie nous proposerons aussi des modèles de communication entre deux nœuds capteurs, nous clôturons nos propositions par des modèles pour la communication entre nœud capteurs (many-to-one), nous achevons le chapitre on donne les indices de performances relatifs aux différents modèles.

2. Files d'attente

Un système ou des unités ou des clients arrivent à un espace d'attente et attendent afin d'acquiescer un service auprès d'un moyen de service si le service n'est pas immédiatement disponible, et à la fin de leur service quittent ce système, est appelé système de file d'attente.

La théorie des files d'attente est une forme de probabilité qui se rapporte aux files d'attente. Elle permet l'analyse des flux entrants et sortants dans un système de files d'attente, et de calculer les divers paramètres de performance du système, comme la probabilité que le service soit immédiatement disponible pour un nouveau client arrivant, le nombre moyen d'unités dans le système et en attente, le temps passe en attente et dans le système. Ainsi, il est possible de prendre des décisions sur la base des paramètres du système, comme le nombre de ressources composant le service par exemple.

2.1 Files d'attente standards

Le modèle de base des files d'attente standard peut être décrit comme suit :

Des clients arrivent suivant un certain processus pour avoir un service. A cet instant, si un serveur est libre le service commence immédiatement. Sinon, il se met en attente, et sera servi lorsque un serveur sera libre et que les clients arrivés avant lui auront été servis (à moins qu'un système de priorité plus complexe ne soit mis en place). Les temps de service suivent une certaine loi de probabilité. Dans le reste de ce mémoire, nous utilisons simplement le mot file d'attente pour désigner une file d'attente standard.

Un système de file d'attente, tel que schématise dans la figure 3.1, possède cinq caractéristiques principales, qui sont : le processus d'entrée (des arrivées), le processus de service (temps de service), le nombre de serveurs composant le service, la capacité du système et la discipline de la file d'attente.

- **Processus des arrivées** : les instants ou les clients individuels s'introduisent dans le système, qui sont généralement inconnus et indépendants, s'appellent les instants d'arrivée. Les inter-arrivées, qui sont les intervalles de temps séparant deux arrivées successives, sont donc des variables aléatoires indépendantes de même loi. Le processus d'arrivée, qui décrit les instants d'arrivée, est alors le processus de comptage associé aux inter-arrivées. Si de plus, la loi est une loi exponentielle de paramètre λ , alors le processus des arrivées est un processus de Poisson de

taux λ . D'autres lois peuvent être utilisées pour décrire les inter-arrivées, comme la loi générale, déterministe, géométrique, etc.

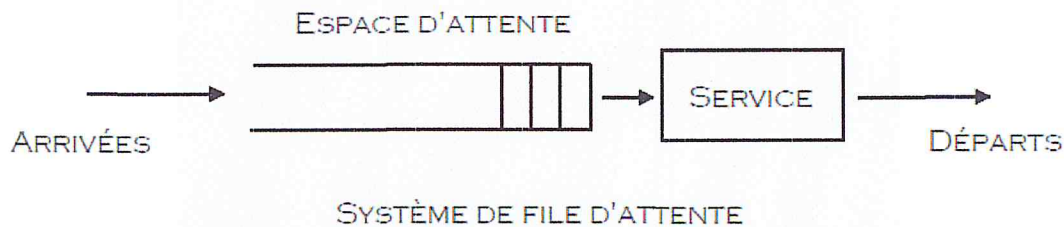


Figure 3.1- Système de file d'attente

- **Temps de service** : les durées de services sont toujours des variables aléatoires indépendantes qui suivent une même loi, qui est souvent, mais non nécessairement la loi exponentielle.
- **Nombre de serveurs** : dans le cas le plus simple, le système contient un seul serveur. Par contre, il peut y avoir plusieurs serveurs qui travaillent en parallèle. Il y a aussi la possibilité qu'un nombre infini de serveurs soient disponibles ; dans la pratique cela signifie que tous les clients reçoivent leur service en parallèle, et il n'y aura pas d'attente dans ce cas.
- **Capacité du système** : la capacité du système correspond au nombre maximum de clients pouvant être admis par le système, y compris les clients en cours de service. Si la taille du système est infinie, il n'y aura aucun blocage de nouveaux clients arrivés. Par contre, si la taille du système est finie, ce qui correspond aux situations réelles, alors des clients peuvent être rejetés.
- **Discipline de la file d'attente** : la sélection du prochain client à servir quand il y a un serveur libre est effectuée selon la discipline de la file d'attente. La règle la plus fréquemment utilisée s'exprime sous la forme « le premier arrive est le premier servi », connue sous l'abréviation Anglo-Saxons FIFO (First In First Out). D'autres disciplines sont évidemment possibles, comme « le dernier arrive est le premier servi »

(LIFO : Last In FirstOut) ou bien la discipline de service aléatoire (Random).

2.2 Files 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 durant 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 [12]. 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.

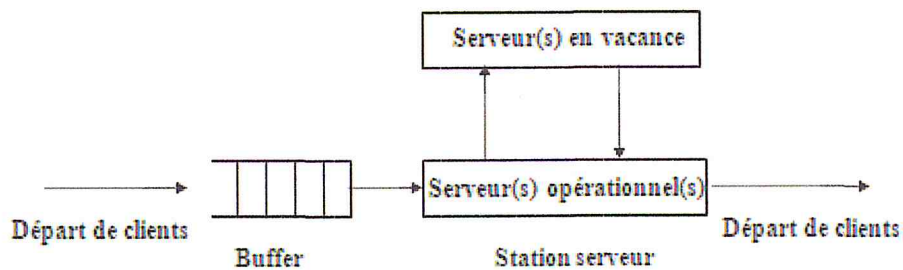


Figure 3.2 - Système de file d'attente avec vacance

2.2.1 Notation

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

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

Où

A : distribution des inter-arrivées.

B : distribution de service.

S : nombre de serveurs.

L : capacité du système.

N : taille de la source de clients.

K_v : discipline de service.

Le K peut être :

E : service exhaustif.

L : service limité.

A : service avec barrière.

B : service de Bernoulli.

N : service n-vacances.

Et le « v » peut correspondre à :

S : vacance unique.

M : vacance multiple.

H : vacance hybride

2.2.2 Politiques de vacance

Une politique de vacance 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.[13,14]

○ 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. Ils existent deux disciplines de service majeures, à savoir, service exhaustif ou non-exhaustif.

- **service exhaustif** : le serveur ne peut prendre une vacance qu'une fois que le système devient vide.
- **service non-exhaustive** : 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 limité** Dans un système d'attente avec vacance a service limite, quand le serveur retourne d'une vacance, il sert un nombre k de clients au maximum, puis débutera une autre vacance
 - ✓ **le service avec barrière** Avec une discipline de ce service, 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

Cette règle indique quand le serveur reprend le service des clients. Il y a deux règles populaires, qui sont la politique de vacance unique et la politique de vacance multiple.

- **Une politique de vacance multiple** exige que le serveur prenne des vacances successives jusqu'à ce qu'il trouve au moins un client en attente dans le système à la fin d'une période de vacance.
- **une politique de service unique** le serveur prend seulement une vacance à la fin de chaque période de service. Après cette vacance, le serveur commence le service des clients dans le système, ou il reste libre si aucun client n'est trouvé dans le système.
- **Les vacances hybrides**, Dans ce cas, si le serveur revient d'une vacance et trouve le système vide il attend alors une durée de temps aléatoire (distribuée exponentiellement) à la fin de la quelle, si aucun client n'est arrivé, le serveur commencera une autre vacance.
- **Vacance multiple adaptatif** (Modified vacation ou Multiple adaptive vacations ou Randomized Vacations): Après une période de service, si le système reste vide, le serveur peut prendre au plus J périodes de vacances successives avant de retourner au service. Le nombre de vacances consommées est considéré comme une variable aléatoire.

Ils existent des règles générales telles que la politique de seuil (N-vacance) et la politique T-vacance Vacance multiple adaptatif

- **une politique N-vacance**, le serveur reprend le service quand le nombre de clients dans le système atteint le seuil N .
- **Une politique aléatoire (T-vacance)**, le serveur reste en vacance pendant une période de longueur T . A la fin de cette période, il reprend le service.

- **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. Vacance d'un capteur sans fil

Dans un capteur sans fil, l'unité de communication (la radio) est la plus gourmande en énergie par rapport aux autres unités. Ainsi, pour de minimiser la consommation d'énergie d'un nœud capteur le Protocol de sommeil/réveil (sleep/wakeup) a été développé, c'est à dire que le capteur peut prendre une vacance pour économiser son énergie. Le nœud capteur possède deux modes de fonctionnement: le mode actif et le mode sommeil (vacance). Lorsqu'il est en mode sommeil, le capteur ne peut pas interagir avec son environnement par conséquent, l'énergie consommée est très basse. Durant le mode actif, le capteur écoute le canal, génère des paquets, reçoit et transmet les paquets de données de ces voisins donc l'énergie consommée est considérable [19].

4. Travaux connexes

La recherche dans le domaine des réseaux de capteurs sans fil en utilisant des modèles formels semble être encore à ses débuts. Nous avons trouvé peu de travaux dans ce Domain, appliquant des modèles de files d'attente, des réseaux de Petri.

Jiang et al. [20] ont utilisé la théorie des files d'attente standards $M/G/1$ avec une politique de vacance (N -vacance) pour optimiser la consommation d'énergie. Dans leur modèle, ils ont considéré des nœuds capteurs avec des buffers à capacité illimitée, et un schéma de communication plusieurs-à-un, dans lequel les nœuds communiquent avec le nœud puits, à un seul saut. Ce nœud possède trois états différents libre, occupé, et transitoire :

- Durant libre, si la fonction de transmission est désactivée, occupé, quand la transmission est activée, démarrage, qui est un état transitoire entre l'état occupé et l'état libre. Durant l'état occupé le nœud peut commencer la transmission des paquets stockés dans le buffer. Dès que le buffer est vide, le

nœud repasse à l'état libre. Les auteurs ont donné des fonctions mesurant la consommation d'énergie et la latence des messages.

Dans [21], les mêmes auteurs ont repris le même modèle avec quelques changements. Ils ont utilisé la politique de vacance Min (N,T) plutôt que la politique de vacance N-vacance. Ils ont considéré seulement les états libre et occupé. Le nœud balance entre les deux états selon les règles suivantes :

- Quand le buffer du capteur est vide, le nœud prend une période de vacance de longueur T.
- Si durant la vacance le nombre de paquets en attente dans le buffer atteint le seuil N, le capteur interrompe la vacance et passe à l'état occuper et active la fonction de transmission. autrement à la fin de la vacance de longueur T si le buffer n'est pas vide, il passe à l'état occupé, sinon il prend une autre période de vacance de longueur T.

La discipline de vacance proposée dans le présent travail à l'avantage de minimiser la latence engendré par la politique de vacance N-vacance dans le cas où le taux d'arrivée est faible et évite la saturation du buffer dans le cas contraire.

A. Lezanska et M.F. Hayat [15], ont utilisé une chaîne de Markov pour l'analyse des performances d'énergie dans les nœuds capteurs sans fils, ils ont introduit un système de service limité à vacances multiples. Pour l'analyse, ils ont utilisé les chaînes de Markov pour l'état du système, le processus de flux, période active et période de vacances sont introduits, à partir de lesquelles les mesures de performance et le gain d'énergie d'un nœud peuvent être calculées, afin de prouver l'exactitude du modèle proposé, ils ont vérifié leurs résultats par des simulations.

Plusieurs techniques ont été proposées pour optimiser la consommation d'énergie dans les RCSF et par conséquent augmenter le temps de vie du réseau, en mettant toute concentration sur les différents aspects des fonctions d'un seul nœud de capteurs sans fil tels que l'agrégation de données, routage efficace, la planification du mode vacance/ actif, et la gestion du buffer. Sleem et al [33], ont proposé une technique d'optimisation d'énergie dans les nœuds capteurs sans fils qui est basé sur la technique N-vacance pour réduire le nombre de transitions de l'état vacance à l'état actif, l'impact de cette technique sur la conservation de l'énergie dans les réseaux de capteurs sans fil a été vérifiée par plusieurs efforts de recherche, il ont analysé l'impact de cette dernière sur des réseaux de capteur sans fils avec des files d'attente sensible au délai, ils ont aussi étudié si il existe une file d'attente optimale de longueur N qui

permet de minimiser la consommation d'énergie ainsi de respecter le délai requis en même temps.

5. Modélisation des RCSF à l'aide des RdPSG

5.1 Modèles pour un seul nœud capteur

5.1.1 Modèle générique (M0)

Avant d'entamer les propositions, nous avons commencé par proposer un modèle générique pour un capteur sans fils en utilisant les RDPSG noté (M0).

Le modèle décrit le comportement d'un nœud capteur durant la phase active de son cycle de vie. Durant cette phase le nœud peut recevoir des paquets à relayer, peut générer des paquets les stocker dans le buffer à capacité limité et les transmet après.

Le modèle (M0) est un modèle de base qui sert à faciliter la description des modèles proposé.

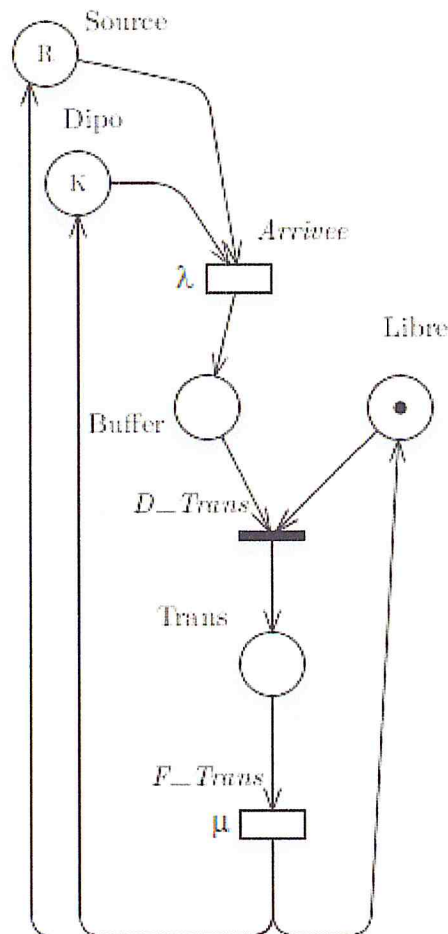


Figure 3.3 Modèle générique(M0)

La figure 3.3 représente le modèle M0 dans lequel :

- La place *Source* représente la source des paquets.
- La place *Dispo* représente la capacité de stockage disponible.
- La place *Buffer* représente le buffer du nœud
- La place *Trans* représente la transmission d'un paquet.
- La 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 :

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

Lors de la réception d'un paquet si au moins un jeton est dans la place *Dispo* le paquet sera stocké dans le Buffer en franchissant la transition *Arrivée*, ainsi un jeton sera produit dans *Buffer*. Ce paquet sera transmis si le module de transmission est libre représenté par un jeton dans la place *libre* en franchissant la transition *D_Trans* qui consommera un jeton de la place *libre*, un autre jeton de la place *Buffer* et produira un jeton dans la place *Trans*. La fin d'une transmission d'un paquet engendre le tire de la transition *F_Trans* qui consomme un jeton de la place *Trans* et en produit 3 jetons : un dans le 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.

Afin d'introduire une stratégie d'économie d'énergie durant la phase active d'un nœud de capteur dans les prochains modèles nous allons introduire la notion de vacance.

Nous avons optés pour que le nœud possède deux états, l'état libre et l'état vacance. Durant l'état libre le capteur peut recevoir des paquets à relayer, générer ces propres paquets et transmet les paquets aux voisins. Durant l'état de vacance la fonction de transmission est désactivée, donc le nœud peut uniquement recevoir et générer les paquets et les stocker dans le buffer jusqu'au prochain réveil et le passage à l'état libre pour pouvoir envoyer les paquets en attente.

Dans le présent travail, nous allons proposer des modèles basés sur le service non-exhaustif.

5.1.2 Modèle de vacance multiple aléatoire avec barrière (M1)

Dans le modèle (M1) nous allons proposer un modèle pour un nœud de capteur sans fils avec la discipline de service avec barrière et vacances multiples aléatoire :

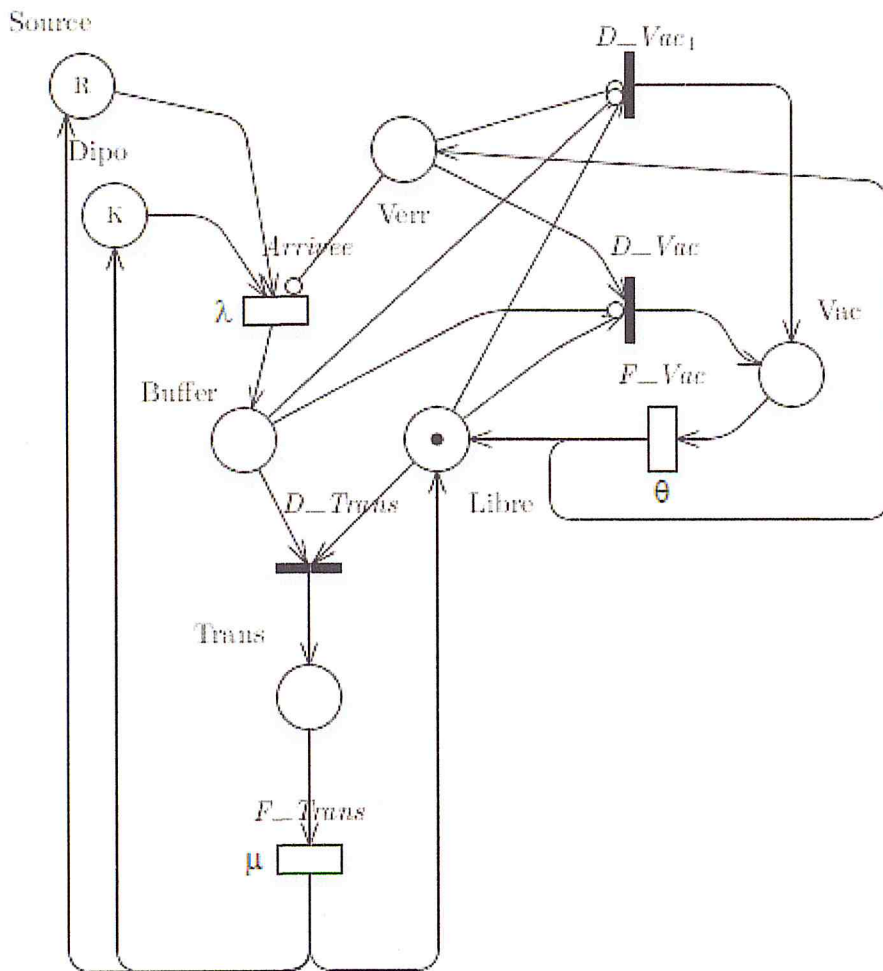


Figure 3.4 Modèle vacance aléatoire multiple avec barrière(M1)

Une discipline de service avec barrière et vacance multiple aléatoire est une politique où le capteur ne servira dès son retour de la vacance que les clients qui étaient présents dans le système, et repartira en vacance aussitôt qu'il aura fini son service. Ces vacances sont de nature multiple, c'est-à-dire qu'à son retour au service, le capteur pourra prolonger sa période de vacance si son buffer est toujours vide

Nous avons modélisé un système avec une telle politique (La figure 3.5) en reprenant le modèle initial M_0 et en ajoutant les places suivantes :

- la place Vac représente le fait que le capteur est en état de vacance
- la place $Verr$ est une sorte de verrouillage pour ne servir que les clients qui étaient dans le système à un instant donné

Nous avons ajouté aussi trois transitions:

- D_Vac1 : elle permet au capteur de prendre une vacance, le franchissement de cette transition absorbera un jeton de la place *libre* dans le cas où les places *buffer* et *verr* sont vide, dans le cas contraire le franchissement est bloqué par deux arc inhibiteur
- D_Vac : elle permet aussi au capteur de prendre une vacance, le franchissement de cette transition absorbera un jeton de la place *libre* et un jeton de la place *verr* qui permettra au capteur de partir en vacances dans le cas où le *buffer* est vide
- F_Vac : elle permet le retour d'une vacance, cette transition sera franchie après une durée aléatoire $1/\theta$

5.1.3 Modèle de vacance unique aléatoire avec barrière (M2)

Dans le modèle (M2) nous allons proposer un modèle pour un nœud de capteur sans fils avec la discipline de service avec barrière et vacances unique aléatoire

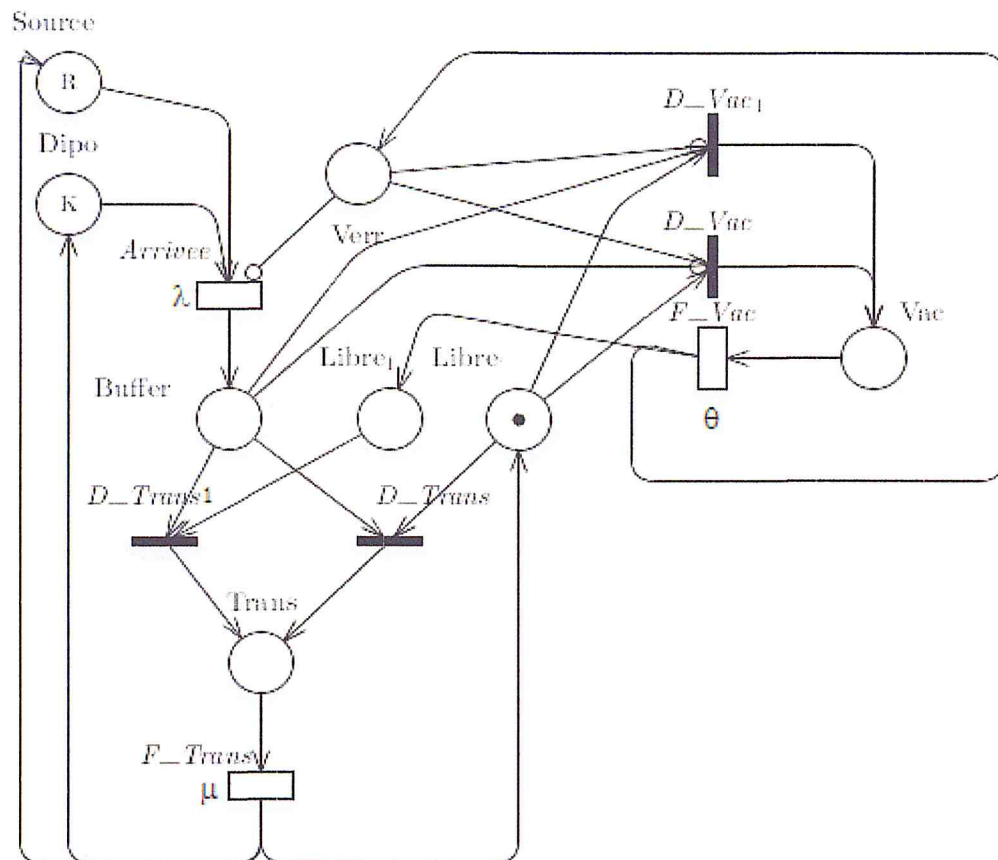


Figure 3.5 Modèle de vacance unique aléatoire avec barrière(M2)

Une discipline de service avec barrière à vacance unique aléatoire est une politique où le capteur ne servira à son retour de la vacance que les clients qui étaient présents dans le système, et repartira en vacance aussitôt qu'il aura fini son service. Ces vacances sont de nature unique, c'est-à-dire qu'à son retour au service, le capteur ne pourra pas prolonger sa période de vacance que s'il a traité ou moins un paquet.

Nous avons modélisé ce système (La figure 3.6) en modifiant le model M1 et en ajoutant la places suivantes :

- *Libre1* cette place exige de traité ou moins un paquet avant de prendre une nouvelle vacance.

Nous avons introduit aussi une nouvelle transition :

- D_trans1 : permet au capteur de transmettre un paquet après une vacance, le franchissement de cette transition absorbera un jeton de la place *libre1* et un jeton du *buffer*

5.1.4 Modèle de vacance unique N-vacance avec barrière(M3)

Dans le modèle (M3) nous allons proposer un modèle pour un nœud de capteur sans fils avec la discipline de service avec barrière et vacances uniques N-vacance :

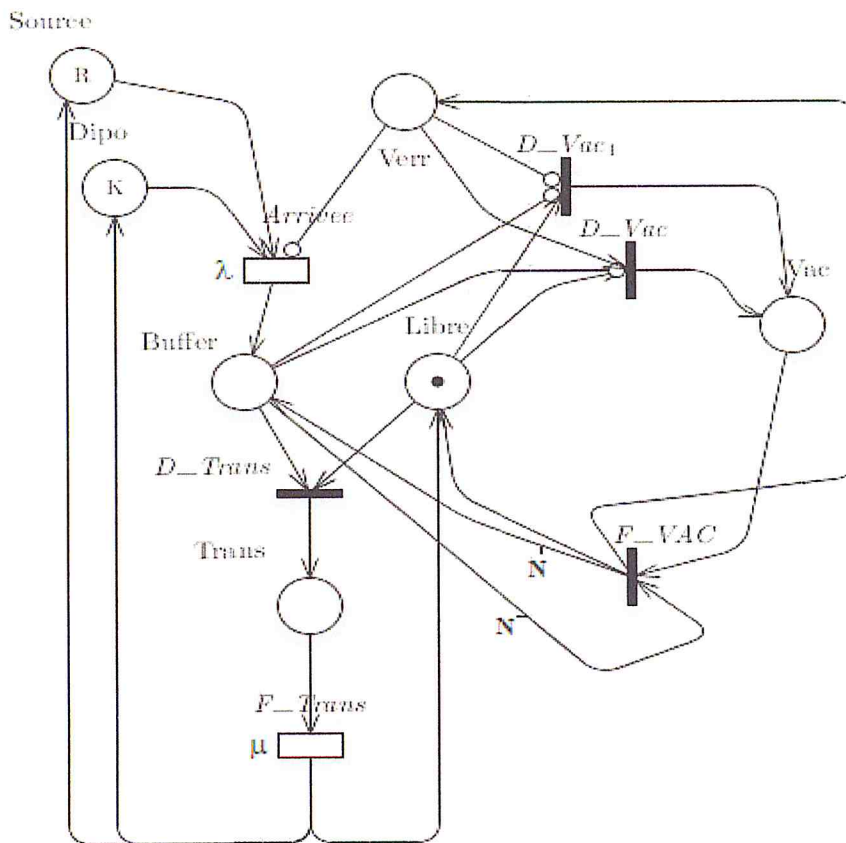


Figure 3.6 Modèle de vacance multiple N-vacance avec barrière(M3)

Dans nous avons modifié par rapport à M1 le règle de fin de vacance au lieu de la vacance de longueur aléatoire nous avons opté pour la politique N-vacance, avec cette discipline, le capteur passe à l'état libre dès que le nombre de paquets en attente dans le buffer atteint le seuil N.

Dans ce model nous avons remplacé la transition temporisé F_vac par





une transition immédiate, on a aussi modifié les règles de franchissement d'une façon quelle absorbera N jeton de la place *buffer* et un jeton de la place *Vac*, elle remet N jeton a la place *buffer* et un jeton a la place *libre*, elle permettra au capteur de passer à l'état libre.

5.1.5 Modèle avec service barrière à vacance multiples adaptatif aléatoire (M4)

Dans le modèle (M4) nous allons proposer un modèle pour un nœud de capteur sans fils avec la discipline de service avec barrière et vacances multiples adaptatif aléatoire :

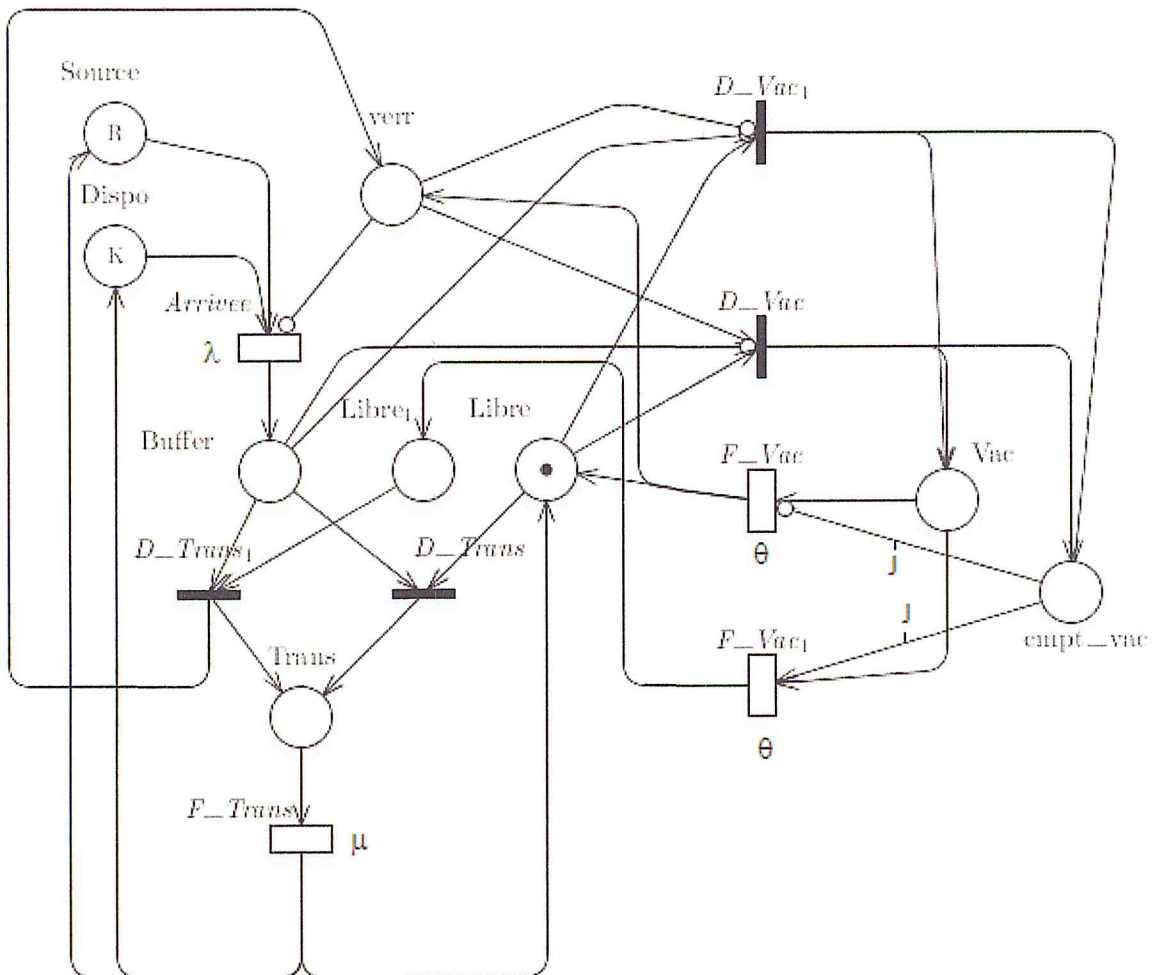


Figure 3.7 Modèle service avec service barrière à vacance multiples adaptatif aléatoire (M4)

Dans ce modèle on se basant sur le modèle M1, nous avons introduit la politique de vacances multiples adaptatives avec cette politique, le capteur

peut prendre au plus J vacances successives si son buffer est vide avant de reprendre le service des paquets

Nous avons ajouté au modèles M1 les places suivantes :

- *cmpt_vac* : il représente le nombre de fois ou le capteur est parti en vacances.
- *Libre1*: oblige le capteur de traité ou moins un paquet avant une nouvelle vacance.

Nous avons ajouté aussi de nouvelles transitions :

- *F_vac1*: cette transition permet au capteur le retour d'une vacance, elle absorbe J jetons de la place *cmpt_vac*, un jeton de *Vac* et un jeton de *Verr* et transmet un jeton à *libre1*
- *D_trans1* : permet au capteur de transmettre un paquet après une vacance, le franchissement de cette transition absorbera un jeton de la place *Libre1* et un jeton de la places *buffer*.

Les règles de franchissement de la transition *F_vacs* ont modifiées par rapport à celles du modèle M1, quand la place *cmpt_vac* atteint J jeton le franchissement sera bloquée par l'arc inhibiteur à poids J .

5.1.1 Modèle avec service limité à vacance multiple aléatoire (M5)

Dans le modèle (M5) nous allons proposer un modèle pour un nœud de capteur sans fils avec la discipline de service limité et vacances multiples aléatoire :

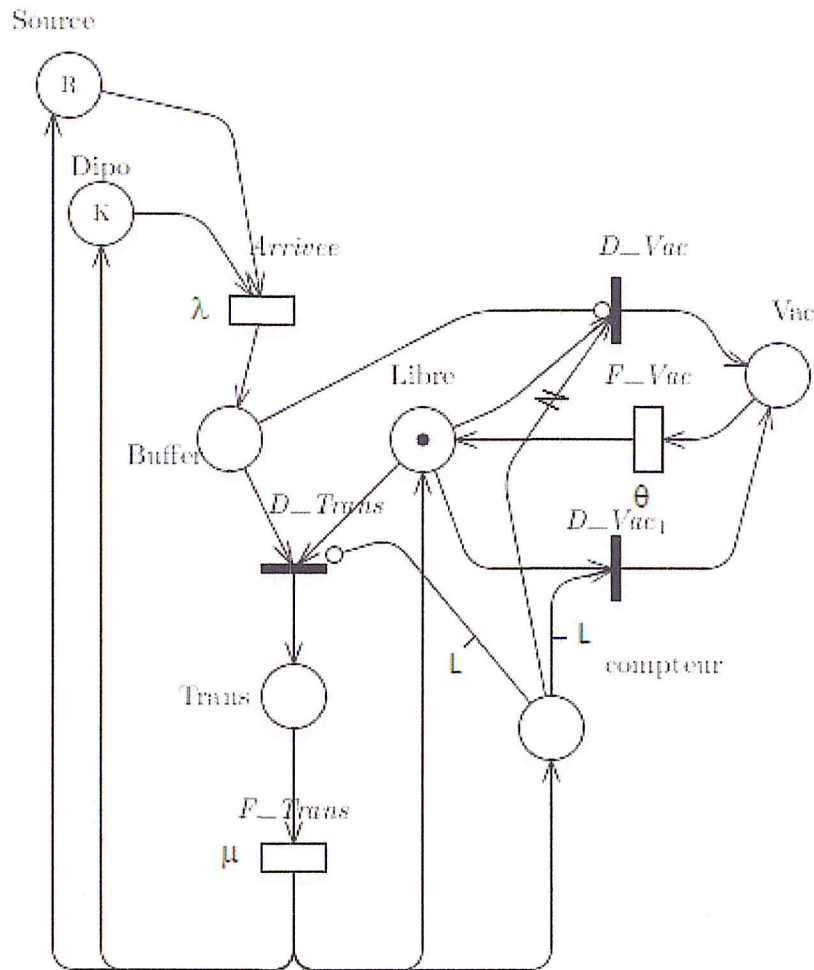


Figure 3.8 Modèle avec service limité à vacance multiple aléatoire (M5)

Une discipline de service limité avec vacances multiples aléatoires est une politique où le capteur prend des vacances quand il a traité L paquets. Ces vacances sont de nature multiple, c'est-à-dire qu'à son retour au service, le capteur prend une autre période de vacance si son buffer est toujours vide.

Nous avons proposé un modèle avec une telle politique (Figure 3.8) en reprenant le model initial M0 et en ajoutant les places suivantes

- la place *compteur* : représente les nombres de paquets que le capteur a transmis.
- la place *Vac* : représente le fait que le capteur est en état de vacances

Sur ce modèle nous avons utilisé l'arc zigzag, ou son poids peut varier, et représente la condition que si le nombre de jetons de la place compteur est

inférieur à L et le buffer et vide le capteur peut partir en vacance, et donc il nous permet de réinitialisé le compteur à zéro.

Nous avons introduit aussi trois transitions:

- D_Vac1 : elle permet au capteur de prendre une vacance, le franchissement de cette transition absorbera un jeton de la place libre et L jeton de la place compteur qui permettra au capteur de partir en vacances
- D_Vac : permet aussi au capteur de prendre une vacance, le franchissement de cette transition absorbera un jeton de la place libre et les jetons disponibles dans la places *compteur* à l'aide de l'arc à zigzag qui nous permet de vider le compteur dans le cas où le buffer est vide, dans le cas contraire le franchissement est bloqué par l'arc inhibiteur
- F_Vac : elle permet le retour d'une vacance, cette transition sera franchie après une durée aléatoire θ

Nous avons également modifié la règle de franchissement de la transition D_Trans on ajoutant l'arc inhibiteur avec le poids L reliant la place *compteur* avec la transition D_trans , de ce fait le franchissement se fait si le nombre de jetons de la place *compteur* est inférieur à L .

5.1.2 Modèle avec service limité à vacance unique aléatoire (M6)

Dans le modèle (M6) nous allons proposer un modèle pour un nœud de capteur sans fils avec la discipline de service limité et vacances unique aléatoire :

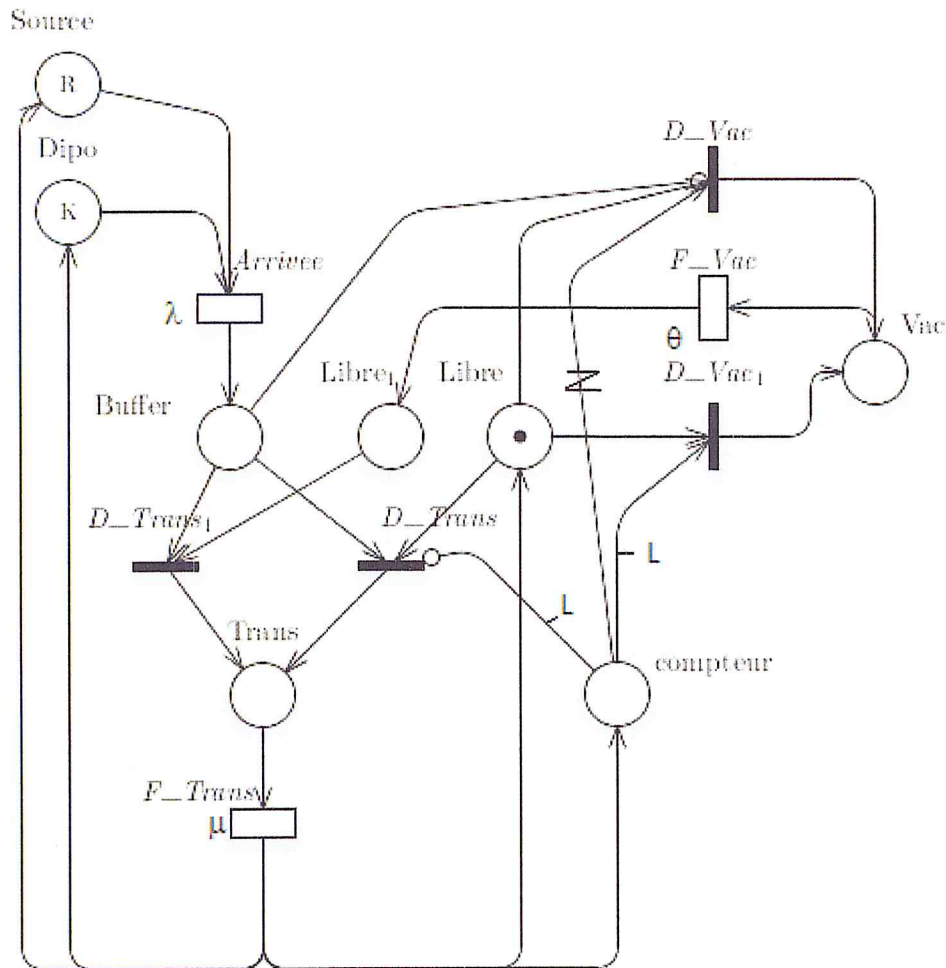


Figure 3.9 Modèle avec service limité à vacance unique aléatoire (M6)

Une discipline de service limité avec vacances unique aléatoires est une politique où le capteur prend des vacances quand il a traité L paquets, ces vacances sont de nature unique, c'est-à-dire qu'à son retour au service, le capteur ne pourra pas prolonger sa période de vacance que s'il a traité ou moins un paquet.

Nous avons proposé un modèle avec une telle politique (Figure 3.9) en modifiant le model M5 et en ajoutant la place suivante :

- *Libre1* : cette place oblige le capteur de traité ou moins un paquet avant de prendre une nouvelle vacance.

Nous avons introduit aussi une nouvelle transition :

- D_trans1 : permet au capteur de transmettre un paquet après une vacance, le franchissement de cette transition absorbera un jeton de la place $libre1$ et un jeton du $buffer$.

5.1.3 Modèle avec service limité à N-vacance unique (m7)

Dans le modèle (M7) nous allons proposer un modèle pour un nœud de capteur sans fils avec la discipline de limité et vacances unique N-vacance:

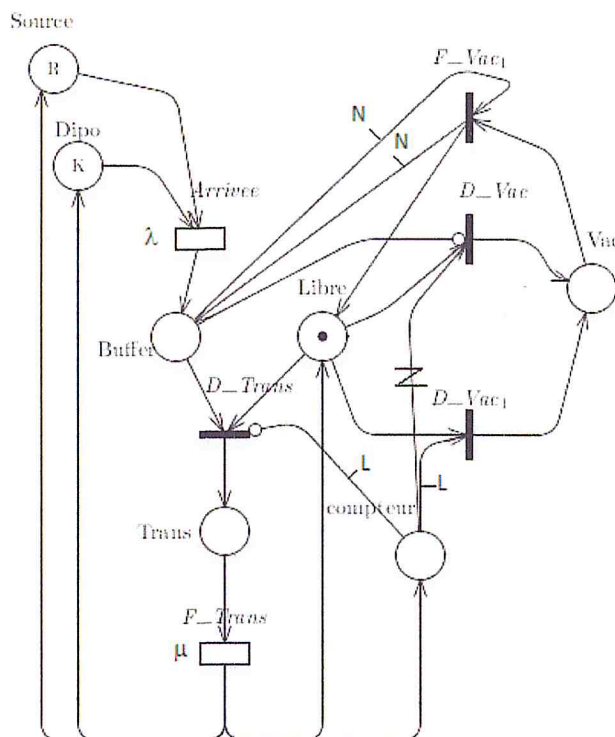


Figure 3.10 Modèle avec service limité à vacance multiple aléatoire (M7)

Dans le présent modèle (M7), nous avons repris le modèle (M5), en adaptant la politique N-vacance comme règles de fin de vacance au lieu des vacances aléatoire.

Dans ce model nous avons fait la même modification que celles dans le modèle (M3).

5.1.4 Modèle avec service limité à vacance multiples adaptatif aléatoire (M8)

Dans le modèle (M8) nous allons proposer un modèle pour un nœud de capteur sans fils avec la discipline de service limité et vacances multiples adaptatif aléatoire :

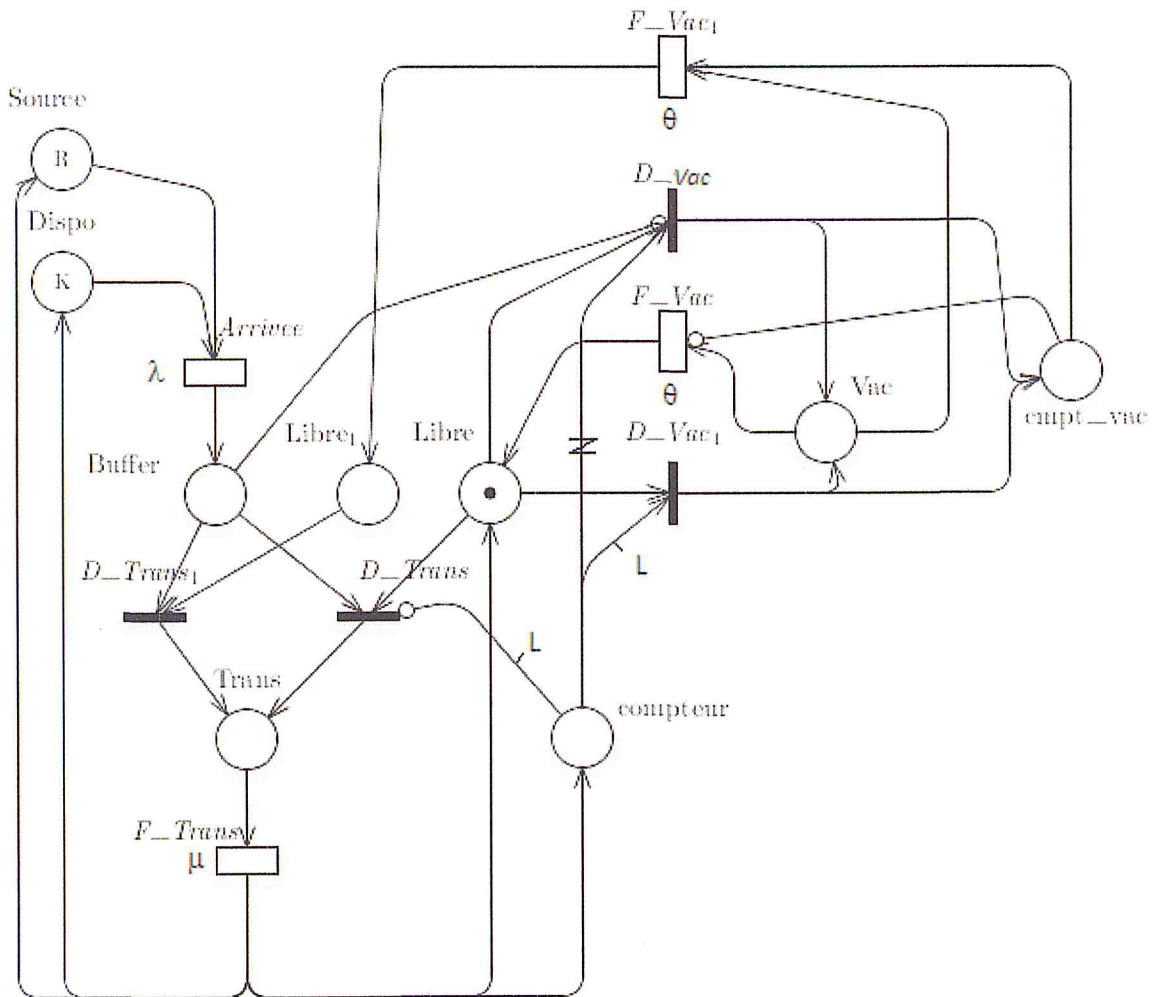


Figure 3.11 Modèle avec service limité à vacance multiples adaptatif aléatoire (M8)

Dans ce modèle (M8), représenté dans la figure 3.11 nous avons repris le modèle M5 (figure 3.8) avec les vacances multiples adaptatifs. Pour ce fait nous avons rajouté les places suivantes :

- *empt_vac* : il représente le nombre de fois ou le capteur est parti en vacances.
- *Libre1*: oblige le capteur de traiter ou moins un paquet avant une

nouvelle vacance.

Et les transitions suivantes :

- F_vac1 : cette transition permet où capteur le retour d'une vacance, elle absorbe J jetons de la place $cmpt_vac$ et un jeton de la place Vac et transmet un jeton à la place $libre1$
- D_trans1 : permet au capteur de transmettre un paquet après une vacance, le franchissement de cette transition absorbera un jeton de la place $libre1$ et un jeton de la place $buffer$.
-

Les règles de franchissement de la transition F_vac sont modifiées par rapport à celles du modèle M5, quand la place $cmpt_vac$ atteint J jeton le franchissement sera bloquée par l'arc inhibiteur à poids J.

5.2 modèle pour deux nœuds capteurs

Les modèles proposés dans la section (5.1) reposent sur une stratégie d'optimisation d'énergie dans seul capteur sans fils uniquement durant la phase active.

Se pendant, les modèles proposé dans cette section propose des stratégies pour l'économie d'énergie dans un scénario de communication entre nœud émetteur et nœud récepteur en globe la phase sommeil et phase active. Pour ce fait, nous avons appliqué la même stratégie issue de la section (5.1) pour l'optimisation durant la phase active.

Concernant la phase sommeil, nous avons analysé l'effet de la règle de passage de la phase active à la phase sommeil du nœud récepteur.

Pour cela on a opté pour les disciplines non-exhaustives (limité et barrière), afin de comparer leurs performances respectifs.

Par la suite nous avons généralisé les modèles proposés dans le scénario émetteur et récepteur pour le cas d'un seul nœud récepteur et plusieurs émetteurs (many-to-one).

5.2.1 Modèle de transmission entre deux nœuds avec service limite à vacance multiple (M9)

Dans le modèle (M9) nous allons proposer un modèle pour deux nœuds de capteurs sans fils (émetteur, récepteur) avec la discipline de service limité et vacances multiples aléatoire pour la communication entre les deux nœuds :

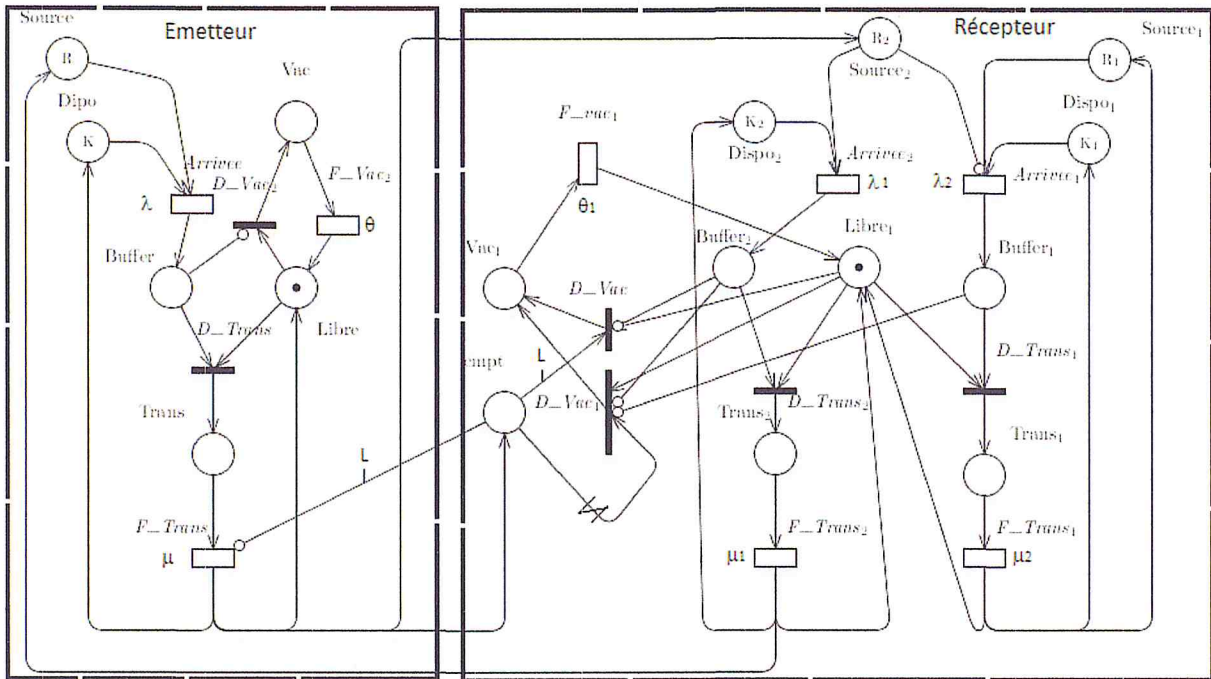


Figure 3.12 Modèle de transmission entre deux nœuds avec service limite à vacance multiple aléatoire (M9) :

Pour le nœud émetteur nous avons utilisé une discipline exhaustive on utilisant les places suivantes :

- La place *Source* représente la source des messages.
- La place *Dispo* représente la capacité de stockage disponible le nœud.
- La place *Buffer* représente le nombre de paquets dans le system en attente de transmission.
- La place *Trans* représente la transmission d'un paquet.
- La place *Libre* représente l'état de l'unité de transmission.
- la place *Vac* représente le fait que le capteur est en état de vacance

Lors de la réception d'un paquet si au moins un jeton est dans la place *Dispo* le paquet sera stocké dans le Buffer en franchissant la transition *Arrivée*,

ainsi un jeton sera produit dans *Buffer*. Ce paquet sera transmis si le module de transmission est libre représenté par un jeton dans la place *libre* en franchissant la transition *D_Trans* qui consommera un jeton de la place *libre*, un autre jeton de la place *Buffer* et remet un jeton dans la place *Trans*. La fin d'une transmission d'un paquet engendre franchissement de la transition *F_Trans* qui consomme un jeton de la place *Trans* cette transition sera bloqué par arc inhibiteur a L poids si la place *compteur* a L jetons, elle remis 3 jetons : un dans le 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, quand le buffer est vide le capteur part en vacances à l'aide de la transition *D_vac2* qui absorbera un jeton de la place *libre* dans le cas contraire la transition sera bloqué par l'arc inhibiteur le retour de la vacance est à l'aide de la transition *F_vac2* après une durée aléatoire elle absorbe un jeton de place *Vac* et remis un jeton à la place *libre*.

Pour la partie de nœud récepteur nous avons utilisé Une discipline de service limité avec vacances multiples aléatoires en utilisant les places suivantes :

- La place *Source1* représente les paquets générés par le capteur.
- La place *Source2* représente les paquets reçus par le capteur
- La place *Dispo1* représente la capacité de stockage disponible dans le nœud pour les paquets générés.
- La place *Dispo2* représente la capacité de stockage disponible dans le nœud pour les paquets reçus.
- La place *Buffer1* représente le nombre de paquets générés dans le system en attente de transmission.
- La place *Buffer2* représente le nombre de paquets reçus dans le system en attente de transmission.
- La place *Trans1* représente la transmission d'un message généré.
- La place *Trans2* représente la transmission d'un message reçu
- La place *Libre1* représente l'état de l'unité de transmission.
- La place *Vac1* représente le fait que le capteur est en état de vacance
- la place *cmpt*: représente les nombres de paquets que le capteur a

transmis.

Et les transitions suivantes :

- la transition *Arrivée1*, qui absorbera un jeton de la place *Source1*, et un jeton de la place *Dispo1*, ainsi un jeton sera remis dans la place *Buffer1*.
- la transition *Arrivée2*, qui absorbera un jeton de la place *Source2*, et un jeton de la place *Dispo2*, ainsi un jeton sera remis dans place *Buffer2*.
- la transition *D_Trans1* qui consommera un jeton de la place *libre1*, un autre jeton de la place *Buffer1* et remis un jeton dans la place *Trans1*.
- la transition *D_Trans2* qui consommera un jeton de la place *libre1*, un autre jeton de la place *Buffer2* et remis un jeton dans la place *Trans2*.
- la transition *F_Trans1* consomme un jeton de la place *Trans1* et remis 3 jetons : un dans le place *Libre1* décrivant le passage de l'état occupé à l'état libre du module radio, un jeton dans la place *Dispo1* qui représente, ainsi, la libération d'une place dans le buffer, un jeton dans la place *Source1*
- la transition *F_Trans2* consomme un jeton de la place *Trans2* et produit 3 jetons : un dans la place *Libre1* décrivant le passage de l'état occupé à l'état libre du module radio, un jeton dans la place *Dispo2* qui représente, ainsi, la libération d'une place dans le buffer, un jeton dans la place *Source* du nœud émetteur
- la transition *D_Vac1* : elle permet au capteur de prendre une vacance, le franchissement de cette transition absorbera un jeton de la place libre et L jeton de la place *cmpt* qui permettra au capteur de partir en vacances si le buffer est vide sinon elle sera bloquée par l'arc inhibiteur.
- La transition *D_Vac* : permet aussi au capteur de prendre une vacance, le franchissement de cette transition absorbera un jeton de la place libre et les jetons disponibles dans la places *cmpt* à l'aide de l'arc zigzag qui nous permet de vider le compteur dans le cas où le

buffer est vide, dans le cas contraire le franchissement est bloqué par l'arc inhibiteur

- La transition F_Vac1 : elle permet le retour d'une vacance, cette transition sera franchie après une durée aléatoire $1/\theta$, elle produit un jeton dans la place *libre1*

5.2.2 Modèle de transmission entre deux nœuds avec service barrière à vacance multiple aléatoire (M10)

Dans le modèle (M10) nous allons proposer un modèle pour deux nœuds de capteurs sans fils (émetteur, récepteur) avec la discipline de service avec barrière et vacances multiples aléatoire pour la communication entre deux nœuds :

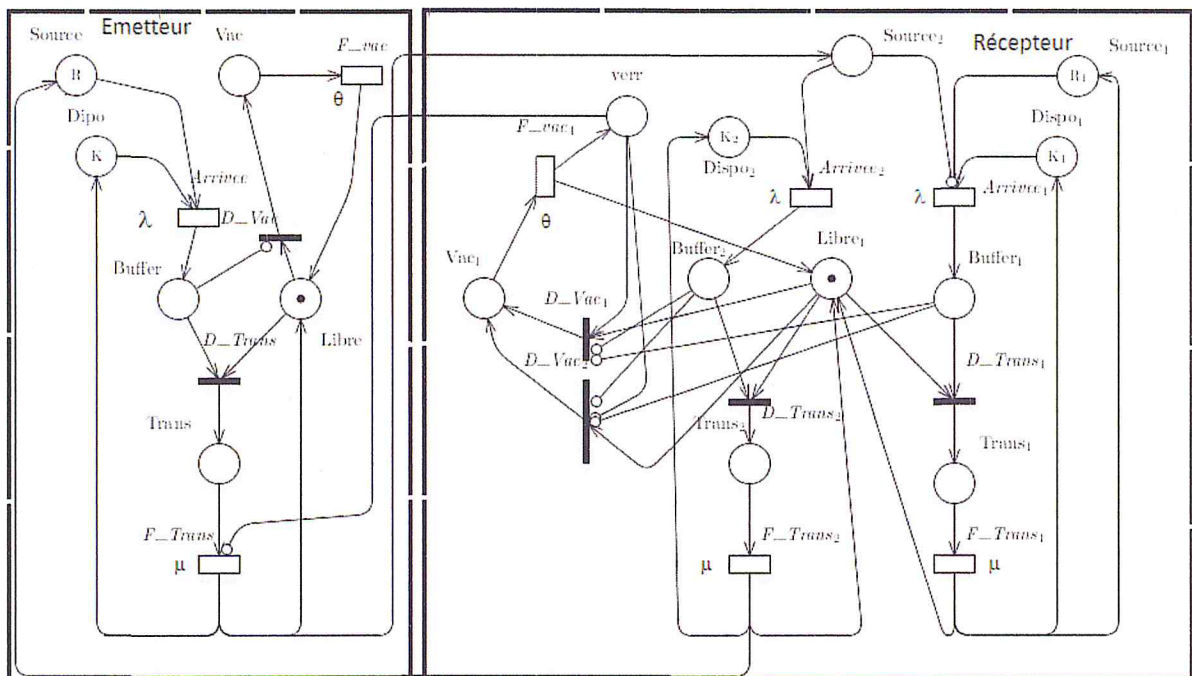


Figure 3.13 Modèle de transmission entre deux nœuds avec service barrière à vacance multiple(M10)

La figure 3.13 représente le modèle M10 dans ce modèle nous avons utilisé les mêmes notions que celles dans le modèle M9 sauf que nous avons changé de discipline:

Pour le nœud émetteur nous avons gardé la même discipline, on a juste modifié les règles de franchissement de la transition F_Trans si la place *Verr* est vide

elle absorbera un jeton de la place *Trans* dans le cas contraire le passage sera bloqué.

Pour le nœud récepteur nous avons utilisé une discipline avec barrière à vacance multiples nous avons changé la place *cmpt* par une place *verr* on modifiant les transitions suivantes :

- *D_Vac1* : elle permet au capteur de prendre une vacance, le franchissement de cette transition absorbera un jeton de la place *libre* et un jeton de la place *verr* qui permettra au capteur de partir en vacances si le buffer est vide sinon elle sera bloquée par l'arc inhibiteur.
- *D_Vac2* : permet aussi au capteur de prendre une vacance, le franchissement sera effectué si la place *verr* et le buffer sont vide, cette transition absorbera un jeton de la place libre.
- *F_Vac* : elle permet le retour d'une vacance, cette transition sera franchie après une durée aléatoire $1/\theta$ elle remet un jeton dans la place *libre1* et un jeton dans la place *verr*.

5.2.3 Modèle de transmission entre plusieurs nœuds et un avec service limite à vacance multiple (M11)

Dans le modèle (M11) nous allons proposer un modèle pour plusieurs nœuds de capteurs à un nœud (many-to-one) avec la discipline de service limité et vacances multiples aléatoire pour la communication entre eux :

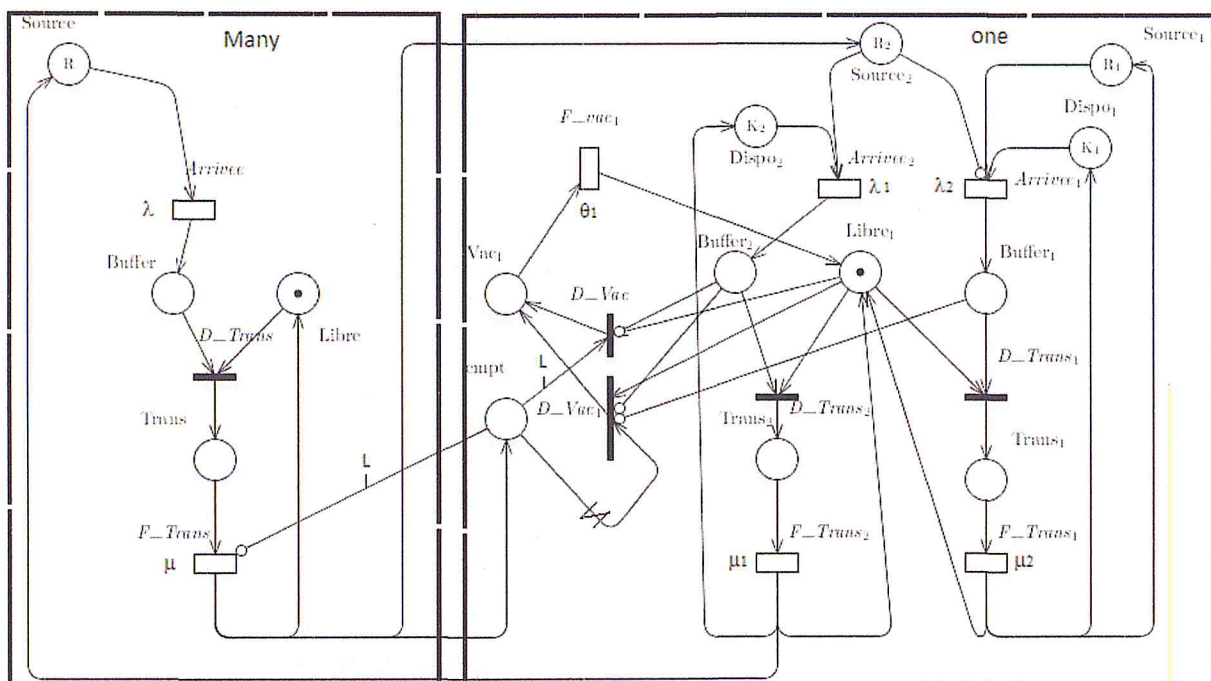


Figure 3.14 Modèle de transmission entre plusieurs nœuds et un avec service limite à vacance multiple (M11)

5.2.4 Modèle de transmission entre plusieurs nœuds à un seul avec service barrière à vacance multiple aléatoire

Dans le modèle (M12) nous allons proposer un modèle pour plusieurs nœuds de capteurs à un seul (many-to-one) avec la discipline de service avec barrière et vacances multiples aléatoire pour la communication entre eux :

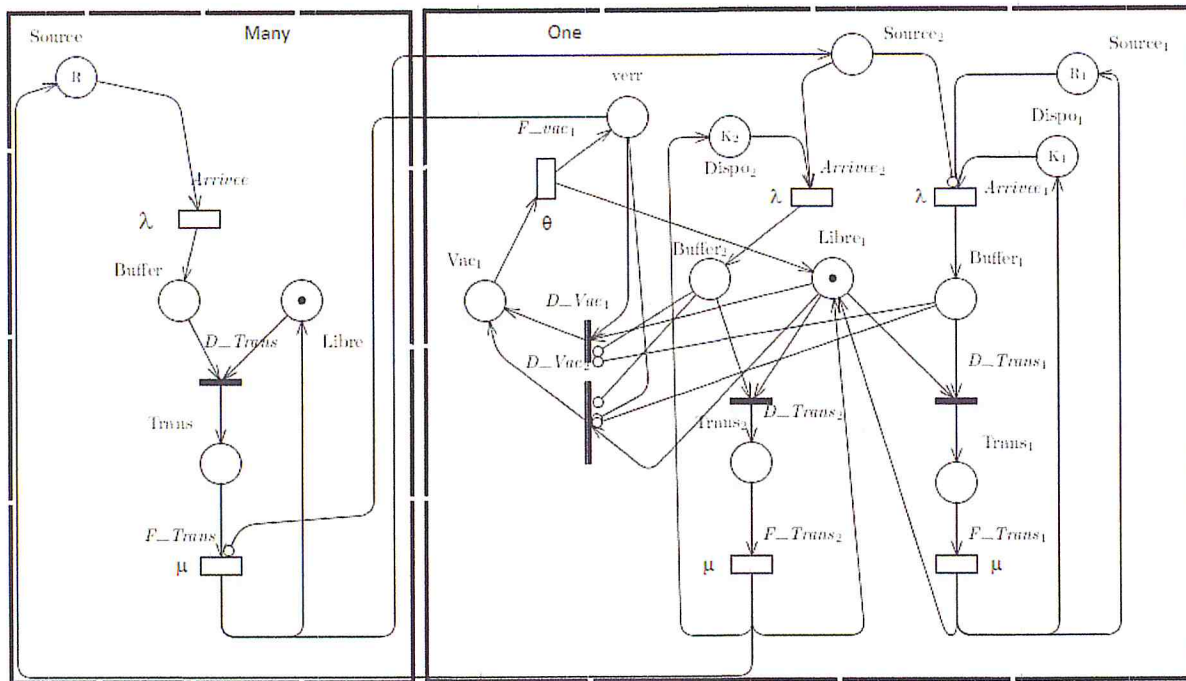


Figure 3.15 Modèle de transmission entre plusieurs nœuds à un seul avec service barrière à vacance multiple aléatoire(M12)

Nous avons généralisé les modèles M11 et M12 à l'aide des modèles respectivement M9 M10, nous avons enlevé la place *dispo* à capacité *K*, pour rendre la capacité de l'émission illimité, car le nombre des nœuds émetteurs est inconnu.

6. Les Indices de performance

En se basant sur les RDPSG des modèles (M1.....12), nous pouvons en décrire les indices de performances tel que : λ c'est le taux d'arrivée, θ c'est le taux de vacance et μ c'est le taux de transmission .

- **Nombre moyen de paquet dans le Buffer, \bar{Q} :** il correspond au nombre moyen de jetons dans la place *Buffer* et il est calculé par :

$$\bar{Q} = \sum_{i: M_i \in A} M_i(Buffer) \cdot \pi_i$$

- **Nombre moyen de paquet dans le capteur, \bar{L} :** il correspond au nombre moyen de jetons dans la place *Buffer* (\bar{Q}) plus le nombre moyen de jetons dans la place *Trans* et il est calculé par :

$$\bar{L} = \bar{Q} + \sum_{i: M_i \in A} M_i(Trans) \cdot \pi_i$$

- **Débit de réception des messages $F^*(Arrivée)$** : il correspond à la fréquence moyenne de franchissement de la transition *Arrivée* et est calculé par :

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

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

$$F^*(F_trans) = \mu \sum_{i: M_i \in A(F_trans)} \pi_i$$

- **Probabilité de saturation du buffer $P_{Saturation}$** : correspond à la probabilité que la capacité du buffer du capteur atteigne la limite *k* de donnée. On d'autres termes ceci représente la surcharge du capteur. Ceci est calculé comme suit :

$$P_{Saturation} = \sum_{i: M_i(Dispo)=0} \pi_i$$

- **Probabilité de vacance P_{Vac}** : correspond à la probabilité que le marquage de la place *Vac* égale a 1. On d'autres termes ceci représente la probabilité mise en vacance du capteur du capteur. Ceci est calculé comme suit :

$$P_{Vac} = \sum_{i: M_i(Vac)=1} \pi_i$$

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

- Dans le cas de vacance aléatoire :

$$Tmoy_v = \frac{1}{Taux\ F_Vac}$$

- Dans le cas de fin de *N_vacance* :

$$Tmoy_v = \frac{N}{Taux\ Arrivée}$$

- **Temps moyen d'état actif $Tmoy_A$** : représente le temps moyen qu'un jeton passe dans la place *Libre*.

$$Tmoy_A = \frac{1}{Taux\ F_Trans}$$

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

$$Tmoy_b = \frac{\bar{q}}{F^* (Arrivée)}$$

- **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 :

$$Tmoy_s = Tmoy_b + \frac{1}{TauxFin_trans}$$

- **Probabilité que le seuil N de donnée soit atteint P_N (modeles avec N_Vacance)** : Correspond à la probabilité que la capacité du *Buffer* du nœud capteur atteigne le seuil N. Ceci est calculé comme suit :

$$P_N = \sum_{i: M_i i: M_i (Buffer=N)} \pi_i$$

7. Conclusion

Dans ce chapitre nous avons étudié la notion de vacance d'un capteur sans fil qui est en rapport avec la mise en veille de l'unité de transmission. Grâce aux réseaux de Petri stochastiques généralisés, nous avons modélisé différentes politiques de vacances non-exhaustives. Les politiques non-exhaustives permettent une bonne expression d'un comportement réel d'un capteur sans fil.

Nous avons proposé plusieurs modèles de vacance pour un seul nœud capteur, et pour deux nœuds qui communiquent entre eux, enfin on a généralisé avec une proposition pour la communication entre plusieurs nœuds avec un seul (many-to-one).

Dans le chapitre suivant nous allons présenter les formules pour calculer la consommation d'énergie et le gain d'énergie, nous allons montrer aussi les différents graphes de la probabilité de vacance, nombre moyen de paquets dans le buffer, temps moyen passé par un paquet dans le buffer, la consommation d'énergie, et le gain d'énergie, en fonction du taux d'arrivée, de la taille du buffer, nombre de paquets dans le buffer (N), du nombre de vacances successives (J), du taux de vacance.

1. Introduction

Dans ce chapitre nous nous intéressons à l'analyse numérique des modèles de vacance conçus dans ce mémoire ainsi qu'aux résultats issus de nos expérimentations.

Notre étude va porter essentiellement sur l'influence du taux d'arrivée des messages, nombre de paquets dans le buffer, du nombre de vacances successive, de la taille du buffer, du taux de vacance, nous allons analyser l'influence des différents politiques de vacances sur les mesures de performances d'un seul nœud capteur ainsi que dans le cas des scénarios de communication de deux nœuds et de plusieurs nœuds capteurs (many-to-one).

Les mesures de performance étudiées sont : le nombre moyen de messages dans le buffer, le temps moyen passé par un paquet dans le buffer, la probabilité qu'un capteur soit en vacance, la consommation d'énergie et le gain d'énergie.

Les résultats numériques de ces expérimentations ont été obtenus après avoir introduit les modèles décrivant les différentes politiques de vacance étudiées et implémenté les formules des indices de performance correspondants, grâce à un outil très performant de spécification, de validation et d'analyse des réseaux de Pétri stochastiques généralisés GreatSPN 2.0.2.[31]

2. Description du GreatSPN

GreatSPN (GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets) [31], est un outil software qui permet la spécification, la validation et l'évaluation des performances des systèmes parallèles, en utilisant le formalisme des RdP stochastiques généralisés et leurs extensions colorées. Il comprend des algorithmes d'analyse efficaces, qui permettent son utilisation dans des applications complexes.

Les principales fonctions et techniques d'analyse implémentées dans l'outil GreatSPN sont :

- Une interface graphique simple à manipuler qui permet à l'utilisateur la saisie du modèle correspondant au système à analyser. Ce choix s'explique par la nature graphique des représentations du formalisme RdP;
- 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.

L'outil permet de calculer automatiquement et implicitement les fréquences de franchissement des transitions. De plus il permet le calcul d'autres indices de performance qui peuvent être explicitement définis par l'utilisateur, tel que : le nombre moyen de marques dans une place donnée, la probabilité d'un certain événement. Une grammaire formelle est utilisée pour la définition des résultats désirés. Les résultats obtenus peuvent être visualisés sur l'interface graphique du logiciel, comme on peut les récupérer à partir du fichier.sta .

3. Mesure de gain

la quantité de gain en matière d'énergie est en fonction de temps passe dans la place vac ,le nombre de transissions d'état sommeil a l'état actif , la consommation totale d'énergie est donnée par EC :

$$ECt = ECv.Pv + ECa.Pa + ECs.Nc + \overline{Q}. EC_{Tx}$$

Tel que :

ECv : l'énergie consommée dans l'état vacance

ECa : l'énergie consommée dans l'état actif

ECs : l'énergie consommée pour un switch de l'état sommeil a l'état actif

EC_{Tx} : l'énergie consommée pour la transmission d'un paquet présent dans le buffer

Pv : probabilité de vacance

Pa : probabilité actif

$$\left\{ \begin{array}{l} P_v = \sum_{i: M_i(vac)=1} \pi_i \\ P_a = 1 - P_v \end{array} \right.$$

Dans notre calcule de la consommation d'énergie on a utilisé les valeurs suivant :

$$ECv=4, ECa=200, ECs=20, ETx=2 \quad [32]$$

Ainsi que nous pouvons déterminer le gain d'énergie par :

$$Gain = \frac{ECa - ECt}{ECa} * 100$$

4. Etude expérimentale

Au cours de cette section nous testons les différents modèles de vacance que nous avons conçues et discutons les différents résultats obtenus en utilisant l’outil GreatSPN. Cette étude expérimentale sert principalement à observer l’impact du taux d’arrivée, nombre de paquet dans le buffer, le nombre de vacances successives, de la taille du buffer et de la durée d’une vacance sur la probabilité de vacance, nombre moyen de message dans le buffer, temps de séjour moyen d’un message dans le système et sur le gain en énergie d’un capteur. Il est à noter que la probabilité de vacance et nombre moyen de paquet dans le buffer ont un impact direct sur le gain d’énergie,

Nous présentons dans ce qui suit les résultats numériques de l’étude de performances des nœuds capteurs sans fil. Nous illustrons l’effet des différentes variations sur les principaux indices de performances.

Le tableau suivant montre les différentes valeurs avec lesquelles nous avons mené nos expérimentations :

	R	R1-R2	K	K1-K2	μ	λ	θ	L	J	N
Figure 4.1	100	-	10	-	10	3	0.01-0.2	4	2	-
Figure 4.2	100	-	10	-	10	3	0.2	4	2-7	-
Figure 4.3	100	-	10	-	10	3	-	4	-	1-10
Figure 4.4	100	-	10	-	10	3	0.01-0.2	4	2	-
Figure 4.5	100	-	10	-	10	3	-	4	-	1-10
Figure 4.6	100	-	10-30	-	10	3	0.2	4	2	10
Figure 4.7	100	-	10	-	10	1-5	0.2	4	5	10
Figure 4.8	100	-	10	-	10	1-5	0.2	4	5	10
Figure 4.9	100	-	10	-	10	1-5	0.2	4	5	10
Figure 4.10	100	-	10	-	10	3	0.01-0.2	4	5	-
Figure 4.11	100	-	10	-	10	3	0.01-0.2	4	5	-
Figure 4.12	100	-	10	-	10	3	0.2	4	-	1-10
Figure 4.13	100	-	10	-	10	3	0.2	4	-	1-00
Figure 4.14	-	50	-	5	10	3	0.01-0.2	2,4	-	-
Figure 4.15	-	50	-	5	10	3	0.01-0.2	2,4	-	-
Figure 4.16	-	50	5	5	10	3	0.01-0.2	2,4	-	-
Figure 4.17	-	50	5	5	10	3	0.01-0.2	2,4	-	-
Figure 4.18	-	50	5	5	10	3	0.01-0.2	2,4	-	-
Figure 4.19	-	50	5	5	10	3	0.01-0.2	3	-	-
Figure 4.20	-	50	5	5	10	3	0.01-0.2	3	-	-
Figure 4.21	-	50	5	5	10	3	0.01-0.2	3	-	-
Figure 4.22	-	50	5	5	10	3	0.01-0.2	3	-	-
Figure 4.23	-	50	5	5	10	3	0.01-0.2	3	-	-

Tableau 4.1 paramètre du réseau

4.1. Etude expérimental pour les modèles pour un seul nœud de capteur

Nous présentons nos résultats numériques de l'étude de performance des modèles d'un seul nœud capteur

M1 : Modèle de vacance multiple aléatoire avec barrière

M2 : Modèle de vacance unique aléatoire avec barrière

M4 : Modèle de service avec barrière à vacance multiple adaptative aléatoire

M5 : Modèle avec service limité à vacance multiple aléatoire

M6 : Modèle avec service limité à vacance unique aléatoire

M8 : Modèle avec service limité à vacance multiple adaptative aléatoire

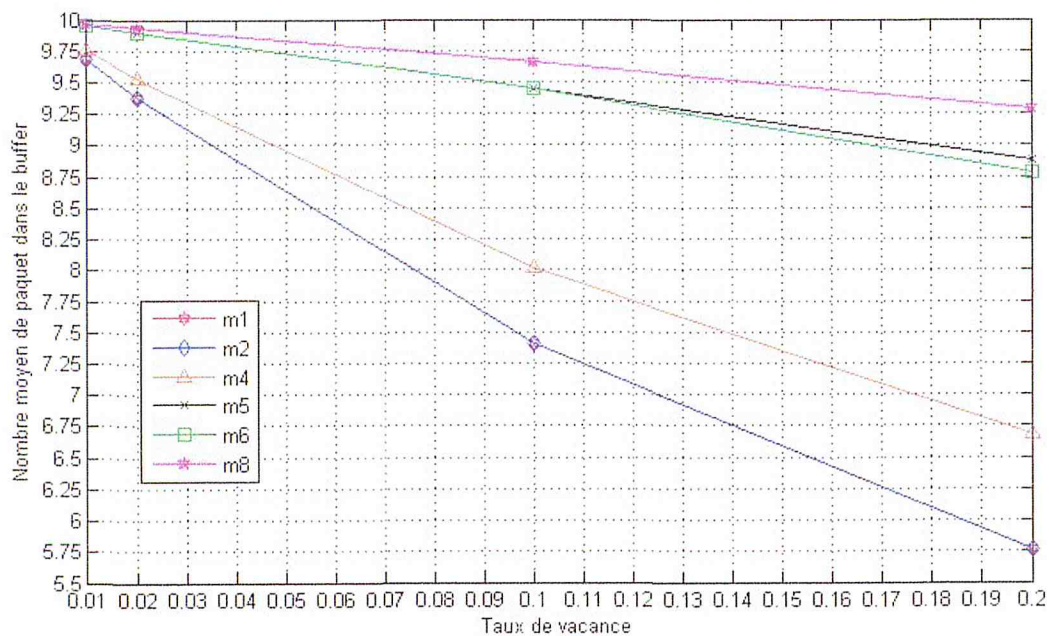


Figure 4.1 nombre moyen de paquet dans le buffer en fonction du taux de vacance

la figure 4.1 montre l'impact du taux de vacance sur le nombre moyen de paquet dans le buffer. on voit clairement que le nombre moyen de paquet dans le buffer décroît avec l'augmentation du taux de vacance on voit aussi que le nombre moyen de paquet des les modeles limités (M5, M6, M8) est largement superieur a celui dans les modeles avec barrière(M1, M2, M4), on remarque aussi qu'on a presque

les même résultats pour les modelés (M1, M2), et pour les modèles (M5, M6) car ils utilisent une politique de vacances multiples et une politique de vacance unique.

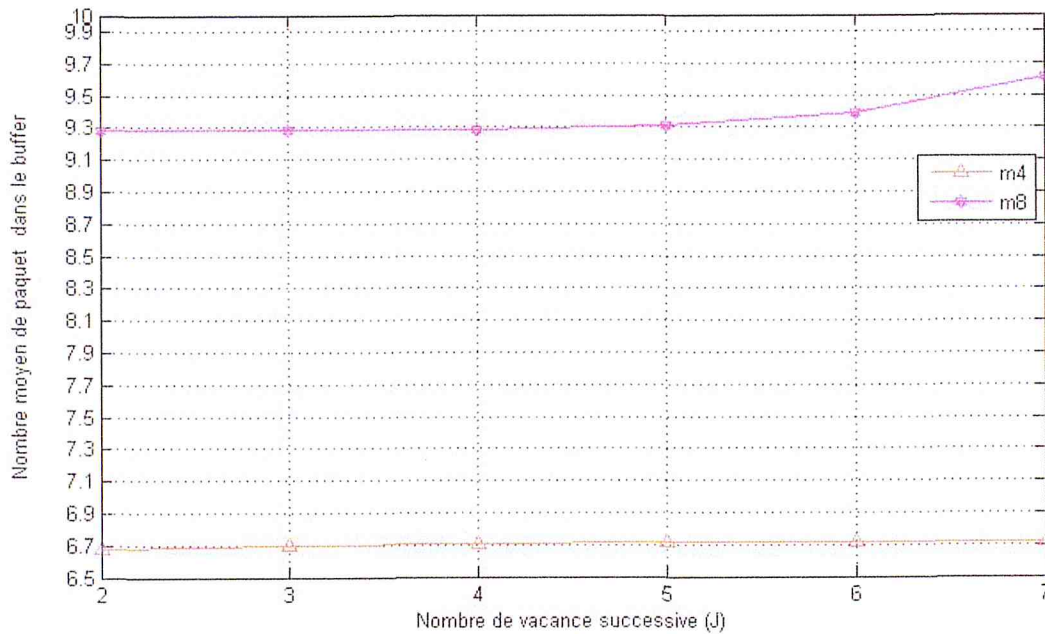


Figure 4.2 nombre moyen de paquet dans le buffer en fonction nombre de vacance successive (J)

figure 4.2 montre l'impact du nombre de vacances successives sur le nombre moyen de paquet dans le buffer, on remarque que le nombre moyen de paquet dans le buffer est constant dans le modèle M4 avec service barrière par rapport aux nombre de vacance successive, on remarque aussi que le nombre de paquet moyen de paquet dans le buffer dans le modèle M8 avec un service limité est stable jusqu'à ce que le nombre de vacance atteint le 5 et là il commence à croitre, on constate que le nombre de vacance successive n'influence pas sur le nombre moyen de paquets dans le buffer dans le modèle avec barrière, par contre quand il atteint la valeur 5 il influence sur le nombre moyen de paquet dans le modèle avec service limité.

M3 : Modèle de vacance unique N-vacance avec barrière

M7 : Modèle avec service limité à N-vacance unique

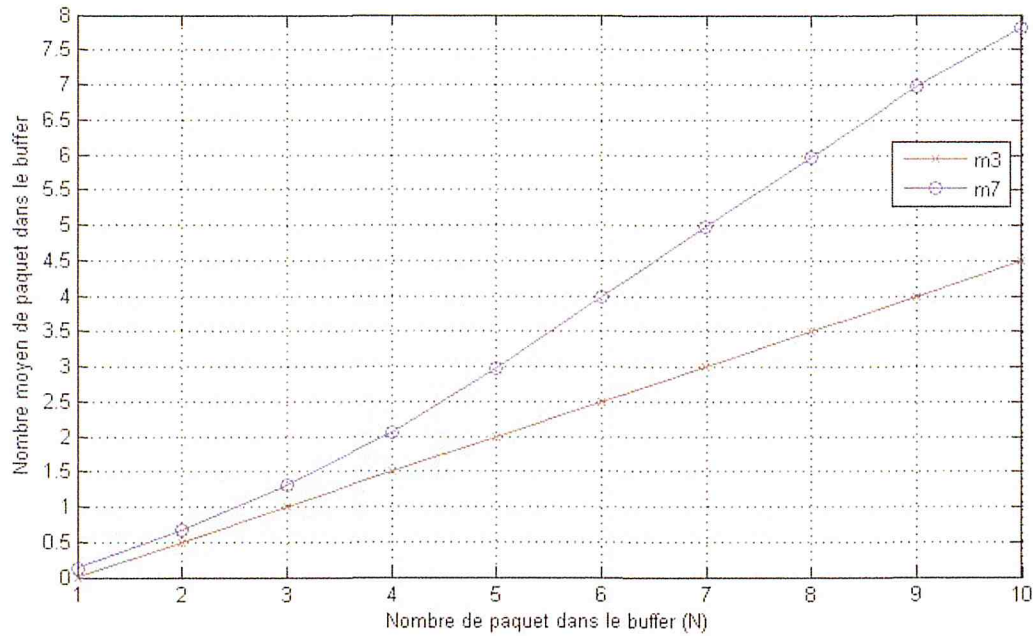


Figure 4.3 nombre moyen de paquet dans le buffer en fonction du nombre de paquet dans le buffer (N)

la figure 4.3 montre que le nombre moyen de paquet augmente avec l'augmentation du nombre de paquet dans le buffer (N), on remarque que le nombre moyen de paquet pour le modèle M7 est supérieur à celui du M4, car le modèle M7 utilise un service limité et le modèle M4 utilise un service avec barrière.

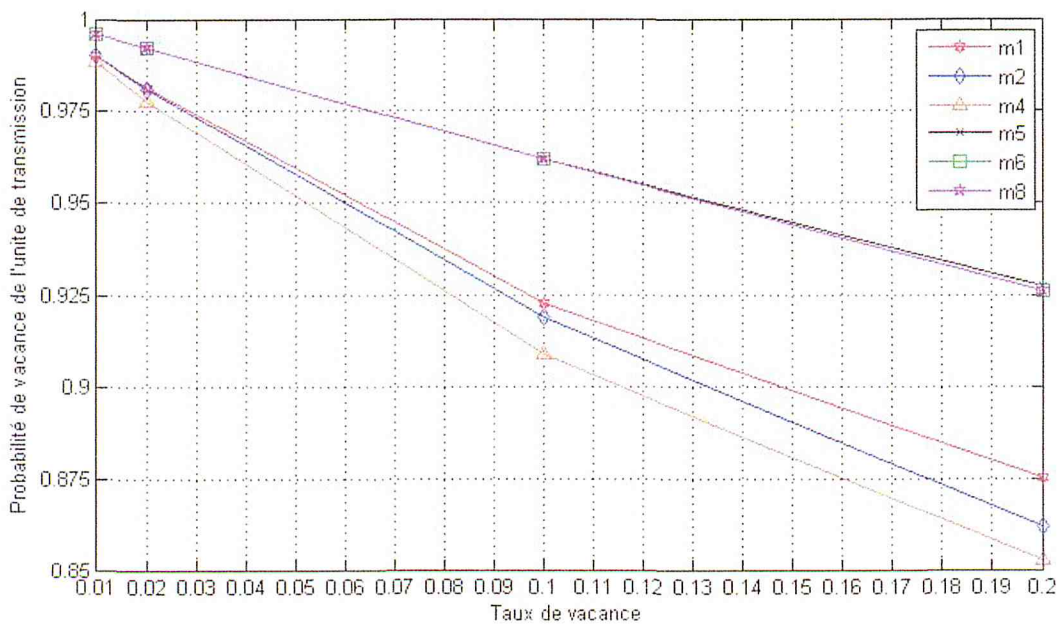


Figure 4.4 probabilité de vacance de l'unité de transmission en fonction du taux de vacance

La figure 4.4 montre que la probabilité de vacance de l'unité de transmission décroît quand le taux de vacance augmente, on voit que les modèles limité (M5, M6, M8) ont la presque la même probabilité de vacance et elle est supérieure à celles des modèles avec barrière (M1, M2, M4), car dans les modèles service limité le capteur part plus souvent en vacance que dans les modèles avec barrière.

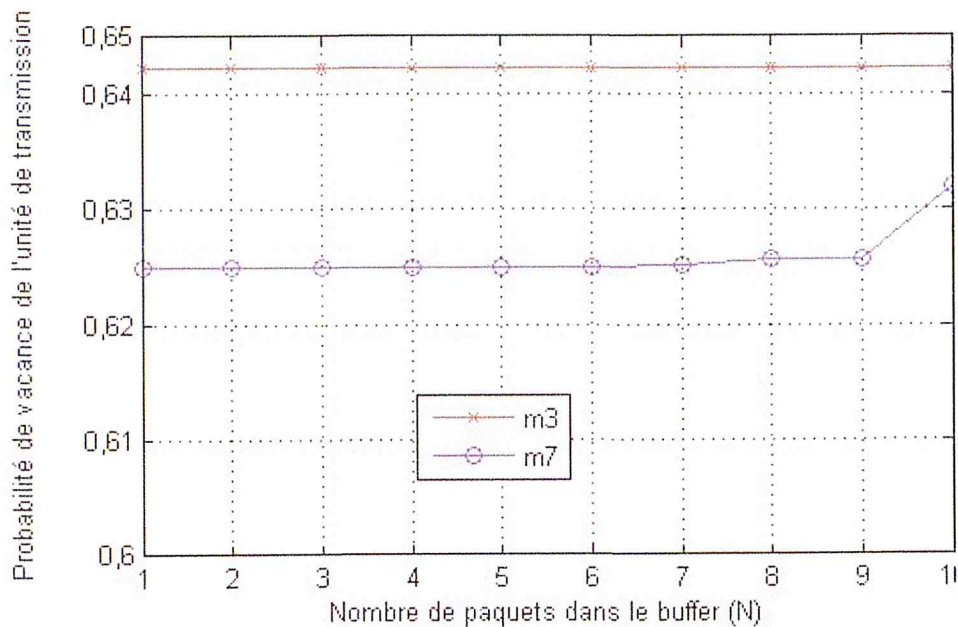


Figure 4.5 probabilité de vacance de l'unité de transmission en fonction du nombre de paquets dans le buffer

la figure 4.5 montre que la probabilité de vacance de l'unité de transmission est stable par rapport aux nombre de paquet dans le buffer pour le modèle M3, pour le modèle M7 elle est stable jusqu'à ce que le $N = 9$ ou elle commence à croître, et elle est supérieur dans le modèle M3 que celle dans le modèle M7, car le modèles M3 avec service avec barrière à son retour de la vacance il traite que les paquets qui étaient dans le buffer à son arrivée et retourne en vacance, par contre le modèles M7 a son retour de la vacance il traite L paquets ou jusqu'à ce que le buffer soit vide.

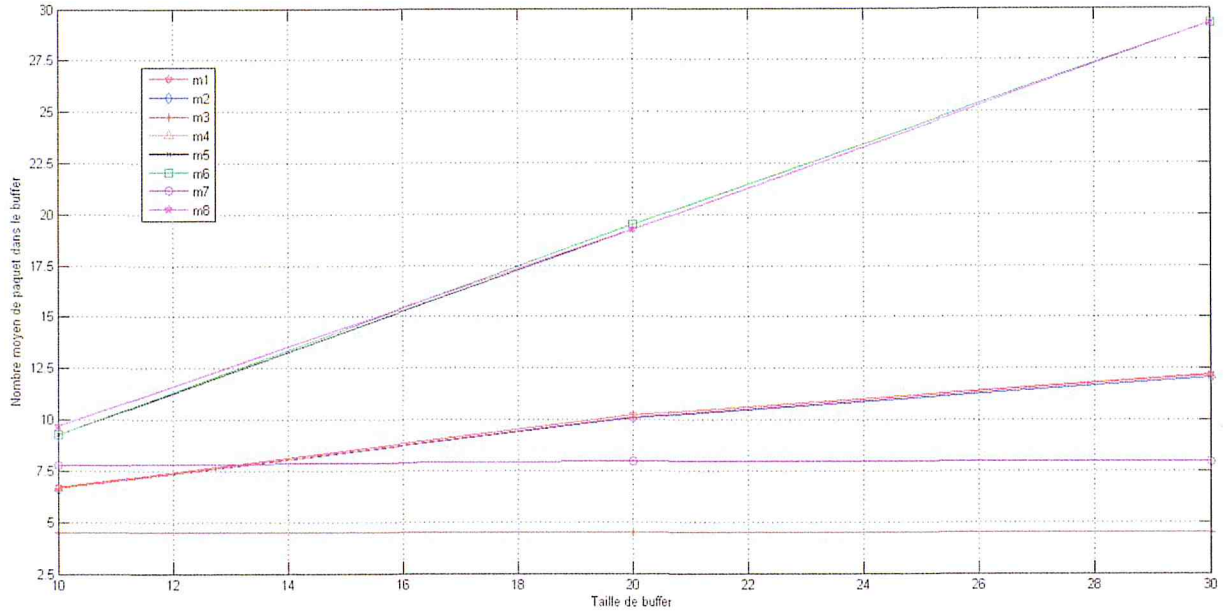


Figure 4.6 nombre moyen de paquet dans le buffer en fonction de la taille du buffer

Dans la figure 4.6 on montre l'influence de la taille du buffer sur le nombre moyen de paquet dans le buffer, on remarque que le nombre moyen de paquet augmente avec l'augmentation de la taille du buffer, sauf pour les modèle M3 et M7 qui sont stable, et sur cette figure on voit également que le nombre moyen de paquet pour les modèle limité(M5 M6 M7 M8) est supérieur à celui des modèles avec barrière(M1 M2 M3 M4).

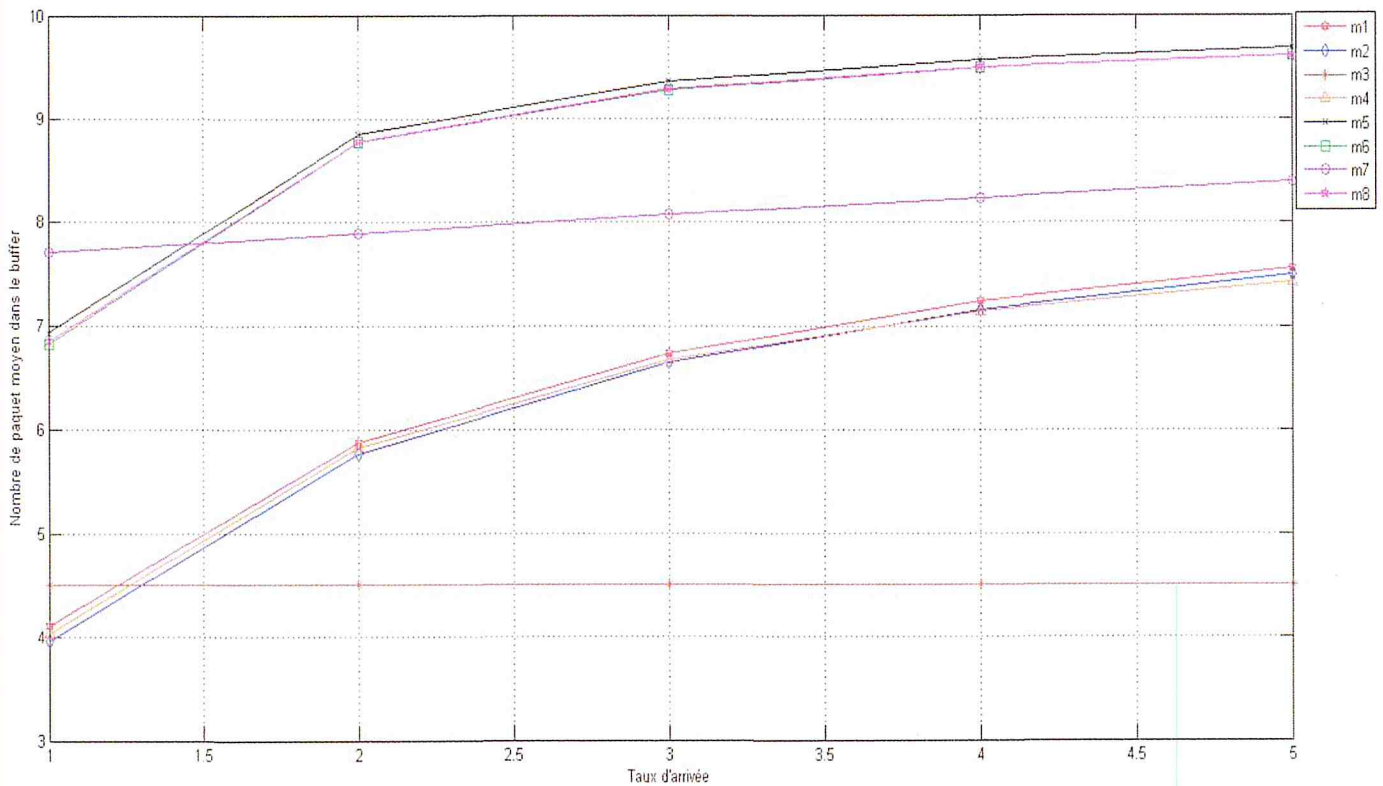


Figure 4.7 nombre moyen de paquet dans le buffer en fonction du taux de d'arrivée

Dans la figure 4.7 on observe que le nombre moyen de paquet croit avec l'augmentation du taux d'arrivée sauf pour le modèle M3 qui est stable, car plus le taux d'arrivée augmente plus le temps d'arrivée diminue et là le nombre moyen de paquet augmente dans le buffer, pour le modèle M3 à vacance unique N-vacance avec barrière il est stable car à son retour de la vacance il traite que les paquets qui étaient dans le buffer à son arrivée et retourne en vacance, donc le taux d'arrivée n'influence pas sur son nombre moyen de paquet dans le buffer.

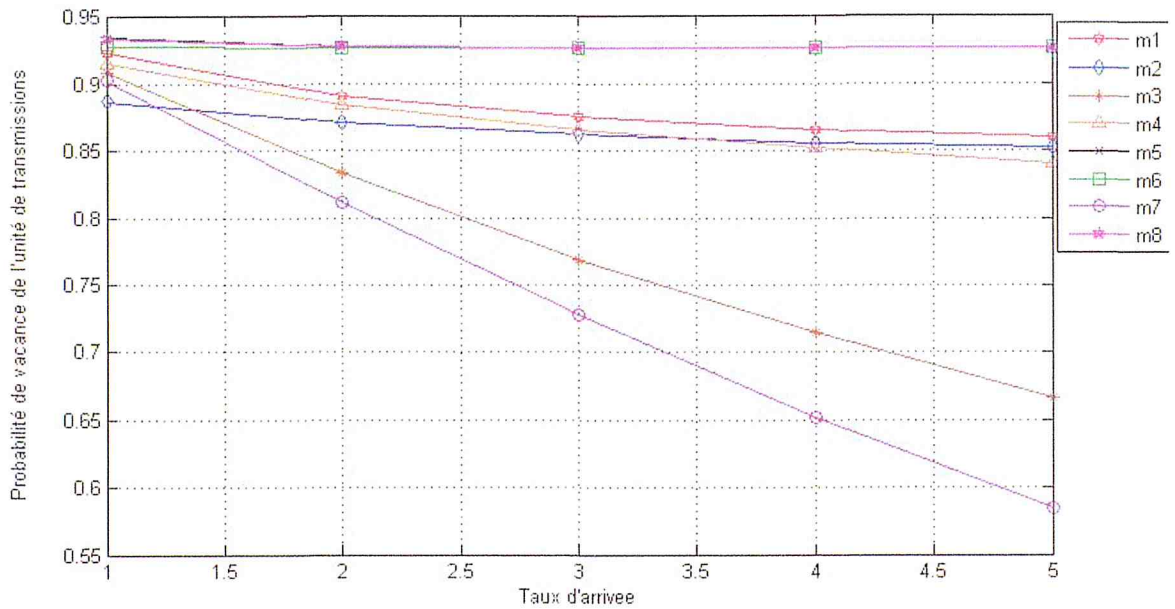


Figure 4.8 probabilité de vacance de l'unité de transmission en fonction du taux d'arrivée

On voit dans la figure 4.8 que la probabilité de vacance de l'unité de transmission décroît avec l'augmentation du taux d'arrivée, sauf pour les modèles (M5, M6, M8) ou elle est presque stable, on remarque que la probabilité de vacance pour les modèles limité (M5 M6 M7 M8) est supérieur à celle des modèles avec barrière (M1 M2 M3 M4).

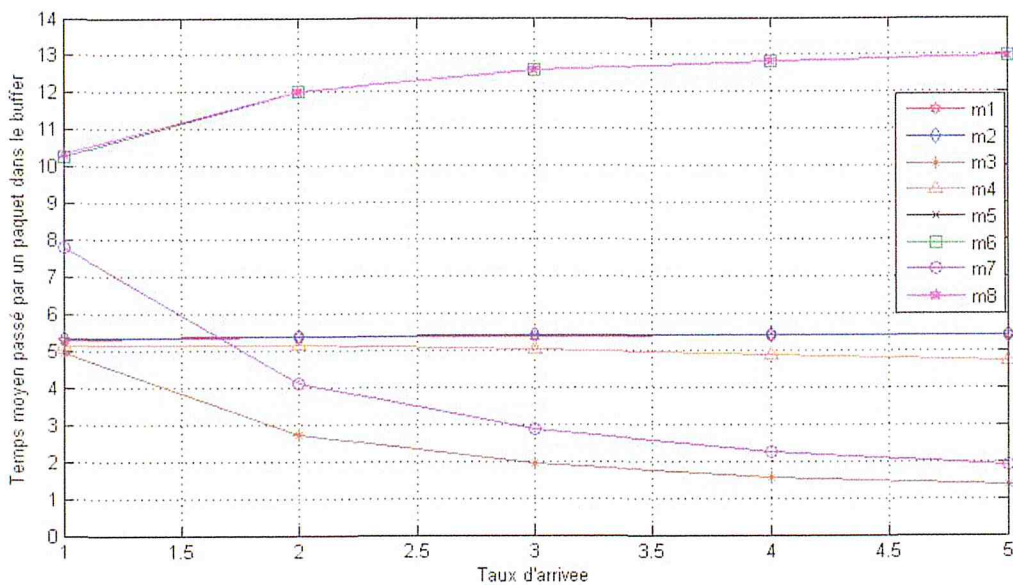


Figure 4.9 temps moyen passé par un paquet dans le buffer en fonction du taux d'arrivée

Dans la figure 4.9 on montre l'influence du taux d'arrivée sur le temps moyen passé par un paquet dans le buffer, on remarque qu'il décroît pour les modèles M7 et M3, il croît pour les modèles M5, M6, M8, et il est stable pour les modèles M1, M2, M4.

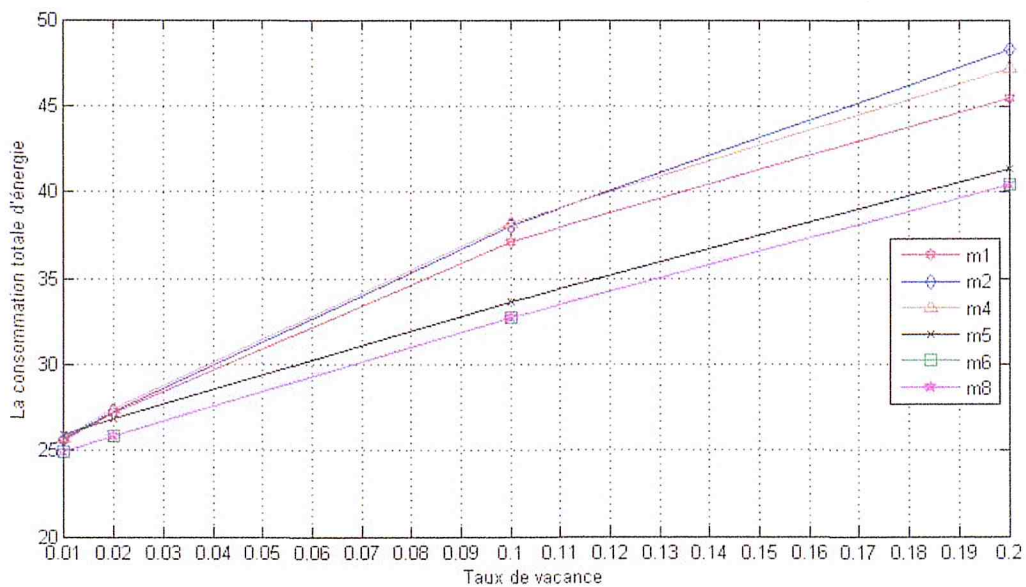


Figure 4.10 la consommation totale d'énergie en fonction du taux de vacance

Dans la figure 4.10 la consommation d'énergie augmente avec l'augmentation du taux de vacance, et on sait que la durée moyenne de vacance = $1/\text{taux de vacance}$ donc la consommation d'énergie diminue avec l'augmentation du temps de vacance, on remarque aussi que les modèles (M5, M6, M8) qui utilisent un service limité consomment moins d'énergie que les modèles (M1, M2, M4) qui utilisent un service avec barrière.

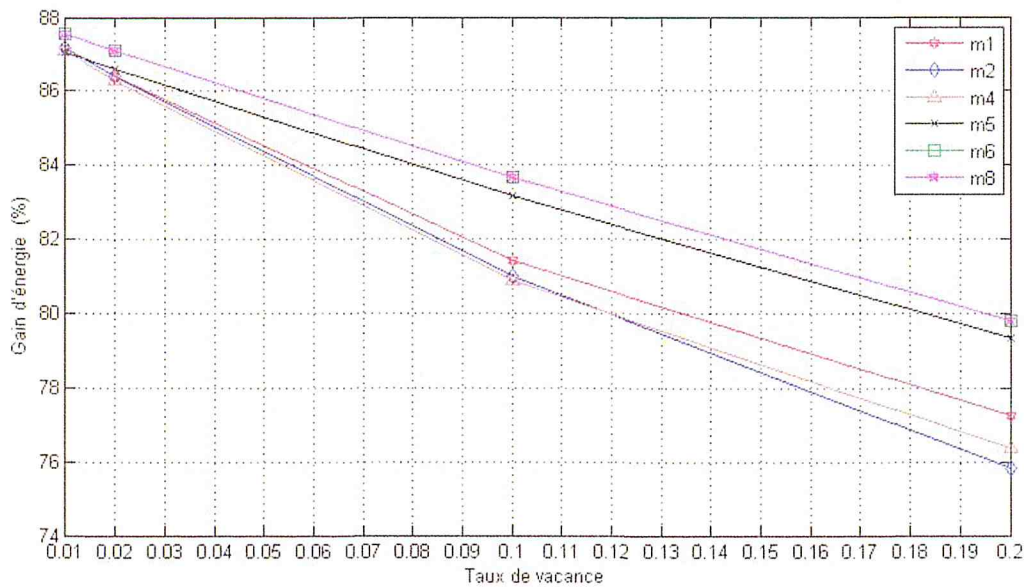


Figure 4.11 gain d'énergie (%) en fonction du taux de vacance

Dans la figure 4.11 on voit l'importante influence du taux de vacance sur le gain d'énergie, on remarque que le gain d'énergie décroît avec l'augmentation du taux de vacance, on sait que durée moyenne de vacance = $1/\text{taux de vacance}$ donc plus que le temps de vacance augment, et la consommation d'énergie diminue d'après la figure (4.10) donc le gain d'énergie augment, les modèles (M1 M2 M4) qui décroissent plus rapidement c'est du à la politique utilisée qui est la politique avec barrière qui partent moins souvent en vacances, ainsi les modèles (M5 M6 M8) présentent un meilleur gain d'énergie c'est du à la politique utilisée qui est la politique de service limité.

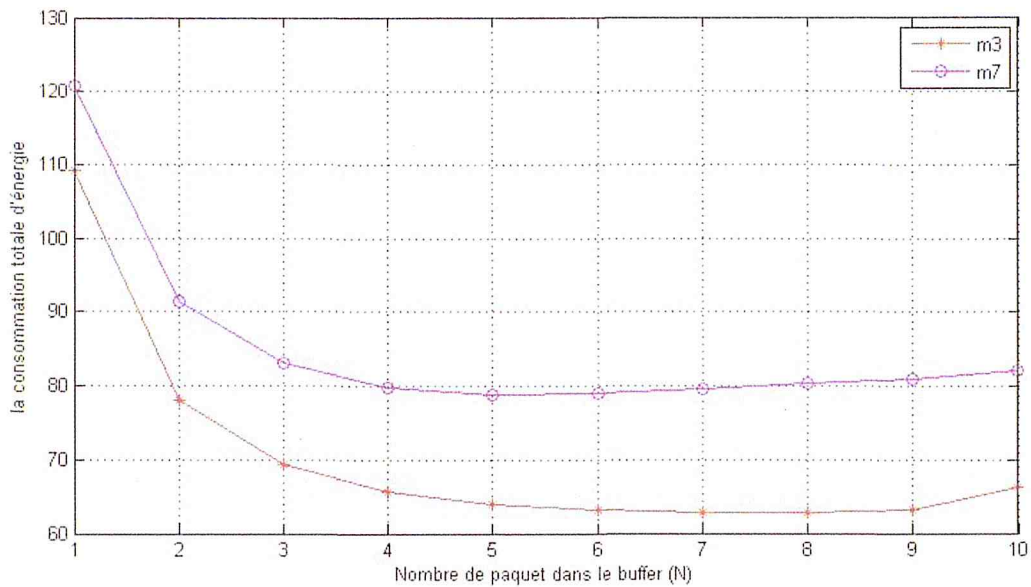


Figure 4.12 la consommation d'énergie en fonction du nombre de paquet dans le buffer (N)

Dans la figure 4.12 on remarque que la consommation d'énergie diminue avec l'augmentation du nombre de paquet dans le buffer, on voit aussi que le modèle M3 avec service avec barrière consomme moins d'énergie que le modèle M7 avec un service limité.

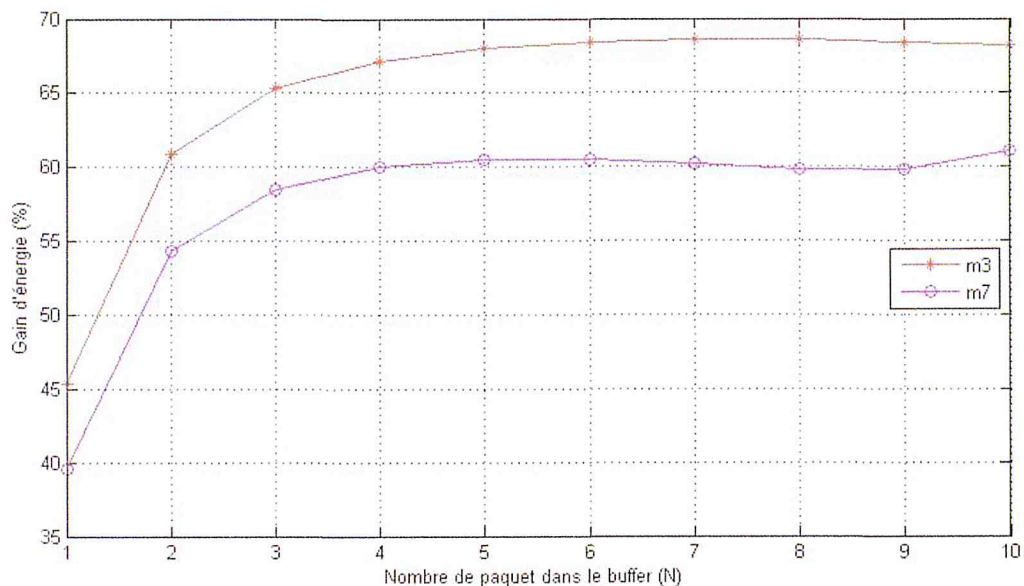


Figure 4.13 gain d'énergie (%) en fonction du nombre de paquet dans le buffer (N)

Dans la figure 4.11 nous avons constaté que le gain d'énergie décroît avec la croissance du taux de vacance, cependant, dans la figure 4.13 on a remarqué que le

gain d'énergie croit avec l'augmentation du nombre de paquet dans le buffer, le gain d'énergie pour le modèle M3 à vacance unique N-vacance avec barrière est plus important que celui pour le modèle M7 avec service limité à N-vacance unique, car la probabilité de vacance de l'unité de transmission pour le modèle M3 est plus grande que celle pour le modèle M7 et c'est ce que nous avons vu dans la figure 4.5.

4.2. Etude expérimental pour les modèles à deux nœuds de capteurs (émetteur, récepteur):

Nous présentons nos résultats numériques de l'étude des performances sur les modèles à deux nœuds capteurs (émetteur, récepteur) :

M9 : Modèle de transmission entre deux nœuds avec service limite à vacance multiple

M10 : Modèle de transmission entre deux nœuds avec service barrière à vacance multiple aléatoire

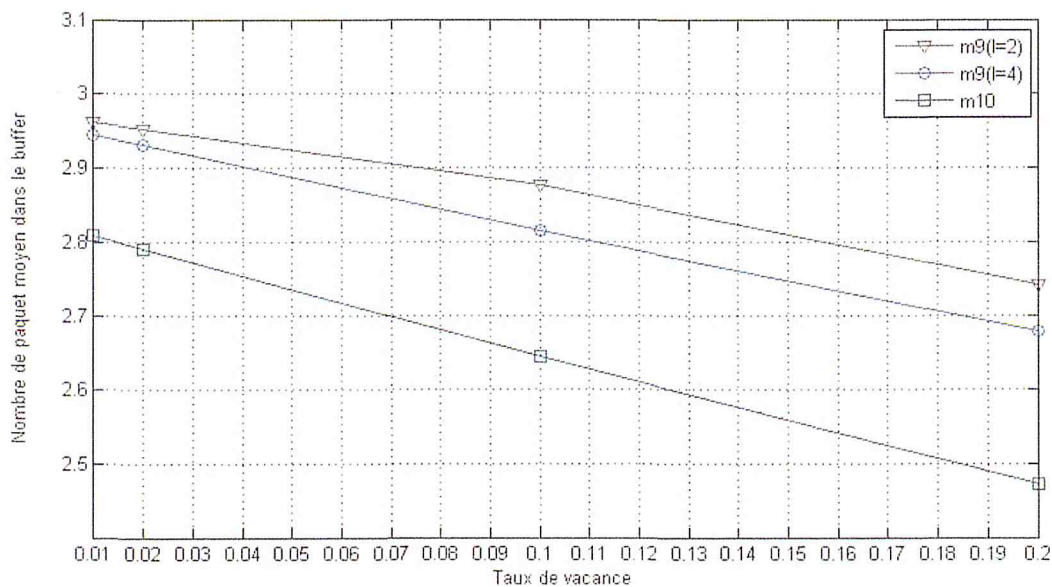


Figure 4.14 le nombre de paquet moyen dans le buffer en fonction du taux de vacance

La figure 4.14 montre que le nombre de paquet moyen décroît avec l'augmentation du taux de vacance, on constate aussi que le nombre de paquet moyen dans les modèles M9 est supérieur à celui du modèle M10, dans la même figure on

remarque également que le nombre moyen de paquet dans le buffer est plus grand dans le modèle M9 (l=2) par rapport au modèle M9 (l=4).

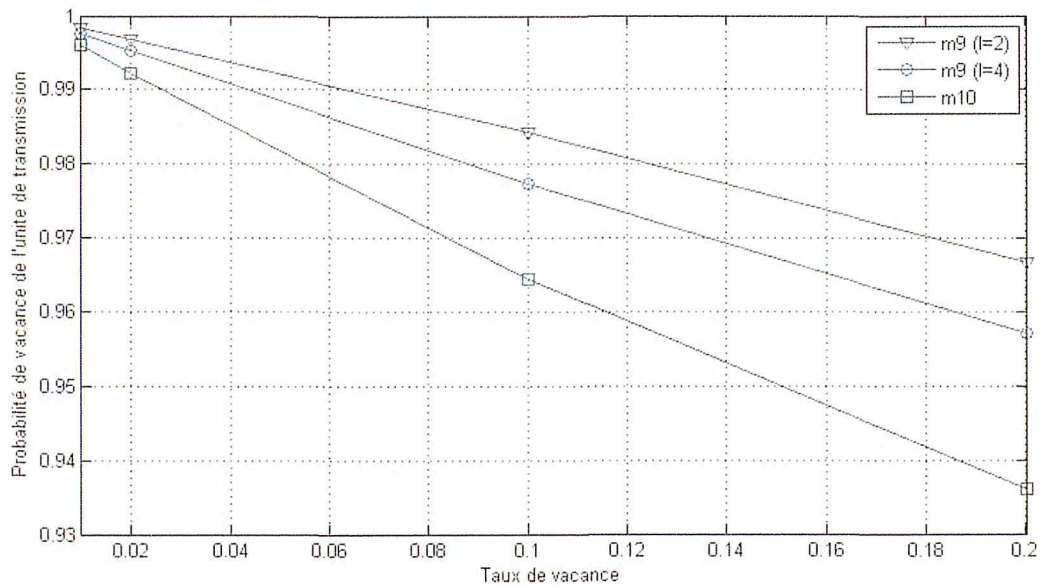


Figure 4.15 probabilité de vacance de l'unité de transmission en fonction du taux de vacance

la figure 4.15 montre l'influence du taux de vacance sur la probabilité de vacance de l'unité de transmission. on remarque que la probabilité de vacance pour les modèles M9 est supérieur à celle du modèle M10, dans la même figure on remarque également que la probabilité de vacance de l'unité de transmission est plus grande dans le modèle M9 (l=2) par rapport au modèle M9 (l=4).

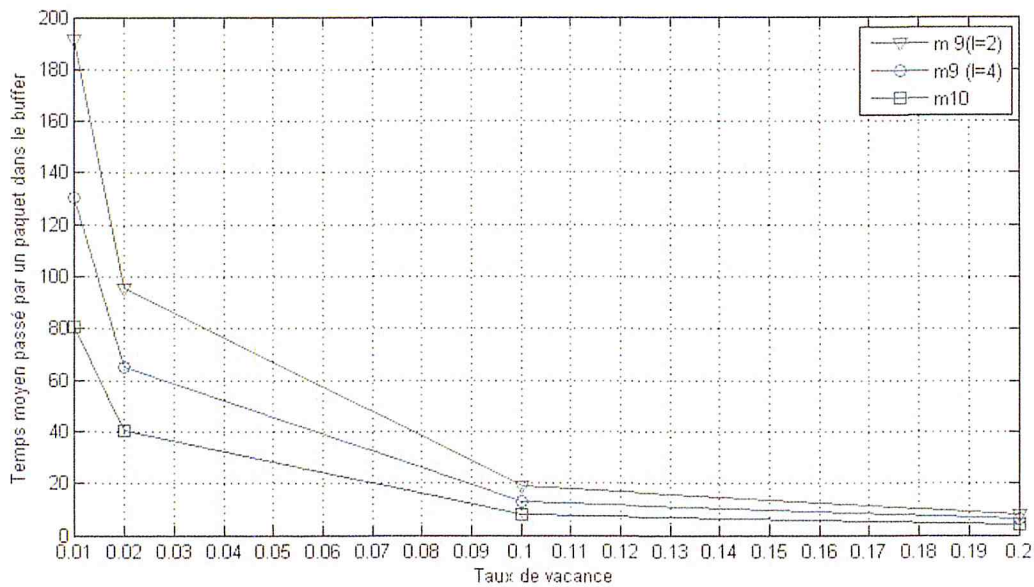


Figure 4.16 temps moyen passé par un paquet dans le buffer en fonction du taux de vacance

Dans la figure 4.16 le temps moyen passé par un paquet dans le buffer décroît en augmentant le taux de vacance. Dans la même figure on remarque également que le temps moyen passé par un paquet dans le buffer est plus grand dans le modèle M9 ($l=2$) par rapport au modèle M9 ($l=4$).

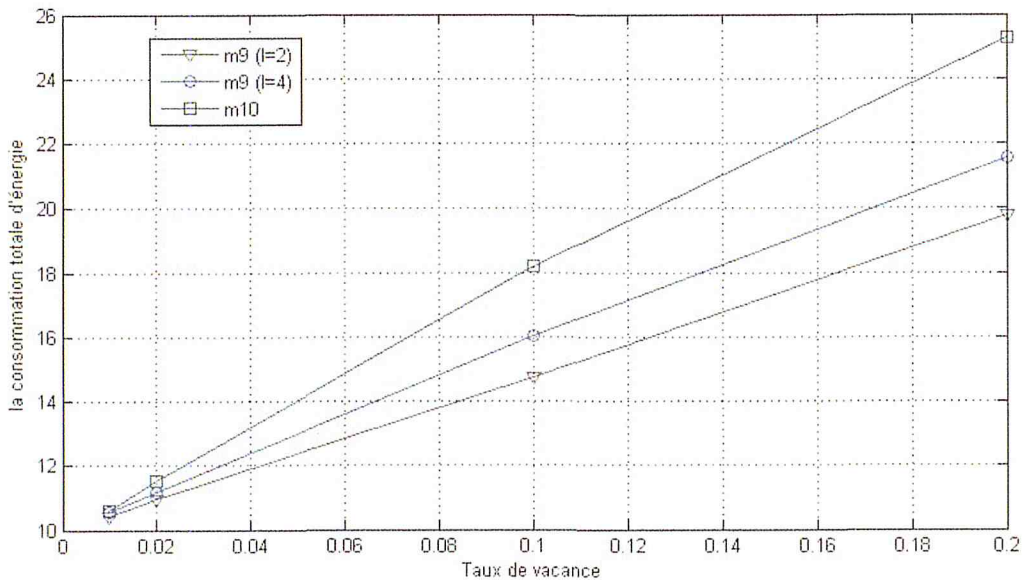


Figure 4.17 la consommation d'énergie en fonction du taux de vacance

La figure 4.17 montre la consommation d'énergie augmente avec l'augmentation du taux de vacance, donc elle va diminuer avec l'augmentation de la durée moyenne de

vacance, le modèle M10 consomme plus d'énergie que les modèles M9, dans la même figure on remarque également que la consommation d'énergie est plus grande dans le modèle M9 (l=4) par rapport au modèle M9 (l=2), c'est du à l'influence de la probabilité de vacance sur la consommation d'énergie.

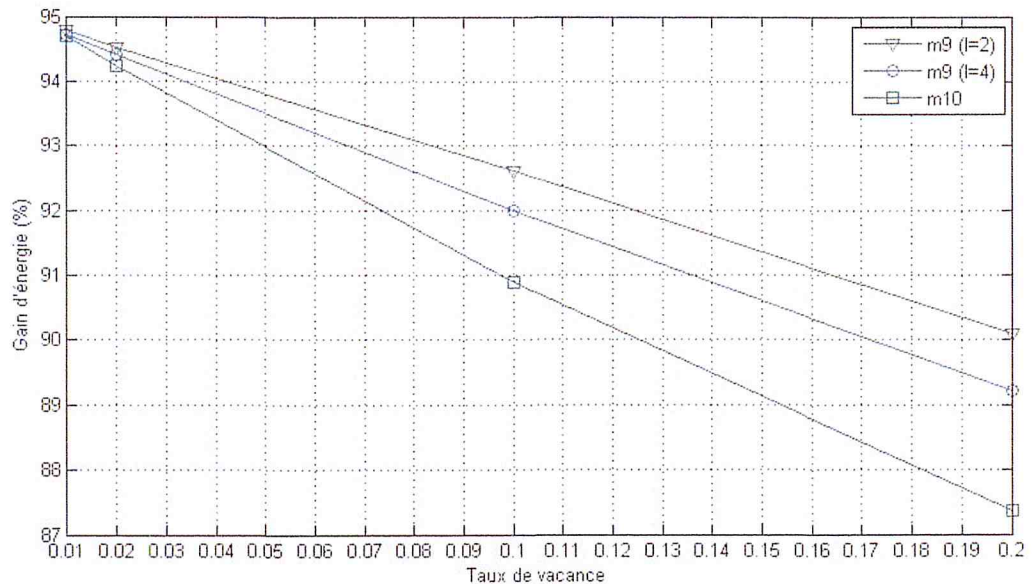


Figure 4.18 gain d'énergie (%) en fonction du taux de vacance

la figure 4.18 montre l'influence du taux de vacance sur le gain d'énergie, on voit que le gain d'énergie décroît en augmentant le taux de vacance, on remarque que les modèles M9 avec service limité donnent un meilleur gain que le modèle M10 avec service barrière, dans la même figure on remarque également que le gain d'énergie est plus important dans le modèle M9 (l=2) par rapport au modèle M9 (l=4). Sa c'est du à l'influence de la consommation d'énergie sur le gain d'énergie

4.3. Etude expérimental pour les modèles de plusieurs nœuds capteur a un (many-to-one):

Nous présentons nos résultats numériques de l'étude des performances sur plusieurs nœuds capteurs à un (many-to-one) :

M11 : Modèle de transmission entre plusieurs nœuds à un avec service limite à vacance multiple

M12 : Modèle de transmission entre plusieurs nœuds a un seul avec service barrière à vacance multiple aléatoire

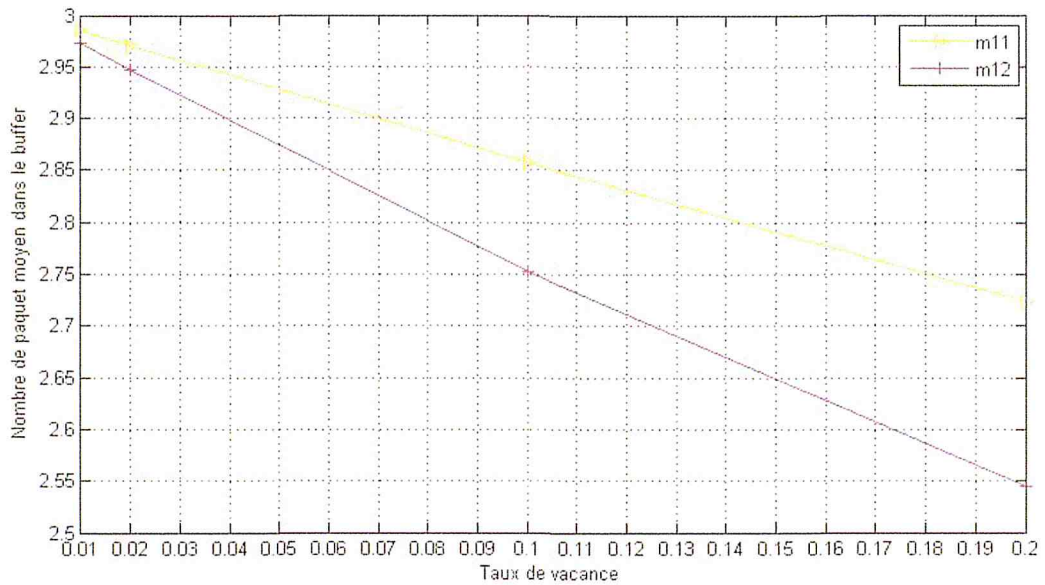


Figure 4.19 nombre de paquet moyen dans le buffer en fonction du taux de vacance

La figure 4.19 montre que le nombre de paquet moyen dans le buffer décroît par rapport au taux de vacance, et on remarque que le nombre de paquet moyen du modèle M11 avec service limité est supérieur à celui du modèle M12 avec service barrière.

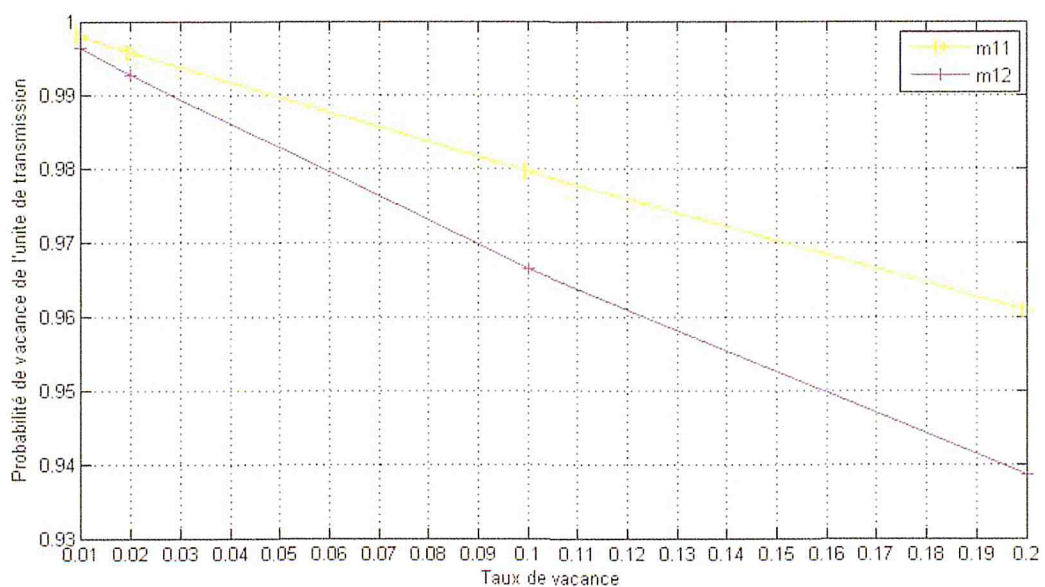


Figure 4.20 la probabilité de vacance de l'unité de transmission en fonction du taux de vacance

La figure 4.20 montre l'influence du taux de vacance sur la probabilité de vacance, on voit que la probabilité décroît en augmentant le taux de vacance, on remarque aussi que la probabilité de vacance du modèle M11 est supérieur à celle du modèle M12.

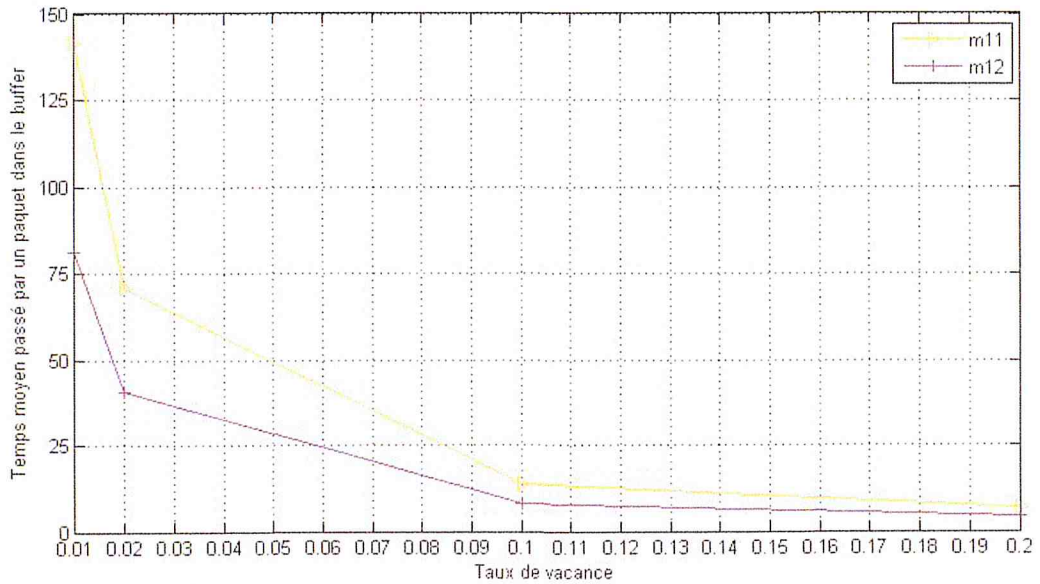


Figure 4.21 temps moyen passé par un paquet dans le buffer en fonction du taux de vacance

la figure 4.21 montre que le temps moyen passé par un paquet dans le buffer diminue en augmentant le taux de vacance, le temps moyen du modèle M11 avec service limité est supérieur à celui du modèle M12 avec service barrière.

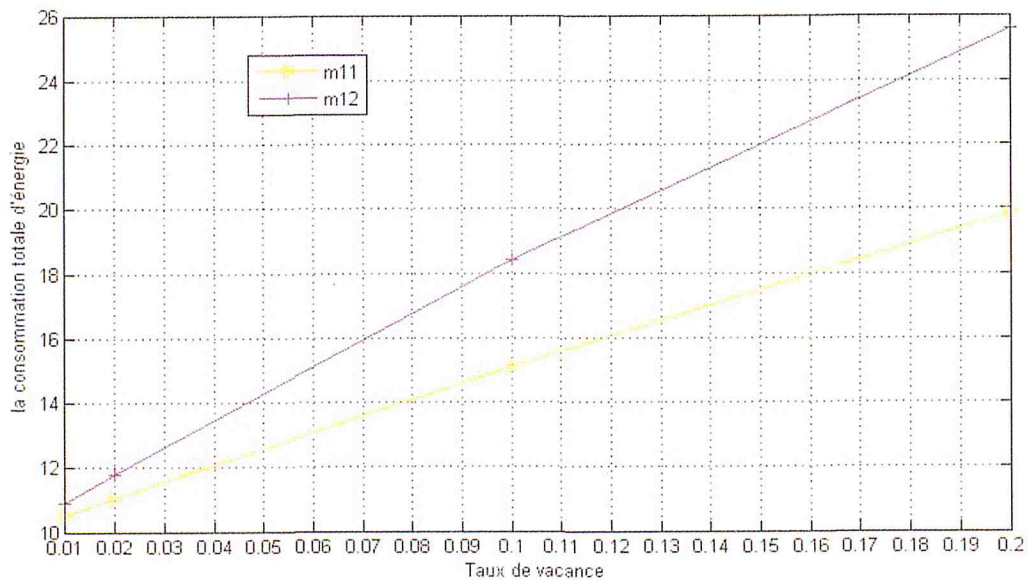


Figure 4.22 la consommation d'énergie en fonction du taux de vacance

Dans la figure 4.22 on remarque que la consommation d'énergie croit avec l'augmentation du taux de vacance, c'est-à-dire qu'elle diminue avec l'augmentation de durée moyenne de vacance, on voit aussi que le modèle M11 avec service limité consomme moins d'énergie que le modèle M12 avec service barrière. cela va influencer sur le gain d'énergie.

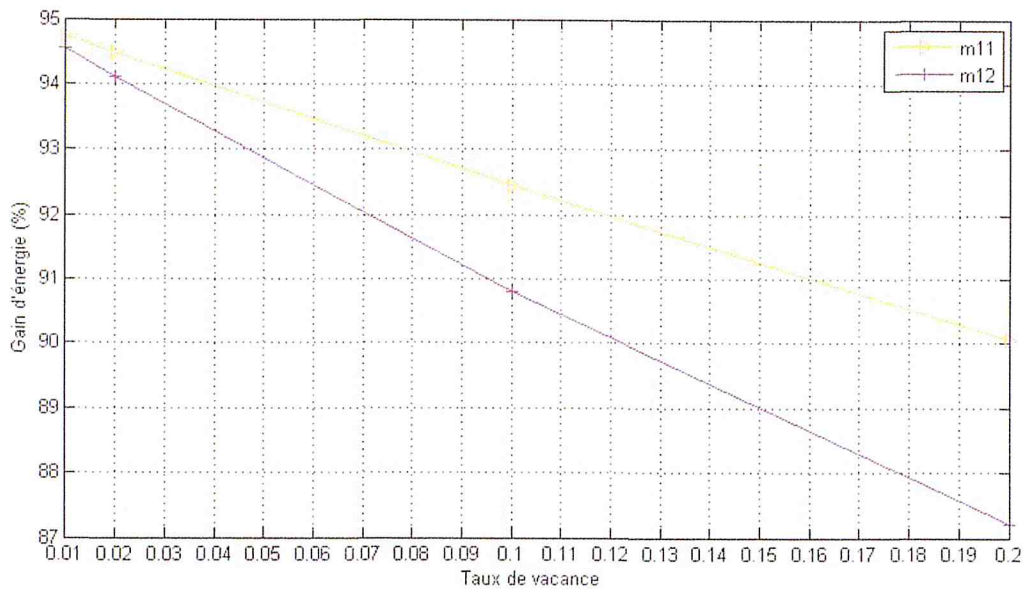


Figure 4.23 gain d'énergie en fonction du taux de vacance

Dans la figure 4.23 on remarque l'influence du taux de vacance sur le gain d'énergie, on voit que le gain décroît en augmentant le taux de vacance, le modèle M11 présente un meilleur gain d'énergie que celui du modèle M12, c'est dû à l'influence de la consommation d'énergie sur le gain d'énergie.

5. Conclusion

Au cours de ce chapitre, nous avons réalisé des calculs numériques en utilisant l'outil GreatSPN, pour montrer l'effet des paramètres du réseau et les disciplines de vacance des capteurs sur les performances

On a appliqué une série d'expérimentations sur les différents modèles que nous avons proposée, nous avons observé l'importance du choix des paramètres et leur impact sur les performances des nœuds capteurs, et par conséquent sur les performances du réseau dans sa totalité

Nous avons pu montrer aussi l'importance et le grand impact de la manière d'exploiter les temps libres des capteurs pour gagner de l'énergie. Les résultats ont montré que dans certains cas, la mise en état de vacance des capteurs sans fil nous offre un meilleur gain en énergie, mais avec une grande augmentation de temps de séjour des messages dans le capteur,

Finalement, nous avons comparé entre les résultats des différentes politiques de vacance, et nous avons constaté que dans certains cas, certaines politiques présentent des meilleurs résultats que d'autres, les modèles avec service limité (M5 M6 M8 M9 M11) donnent des meilleurs résultats que les modèles avec barrière (M1 M2 M3 M10 M12), par contre pour les modèles avec la politique N-vacance le modèle (M3) avec service barrière donne des meilleurs résultats que le modèles (M7) avec service limité.

Conclusion générale

Les réseaux de capteurs constituent un axe de recherche très fertile et peuvent être appliqués dans plusieurs domaines différents, Cependant ces réseaux ont leurs spécificités résultant de leur nature. Ils ont des ressources matérielles très restreintes, comme la capacité de communication, l'espace de stockage, la vitesse de calcul et la source d'énergie. De ce fait l'analyse et l'évaluation des performances de ces réseaux est très important avant tous déploiement. Généralement, l'évaluation des performances des RCSF se fait par le biais des simulateurs à temps discret tel que NS2. Une autre alternative à l'usage des simulateurs c'est l'usage des méthodes formelles qui permettent d'obtenir des résultats exacts.

Parmi les méthodes formelles, nous avons opté pour les réseaux de Petri. Cet outil graphique et mathématique assure une forte capacité de description, de modélisation et d'évaluation des performances de systèmes complexes caractérisés par la concurrence, l'exclusion mutuelle, la synchronisation et le conflit. Une extension à ce modèle est les réseaux de Petri stochastiques généralisé. L'utilisation de ces derniers nous a permis de vérifier à la fois les propriétés qualitatives et quantitatives du système considéré. Afin de réduire la consommation de l'énergie au niveau des nœuds capteur, nous avons proposés d'introduire la notion de vacance issue de la théorie des files d'attente avec vacance durant le temps d'inactivité de la phase actif. Nous avons considérés différentes discipline de vacances non exhaustives.

En se basons sur les RDPSG, nous avons modélisé dans la première partie un seul nœud capteur avec les différentes politiques vacances et services non exhaustif .Dans la deuxième partie nous avons proposé des modèles pour deux nœud capteur dans un scénario émetteur et récepteur. A la fin on a généralisé ces dernier modèles dans un scénario plusieurs à un.

Grâce à l'outil d'évaluation des paramètres de performances des réseaux de Pétri, GreatSPN [31], nous avons pu expérimenter et tester nos différentes modélisations et nous avons pu voir l'impact du taux de réception de message, le taux de transmission, la taille du buffer et le taux de fin de vacance sur la consommation d'énergie de capteur, ainsi que sur ces performance. Nous avons de plus étudié le gain en énergie et nous avons constaté que la probabilité de vacance de l'unité de transmission, la durée d'une vacance, le nombre de paquet moyen dans le buffer et le nombre de cycle influent directement sur ce dernier. Enfin, nous avons vu que la politique avec service limité présente des meilleurs résultats que celle avec barrière, sauf pour le model a N-vacance.

Enfin pour poursuivre ce travail nous proposons comme perspective, de valider nos propositions par un simulateur à un temps discret (NS2,...), intégra les propositions dans le protocole mac dédié aux RCSF. Une autre perspective serait d'intégrer les propositions autant que sous couche dans la couche liaison.

Bibliographie

- [01] M. LEHSAINI, " Diffusion et couverture basées sur le clustering dans les réseaux de capteurs: application à la domotique.", thèse de doctorat en informatique, Université A.B Tlemcen Faculté des Sciences pour l'Ingénieur && Université de Franche-Comté U.F.R Sciences et Techniques École Doctorale SPIM, 2009.
- [02] Yassine BOUFENNECHE, " Influence des messages répétés et des vacances sur les performances et la consommation d'énergie dans les réseaux de capteurs sans fil ", thèse de magister en informatique, U.S.T.H.B , Novembre 2014
- [03] Abdallah Makhoul, "Réseaux de capteurs : localisation, couverture et fusion de données". Thèse de doctorat en informatique, Laboratoire d'Informatique de l'Université de Franche-Comté, Ecole Doctorale Sciences Pour l'Ingénieur et Microtechniques, Novembre 2008
- [04] M. A. Martin, "Wireless Sensor Networks : Technology and Protocols .InTech, " 2012.
- [05] I. Akyildiz, W. Su, E. Cayirci, Y. Sankarasubramaniam. "A survey on sensor Networks", *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, Georgia Institute of Technology, Atlanta, USA. Août 2002.
- [06] D. J. Malan, M. Welsh, and M. D. Smith, "A Public-Key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography", Proc. 1st IEEE Int'l. Conf. Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, Oct.2004.
- [07] Rachid BECHAR , "Surveillance distribuée d'un réseau de capteurs sans fil à base d'agents", thèse de doctorat en informatique, Université d'oran ,avril 2015
- [08] T. RAULT , A. BOUABDALLAH ET Y. CHALLAL, "Energy efficiency in wireless Sensor networks: A top-down survey ", *Computer Networks*, Vol.67, 2014
- [09] H. CUI, C. FENG, R. P. LIU, X. HUANG, AND Y. LIU. " Prolonging Network Lifetime in Two-level Heterogeneous Wireless Sensor Networks 2012 ".International Symposium on Communications and Information Technologies (ISCIT), 2012.
- [10] R. N. ENAM , S. MISBAHUDDIN, AND M. IMAM. " Energy Efficient Round Rotation Method for a Random Cluster Based WSN 2012" .International Conference on Collaboration Technologies and Systems (CTS) , 2012
- [11] J-S. KIM, S-Y CHOI, S-J HAN, AND J-H CHOI. " Alternative Cluster Head Selection Protocol for Energy Efficiency in Wireless Sensor Networks ". *Software Technologies for Future Dependable Distributed Systems* , 2009

- [12] N. Tian and Z. G. Zhang, "Vacation Queueing Models : Theory and Applications" (International Series in Operations Research & Management Science). Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006.
- [13] B. Doshi, "Queueing systems with vacations - A survey," Queueing Systems, vol.1, no. 1, pp. 29–66, 1986.
- [14] H. Takagi, "Queueing Analysis, Vol. 1 : Vacation and Priority Systems," NorthHolland, Amsterdam, 1991.
- [15] A. Lezanska and M.F. Hayat , " Performance Modelling of Energy Efficient Wireless Nodes " IEEE 2011.
- [16] Merlin, P. M." A Study of the recoverability of computer systems". University of California: PhD thesis. 1974
- [17] Sifakis, J,"Use of Petri Nets for Performance Evaluation" Dans E. G. H. Beilner, Measuring Modelling and Evaluation Computer Systems (pp. 75-93). Amsterdam: Eds.1977
- [18] Ramchandani, C. "Analysis of Asynchronous Concurrent Systems by Time Petri Nets". MIT, Cambridge: PhD thesis.1974
- [19] S. Jin , W. Yue and Q. Sun "performance analysis of the sleep/wakeup protocol in a wireless sensor network" International Journal of Innovative Computing, Information and Control ICIC International Volume 8, Number 5(B), Mai 2012 .
- [20] F. C. Jiang, D. C. Huang, and K. H. 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, QTNA '09, (Fusionopolis, Singapore), 2009.
- [21] F. C. Jiang, C. T. Yang, K. H. Wang, and D. C. Huang, "Design Framework to Optimize Power Consumption and Latency Delay for Sensor Nodes Using Min(N,T) Policy M/G/1 Queueing Models," in 5th International Conference on Future Information Technology, FutureTech' 2010, pp. 1–8, May 2010.
- [22] M. Diaz, Les réseaux de Petri : Modèles fondamentaux. Paris : Hermès Science Publications, 2001.
- [23] G. W. Brams, Réseaux de Petri : Théorie et Pratique, volume 1. Paris : Masson, 1983.
- [24] W. Khansa, Réseaux de Petri P-temporels : contribution à l'étude des systèmes à évènements discrets. PhD thesis, Université de Savoie, 1997.

- [25] N. Gharbi, Evaluation des performances et de la fiabilité des systèmes Multi-classes avec Rappel à l'aide des réseaux de Petri Stochastiques colorés. PhD thesis, USTHB, Algiers, 2007.
- [26] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, Modeling with Generalized stochastic Petri Nets. John Wiley & Sons, 1995.
- [27] G. Balbo, "Introduction to generalized stochastic Petri nets," in Formal methods for performance evaluation, pp. 83–131, Springer, 2007.
- [28] Vidal-Naquet, G., & Choquet-Geniet, A. (1992). Réseaux de petri et systèmes parallèle Paris: Editions Armand Colin.
- [29] Symons, F. J. "Modeling and Analysis of Communication Protocols Using Numerical Petri Nets". Universitu of Essex: PhD thesis.1978
- [30] Molloy, M. K. " Performance Analysus Using Stochastic Petri Nets". Dans IEEE Transaction on Computers Vol. 31 (pp. 913-917).1982
- [31] G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaud, "GreatSPN 1.7 : GRaphical Editor and Analyser for Timed and Stochastic Petri Nets," Performance Evaluation, vol. 24, no. 1-2, pp. 47–68, 1995.
- [32] F. C. Jiang, C. T. Yang, D. C. Huang, and F.Y.Leu, "Lifetime elongation for wireless sensor network using queue-based approaches" ,J Supercomput (2012) 59:1312–1335.
- [33] A. Sleeml, M.Bao, W. Ll "Analyzing the Impact of the N-Policy on Quality of Service in Wireless Sensor Networks", IEEE 2014